An Efficient and Secure Authentication Scheme with Session Key Negotiation for Timely Application of WSNs

Jiping Li[1], Yuanyuan Zhang[1], Lixiang Shen[1], Jing Cao[1], Wenwu Xie[2], Yi Zheng[3], and Shouyin Liu[4]

[1] School of Computer Science and Information Engineering, Changzhou Institute of Technology,
Changzhou 213032, China
[e-mail: lijp@czu.cn, zhangyy@czu.cn, caoj@czu.cn, shenlx@czu.cn]

[2] Department of Information Science and Engineering, Hunan Institute of Science and Technology,
Yueyang 410006, China,
[e-mail: gavinxie@hnist.edu.cn]

[3] School of Artificial Intelligence, Jianghan University, Wuhan 430056, China
[e-mail : zheng_saber@163.com]

[4] School of Physical Science and Technology, Central China Normal University, Wuhan 430079, China
[e-mail : syliu@mail.ccnu.edu.cn]

*Corresponding author: Jiping Li

*Received November 8, 2023; revised December 6, 2023; revised December 31, 2023;
accepted February 25, 2024; published March 31, 2024*

Abstract

For Internet of Things, it is more preferred to have immediate access to environment information from sensor nodes (SNs) rather than from gateway nodes (GWNs). To fulfill the goal, mutual authentication scheme between user and SNs with session key (SK) negotiation is more suitable. However, this is a challenging task due to the constrained power, computation, communication and storage resources of SNs. Though lots of authentication schemes with SK negotiation have been designed to deal with it, they are still insufficiently secure and/or efficient, and some even have serious vulnerabilities. Therefore, we design an efficient secure authentication scheme with session key negotiation (eSAS2KN) for wireless sensor networks (WSNs) utilizing fuzzy extractor technique, hash function and bitwise exclusive-or lightweight operations. In the eSAS2KN, user and SNs are mutually authenticated with anonymity, and an SK is negotiated for their direct and instant communications subsequently. To prove the security of eSAS2KN, we give detailed informal security analysis, carry out logical verification by applying BAN logic, present formal security proof by employing Real-Or-Random (ROR) model, and implement formal security verification by using AVISPA tool. Finally, computation and communication costs comparison show the eSAS2kN is more efficient and secure for practical application.

Keywords: Mutual authentication, Session key negotiation, Wireless Sensor Networks, AVISPA, ROR model

This research was supported by the National Natural Science Foundation of China under Grant No. 62372070 and 62372069.

http://doi.org/10.3837/tiis.2024.03.015 **ISSN : 1976-7277**

1. Introduction

In Internet of Things environment, wireless sensor networks (WSNs) act as a bridge, which links the real physical world and impalpable virtual world, and provide the possibility and feasibility to observe and analyze the monitoring objects with a good resolution [1]. Nowadays, WSNs have been widely applied in environment monitoring, disaster alert, healthcare, and military sensing and tracking [2-7]. The WSNs are consisted of a great many scattered sensor nodes (SNs), which are deployed to gather environmental information and transmit it to gateway nodes (GWNs) wirelessly. In WSNs, the GWNs are the most powerful nodes with rich computation and storage resources. However, the SNs are terribly resource-constraint, for example, with insufficient memory capacity, weak computing capability, and limited transmission range. Since WSNs are often randomly scattered in an unattended specific area, an adversary may easily capture a sensor node and extract the secrets from its tamper-prone memory with cost effectiveness. In addition, owing to WSN's open and wireless communication nature, malicious SNs may intercept, replay and even modify the transmitted messages. Moreover, user’s privacy, including user's identity, gender, access time and access habits, is vulnerable to leakage. Therefore, how to ensure the safety and stabilization of WSNs and how to prevent unauthorized disclosure of user's privacy becomes the most important and critical issue.

Generally, users can access the collected data from the GWNs. However, for some emergency scenarios, users hope to obtain environment information timely from some specific SNs instead of from the GWNs. For these cases, the most efficient and feasible method is mutual authentication with session key (SK) negotiation. For confidential and integral data transmission, user and SNs should be mutually authenticated, and a shared SK should be negotiated for subsequent secure data transmission. That is to say, the SNs should have the ability of verifying the user's legitimacy based on the transmitted packets from the user. Meanwhile, the user should also have the ability of verifying the legitimacy of SNs based on the received packets from the SNs. Moreover, SK should be securely negotiated and allocated to the user and the SNs after the mutual authentication has been achieved.

However, it’s so challenging to achieve mutual authentication with SK negotiation due to the SNs' poor power supply, computation ability, communication ability and storage capacity. The conventional schemes are not suitable for WSNs, which is a distributed ad-hoc like networks consisted of a variety of resource-constrained SNs. The most common security mechanism for WSNs is symmetric cryptography [8-9]. These solutions are efficient and simple to implement, however, the same symmetric key for different sessions will make the SNs susceptible to higher risks in unsupervised and unprotected environments [10-11]. Moreover, the solutions are hard to establish a shared secret key beforehand and have nonsupport of non-repudiation. Therefore, some researches focus on public-key cryptography solutions. These solutions do not need to distribute keys in advance and to share pairwise keys. However, these public-key cryptography solutions are too computationally expensive for the resource-constrained SNs if not accelerated by adopting cryptographic hardware. Recently, new trust models have been proposed to secure emergency message dissemination in VANETs, and trust evaluation scheme for federated learning in a digital twin for mobile networks [12-14].

Since a person’s biometrics may vary slightly occasionally [15], high rejection probability will happen in the login phase if conventional bio-hashing technique is adopted [16]. For successful biometric verification in login phase, fuzzy extractor technique [17] is adopted in our efficient secure authentication scheme with session key negotiation (eSAS2KN). Due to the smartcard’s characteristics of convenience and security, and the user’s biometric characteristics of uniqueness, therefore, we aim to design a more lightweight three-factor authentication in terms of password, smartcard and user's biometrics to achieve mutual authentication and session key negotiation utilizing the lightweight cryptographic primitives such as hash and XOR operations.

The main contributions of our work are five folds. 1) A computing-efficient eSAS2KN is proposed for multi-gateway WSNs, which achieves mutual authentication of user, gateway node, and sensor node, as well as negotiating a session key for timely and direct communications between user and the specific sensor nodes. 2) The anonymous three-factor authentication with session key negotiation scheme not only strengthens the security and privacy of the eSAS2KN, but also decreases the user's login-rejection probability in the login phase. 3) The session key is both verified in Msg3 when sensor node is authenticated by gateway node, and verified in Msg4 when the gateway node is authenticated by user, respectively, which greatly improves the confidentiality of real-time communications between user and the specific sensor nodes. 4) Informal security analysis, BAN logical verification, ROR-based formal security proof, and AVISPA-based formal security verification are implemented to prove the security of the eSAS2KN; and 5) Computation overhead and communication overhead are compared with those of relevant schemes to show the efficiency of the proposed eSAS2KN.

2. Related Work

For legitimate access of WSNs and ensuring the confidentiality and reliability of the transmitted information, a variety of mutual authentication with SK negotiation schemes, which are supplemented with long-term secret keys stored in smart card (SC), have been developed in the past decades [18–28].

In 2009, Das [18] first proposed an authentication with SK negotiation protocol between user and SNs by using password and smart card (SC) in WSNs. The protocol well suits the environment of WSNs due to its low computation cost. However, Das's protocol was found not secure against several attacks by several researchers. In 2010, Khan et al. [19] first stated Das's protocol has no password change operations, and more likely to incur privileged-insider attacks as well as GWNs bypassing attacks. Chen et al. [20] also indicated Das's protocol failed in mutual authentication of two communication parties, and not immune to parallel session attacks. Meanwhile, He et al. [21] also indicated Das's protocol easily suffers from impersonation and privileged-insider attacks.

In the following years, several works focus on solving the mutual authentication with session key negotiation in WSNs. In 2011, Yeh et al. [22] indicated several security defects existing in Das's protocol, and proposed a strengthened authentication scheme with SK negotiation by applying Elliptic Curve Cryptosystem. Nevertheless, several researchers claimed that Yeh et al.'s scheme is time consuming due to scalar multiplications on elliptic curve, and still more likely to incur several attacks. In 2012, several security vulnerabilities such as lack of key negotiation, captured sensor node impersonation (CSNI) attacks, and stolen/lost smart card (SLSC) attacks in both Das's protocol and its derivatives [19, 20] were indicated by Vaidya et al. [23]. In 2013, Xue et al. [24] put forward a scheme, in which user, GWNs and SNs can mutually authenticate each other by applying temporary credential, hash and XOR operations. The scheme has more security features and high security level with little increase in computation and communication costs, and storage capacity. In 2014, Kim et al. [25] claimed Vaidya et al.'s protocol is more likely to suffer from impersonation attacks and GWNs bypassing attacks. He also proposed an enhanced lightweight authentication scheme with SK negotiation. In 2015, Chang et al. [26] indicated Kim et al.'s protocol is likely to suffer from man-in-the-middle (MITM) attacks, CSNI attacks, SLSC attacks, user's privacy leakage, and SK violation attacks. To eradicate these security pitfalls, He also devised an improved authentication scheme with SK negotiation by using dynamic identity. In 2016, Park et al. [27] indicated Chang et al.'s protocol is still vulnerable to offline password guessing (OPG) attacks, secure issues in perfect forward, and incorrectness of password change, and also designed an enhanced authentication scheme with SK negotiation. In 2017, Jung et al. [28] claimed Chang et al.'s scheme is vulnerable to OPG attacks, user impersonation attacks, SK compromising attacks. Moreover, the scheme has no SK verification, and has high load on GWNs. To eliminate these security vulnerabilities or defects, He also designed an improved authentication scheme with SK negotiation for WSNs environments.

In recent years, more attentions are paid to this subject. In 2018, Amin et al. [29] proposed a MBS-UAKA protocol for WSNs with multiple base station. The user authentication with key agreement achieves secure communication and authentication. In 2019, Soni et al. [30] designed an improved scheme, which efficiently eliminates both active and passive attacks, used for patient monitoring WSNs. In 2020, Ali et al. [31] proposed a robust scheme for secure communications of WSNs-based healthcare system, which achieves authentication and access control. In 2021, Wu et al. [32] put forward a fresh three-factor authentication scheme for WSNs. In 2022, Dai et al. [33] also put forward a three-factor authentication scheme based on ECC technique for multi-gateway WSNs. In spite of great improvements, these schemes are still insufficiently secure or efficient, and some fail in achieving user anonymity and untraceability. Some cannot achieve lightweight because of their heavy computation costs and communication costs. More seriously, some are vulnerable to SK leakage attacks. Moreover, the biometrics of the same person may slightly vary occasionally [15], therefore, high rejection probability occurs if conventional bio-hashing technique is applied in the design of authentication protocols. In addition, biometric data is vulnerable to a variety of noise in the phase of data acquisition. Worse yet, the regeneration of user's real biometrics may succeed in cheating GWNs or SNs in common practice.

To overcome these defects in biometric data acquisition, we resort to fuzzy extractor method [17, 34–36] to generate a random string with uniform distribution and a public parameter according to its input biometrics within a given error tolerance. Therefore, we are greatly inspired to design the eSAS2KN by utilizing the fuzzy extractor, smartcard, lightweight operations such as hash function, and bitwise exclusive-or computation for multi-gateway WSNs. The novelty of eSAS2KN lies in four folds. 1) Mutual authentications are achieved between any two of a user, a GWN, and a SN, and an SK between a user and the specific SN is negotiated for their timely, direct and subsequent secure communications; 2) User's anonymity and privacy protection are achieved; 3) Session key establishment and verification are incorporated into the authentication phases, which strengthens the communication confidentiality between user and the specific SN, and 4) The user's login-rejection probability is decreased in the login phase.

3. Network Model and Threat Model

For clearly elaborate the proposed eSAS2KN, we first present network model, and then the threat model in this section.

3.1 Network Model

In the eSAS2KN, there are K gateway nodes (GWNs), each of which severs J sensor nodes (SNs) scattered in the vicinity of GWN_k, as shown in **Fig. 1**. The resource-constrained SNs in the specific area are used to harvest the desired environment information. Each GWN is deployed in the center of J SNs to aggregate the collected environment information and forward it to the specific user U_i. The U_i in the vicinity of GWN_k and the sensor node S_j can obtain the environment information from the specific GWN_k or directly from the specific S_j for real-time applications. In addition, all the users, GWNs and SNs are synchronized with the same clock.

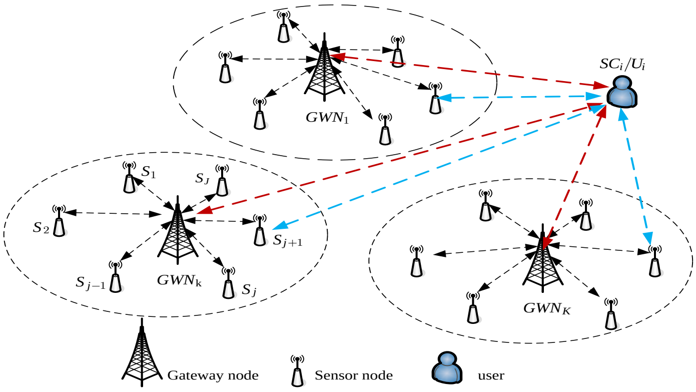


Fig. 1. Network model of eSAS2KN

3.2 Threat Model

In eSAS2KN, all the SNs are assumed to be untrustworthy, however, all the GWNs are trustworthy and cannot be compromised. The DY model [37] is employed to evaluate the security of eSAS2KN. Under this threat model, any node can launch communications with the other via public (insecure) channels. Any adversary has the ability to intercept the transmitted messages, alter or even delete the message contents, and inject bogus messages via public channels. Moreover, the secrets stored in the memory of a legitimate user's SC can be extracted through power analysis [38–41]. However, the secrets stored in the memory of any GWN cannot be extracted through power analysis.

In the eSAS2KN, CK model is also utilized to analyze and evaluate the security of key-exchange protocols [42,43]. Under this threat model, any adversary has the ability to send messages, compromise secrets including the SK, secret key, and session state. As a result, temporary session secrets, SK and long-term private keys may be leaked in the phase of key exchange, which will directly threaten the other previous and/or future session keys [44].

4. The eSAS2KN Scheme

In the eSAS2KN, mutual authentication is achieved among U_i, GWN_k and S_j . In addition, an SK is negotiated between U_i and S_j under the coordination of GWN_k after mutual authentication. With the computed SK, U_i and S_j can achieve timely, direct and secure communications without the $GWN_k's$ intervention. The main notations and their corresponding descriptions are listed in **Table 1**.

Table 1. Notations and corresponding description

Notations	Descriptions
U_i, S_j, GWN_k, SC_i	The i^{th} user, j^{th} sensor node, k^{th} gateway node, and i^{th} smart card, respectively
ID_i, PW_i, BIO_i	The i^{th} user's identity, password and biometric template, respectively
SID_j, IDS_i	Identities of S_j and SC_i, respectively
SK_i, SK_j	Session keys computed at U_i and S_j ends, respectively
K	Long time secret known only to GWN_k
$h(\cdot)$, $\oplus$, $\|\|$	Hash function, XOR, and concatenation operations, respectively
$HD(BIO_i, BIO_i^*)$	Hamming distance between BIO_i and BIO_i^*
$Gen(\cdot)$	Probabilistic generation function of fuzzy extractor
$Rep(\cdot)$	Deterministic reproduction function of fuzzy extractor
σ_i, τ_i	Biometric secret key and public reproduction parameter, respectively
ϵ	Error tolerance threshold used in $Rep(\cdot)$
ΔT	The maximum transmission delay
$T_i^{(\#)}$, $T_{G_k}^{(\#)}$, $T_j^{(\#)}$	Current timestamp of U_i, GWN_k and S_j, respectively

4.1 Registration Phase

The registration phase of eSAS2KN consists of two parts. One is for user registration, the other for sensor node registration, which is shown in **Fig. 2**.

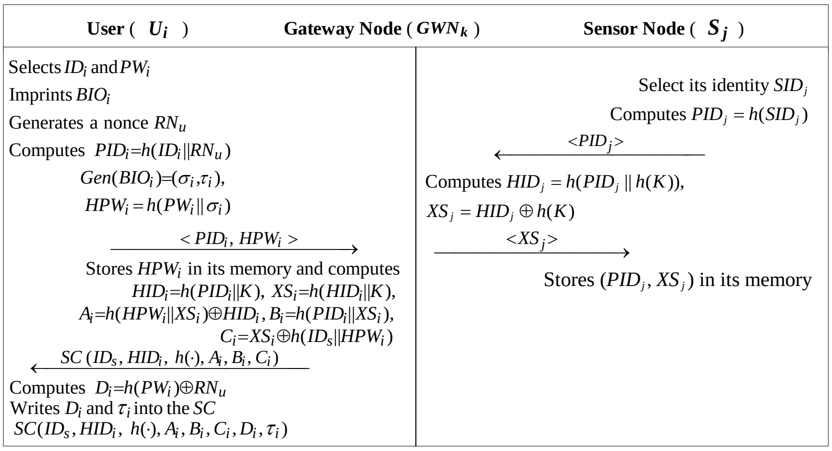

User (U_i)
Gateway Node (GWN_k)
Sensor Node (S_j)
Selects ID_i and PW_i
Imprints BIO_i
Generates a nonce RN_u
Computes $PID_i = h(ID_i \| RN_u)$
$Gen(BIO_i) = (\sigma_i, \tau_i)$,
$HPW_i = h(PW_i \| \sigma_i)$
$< PID_i, HPW_i >$
Stores HPW_i in its memory and computes
$HID_i = h(PID_i \| K)$, $XS_i = h(HID_i \| K)$,
$A_i = h(HPW_i \| XS_i) \oplus HID_i$, $B_i = h(PID_i \| XS_i)$,
$C_i = XS_i \oplus h(ID_S \| HPW_i)$
$SC\ (ID_S, HID_i,\ h(\cdot), A_i, B_i, C_i)$
Computes $D_i = h(PW_i) \oplus RN_u$
Writes D_i and τ_i into the SC
$SC(ID_S, HID_i,\ h(\cdot), A_i, B_i, C_i, D_i, \tau_i)$
Select its identity SID_j
Computes $PID_j = h(SID_j)$
$<PID_j>$
Computes $HID_j = h(PID_j \| h(K))$,
$XS_j = HID_j \oplus h(K)$
$<XS_j>$
Stores (PID_j, XS_j) in its memory

Fig. 2. Registration process of eSAS2KN.

4.1.1 User Registration

For registration, U_i transmits a request message to the nearest GWN_k through a secure channel. Succeeded in verifying the $U_i's$ identity, the GWN_k issues an SC_i to U_i. The user registration process is described in the following steps, which are illustrated in the left part of **Fig. 2**.

UR1: U_i selects his ID_i and PW_i, then imprints his BIO_i on a biometric sensor.
UR2: U_i generates a nonce RN_i, then computes $PID_i = h(ID_i \parallel RN_i)$.
UR3: U_i computes $Gen(BIO_i) = (\sigma_i, \tau_i)$, $HPW_i = h(PW_i \parallel \sigma_i)$, and then sends the message $< PID_i, HPW_i >$ to the GWN_k for registration.
UR4: The GWN_k stores HPW_i in its memory and calculates $HID_i = h(PID_i \| K)$, $XS_i = h(HID_i \| K)$, $A_i = h(HPW_i \| XS_i) \oplus HID_i$, $B_i = h(PID_i \| XS_i)$ and $C_i = XS_i \oplus h(IDS_i \| HPW_i)$, and then writes $(IDS_i, HID_i, h(\cdot), A_i, B_i, C_i)$ to the memory of SC_i, and finally issues SC_i to U_i via a secure channel.
UR5: U_i calculates $D_i = h(PW_i) \oplus RN_i$, then writes D_i and τ_i into the SC_i.

4.1.2 Sensor Node Registration

For sensor node registration, S_j first sends its hashed identity PID_j to the nearest GWN_k. On receiving the registration request from S_j, the GWN_k stores PID_j in its memory, calculates HID_j, A_j and B_j, and then sends them to S_j in secure means. After receiving the parameters HID_j, A_j and B_j, the S_j writes them to its memory. The detailed registration phase of sensor node is presented in the following steps, which are illustrated in the right part of **Fig. 2**.
SR1: S_j chooses SID_j as its identity, generates a random nonce RN_j, and calculates $PID_j = h(SID_j \| RN_j)$, then sends PID_j to GWN_k for registration through a secure channel.
SR2: GWN_k stores PID_j in its memory, calculates $HID_j = h(PID_j \| K)$, $XS_j = h(HID_j \| K)$, $A_j = h(XS_j \| RN_k) \oplus HID_j$, $B_j = XS_j \oplus h(PID_j)$, and then sends $< HID_j, A_j, B_j >$ to S_j through a secure channel.
SR3: S_j stores the parameters (HID_j, A_j, B_j) in its memory.

4.2 Login and Authentication Phase

This phase is used to achieve mutual authentication and session key negotiation between U_i and S_j with the coordination of GWN_k. The detailed process of login, authentication, and session key negotiation is illustrated in the following steps, which are shown in **Fig. 3**.
LA1: U_i puts SC_i on a card reader, types in its ID_i and PW_i, then imprints his BIO_i^* through a biometric sensor. (BIO_i^* means that the input biometrics are slightly different from the original BIO_i)
LA2: SC_i computes $\sigma_i^* = Rep(BIO_i^*, \tau_i)$ provided that $HD(BIO_i, BIO_i^*) < t$, $HPW_i^* = h(PW_i \| \sigma_i^*)$, $RN_i = D_i \oplus h(PW_i)$, $PID_i = h(ID_i \| RN_i)$, $XS_i^* = C_i \oplus h(IDS_i \| HPW_i^*)$ and $B_i^* = h(PID_i \| XS_i^*)$.
LA3: SC_i verifies $B_i^* = ?B_i$. If not satisfied, SC_i terminates the login, authentication, and session key phase; otherwise, SC_i calculates $k_i = h(XS_i \| T_i^{(1)})$, $DID_i = h(HPW_i \| XS_i) \oplus k_i$, and $M_{U_i,G_k} = h(A_i \| XS_i \| T_i^{(1)})$, where G_k refers to GWN_k for short.
LA4: U_i transmits the message $Msg1 : < DID_i, HID_i, M_{U_i,G_k}, T_i^{(1)} >$ to the GWN_k as a login request.

LA5: Upon receiving the login request, GWN_k checks $| T_{G_k}^{(1)} - T_i^{(1)} | \leq \Delta T$. If not satisfied, the login, authentication, and session key negotiation process stop; otherwise, goes to the next step.

LA6: GWN_k calculates $XS_i = h(HID_i \| K)$, $k_i = h(XS_i \| T_i^{(1)})$, $A_i = DID_i \oplus k_i \oplus HID_i$, and $M_{U_i,G_k}^{*} = h(A_i \| XS_i \| T_i^{(1)})$, then verifies $M_{U_i,G_k}^{*} =? M_{U_i,G_k}$. If does not hold, the login, authentication, and session key negotiation process stop; otherwise, goes to the next step.

LA7: GWN_k generates a nonce RN_k , gets PID_j and K from its memory, and then computes $HID_j = h(PID_j \| K)$, $XS_j = h(HID_j \oplus K)$, $M_j = h(XS_j \| RN_k) \oplus HID_i$, $N_j = h(PID_j \| XS_j) \oplus HID_j$, and $M_{G_k,S_j} = h(HID_i \| M_j \| N_j \| T_{G_k}^{(2)})$, and finally transmits the message $Msg2 :< HID_i, PID_j, M_{G_k,S_j}, T_{G_k}^{(2)} >$ to S_j .

LA8: On receiving $Msg2$ from GWN_k , S_j checks $| T_j^{(1)} - T_{Gk}^{(2)} | \leq \Delta T$. If not satisfied, the login and authentication process stops; otherwise, S_j computes $M_j = A_j \oplus HID_j \oplus HID_i$, $XS_j = B_j \oplus h(PID_j)$, $N_j = h(PID_j \| XS_j) \oplus HID_j$, and $M_{G_k,S_j}^{*} = h(HID_i \| M_j \| N_j \| T_{G_k}^{(2)})$.

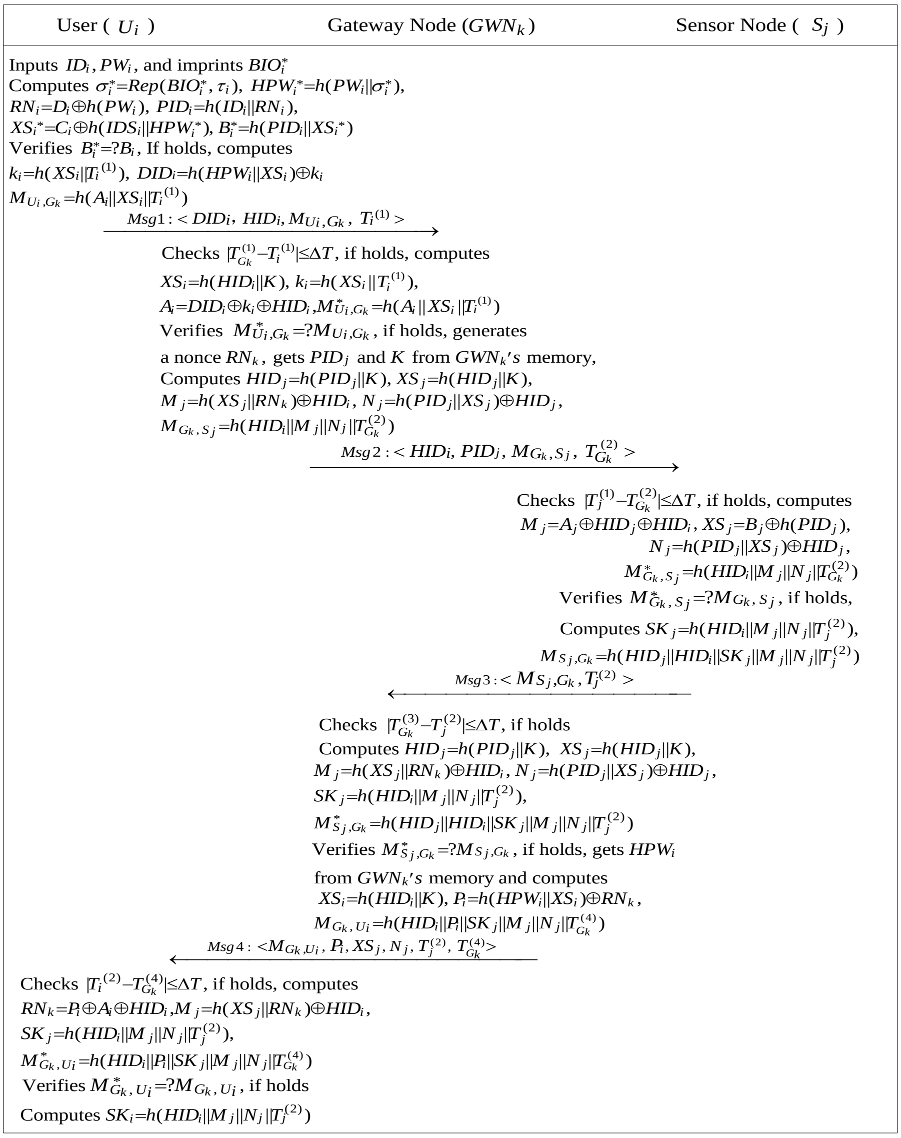
LA9: S_j verifies $M_{G_k,S_j}^{*} =? M_{G_k,S_j}$. If not satisfied, the login and authentication process stops; otherwise, S_j calculates $SK_j = h(HID_i \| M_j \| N_j \| T_j^{(2)})$, and $M_{S_j,G_k} = h(HID_j \| HID_i \| SK_j \| M_j \| N_j \| T_j^{(2)})$, then transmits the message $Msg3 :< M_{S_j,G_k}, HID_i, T_j^{(2)} >$ to GWN_k .

LA10: On receiving $Msg3$, GWN_k checks $| T_{G_k}^{(3)} - T_j^{(2)} | \leq \Delta T$. If not satisfied, the login and authentication process stops; otherwise, GWN_k calculates $HID_j = h(PID_j \| K)$, $XS_j = h(HID_j \| K)$, $M_j = h(XS_j \| RN_k) \oplus HID_i$, $N_j = h(PID_j \| XS_j) \oplus HID_j$, $SK_j = h(HID_i \| M_j \| N_j \| T_j^{(2)})$, and $M_{S_j,G_k}^{*} = h(HID_j \| HID_i \| SK_j \| M_j \| N_j \| T_j^{(2)})$.

LA11: GWN_k verifies $M_{S_j,G_k}^{*} =? M_{S_j,G_k}$. If not satisfied, the authentication process stops; otherwise, GWN_k gets HPW_i and K from its memory and computes $XS_i = h(HID_i \| K)$, $P_i = h(HPW_i \| XS_i) \oplus RN_k$ and $M_{G_k,U_i} = h(HID_i \| P_i \| SK_j \| M_j \| N_j \| T_{G_k}^{(4)})$,then transmits the message $Msg4 :< M_{G_k,U_i}, HID_i, P_i, XS_j, N_j, T_j^{(2)}, T_{G_k}^{(4)} >$ to U_i .

LA12: On receiving $Msg4$, U_i checks $|T_i^{(2)} - T_{G_k}^{(4)}| \leq \Delta T$. If not satisfied, the login and authentication process stops; otherwise, U_i computes $RN_k = P_i \oplus A_i \oplus HID_i$, $M_j = h(XS_j \| RN_k) \oplus HID_i$, $SK_j = h(HID_i \| M_j \| N_j \| T_j^{(2)})$ and $M_{G_k,U_i}^{*} = h(HID_i \| P_i \| SK_j \| M_j \| N_j \| T_{G_k}^{(4)})$.

LA13: U_i verifies $M_{G_k,U_i}^{*} =? M_{G_k,U_i}$ .If not satisfied, the login and authentication process stop; otherwise, U_i computes $SK_i = h(HID_i \| M_j \| N_j \| T_j^{(2)})$. Now, mutual authentications among U_i , GWN_k and S_j are achieved, and session keys SK_j and SK_i are computed from S_j and U_i , respectively.



User (U_i) Gateway Node (GWN_k) Sensor Node (S_j)

Inputs ID_i, PW_i, and imprints BIO_i^*
Computes $\sigma_i^* = Rep(BIO_i^*, \tau_i)$, $HPW_i^* = h(PW_i || \sigma_i^*)$,
$RN_i = D_i \oplus h(PW_i)$, $PID_i = h(ID_i || RN_i)$,
$XS_i^* = C_i \oplus h(IDS_i || HPW_i^*)$, $B_i^* = h(PID_i || XS_i^*)$
Verifies $B_i^* =? B_i$, If holds, computes
$k_i = h(XS_i || T_i^{(1)})$, $DID_i = h(HPW_i || XS_i) \oplus k_i$
$M_{U_i,G_k} = h(A_i || XS_i || T_i^{(1)})$

$\xrightarrow{Msg1 : < DID_i,\ HID_i, M_{U_i,G_k},\ T_i^{(1)} >}$

Checks $|T_{G_k}^{(1)} - T_i^{(1)}| \le \Delta T$, if holds, computes
$XS_i = h(HID_i || K)$, $k_i = h(XS_i || T_i^{(1)})$,
$A_i = DID_i \oplus k_i \oplus HID_i$, $M_{U_i,G_k}^* = h(A_i || XS_i || T_i^{(1)})$
Verifies $M_{U_i,G_k}^* =? M_{U_i,G_k}$, if holds, generates
a nonce RN_k, gets PID_j and K from $GWN_k's$ memory,
Computes $HID_j = h(PID_j || K)$, $XS_j = h(HID_j || K)$,
$M_j = h(XS_j || RN_k) \oplus HID_i$, $N_j = h(PID_j || XS_j) \oplus HID_j$,
$M_{G_k,S_j} = h(HID_i || M_j || N_j || T_{G_k}^{(2)})$

$\xrightarrow{Msg2 : < HID_i,\ PID_j,\ M_{G_k,S_j},\ T_{G_k}^{(2)} >}$

Checks $|T_j^{(1)} - T_{G_k}^{(2)}| \le \Delta T$, if holds, computes
$M_j = A_j \oplus HID_j \oplus HID_i$, $XS_j = B_j \oplus h(PID_j)$,
$N_j = h(PID_j || XS_j) \oplus HID_j$,
$M_{G_k,S_j}^* = h(HID_i || M_j || N_j || T_{G_k}^{(2)})$
Verifies $M_{G_k,S_j}^* =? M_{G_k,S_j}$, if holds,
Computes $SK_j = h(HID_i || M_j || N_j || T_j^{(2)})$,
$M_{S_j,G_k} = h(HID_j || HID_i || SK_j || M_j || N_j || T_j^{(2)})$

$\xleftarrow{Msg3 : < M_{S_j,G_k}, T_j^{(2)} >}$

Checks $|T_{G_k}^{(3)} - T_j^{(2)}| \le \Delta T$, if holds
Computes $HID_j = h(PID_j || K)$, $XS_j = h(HID_j || K)$,
$M_j = h(XS_j || RN_k) \oplus HID_i$, $N_j = h(PID_j || XS_j) \oplus HID_j$,
$SK_j = h(HID_i || M_j || N_j || T_j^{(2)})$,
$M_{S_j,G_k}^* = h(HID_j || HID_i || SK_j || M_j || N_j || T_j^{(2)})$
Verifies $M_{S_j,G_k}^* =? M_{S_j,G_k}$, if holds, gets HPW_i
from $GWN_k's$ memory and computes
$XS_i = h(HID_i || K)$, $P_i = h(HPW_i || XS_i) \oplus RN_k$,
$M_{G_k,U_i} = h(HID_i || P_i || SK_j || M_j || N_j || T_{G_k}^{(4)})$

$\xleftarrow{Msg4 : < M_{G_k,U_i}, P_i, XS_j, N_j, T_j^{(2)}, T_{G_k}^{(4)} >}$

Checks $|T_i^{(2)} - T_{G_k}^{(4)}| \le \Delta T$, if holds, computes
$RN_k = P_i \oplus A_i \oplus HID_i$, $M_j = h(XS_j || RN_k) \oplus HID_i$,
$SK_j = h(HID_i || M_j || N_j || T_j^{(2)})$,
$M_{G_k,U_i}^* = h(HID_i || P_i || SK_j || M_j || N_j || T_{G_k}^{(4)})$
Verifies $M_{G_k,U_i}^* =? M_{G_k,U_i}$, if holds
Computes $SK_i = h(HID_i || M_j || N_j || T_j^{(2)})$

Fig. 3. Login and authentication process of eSAS2KN

4.3 Password and Biometrics Updating Phase

For security consideration, a legitimate user hopes to alter his/her current password to a new one, and/or to change the current memory-stored biometric template to a new one. To this end, the eSAS2KN provides password and biometrics updating function. The detailed password and biometrics updating process is described as the following steps, which are illustrated in **Fig. 4**.

User (U_i)	**Smartcard (SC_i)**
Inputs the current ID_i ,PW_i	
Imprints the current BIO_i^*	Computes $\sigma_i^*=Rep(BIO_i^*,\ \tau_i)$, $HPW_i^*=h(PW_i\,\|\,\sigma_i^*)$, $RN_i=D_i\oplus h(PW_i)$, $PID_i=h(ID_i\|RN_i)$, $XS_i^*=C_i\oplus h(IDS_i\|HPW_i^*)$, $B_i^*=h(PID_i\|XS_i^*)$ and verifies $B_i^*?=B_i$
Imprints the new BIO_i^{new}	
Inputs the new PW_i^{new}	Computes $\sigma_i^{new}=Rep(BIO_i^{new},\tau_i)$, $HPW_i^{new}=h(PW_i^{new}\|\,\sigma_i^{new})$, $RN_i^{new}=D_i\oplus h(PW_i^{new})$, $PID_i^{new}=h(ID_i\|RN_i^{new})$, $XS_i^{new}=C_i\oplus h(IDS_i\,\|HPW_i^{new})$, $A_i^{new}=h(HPW_i^{new}\|XS_i^{new})\oplus HID_i$, $B_i^{new}=h(PID_i^{new}\|XS_i^{new})$, $C_i^{new}=XS_i^{new}\oplus h(IDS_i\,\|HPW_i^{new})$, $D_i^{new}=h(PW_i^{new})\oplus RN_i^{new}$, $HID_i^{new}=A_i^{new}\oplus h(HPW_i^{new}\,\|XS_i^{new})$
Replaces A_i, B_i, C_i, D_i and HID_i with A_i^{new}, B_i^{new}, C_i^{new}, D_i^{new} and HID_i^{new}, respectively.	

Fig. 4. Password and biometrics updating process of eSAS2KN.

U1: U_i puts SC_i on a card reader, types in the currently used ID_i and PW_i, and imprints his BIO_i^*.

U2: SC_i computes $\sigma_i^*=Rep(BIO_i^*,\tau_i)$, $HPW_i^*=h(PW_i\,\|\,\sigma_i^*)$, $RN_i=D_i\oplus h(PW_i)$, $PID_i=h(ID_i\|RN_i)$, $XS_i^*=C_i\oplus h(IDS_i\|HPW_i^*)$ and $B_i^*=h(PID_i\,\|\,XS_i^*)$.

U3: SC_i verifies $B_i^*=?B_i$. If unsuccessful, the password and biometrics updating process stops; otherwise, the next step proceeds.

U4: U_i imprints its new biometrics BIO_i^{new} and/or input a new password PW_i^{new}, then SC_i computes $\sigma_i^{new}=Rep(BIO_i^{new},\tau_i)$, $HPW_i^{new}=h(PW_i^{new}\,\|\,\sigma_i^{new})$, $RN_i^{new}=D_i\oplus h(PW_i^{new})$, $PID_i^{new}=h(ID_i\,\|\,RN_i^{new})$, $XS_i^{new}=C_i\oplus h(IDS_i\,\|\,HPW_i^{new})$, $A_i^{new}=h(HPW_i^{new}\,\|\,XS_i^{new})\oplus HID_i$, $B_i^{new}=h(PID_i^{new}\|XS_i^{new})$, $C_i^{new}=XS_i^{new}\oplus h(IDS_i\|HPW_i^{new})$, $D_i^{new}=h(PW_i^{new})\oplus RN_i^{new}$, and $HID_i^{new}=A_i^{new}\oplus h(HPW_i^{new}\,\|\,XS_i^{new})$.

U5: SC_i/U_i replaces the current A_i, B_i, C_i, D_i and HID_i with the newly computed A_i^{new}, B_i^{new}, C_i^{new}, D_i^{new}, and HID_i^{new}, respectively. Finally, the parameters $(IDS_i, HID_i^{new}, h(\cdot), A_i^{new}, B_i^{new}, C_i^{new}, D_i^{new}, \tau_i)$ are stored in SC_i.

5. Security Analysis

To evaluate the scheme's security, we give detailed informal security analysis, formal security proof, as well as formal security verification of eSAS2KN.

5.1 Informal Security Analysis

User Anonymity: In eSAS2KN, an attacker $\mathcal{A}$ cannot obtain ID_i even though he/she obtains HID_i from the extracted parameters $(IDS_i, HID_i, h(\cdot), A_i, B_i, C_i, D_i, \tau_i)$ by power analysis attack [38-41] from the memory of $SLSC$ or from the intercepted messages $Msg1$, $Msg2$, $Msg3$ and/or $Msg4$. Since the secret K is only possessed by the trusted GWN_k, $\mathcal{A}$ cannot obtain it. Without K, $\mathcal{A}$ cannot guess PID_i from $HID_i=h(PID_i\|K)$. Without RN_i and PID_i, $\mathcal{A}$ cannot guess ID_i from $PID_i=h(ID_i\,\|\,RN_i)$. Therefore, the proposed eSAS2KN has the feature of user anonymity.

Man-in-the-Middle Attack (MITM)***:*** If $\mathcal{A}$ wants to launch a MITM attack to cheat GWN_k through forging/altering the transmitted $Msg1$, he/she must obtain a legitimate user's SC_i, ID_i, PW_i and BIO_i. Without BIO_i, $\mathcal{A}$ cannot compute $U_i's$ biometric secret σ_i according to $\sigma_i^*=Rep(BIO_i^*,\tau_i)$. Without $U_i's$ password PW_i and smartcard SC_i, $\mathcal{A}$ cannot compute the nonce RN_i according to $RN_i = D_i \oplus h(PW_i)$. Without password PW_i and biometric secret σ_i, $\mathcal{A}$ cannot compute $U_i' s$ hash password HPW_i^* according to $HPW_i^* = h(PW_i \| \sigma_i^*)$. Without HPW_i^* and SC_i, $\mathcal{A}$ cannot compute XS_i^* according to $XS_i^* = C_i \oplus h(IDS_i \| HPW_i^*)$. If $\mathcal{A}$ cannot obtain XS_i and HPW_i, he/she will not compute DID_i and M_{U_i,G_k} according to $DID_i = h(HPW_i \| S_i) \oplus h(XS_i \| T_i^{(1)})$ and $M_{U_i,G_k} = h(A_i \| XS_i \| T_i^{(1)})$, respectively. Therefore, $\mathcal{A}$ cannot forge $Msg1\text{:} < DID_i, HID_i, M_{U_i,G_k}, T_i^{(1)} >$ to cheat GWN_k.

If $\mathcal{A}$ wants to launch a MITM attack to cheat U_i through forging/altering the transmitted $Msg4$, he/she must obtain PID_j and secret K. However, all the $GWNs$ are trustworthy and cannot be compromised, so $\mathcal{A}$ cannot extract PID_j and K from the memory of GWN_k. Without PID_j and K, $\mathcal{A}$ cannot compute HID_j according to $HID_j = h(PID_j \| K)$ and XS_j according to, $XS_j = h(HID_j \| K)$ respectively. Without HID_j and XS_j, $\mathcal{A}$ cannot compute M_j and N_j according to $M_j = h(XS_j \| RN_k) \oplus HID_i$ and $N_j = h(PID_j \| XS_j) \oplus HID_j$, respectively. Without M_j and N_j, $\mathcal{A}$ cannot compute M_{G_k,U_i} according to $M_{G_k,U_i} = h(HID_i \| P_i \| SK_j \| M_j \| N_j \| T_{G_k}^{(4)})$, let alone to forge $Msg4 :< M_{G_k,U_i}, HID_i, P_i, XS_j, N_j, T_j^{(2)}, T_{G_k}^{(4)} >$ to cheat U_i.

If $\mathcal{A}$ wants to launch MITM attacks to cheat S_j through forging/altering the transmitted $Msg2$, he/she must obtain PID_j and the secret K. However, PID_j and K are stored in the memory of the trusted GWN_k .Without PID_j and K, $\mathcal{A}$ cannot computes M_j and N_j according to $M_j = h(XS_j \| RN_k) \oplus HID_i$ and $N_j = h(PID_j \| XS_j) \oplus HID_j$, respectively. Without M_j and N_j, $\mathcal{A}$ cannot compute $M_{G_k,S_j} = h(HID_i \| M_j \| N_j \| T_{G_k}^{(2)})$, let alone to forge $Msg2\text{:}< HID_i, PID_j, M_{G_k,S_j}, T_{G_k}^{(2)} >$ to cheat S_j.

If $\mathcal{A}$ wants to launch a MITM attack to cheat GWN_k through forging/altering the transmitted $Msg3$, he/she must obtain M_j and N_j to compute $SK_j = h(HID_i \| M_j \| N_j \| T_j^{(2)})$. However, $\mathcal{A}$ cannot obtain M_j and N_j, which has been analyzed above. Without SK_j, $\mathcal{A}$ cannot compute M_{S_j,G_k} according to $M_{S_j,G_k} = h(HID_j \| HID_i \| SK_j \| M_j \| N_j \| T_j^{(2)})$. Therefore, $\mathcal{A}$ cannot forge $Msg3 :< M_{S_j,G_k}, HID_i, T_j^{(2)} >$ to cheat GWN_k.

Based on the above analysis, it can be concluded that MITM attacks are resisted in eSAS2KN.

Mutual Authentication with Key Agreement: In the eSAS2KN, U_i is authenticated by GWN_k through verifying $Msg1\text{:} < DID_i, HID_i, M_{U_i,G_k}, T_i^{(1)} >$ by checking $M_{U_i,G_k}^* = ?M_{U_i,G_k}$. Similarly, S_j is authenticated by GWN_k through verifying $Msg3 :< M_{S_j,G_k}, HID_i, T_j^{(2)} >$ by checking $M_{S_j,G_k}^* = ?M_{S_j,G_k}$. In addition, GWN_k is authenticated by U_i through verifying $Msg4\text{:}< M_{G_k,U_i}, HID_i, P_i, XS_j, N_j, T_j^{(2)}, T_{G_k}^{(4)} >$ by checking $M_{G_k,U_i}^* = ?M_{G_k,U_i}$. Likewise, GWN_k is authenticated by S_j through verifying $Msg2\text{:} < HID_i, PID_j, M_{G_k,S_j}, T_{G_k}^{(2)} >$ by checking $M_{G_k,S_j}^* = ?M_{G_k,S_j}$. Moreover, SK_j and SK_i can be computed at S_j and U_i ends, respectively, for their subsequent secure communications. In view of the above analysis, it can be concluded that the proposed eSAS2KN has the feature of mutual authentication and achievement of SK negotiation.

Withstands Replay Attack: In the eSAS2KN, all the sent or received messages are labelled with the sender's current timestamps, such as $T_i^{(1)}$, $T_G^{(1)}$ and $T_j^{(1)}$. It is impossible for $\mathcal{A}$ to login to GWN_k, and to tamper the intercepted messages to cheat S_j or U_i as legal GWN_k, or to cheat GWN_k as a legal S_j or U_i. Therefore, the proposed eSAS2KN has the feature of withstanding replay attacks.

Withstands SLSC Attack: In the eSAS2KN, even if $\mathcal{A}$ succeeds in extracting the parameters $(IDS_i, HID_i, h(\cdot), A_i, B_i, C_i, D_i, \tau_i)$ stored in the stolen or lost SC_i through power analysis attack [38-41], he/she still has no chance to launch malicious attacks. If $\mathcal{A}$ wants to guess the user's ID_i according to $PID_i = h(ID_i \| RN_i)$, he/she must obtain the secret K and

$h(PW_i)$ in advance. If $\mathcal{A}$ has the secret K, he/she can guess PID_i according to $HID_i = h(PID_i \| K)$ with the extracted HID_i from the stolen or lost SC_i. If $\mathcal{A}$ has $h(PW_i)$, he/she can computes RN_i according to $RN_i = D_i \oplus h(PW_i)$ with the extracted D_i from the stolen or lost SC_i. However, due to the trustworthiness of GWN_k, $\mathcal{A}$ cannot obtain the secret K. Without knowing the random nonce RN_i, $\mathcal{A}$ cannot compute $h(PW_i)$ according to $h(PW_i) = D_i \oplus RN_i$. Therefore, $\mathcal{A}$ cannot guess the user's ID_i according to $PID_i = h(ID_i \| RN_i)$. Not knowing ID_i, $\mathcal{A}$ cannot imitate a legitimate user to cheat S_j or GWN_k. Therefore, the proposed eSAS2KN has the feature of withstanding SLSC attacks.

Withstands OPG Attack: In the eSAS2KN, if $\mathcal{A}$ hopes to guess offline the correct PW_i according to $HPW_i = h(PW_i \| \sigma_i)$, he/she has to obtain HPW_i and a legitimate user's biometric secret key σ_i. However, $\mathcal{A}$ cannot obtain HPW_i and σ_i. The HPW_i cannot be extracted because it is stored in the memory of GWN_k, which is a trustworthy node. In addition, without a legitimate user's BIO_i, $\mathcal{A}$ will unachievably compute σ_i according to $Gen(BIO_i) = (\sigma_i, \tau_i)$. Therefore, the proposed eSAS2KN has the feature of withstanding OPG attacks.

Withstands CSNI Attack: In the eSAS2KN, if S_j is compromised by a malicious $\mathcal{A}$, the parameters (HID_j, A_j, B_j) stored in $S_j{}'s$ memory are all known to $\mathcal{A}$. However, $\mathcal{A}$ cannot compute the secret information of both U_i and GWN_k. Suppose that $SK_i = h(HID_i \| M_j \| N_j \| T_j^{(2)})$ is computed by U_i in the current session. However, the old or the future session keys will not be known to $\mathcal{A}$. When the comprised S_j receives the request $<HID_i, PID_j, M_{G_k,S_j}, T_{G_k}^{(2)}>$ from GWN_k, $\mathcal{A}$ can retrieve M_j, XS_j, N_j. However, these parameters have no relation to U_i. Therefore, $\mathcal{A}$ will not be able to impersonate U_i in the future. Although $\mathcal{A}$ can successfully retrieve XS_j from $GWN_k{}'s$ request to the compromised S_j, $\mathcal{A}$ cannot derive the $GWN_k{}'s$ secret K according to $XS_j = h(HID_j \| K)$. Without K, $\mathcal{A}$ will not be able to impersonate the GWN_k. Therefore, the proposed eSAS2KN has the feature of withstanding $CSNI$ attacks.

Withstands GWNs Bypassing Attack: In the eSAS2KN, even if $\mathcal{A}$ captures a legal user's SC_i and succeeds in extracting the parameters $(IDS_i, HID_i, h(\cdot), A_i, B_i, C_i, D_i, \tau_i)$ stored in the memory of SC_i, he/she cannot cheat S_j by impersonating GWN_k. Since $\mathcal{A}$ cannot obtain PID_j and the secret K, which is owned only by the trusted GWN_k, he/she cannot compute HID_j according to $HID_j = h(PID_j \| K)$, let alone XS_j. Without XS_j, $\mathcal{A}$ cannot compute M_j and N_j. Without M_j and N_j, $\mathcal{A}$ will fail to compute M_{G_k,S_j} according to $M_{G_k,S_j} = h(HID_i || M_j \| N_j \| T_{G_k}^{(2)})$, let alone forges a valid message $<HID_i, PID_j, M_{G_k,S_j}, T_G^{(2)}>$ to cheat S_j. Therefore, the proposed eSAS2KN has the feature of withstanding GWNs bypassing attacks.

Provides Password Verification Process: In the eSAS2KN, given the possibility of incorrect password input, we adopt password verification process by verifying $B_i^* = ?B_i$ at the beginning of login process. In addition, given the fact that a person's biometrics may be slightly different from the original one once in a while [15], therefore, we resort to fuzzy extractor technique instead of conventional bio-hashing techniques to decrease $U_i{}'s$ high rejection rate in the login phase of eSAS2KN. Therefore, the two methods contribute greatly to the robustness of eSAS2KN.

Provides Session Key Verification Process: In the eSAS2KN, S_j computes M_{S_j,G_k} according to $M_{S_j,G_k} = h(HID_j \| HID_i \| SK_j \| M_j \| N_j \| T_j^{(2)})$ with the computed session key SK_j, and then transmits $Msg3$: $<M_{S_j,G_k}, HID_i, T_j^{(2)}>$ to GWN_k. Since the session key SK_j is concatenated in M_{S_j,G_k} of $Msg3$, from the perspective of GWN_k, the verification of SK_j and authentication of S_j are both achieved by verifying $M_{S_j,G_k}^* = ?M_{S_j,G_k}$ .In the same way,

GWN_k computes $M_{G_k,U_i}=h(HID_i\|P_i\ \|SK_j\|M_j\|N_j\|T_{G_k}^{(4)})$ with the computed session key SK_j , and then transmits $Msg4{:}<M_{G_k,U_i}, HID_i,\ P_i, XS_j, N_j, T_j^{(2)}, T_{G_k}^{(4)}>$ to U_i . Since SK_j is also concatenated in M_{G_k,U_i} , from the perspective of U_i , the verification of SK_j and authentication of GWN_k are both achieved by verifying $M_{G_k,U_i}^{*}\ ?= M_{G_k,U_i}$. Therefore, the eSAS2KN has the security feature of providing SK verification.

Withstands Privileged-insider Attack: In the eSAS2KN, ID_i and PW_i sent to GWN_k for registration are both in encrypted forms with $PID_i=h(ID_i\|RN_i)$ and $HPW_i=h(PW_i\|\sigma_i)$, respectively. A privileged insider attacker cannot identify the registered user's ID_i and PW_i . Therefore, the proposed eSAS2KN has the feature of withstanding privileged-insider attacks.

Provides Session Key Security: In the eSAS2KN, $\mathcal{A}$ cannot calculate SK_i and SK_j . To calculate SK_i/SK_j according to $SK_i / SK_j = h(HID_i \| M_j \| N_j \| T_j^{(2)})$, $\mathcal{A}$ must compute M_j and N_j . However, RN_k is random nonce generated by GWN_k , which is known only to the trusted GWN_k , $\mathcal{A}$ cannot obtain it. Moreover, due to the trustworthiness of GWN_k , $\mathcal{A}$ cannot extract PID_j from the memory of GWN_k . Therefore, without RN_k and PID_j , $\mathcal{A}$ cannot compute M_j and N_j according to $M_j=h(XS_j\|RN_k)\oplus HID_i$ and $N_j=h(PID_j\|XS_j)\oplus HID_j$, respectively. Therefore, $\mathcal{A}$ cannot compute SK_i/SK_j from both U_i and S_j ends. This means that the proposed eSAS2KN has the security feature of providing SK security.

5.2 Formal Security Proof and Verification

In the following subsections, BAN logic-based verification, ROR model-based formal security proof, and AVISPA-based formal security verification are presented in detail to show the security of the proposed eSAS2KN.

5.2.1 Logical Verification

In this subsection, the well-known BAN logic, regarded as a distinguished tool to give logical verification of cryptographic protocols, is used to prove the legitimacy of SK, which is computed at both U_i and S_j ends.

Basic Notations

- $U \lhd C$: C is received by U .
- $U \mid\equiv C$: C is believed by U .
- $\#(C)$: C is fresh.
- $U \mid\sim C$: C is once sent by U .
- $U \overset{K}{\longleftrightarrow} S$: U and S share the secret key K .
- $U \mid\Rightarrow C$: U has jurisdiction over C .
- (C, M) : C or M is a part of (C, M) .
- $\{C\}_K$: C is encrypted with K .

Logic Rules

- **R1** (Message-meaning rule): $\frac{U\mid\equiv U\overset{K}{\longleftrightarrow}S,\ U\lhd\{C\}_K}{U\mid\equiv S\mid\sim C}$, If U believes he shares a secret key K with S , and receives an encrypted C with K , then he believes C is once sent by S .
- **R2** (Nonce-verification rule): $\frac{U\mid\equiv\#(C), U\mid\equiv S\mid\sim C}{U\mid\equiv S\mid\equiv C}$, If U believes C is fresh, and believes S once sent C , then U believes S believe C .
- **R3** (Believe rule) : $\frac{U\mid\equiv C,\ U\mid\equiv M}{U\mid\equiv(C,M)}$, if U believes C and M , then he believes (C,M).
- **R4** (Freshness rule): $\frac{U\mid\equiv\#(C)}{U\mid\equiv\#(C,M)}$, if U believes C is fresh, then he believes the freshness of

$(C,\, M)$.

Goals

In the following logic verification process, we aim at achieving the following four goals.

- **Goal** 1: $U_i|\equiv(U_i \overset{SK_i/SK_j}{\longleftrightarrow} S_j)$
- **Goal** 2: $S_j|\equiv(U_i \overset{SK_i/SK_j}{\longleftrightarrow} S_j)$
- **Goal** 3: $U_i|\equiv S_j|\equiv(U_i \overset{SK_i/SK_j}{\longleftrightarrow} S_j)$
- **Goal** 4: $S_j|\equiv U_i|\equiv(U_i \overset{SK_i/SK_j}{\longleftrightarrow} S_j)$

Assumptions

To well analyze the eSAS2KN, the listed assumptions are considered in the proof process.

- **A1**: $GWN_k|\equiv\#(T_i^{(1)})$
- **A2**: $S_j|\equiv\#(T_{G_k}^{(2)})$
- **A3**: $GWN_k|\equiv\#(T_j^{(2)})$
- **A4**: $U_i|\equiv\#(T_{G_k}^{(4)})$
- **A5**: $GWN_k|\equiv(GWN_k \overset{PID_j}{\longleftrightarrow} S_j)$
- **A6:** $S_j|\equiv(S_j \overset{PID_j}{\longleftrightarrow} GWN_k)$
- **A7**: $U_i|\equiv(U_i \overset{HPW_i}{\longleftrightarrow} GWN_k)$
- **A8**: $GWN_k|\equiv(GWN_k \overset{HPW_i}{\longleftrightarrow} U_i)$
- **A9**: $U_i|\equiv S_j|\Rightarrow(U_i \overset{SK_i/SK_j}{\longleftrightarrow} S_j)$
- **A10**: $S_j|\equiv U_i|\Rightarrow(U_i \overset{SK_i/SK_j}{\longleftrightarrow} S_j)$

Ideal Forms Conversion

Before logic verification, all transmitted messages are transformed to ideal forms as follows.

- **Msg1**: $U_i \to GWN_k\ :\{HID_i,\, K,\, T_i^{(1)}\}_{HPW_i}$
- **Msg2**: $GWN_k \to S_j\ :\{HID_i,\, M_j,\, N_j,\, T_{G_k}^{(2)}\}_{PID_j}$
- **Msg3**: $S_j \to GWN_k\ :\{HID_i,\, K,\, M_j,\, N_j,\, T_j^{(2)}\}_{PID_j}$
- **Msg4**: $GWN_k \to U_i\ :\{HID_i,\, M_j,\, N_j,\, T_j^{(2)},\, K,\, T_{G_k}^{(4)}\}_{HPW_i}$

Logical Verification of eSAS2KN

To well describe the verification process, predefined information, including four rules, ten assumptions and four messages are used as follows.

- According to $Mgs1$, V1 is derived as **V1**: $GWN_k \lhd \{HID_i,\, K,\, T_i^{(1)}\}_{HPW_i}$
- According to A8 and R1, V2 is derived as **V2**: $GWN_k\ |\equiv\ U_i\ |\sim\ (HID_i,\, K,\, T_i^{(1)})$
- According to A1 and R3, V3 is derived as **V3**: $GWN_k\ |\equiv\ \#(HID_i,\, K,\, T_i^{(1)})$
- According to V2, V3 and R2, V4 is derived as **V4**: $GWN_k\ |\equiv U_i\ |\equiv\ (HID_i,\, K,\, T_i^{(1)})$
- According to $Msg2$, V5 is derived as **V5**: $S_j\ \lhd\ \{HID_i,\, M_j,\, N_j,\, T_{G_k}^{(2)}\}_{PID_j}$
- According to A6 and R1, V6 is derived as **V6**: $S_j\ |\equiv\ GWN_k\ |\sim\ (HID_i,\, M_j,\, N_j,\, T_{G_k}^{(2)})$
- According to A2 and R3, V7 is derived as **V7**: $S_j\ |\equiv\ \#(HID_i,\, M_j,\, N_j,\, T_{G_k}^{(2)})$
- According to V6, V7 and R2, V8 is derived as **V8**: $S_j|\equiv GWN_k|\equiv(HID_i,\, M_j,\, N_j,\, T_{G_k}^{(2)})$
- According to $Msg3$, V9 is derived as **V9**: $GWN_k\ \lhd\ \{HID_i,\, K,\, M_j,\, N_j,\, T_j^{(2)}\}_{PID_j}$
- According to A5 and R1, V10 is derived as **V10**: $GWN_k|\equiv S_j|\sim(HID_i,\, K,\, M_j,\, N_j,\, T_j^{(2)})$
- According to A3 and R3, V11 is derived as **V11**: $GWN_k\ |\equiv\ \#(HID_i,\, K,\, M_j,\, N_j,\, T_j^{(2)})$

- According to V10, V11 and R2, V12 is derived as
  V12: $GWN_k \mid\equiv S_j \mid\equiv (HID_i, K, M_j, N_j, T_j^{(2)})$
- According to $Msg4$, V13 is derived as **V13**: $U_i \lhd \{HID_i, M_j, N_j, K, T_j^{(2)}, T_{G_k}^{(4)}\}_{HPW_i}$
- According to A7 and R1, V14 is derived as **V14**: $U_i \mid\equiv GWN_k \mid\sim (HID_i, M_j, N_j, K, T_j^{(2)}, T_{G_k}^{(4)})$
- According to A4 and R3, V15 is derived as **V15**: $U_i \mid\equiv \#(HID_i, M_j, N_j, K, T_j^{(2)}, T_{G_k}^{(4)})$
- According to V14, V15 and R2, V16 is derived as follows.
  V16: $U_i \mid\equiv GWN_k \mid\equiv (HID_i, M_j, N_j, K, T_j^{(2)})$
- According to V12, V16 and $SK_i / SK_j = h(HID_i || M_j || N_j || T_j^{(2)})$, V17 is derived as follows.
  V17: $U_i \mid\equiv S_j \mid\equiv (U_i \xleftrightarrow{SK_i / SK_j} S_j)$ $(Goal_3)$
- According to V8, V4 and $SK_i / SK_j = h(HID_i \parallel M_j \parallel N_j \parallel T_j^{(2)})$, V18 is derived as follows.
  V18: $S_j \mid\equiv U_i \mid\equiv (U_i \xleftrightarrow{SK_i / SK_j} S_j)$ $(Goal_4)$
- According to A9, V17 and R4, V19 is derived as follows.
  V19: $U_i \mid\equiv (U_i \xleftrightarrow{SK_i / SK_j} S_j)$ $(Goal_1)$
- According to A10, V18 and R4, V20 is derived as follows.
  V20: $S_j \mid\equiv (U_i \xleftrightarrow{SK_i / SK_j} S_j)$ $(Goal_2)$

It can be clearly drawn from the above demonstrations that U_i, GWN_k and S_j are mutually authenticated, and SK is negotiated and shared between U_i and S_j.

5.2.2 Formal Security Proof

To demonstrate the SK security of eSAS2KN, we implement formal security proof by applying Real-Or-Random (ROR) model. The definition of ROR model is presented as follows.

Participants: Let $\Pi_{U_i}^{u}$, $\Pi_{S_j}^{s}$ and $\Pi_{GWN_k}^{g}$ be the instances of U_i, S_j and GWN_k, respectively.

Acceptedstate: On receiving the last anticipated message, instance Π^t transits to the accepted state. When all the messages received and sent by Π^t are concatenated in order, it will represent session identification for the current session.

Partnering: Instances Π^{t1} and Π^{t2} are deemed as partner to each other if 1) Π^{t1} and Π^{t2} are both in accepted state; 2) Π^{t1} and Π^{t2} are mutually authenticated by each other and share the same session identification; and 3) Π^{t1} and Π^{t2} are mutual partners of each other.

Freshness: $\Pi_{U_i}^{u}$ and $\Pi_{S_j}^{s}$ are deemed as freshness if SK computed from U_i and S_j ends is kept from disclosure to $\mathcal{A}$ by using the Reveal queries $Reveal(\Pi^u)$ and $Reveal(\Pi^s)$.

Adversary: In ROR model, DY threat model is employed for formal security proof. $\mathcal{A}$ has the ability to full control over all the communications. That is to say, $\mathcal{A}$ can intercept, alter, delete and inject forged information through the following queries.

$Execute(\Pi^u / \Pi^s, \Pi^g)$ is used to model a passive intercepting attack, in which $\mathcal{A}$ can read the transmitted messages between the legitimate U_i / S_j and GWN_k.

$Reveal(\Pi^u)$ is used to model the current SK generated by Π^u and its partner Π^s is disclosed to $\mathcal{A}$. If $\mathcal{A}$ cannot reveal SK between Π^u and Π^s using the query $Reveal(\Pi^u)$, then SK is secure.

$Send(\Pi^u / \Pi^s / \Pi^g, Msg)$ is used to models an active attack, in which Msg can be sent to the participant $\Pi^u / \Pi^s / \Pi^g$ by $\mathcal{A}$.

$CorruptSN(\Pi_{S_j}^{s})$ is used to model an active captured sensor node attack, in which the secret credentials, HID_j, A_j and B_j stored in the memory of S_j, are known to $\mathcal{A}$.

$CorruptSC(\Pi^u)$ is used to model an active stolen or lost smartcard attack, in which the information $(IDS_i,\ HID_i,\ h(\cdot),\ A_i,\ B_i,\ C_i,\ D_i,\ \tau_i)$ stored in the $SC_i's$ memory is known to $\mathcal{A}$.

$Test(\Pi^u/\Pi^s)$ is used to model the semantic security of SK between U_i and S_j adhering to the ROR model's indistinguishability style. An impartial coin c needs to be throwed ahead of starting experiment. If $\mathcal{A}$ executes this query and the generated SK is of freshness, then the instance Π^u/Π^s returns an SK when $c=1$ or a random number when $c=0$. For other cases, the query returns a null value.

Semantic security of SK: In the ROR model, $\mathcal{A}$ makes as many as $Test$ queries to either $\Pi^u_{U_i}$ or $\Pi^s_{S_j}$ as necessary to distinguish the instance $\Pi^u_{U_i}{}'s$ or $\Pi^s_{S_j}{}'s$ real SK from a random key. The output of $Test$ queries must be consistent or uniform to random bit c. Once completing the game, $\mathcal{A}$ returns a guessed bit c' and wins the game (denoted as $Succ$) if $c'=c$. The gained advantage of $\mathcal{A}$ to break the semantic security of eSAS2KN is defined as $Adv_{\mathcal{A}}^{eSAS2KN}=|2Pr[Succ]-1|$. The proposed eSAS2KN is secure if $Adv_{\mathcal{A}}^{eSAS2KN}\le\upsilon$, for the run time t and sufficiently small value $\upsilon>0$.

Random oracle: The one-way cryptographic hash function $h(\cdot)$ is used to model the random oracle, say $\mathcal{H}$, which can be accessed by all communicating parties and $\mathcal{A}$.

Theorem 1: Let $\mathcal{A}$ be an adversary running in a polynomial time t against eSAS2KN in the ROR model, then, $Adv_{\mathcal{A}}^{eSAS2KN}(t)\le\frac{q_h^2}{|\mathcal{H}|}+\frac{q_{send}}{2^{l-1}|\mathcal{PD}|}$, where q_h, $|\mathcal{H}|$, q_{send}, $|\mathcal{PD}|$ and l denote the number of hash queries, the range space of $h(\cdot)$, the number of send queries, the size of uniformly distributed password dictionary, and the number of bits present in σ_i, respectively.

Proof: To demonstrate the Theorem 1, we define five games, G_i ($i=0,\ 1,\ \cdots,\ 4$) in sequence. Let $Succ_i$ be the event that the bit c of the throwed impartial coin in G_i is successfully guessed by $\mathcal{A}$. The five games are defined as below in details.

Game G₀: G_0 represents an actual attack in the ROR model launched by $\mathcal{A}$ against eSAS2KN scheme, in which $\mathcal{A}$ selects a bit c in advance at the beginning of the game G_0. Therefore, $\mathcal{A}'s$ advantage is obtained

$$Adv_{\mathcal{A}}^{eSAS2KN}(t)=|2Pr[Succ_0]-1| \tag{1}$$

Game G₁: G_1 denotes an intercepting attack launched by $\mathcal{A}$ through running $Execute(\Pi^u/\Pi^s,\Pi^g)$ to obtain the messages $Msg1:<DID_i,\ HID_i,\ M_{U_i,G_k},\ T_i^{(1)}>$, $Msg2:<HID_i,PID_j,M_{G_k,S_j},T_{G_k}^{(2)}>$, $Msg3:<M_{S_j,G_k},HID_i,\ T_j^{(2)}>$. The output of $Test(\Pi^u/\Pi^s)$ is examined whether the SK between U_i and S_j is a key or a random number. In the eSAS2KN, the SK is computed according to $SK_{i/j}=h(HID_i||M_j||N_j||T_j^{(2)})$ and the intercepted messages do not reveal the secret parameters M_j and N_j since $\mathcal{A}$ cannot obtain the random nonce RN_k. Therefore, the probability of $\mathcal{A}'s$ winning G_1 by the eavesdropping attack is not increased, and equal to that of winning G_0 as below.

$$Pr[Succ_1]=Pr[Succ_0] \tag{2}$$

Game G₂: G_2 denotes an active attack launched by $\mathcal{A}$ through running $Send(\Pi^u/\Pi^s/\Pi^g,Msg)$ and hash queries aiming to trick a legal instance into accepting an illegal message. To create hash collisions, $\mathcal{A}$ can make any number of hash queries. However, the current timestamps are attached with all the transmitted messages. Therefore, it is infeasible for $\mathcal{A}$ find hash collision occurrence in a polynomial time through running send and hash queries. By using birthday paradox theory to find hash collision, the following result is obtained.

$$|Pr[Succ_1]-Pr[Succ_2]|\leq\frac{q_h^2}{2|\mathcal{H}|} \tag{3}$$

Game G3: In G_3, $\mathcal{A}$ launches an active attack by executing $CorruptSC(\Pi^u)$ and extracts all the secret credentials, i.e., $(IDS_i, HID_i, h(\cdot), A_i, B_i, C_i, D_i, \tau_i)$, stored in the memory of the stolen or lost SC_i. By using dictionary attack, $\mathcal{A}$ can guess PW_i according to the information extracted from SC_i. Due to the use of fuzzy extractor, which is used to compute $HPW_i = h(PW_i \| \sigma_i)$ by extracting $U_i{}'$s biometric secret key σ_i, it allows for the retrieval of at most l nearly random bits. Therefore, the probability of $\mathcal{A}$ guessing $\sigma_i \in \{0,1\}^l$ is $1/2^l$ approximately. Due to the number limitation of permitted wrong password entries, we can obtain the following result.

$$\left| Pr[Succ_2]-Pr[Succ_3] \right| \leq \frac{q_{send}}{2^l|\mathcal{PD}|} \tag{4}$$

Game G4: In G_4, $\mathcal{A}$ launches a sensor node capture attack by executing $CorruptSN(\Pi^s_{S_j})$ and extracts all the secret credentials $\{HID_j, A_j, B_j\}$, which are stored in the memory of S_j. However, all the extracted secret credentials by using $CorruptSN(\Pi^s_{S_j})$ queries have no help in deriving the session key $SK_{i/j}$ because the derivation of HID_i needs $U_i{}'$s identity ID_i, the random nonce RN_i generated by U_i, and $GWN_k{}'$s long-time secret K. In addition, the derivation of M_j needs the random nonce RN_k generated by GWN_k. Therefore, the probability of $\mathcal{A}'s$ winning G_4 by executing $CorruptSN(\Pi^s_{S_j})$ queries is not increased, and equal to that of winning G_3 as below.

$$Pr[Succ_3]=Pr[Succ_4] \tag{5}$$

In order to break the semantic security of the proposed eSAS2KN, $\mathcal{A}$ tries to run all the oracle queries. However, $\mathcal{A}$ can only guess the bit c at last for winning the game after $Test(\Pi^u/\Pi^s)$ query. Therefore, the following result is obtained.

$$Pr[Succ_4]=\frac{1}{2} \tag{6}$$

According to equations (1), (2) and (6), the following result is obtained.

$$\frac{1}{2}Adv_{\mathcal{A}}^{SAS2KN}(t) = \left|Pr[Succ_0]-1/2\right| = \left|Pr[Succ_1]-Pr[Succ_4]\right| \tag{7}$$

By using triangular inequality, the following result can be obtained.

$$\begin{aligned}\left|Pr[Succ_1]-Pr[Succ_4]\right| &\leq \left|Pr[Succ_1]-Pr[Succ_2]\right|+\left|Pr[Succ_2]-Pr[Succ_4]\right| \\ &\leq \left|Pr[Succ_1]-Pr[Succ_2]\right|+\left|Pr[Succ_2]-Pr[Succ_3]\right| \\ &\quad +\left|Pr[Succ_3]-Pr[Succ_4]\right|\end{aligned} \tag{8}$$

According to equations (7), (3), (4) and (5), the result can be derived as follows.

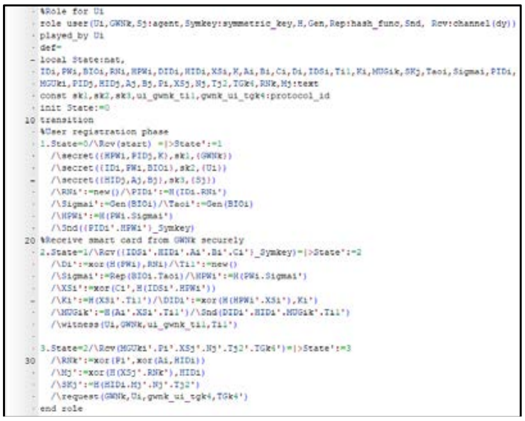
$$Adv_{\mathcal{A}}^{eSAS2KN}(t) \leq \frac{q_h^2}{|\mathcal{H}|} + \frac{q_{send}}{2^{l-1}|\mathcal{PD}|} \tag{9}$$

5.2.3 Formal Security Verification

In this subsection, we implement formal security verification by using AVISPA, which is a widely used common tool to automatically verify the security of Internet protocols [45-47]. In the AVISPA, a user can select different verification techniques to check the security of the same security protocol [48].

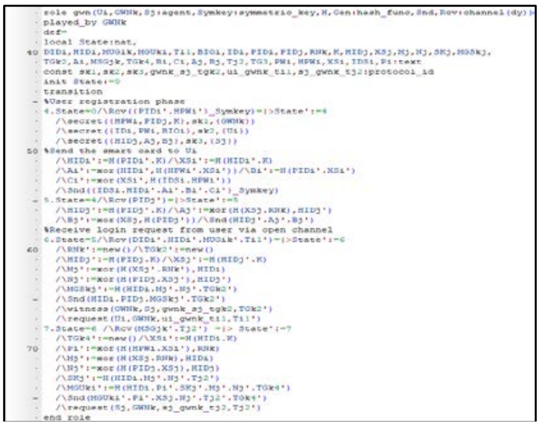
The tool employs a role-based formal language, called High-Level Protocol Specification Language (HLPSL) to specify some protocols and their possessed security properties. In addition, the tool is a modular and expressive language and can run On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) to conduct a large number of most advanced automatic analysis techniques.

To conduct eSAS2KN's formal security verification, we describe the scheme by utilizing the role-oriented HLPSL. In [49-50], some specifications involved in AVISPA and HLPSL are addressed in detail. For simulation, we firstly describe the roles of different objects in HLPSL. The basic roles of U_i , GWN_k and S_j are shown in **Fig. 5**, **Fig. 6** and **Fig. 7**, respectively. The mandatory roles of session, goal and environment are shown in **Fig. 8**. Then, we simulate eSAS2KN in an animator, called SPAN, for the tool AVISPA. With the consideration of no supporting XOR operations for SATMC and TA4SP back-ends in SPAN [51], therefore, we simulate eSAS2KN's security merely by employing the OFMC and CL-AtSe back-ends. The simulation results in **Fig. 9** demonstrate the proposed eSAS2KN is safe.



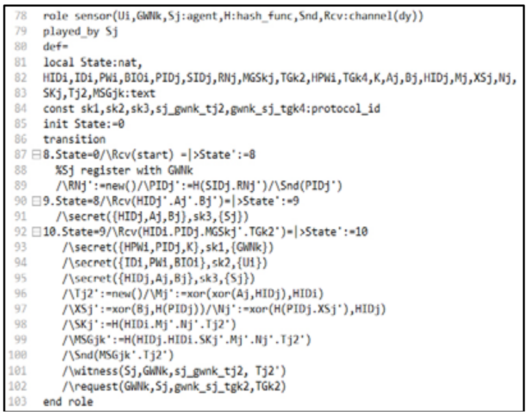
```
%Role for Ui
role user(Ui,GWNk,Sj:agent,Symkey:symmetric_key,H,Gen,Rep:hash_func,Snd, Rcv:channel(dy))
played_by Ui
def=
local State:nat,
IDi,PWi,BIOi,RNi,HPWi,DIDi,HIDi,XSi,K,Ai,Bi,Ci,Di,IDSi,Ti1,Ki,MUGik,SKj,Taoi,Sigmai,PIDi,
MGUki,PIDj,HIDj,Aj,Bj,Pi,XSj,Nj,Tj2,TGk4,RNk,Mj:text
const sk1,sk2,sk3,ui_gwnk_ti1,gwnk_ui_tgk4:protocol_id
init State:=0
transition
%User registration phase
1.State=0/\Rcv(start) =|>State':=1
  /\secret({HPWi,PIDj,K},sk1,{GWNk})
  /\secret({IDi,PWi,BIOi},sk2,{Ui})
  /\secret({HIDj,Aj,Bj},sk3,{Sj})
  /\RNi':=new()/\PIDi':=H(IDi.RNi')
  /\Sigmai':=Gen(BIOi)/\Taoi':=Gen(BIOi)
  /\HPWi':=H(PWi.Sigmai')
  /\Snd({PIDi'.HPWi'}_Symkey)
%Receive smart card from GWNk securely
2.State=1/\Rcv({IDSi'.HIDi'.Ai'.Bi'.Ci'}_Symkey)=|>State':=2
  /\Di':=xor(H(PWi),RNi)/\Ti1':=new()
  /\Sigmai':=Rep(BIOi.Taoi)/\HPWi':=H(PWi.Sigmai')
  /\XSi':=xor(Ci',H(IDSi'.HPWi'))
  /\Ki':=H(XSi'.Ti1')/\DIDi':=xor(H(HPWi'.XSi'),Ki')
  /\MUGik':=H(Ai'.XSi'.Ti1')/\Snd(DIDi'.HIDi'.MUGik'.Ti1')
  /\witness(Ui,GWNk,ui_gwnk_ti1,Ti1')

3.State=2/\Rcv(MGUki'.Pi'.XSj'.Nj'.Tj2'.TGk4')=|>State':=3
  /\RNk':=xor(Pi',xor(Ai,HIDi))
  /\Mj':=xor(H(XSj'.RNk'),HIDi)
  /\SKj':=H(HIDi.Mj'.Nj'.Tj2')
  /\request(GWNk,Ui,gwnk_ui_tgk4,TGk4')
end role
```

Fig. 5. Basic role of U_i in HLPSL.



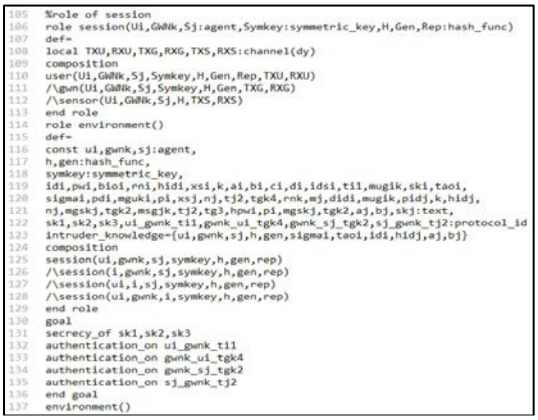
```
role gwn(Ui,GWNk,Sj:agent,Symkey:symmetric_key,H,Gen:hash_func,Snd,Rcv:channel(dy))
played_by GWNk
def=
local State:nat,
DIDi,HIDi,MUGik,MGUki,Ti1,BIOi,IDi,PIDi,PIDj,RNk,K,HIDj,XSj,Mj,Nj,SKj,MGSkj,
TGk2,Ai,MSGjk,TGk4,Bi,Ci,Aj,Bj,Tj2,TG3,PWi,HPWi,XSi,IDSi,Pi:text
const sk1,sk2,sk3,gwnk_sj_tgk2,ui_gwnk_ti1,sj_gwnk_tj2:protocol_id
init State:=0
transition
%User registration phase
4.State=0/\Rcv({PIDi'.HPWi'}_Symkey)=|>State':=4
  /\secret({HPWi,PIDj,K},sk1,{GWNk})
  /\secret({IDi,PWi,BIOi},sk2,{Ui})
  /\secret({HIDj,Aj,Bj},sk3,{Sj})
%Send the smart card to Ui
  /\HIDi':=H(PIDi'.K)/\XSi':=H(HIDi'.K)
  /\Ai':=xor(HIDi',H(HPWi'.XSi'))/\Bi':=H(PIDi'.XSi')
  /\Ci':=xor(XSi',H(IDSi.HPWi'))
  /\Snd({IDSi.HIDi'.Ai'.Bi'.Ci'}_Symkey)
5.State=4/\Rcv(PIDj')=|>State':=5
  /\HIDj':=H(PIDj'.K)/\Aj':=xor(H(XSj.RNk),HIDj')
  /\Bj':=xor(XSj,H(PIDj'))/\Snd(HIDj'.Aj'.Bj')
%Receive login request from user via open channel
6.State=5/\Rcv(DIDi'.HIDi'.MUGik'.Ti1')=|>State':=6
  /\RNk':=new()/\TGk2':=new()
  /\HIDj':=H(PIDj.K)/\XSj':=H(HIDj'.K)
  /\Mj':=xor(H(XSj'.RNk'),HIDi)
  /\Nj':=xor(H(PIDj.XSj'),HIDj')
  /\MGSkj':=H(HIDi.Mj'.Nj'.TGk2')
  /\Snd(HIDi.PIDj.MGSkj'.TGk2')
  /\witness(GWNk,Sj,gwnk_sj_tgk2,TGk2')
  /\request(Ui,GWNk,ui_gwnk_ti1,Ti1')
7.State=6 /\Rcv(MSGjk'.Tj2') =|> State':=7
  /\TGk4':=new()/\XSi':=H(HIDi.K)
  /\Pi':=xor(H(HPWi.XSi'),RNk)
  /\Mj':=xor(H(XSj.RNk),HIDi)
  /\Nj':=xor(H(PIDj.XSj),HIDj)
  /\SKj':=H(HIDi.Mj'.Nj'.Tj2')
  /\MGUki':=H(HIDi.Pi'.SKj'.Mj'.Nj'.TGk4')
  /\Snd(MGUki'.Pi'.XSj.Nj'.Tj2'.TGk4')
  /\request(Sj,GWNk,sj_gwnk_tj2,Tj2')
end role
```

Fig. 6. Basic role of GWN_k in HLPSL.



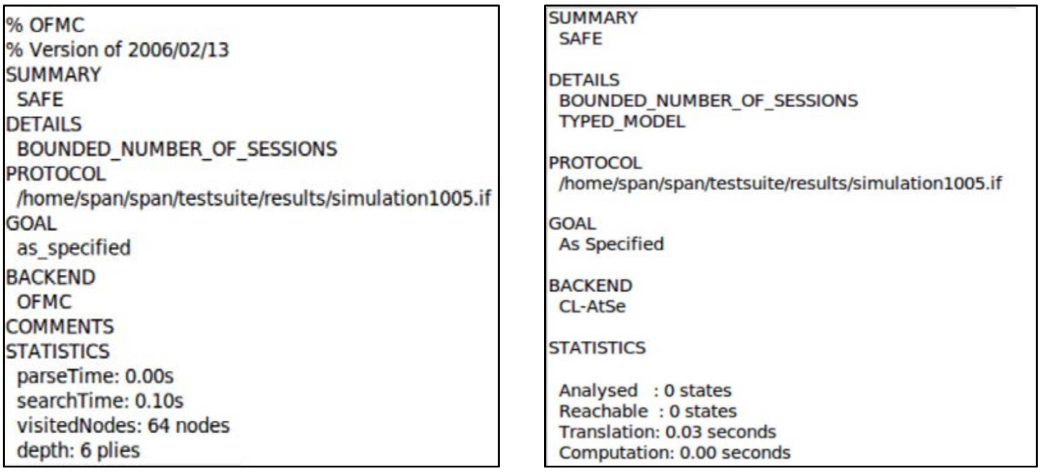
```
role sensor(Ui,GWNk,Sj:agent,H:hash_func,Snd,Rcv:channel(dy))
played_by Sj
def=
local State:nat,
HIDi,IDi,PWi,BIOi,PIDj,SIDj,RNj,MGSkj,TGk2,HPWi,TGk4,K,Aj,Bj,HIDj,Mj,XSj,Nj,
SKj,Tj2,MSGjk:text
const sk1,sk2,sk3,sj_gwnk_tj2,gwnk_sj_tgk4:protocol_id
init State:=0
transition
8.State=0/\Rcv(start) =|>State':=8
  %Sj register with GWNk
  /\RNj':=new()/\PIDj':=H(SIDj.RNj')/\Snd(PIDj')
9.State=8/\Rcv(HIDj'.Aj'.Bj')=|>State':=9
  /\secret({HIDj,Aj,Bj},sk3,{Sj})
10.State=9/\Rcv(HIDi.PIDj.MGSkj'.TGk2')=|>State':=10
   /\secret({HPWi,PIDj,K},sk1,{GWNk})
   /\secret({IDi,PWi,BIOi},sk2,{Ui})
   /\secret({HIDj,Aj,Bj},sk3,{Sj})
   /\Tj2':=new()/\Mj':=xor(xor(Aj,HIDj),HIDi)
   /\XSj':=xor(Bj,H(PIDj))/\Nj':=xor(H(PIDj.XSj'),HIDj)
   /\SKj':=H(HIDi.Mj'.Nj'.Tj2')
   /\MSGjk':=H(HIDj.HIDi.SKj'.Mj'.Nj'.Tj2')
   /\Snd(MSGjk'.Tj2')
   /\witness(Sj,GWNk,sj_gwnk_tj2, Tj2')
   /\request(GWNk,Sj,gwnk_sj_tgk2,TGk2)
end role
```

Fig. 7. Basic role of S_j in HLPSL.



```
%role of session
role session(Ui,GWNk,Sj:agent,Symkey:symmetric_key,H,Gen,Rep:hash_func)
def=
local TXU,RXU,TXG,RXG,TXS,RXS:channel(dy)
composition
user(Ui,GWNk,Sj,Symkey,H,Gen,Rep,TXU,RXU)
/\gwn(Ui,GWNk,Sj,Symkey,H,Gen,TXG,RXG)
/\sensor(Ui,GWNk,Sj,H,TXS,RXS)
end role
role environment()
def=
const ui,gwnk,sj:agent,
h,gen:hash_func,
symkey:symmetric_key,
idi,pwi,bioi,rni,hidi,xsi,k,ai,bi,ci,di,idsi,ti1,mugik,ski,taoi,
sigmai,pdi,mguki,pi,xsj,nj,tj2,tgk4,rnk,mj,didi,mugik,pidj,k,hidj,
nj,mgskj,tgk2,msgjk,tj2,tg3,hpwi,pi,mgskj,tgk2,aj,bj,skj:text,
sk1,sk2,sk3,ui_gwnk_ti1,gwnk_ui_tgk4,gwnk_sj_tgk2,sj_gwnk_tj2:protocol_id
intruder_knowledge={ui,gwnk,sj,h,gen,sigmai,taoi,idi,hidj,aj,bj}
composition
session(ui,gwnk,sj,symkey,h,gen,rep)
/\session(i,gwnk,sj,symkey,h,gen,rep)
/\session(ui,i,sj,symkey,h,gen,rep)
/\session(ui,gwnk,i,symkey,h,gen,rep)
end role
goal
secrecy_of sk1,sk2,sk3
authentication_on ui_gwnk_ti1
authentication_on gwnk_ui_tgk4
authentication_on gwnk_sj_tgk2
authentication_on sj_gwnk_tj2
end goal
environment()
```

Fig. 8. Roles of session, goal and environment in HLPSL.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/simulation1005.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.10s
  visitedNodes: 64 nodes
  depth: 6 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/simulation1005.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.03 seconds
  Computation: 0.00 seconds
```

Fig. 9. Simulation results of eSAS2KN under OFMC and CL-AtSe backends.

6. Security and Performance Comparison

In this section, security and performance comparisons are conducted with some related protocols in the light of security features, computation costs, and communication costs.

6.1 Security features

The comparison in terms of security feature is conducted with some related protocols, which is illustrated in **Table 2**. The comparison shows that the high security of our protocol over other related schemes and can be applied in practical applications.

Table 2. Comparison of security features.

Security features	Jung [28]	Das [35]	Maurya [36]	Amin [29]	Soni [30]	Ali [31]	Wu [32]	Dai [33]	ours
User anonymity	✓	×	×	✓	✓	✓	✓	✓	✓
Resist MITM attack	×	✓	✓	×	✓	✓	✓	✓	✓
Mutual authentication	✓	×	✓	✓	✓	✓	✓	✓	✓
DoS attack	✓	✓	✓	✓	✓	✓	×	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
SLSC attack	✓	×	✓	✓	✓	✓	✓	✓	✓
OPG attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
CSNI attack	✓	✓	✓	×	✓	×	✓	✓	✓
User impersonation attack	✓	✓	✓	✓	×	✓	✓	✓	✓
GWN bypassing attack	✓	×	×	×	×	✓	✓	×	✓
Password verification	✓	✓	×	✓	✓	✓	×	✓	✓
SK verification	✓	✓	×	×	✓	×	✓	×	✓
Privileged insider attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Provide SK security	✓	×	✓	×	✓	×	✓	✓	✓
Efficient password change	✓	✓	✓	✓	✓	✓	✓	✓	✓
Offline ID guessing attack	✓	×	✓	×	✓	×	✓	✓	✓
Formal security proof	×	×	✓	×	×	×	×	×	✓
Resist user rejection in login phase	×	✓	✓	×	✓	✓	×	×	✓

6.2 Computation costs

For clear analysis and description of each protocol's computation costs, we give the following denotations. T_H denotes the time to run the one-way hash operation, T_X to conduct XOR operation, T_E to conduct an ECC point multiplication operation, T_F to conduct a fuzzy extractor operation, T_B to conduct Bio-Hash operation, and to run symmetric-key encryption or decryption. According to the experiment results in [52-54], T_H , T_E , T_F , T_B and T_S are 0.0005s, 0.063075s, 0.063075s, 0.063075s and 0.0087s, respectively. The bitwise XOR operation is so lightweight and its computing time can be negligible, therefore, we do not consider them in the final approximate total computation costs. The computation costs in the login phase, authentication and key negotiation phase, and final approximate total computation costs of different schemes are shown in **Table 3**.

Table 3. Computation costs comparison with related protocols.

Scheme	**Phase**		**Total computation costs**
	Login	**Authentication and key negotiation**	
Jung[28]	$5T_H+3T_X+2T_B$	$15T_H+9T_X$	$20T_H+12T_X+2T_B \approx 0.13615$s
Das[35]	$3T_H+T_X+T_F$	$9T_H+3T_X+4T_S$	$12T_H+4T_X+T_F+4T_S \approx 0.103875$s
Maurya[36]	$3T_H+2T_X+T_F+2T_E$	$T_H+T_X+3T_E+5T_S$	$4T_H+3T_X+5T_E+T_F+5T_S \approx 0.42395$s
Amin[29]	$8T_H+5T_X+T_F$	$35T_H+17T_X$	$43T_H+22T_X+T_F \approx 0.084575$
Soni[30]	$8T_H+5T_X+T_F+2T_E$	$25T_H+9T_X+4T_E$	$33T_H+14T_X+6T_E+T_F \approx 0.458025$s
Ali[31]	$2T_H+4T_X+T_F+2T_E$	$4T_H+18T_X+2T_E$	$6T_H+22T_X+4T_E+T_F \approx 0.318375$s
Wu[32]	$8T_H+7T_X+T_F$	$24T_H+13T_X$	$32T_H+20T_X+T_F \approx 0.079075$s
Dai[33]	$5T_H+3T_X+2T_E+T_B$	$24T_H+11T_X+6T_E$	$29T_H+14T_X+8T_E+T_B \approx 0.582175$s
Ours	$8T_H+3T_X+T_F$	$26T_H+14T_X$	$34T_H+17T_X+T_F \approx 0.080075$s

From the **Table 3**, we can clearly conclude that our protocol performs much better efficiency than Jung [28], Das [35], Maurya [36], Amin [29], Soni [30], Ali [31] and Dai [33] et. al.'s protocols. Even though our protocol has slightly more computing time than Wu [32], it is yet admissible because eSAS2KN provides much more security features than Wu [32] et. al.'s protocol.

6.3 Communication Costs

For clear analysis and comparison of communication costs of different schemes, the lengths of identity, password, random number/string, error tolerance threshold, request, response, symmetric encryption/decryption, probabilistic generation function and deterministic reproduction function for fuzzy extractor are assumed to be 128 bits, the lengths of credential and timestamps are 32 bits, and the lengths of hash function, bio-hash function, as well as ECC encryption/decryption operation are 160 bits. The communication costs in the login phase, authentication and key negotiation phase, as well as the total communication costs in bits of different schemes are summarized in **Table 4**.

Table 4. Communication costs comparison with related protocols.

Scheme	**Phase**		**Total communication costs (in bits)**
	Login	**Authentication and key negotiation**	
Jung[28]	512	1056	1568
Das[35]	256	704	960
Maurya[36]	416	256	672
Amin[29]	576	3456	4032
Soni[30]	672	1280	1952
Ali[31]	608	1216	1824
Wu[32]	960	1504	2464
Dai[33]	672	1376	2048
Ours	512	1408	1920

From the **Table 4**, we can clearly conclude that our protocol has much better communication efficiency than Amin [29], Soni [30], Wu [32] and Dai [33] et al.'s protocols. Although eSAS2KN provides less communication efficiency than Jung [28], Das [35], Maurya [36] and Ali [31], it yet provides much more security and functionality features, which is presented in **Table 2**.

7. Conclusions

In this paper, we developed a scheme (eSAS2KN) that enables lightweight mutual authentication as well as session key establishment. Due to the adoption of fuzzy extractor technique, user's high rejection probability can be avoided in the login phase. Informal security analysis, BAN logic verification, formal security proof and verification demonstrate that the proposed eSAS2KN is safe. More importantly, the eSAS2KN is developed with only lightweight hash operations and XOR operations, which make it more lightweight and more efficient. Performance comparison with the competitive schemes shows that the eSAS2KN is more suitable for real-time communications between users and sensor node for multi-gateway WSNs, and can easily be implemented for a practical application.

References

[1] K.-A. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 577–601, Jul. 2016. Article (CrossRef Link)

[2] F. A. Silva, "Industrial Wireless Sensor Networks: Applications, Protocols, and Standards," *IEEE Industrial Electronics Magazine*, vol. 8, no. 4, pp. 67-68, Dec. 2014. Article (CrossRef Link)

[3] H. Xie, Z. Yan, Z. Yao et. al., "Data collection for security measurement in wireless sensor networks: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2205-2224, Apr. 2019. Article (CrossRef Link)

[4] P. Rawat, K. D. Singh, "Wireless sensor networks: a survey on recent developments and potential synergies," *Journal of Supercomputing*, vol. 68, pp.1-48, Apr. 2014. Article (CrossRef Link)

[5] D. He, S. Chan, S. Tang, "A novel and lightweight system to secure wireless medical sensor networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 316-326, Jan. 2014. Article (CrossRef Link)

[6] P. Gope, T. Hwang, “Bsn-care: A secure IoT-based modern healthcare system using body sensor network,” *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1368-1376, Mar. 2016. Article (CrossRef Link)

[7] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, M. K. Khan, “A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity,” *Security and Communication Networks*, vol. 9, no. 15, pp. 2643-2655, Feb. 2016. Article (CrossRef Link)

[8] M. S. Yousefpoor, H. Barati, “Dynamic key management algorithms in wireless sensor networks: A survey,” *Computer Communications*, vol. 134, pp.52-69, Jan. 2019. Article (CrossRef Link)

[9] M. Wazid, A. K. Das et. al., “Authenticated key management protocol for cloud-assisted body area sensor networks,” *Journal of Network and Computer Applications*, vol. 123, pp. 112-126, Dec. 2018. Article (CrossRef Link)

[10] S. Athmani, A. Bilami et. al., “Edak: An efficient dynamic authentication and key management mechanism for heterogeneous WSNs,” *Future Generation Computer Systems*, vol. 92, pp. 789-799, Mar. 2019. Article (CrossRef Link)

[11] S.-H. Seo, J. Won et. al., “Effective key management in dynamic wireless sensor networks,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 371-383, Feb. 2015. Article (CrossRef Link)

[12] Z. Liu, J. Wen, J. Ma, et. al., “TCEMD: A trust cascading-based emergency message dissemination model in VANETs,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4028-4048, May 2020. Article (CrossRef Link)

[13] Z. Liu, F. Huang, J. Weng et. al., “BTMPP: Balancing trust management and privacy preservation for emergency message dissemination in vehicular networks,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5386-5407, Apr. 2021. Article (CrossRef Link)

[14] J. Guo, Z. Liu, S. Tian et. al., “TFL-DT: A trust evaluation scheme for federated learning in digital twin for mobile networks,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no.11, pp. 3548-3560, Nov. 2023. Article (CrossRef Link)

[15] A. K. Das, “Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards,” *IET Information Security*, vol. 5, no. 3, pp. 145-151, Apr. 2011. Article (CrossRef Link)

[16] C.-C. Chang, N.-T. Nguyen, “An untraceable biometric-based multi-server authenticated key agreement protocol with revocation,” *Wireless Personal Communications*, vol. 90, pp. 1695-1715, Jun. 2016. Article (CrossRef Link)

[17] Y. Dodis, R. Ostrovsky, L. Reyzin, et. al., “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” *SIAM journal on computing*, vol. 38, no. 1, pp. 97-139, 2008. Article (CrossRef Link)

[18] M. L. Das, “Two-factor user authentication in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp.1086-1090, Mar. 2009. Article (CrossRef Link)

[19] M. K. Khan, K. Alghathbar, “Cryptanalysis and security improvements of ‘two-factor user authentication in wireless sensor networks’,” *Sensors*, vol. 10, no. 3, pp. 2450-2459, Mar. 2010. Article (CrossRef Link)

[20] T.-H.Chen, W.-K.Shih, “A robust mutual authentication protocol for wireless sensor networks,” *ETRI Journal*, vol. 32, no. 5, pp. 704-712, Oct. 2010. Article (CrossRef Link)

[21] D. He, Y. Gao, et. al., “An enhanced two-factor user authentication scheme in wireless sensor networks,” *Ad-Hoc and Sensor Wireless Networks*, vol. 10, no. 4, pp. 343-359, 2010. Article (CrossRef Link)

[22] Yeh H.-L., Chen T.-H., Liu P.-C., et. al., “A secured authentication protocol for wireless sensor networks using elliptic curves cryptography,” *Sensors*, vol. 11, no. 5, pp. 4767-4779, May 2011. Article (CrossRef Link)

[23] B. Vaidya, D. Makrakis, H. Mouftah, “Two-factor mutual authentication with key agreement in wireless sensor networks,” *Security and Communication Networks*, vol. 9, no. 2, pp. 171-183, Jan. 2016. Article (CrossRef Link)

[24] K. Xue, C. Ma, P. Hong, R. Ding, “A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks,” *Journal of Network and Computer Applications*,

vol. 36, no. 1, pp. 316-323, Jan. 2013. Article (CrossRef Link)

[25] J. Kim, D. Lee et. al., "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6443–6462, Apr. 2014. Article (CrossRef Link)

[26] I.-P. Chang, T.-F. Lee et. al., "Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks," *Sensors*, vol. 15, no. 12, pp. 29841–29854, Nov. 2015. Article (CrossRef Link)

[27] Y. Park, Y. Park, "Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 12, Dec. 2016. Article (CrossRef Link)

[28] J. Jung, J. Moon et. al., "Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks," *Sensors*, vol. 17, no. 3, pp. 644, Mar. 2017. Article (CrossRef Link)

[29] R. Amin, S. H. Islam, G. Biswas, M. S. Obaidat, "A robust mutual authentication protocol for WSN with multiple base-stations," *Ad Hoc Networks*, vol. 75-76, pp. 1-18, Jun. 2018. Article (CrossRef Link)

[30] P. Soni, A. K. Pal, S. H. Islam, "An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system," *Computer Methods and Programs in Biomedicine*, vol. 182, pp. 105054, Dec. 2019. Article (CrossRef Link)

[31] Z. Ali, A. Ghani et. al., "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, pp.102502, Jun. 2020. Article (CrossRef Link)

[32] F. Wu, X. Li et. al., "A novel three-factor authentication protocol for wireless sensor networks with IoT notion," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1120-1129, Mar. 2021. Article (CrossRef Link)

[33] C. Dai, Z. Xu, "A secure three-factor authentication scheme for multi gateway wireless sensor networks based on elliptic curve cryptography," *Ad Hoc Networks*, vol. 127, pp. 102768, Mar. 2022. Article (CrossRef Link)

[34] D. He, N. Kumar et.al., "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 30-37, Feb. 2014. Article (CrossRef Link)

[35] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems*, vol. 30, no.1, pp. e2933, Jan. 2017. Article (CrossRef Link)

[36] A. K. Maurya, 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, pp. 136, Oct. 2017. Article (CrossRef Link)

[37] D. Dolev, A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp.198-208, Mar. 1983. Article (CrossRef Link)

[38] J. Lin, W. Yu et. al., "A survey on internet of things: architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017. Article (CrossRef Link)

[39] Q. Yang, P. Gasti et. al., "On inferring browsing activity on smartphones via SUB power analysis side-channel," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1056-1066, May 2017. Article (CrossRef Link)

[40] S. R. Shanmugham, S. Paramasivam, "Survey on power analysis attacks and its impact on intelligent sensor networks," *IET Wireless Sensor Systems*, vol. 8, no. 6, pp. 295-304, Dec. 2018. Article (CrossRef Link)

[41] R. Lumbiarres-Lopez, M. López-Garc´ıa, E. Canto-Navarro, "Hardware architecture implemented on FPGA for protecting cryptographic keys against side-channel attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, vol. 5, pp. 898-905, Oct. 2018. Article (CrossRef Link)

[42] R. Canetti, H. Krawczyk, “Analysis of key-exchange protocols and their use for building secure channels,” *Presented at Eurocrypt 2001*. [Online]. Available: https://eprint.iacr.org/2001/040

[43] R. Canetti, H. Krawczyk, “Universally Composable Notions of Key Exchange and Secure Channels,” in *Proc. of EUROCRYPT 2002: Advances in Cryptology — EUROCRYPT 2002*, pp. 337-351, Apri. 2002. Article (CrossRef Link)

[44] V. Odelu, A. K. Das et. al., “Provably secure authenticated key agreement scheme for smart grid,” *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1900-1910, May 2018. Article (CrossRef Link)

[45] M. Wazid, A. K. Das et. al., “Secure authentication scheme for medicine anti-counterfeiting system in IoT environment,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1634–1646, Oct. 2017. Article (CrossRef Link)

[46] S. Chatterjee, S. Roy et. al., “Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 824-839, Oct. 2018. Article (CrossRef Link)

[47] V. Odelu, A. K. Das, A. Goswami, “A secure biometrics-based multi-server authentication protocol using smart cards,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp.1953-1966, Sep. 2015. Article (CrossRef Link)

[48] A. K. Das, “A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications,” *Networking Science*, vol. 2, pp.12-27, May 2013. Article (CrossRef Link)

[49] A security protocol animator for AVISPA, [Online]. Available: https://people.irisa.fr/Thomas.Genet/span/, Accessed on: 2023, August 17.

[50] A. Armando, D. Basin, Y. Boichut, et al., “The AVISPA tool for the automated validation of internet security protocols and applications,” in *Proc. of 17th Int. conf. on computer aided verification (CAV2005)*, Edinburgh, Scotland, Jul. pp. 6-10, 2005.

[51] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K.-K. R. Choo, Y. Park, “Design of secure and lightweight authentication protocol for wearable devices environment,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1310-1322, Jul. 2018. Article (CrossRef Link)

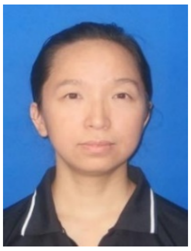
[52] D. He, N. Kumar et. al., “Enhanced three-factor security protocol for consumer USB mass storage devices,” *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 30-37, Feb. 2014. Article (CrossRef Link)

[53] CT Li, MS Hwang, YP Chu, “A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks,” *Computer Communications*, vol. 31, no. 12, pp. 2803-2814, Jul. 2008. Article (CrossRef Link)

[54] W. Li, Q. Wen et. al., “An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network,” *Computer Communications*, vol. 35, no. 2, pp. 188-195, Jan. 2012. Article (CrossRef Link)

Jiping Li received the Ph. D. degree in Radio Physics from Central China Normal University, Wuhan, China, in 2012, and M.Sc. degree in Computer Science and Technology from Ocean University of China, Qingdao, China, in 2006. He is currently a professor with the School of Computer Science and Information Engineering, Changzhou Institute Technology, Changzhou, China. His research interests include the security of Internet of Vehicles, the application and security of Industrial Internet of Things, wireless resource allocation, and the security of Wireless Networks.

Yuanyuan Zhang received the B.E. degree in Electronic Information Engineering and B.A. degree in Business English from Jiangsu University of Technology in 2017, and Ph.D degree in Traffic Information Engineering and Control from Shanghai Maritime University in 2023. From 2021 to 2022, she was a visiting Ph.D. Student in Electrical and Computer Engineering with the University of Victoria, Canada. She is currently a lecturer with the School of Computer and Information Engineering, Changzhou Institute of Technology. Her research interests include wireless sensor networks, underwater acoustic sensor networks, data processing, target localization and tracking.



Lixiang Shen received the Ph.D. degree in Computer Science and Technology from Northwestern Polytechnical University in 2021. She is currently a vice Professor with School of Computer Science and Information Engineering, Changzhou Institute of Technology, Changzhou, China. And her current research interests include the hardware security and cyber security.

Jing Cao received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University in 2018. She is currently a lecturer with the School of Computer Science and Information Engineering, Changzhou Institute of Technology, Changzhou, China. Her current research interest includes resource allocation of wireless network, intelligent IOT, and QoS optimization.

Wenwu Xie received the B.S., M.S., and Ph.D. degrees in communication engineering from Central China Normal University, Wuhan, China, in 2004, 2007, and 2017, respectively. He is currently an Associate Professor with the School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang, Hunan, China. His research interests include communication algorithm and control algorithm.

Yi Zheng received the M.S. and Ph.D. degree from the College of Physical Science and Technology, Central China Normal University, China, in 2018 and 2022. He is currently a Lecturer with the School of Artificial Intelligence, Jianghan University, Wuhan, China. His research interests include intelligent communication and deep learning.

Shouyin Liu received the B.S. degree in Physics and his M.S. degree in Radio Electronics from Central China Normal University, Wuhan, China, in 1985 and 1988, respectively. He received the Ph.D. degree in Electronic Communication Engineering from Hanyang University, Korea, in 2005. Since 2004, he has been a professor at Central China Normal University. His current research interests include digital communication, WSN, and location techniques.