A Secure Authentication Scheme for Wireless Sensor Networks Based on DAC and Intel SGX

Xin Liu, Zhenbin Guo, Jun Ma, and Yuchen Song

***Abstract*—Wireless sensor networks (WSNs) are widely implemented in military, intelligent medical, intelligent transportation, space exploration, and other fields. However, the authentication and communication of WSNs are carried out in the harsh external environment via a public channel, which is more vulnerable to various attacks than the traditional networks. Authentication is the key technology of security measures, but a common authentication scheme is not suitable for WSNs due to limitations of memory, computing, and energy consumption. Therefore, designing a secure and efficient authentication scheme is essential in WSNs. In recent years, several studies proved that the dynamic authentication credential (DAC) is efficient for enhancing the security of the authentication scheme. This article designs a secure authentication scheme for WSNs based on DAC and Intel software guard extensions (SGX). Compared with other DAC authentication schemes, our scheme provides the confirmation action of DAC rotation, which can effectively prevent the asynchronous update problem caused by packet loss. In order to resist the privileged user attack and the authentication table leakage attack, we choose the SGX, which can protect the data in use, as the trusted execution environment in the gateway node, and we adopt SGX to store the master key for protecting the authentication table. Finally, the security of our authentication scheme is verified by BAN logic, the simulation tool AVISPA, and informal security analysis. Through the consumption overhead analysis, the NS3 simulation result, and detailed comparison with other recent schemes, we conclude that our scheme is practical and achieves better security with less overhead.**

***Index Terms*—Confidential computing, cyber security, dynamic authentication credential (DAC), software guard extensions (SGX), wireless sensor networks (WSNs).**

I. Introduction

With the development of communication technology and the widely used Internet of Things (IoT) [1], wireless sensor networks (WSNs) have developed rapidly as the key technology of the IoT [2]. In the past few years, WSNs have been widely implemented in many fields, such as environment, military, intelligent medical [3], intelligent transportation, space exploration, and other fields [4] because of their low cost, easy deployment, and high flexibility. WSNs are generally composed of remote users, gateway nodes (GWNs), and multiple sensor nodes (SNs), which are randomly distributed in a certain area [5]. The SNs are responsible for collecting and storing information and the GWN manages all the SNs to provide services to legitimate users. As the authentication and communication of WSNs are carried out in the public channel [6], WSNs are more vulnerable to tracking attacks, replay attacks, eavesdropping attacks, and other kinds of attacks [7] from an adversary than traditional communications networks [8]. Moreover, the SNs of WSNs are usually deployed in a harsh external environment and limited by memory, computation, and energy consumption. Therefore, it is particularly important to design an efficient and secure authentication scheme in WSNs to achieve identity authentication and secure communication in the public channel.

However, most of the research only implements identity authentication and key agreement. These schemes do not dynamically rotate the credential parameters stored in a smart card, GWN, and SN. The consequence is that the users and the SN will not recognize whether their authentication credentials have been leaked. If the damaged authentication credentials are not updated in time, it may lead to disastrous consequences [9]. Similarly, the traditional scheme based on dynamic authentication credentials (DACs) does not consider the actual problem of packet loss. Once the packet is lost, the credential update is not synchronized and will lead to the next login failure. Due to the practical and security requirements, WSNs need to provide authentication between multiple users and multiple SNs. Therefore, the authentication schemes based on DAC often need to store the authentication table that maps roles to access in the memory of the GWN. However, the authentication table is vulnerable to the privileged user attack and table lost attack. Once the authentication table is obtained by an adversary or privileged users become the adversary, the adversary can guess the credential and sensitive information of other legal users and launch an impersonation attack. With the increase in software complexity, the adversary can use malware such as Trojans to obtain authentication tables and secret keys. Therefore, a hardware-based trusted execution environment (TEE) for storing the authentication table and secret keys is indispensable. Several researchers have adopted a trusted platform module (TPM) to protect the sensitive

Manuscript received February 10, 2021; revised May 28, 2021 and June 30, 2021; accepted July 13, 2021. Date of publication July 19, 2021; date of current version February 21, 2022. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2020-6; in part by the Gansu Province Key Research and Development under Grant 20YF3FA024; and in part by the Lanzhou Science and Technology Project Funding under Grant 2019-4-47. *(Corresponding author: Xin Liu.)*

Xin Liu and Jun Ma are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China (e-mail: xinl@lzu.edu.cn; junma@lzu.edu.cn).

Zhenbin Guo is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China (e-mail: guozhb19@lzu.edu.cn).

Yuchen Song is with the Department of Information Technology Laboratory, Communication and Information Center, State Administration of Work Safety, Beijing 100013, China (e-mail: songych2020@163.com).

Digital Object Identifier 10.1109/JIOT.2021.3097996

information in the GWN. However, the most widely used TEE, TPM, is cumbersome, inefficient, and unable to protect the data in use. Unfortunately, the information and secret authentication credentials are always in use. Therefore, TPM is not suitable for protecting the authentication tables and credentials.

In this article, we adopt the Intel software guard extensions (SGX) and DAC together to improve the security of the authentication table. SGX is a secure chip that can be used for protecting data in use from the computing environment. As the SGX is limited by the storage and I/O capacity, we only use SGX to store the master GWN key, which is called through the security interface. If an adversary who obtained the authentication table stored in the GWN would be unable to obtain the authentication credential and sensitive information because the dynamic parameter stored information is encrypted by the master key stored in SGX. Hence, our scheme can resist both the authentication table leakage attack and privileged user attack. Finally, through the detailed proof and comparison with recent schemes, we prove that our authentication scheme achieves a better tradeoff between security and performance.

II. Related Work

Since the birth of WSNs, many experts and researchers have proposed many kinds of identity authentication schemes to protect user privacy and communication security in WSNs. In this section, we review the relevant work of these schemes.

In the beginning, Das [10] designed a typical two-factor authentication protocol that integrates a user password and smart card information for user authentication. Das proposed that this scheme could achieve better security and efficiency authentication than earlier schemes. However, Khan and Alghathbar [11] pointed out that the scheme designed by Das was vulnerable to privileged user attacks and gateway bypassing attacks. Similarly, Vaidya *et al.* [12] have proved that the schemes of Khan *et al.* cannot resist stolen smart card attacks. Kim *et al.* [13] designed an authentication protocol that stored information with ciphertext in a smart card to resist stolen smart card attacks. However, Li *et al.* [14] proposed that the authentication scheme of Kim *et al.* cannot resist offline guessing attacks and SN impersonation attacks. To achieve secure mutual authentication and anonymity, Yu *et al.* [15] designed a protocol based on a dynamic session key that could effectively resist man-in-the-middle attacks and replay attacks from an adversary. Unfortunately, Sadri and Asaar [16] pointed that the scheme of Yu *et al.* is easy to attack by offline guessing attacks and impersonation attacks. To improve the safety and efficiency of the WSNs, Amin *et al.* [17] designed a three-factor authenticated scheme with anonymity preserving for WSNs and added biometric factors into the authentication process. Amin's scheme not only solves the vulnerability of offline guessing attacks but also solves the shortcomings of some cryptographic attacks. Unfortunately, Ostad-Sharif *et al.* [18] pointed out that the scheme of Amin *et al.* cannot resist replay attacks and forward secrecy is not enough. Therefore, Ostad-Sharif *et al.* designed a lightweight security scheme to solve these problems. Afterward, Chen *et al.* [19] pointed that the scheme of Ostad-Sharif *et al.* also has many flaws in the authentication and key agreement phase. In response to these deficiencies, Chen *et al.* designed a security authentication scheme which is based on time credentials and dynamic identity. Likewise, Chang and Le [20] designed an authentication scheme that can provide sufficient forward secrecy. Then, Amin *et al.* [21] pointed that the scheme of Chang *et al.* is vulnerable to user tracking attacks, offline guessing attacks, and stolen smart card attacks. Ultimately, Amin *et al.* designed an untraceable anonymous three-factor authentication scheme that was based on the scheme of Chang *et al.* and also proposed a new realistic architecture for WSNs.

In recent years, many studies proved that the DAC is an efficient measure for improving the security of authentication schemes. Chang *et al.* [22] presented an authentication scheme with dynamic identity. However, Yang *et al.* [23] showed that the scheme of Chang *et al.* cannot provide enough forward secrecy and also has high computation overhead. Therefore, Yang *et al.* designed an authentication scheme with DACs. The authentication certificate is updated periodically in each session, and the scheme only requires a one-way hash function and an XOR operation. However, with the increase of software complexity and more sophisticated attackers, a hardware TEE must be introduced to improve the security of WSNs. Agrawal *et al.* [24] proposed a new program integrity verification scheme based on TPM, which verifies identity by comparing the program memory content of the SN before and after capture. Similarly, Fu and Peng [25] presented a hardware-based remote authentication scheme. In their scheme, each SN is equipped with a TPM. Meanwhile, Tan *et al.* [26] proposed and implemented a remote authentication scheme for detecting unauthorized tampering in the application codes running on SNs with the assistance of TPM. Unfortunately, it was not difficult to determine that TPM can only provide static protection, which is not suitable for the DAC update operation in WSNs.

Therefore, many researchers have focused on SGX technology, which is more suitable for WSNs. In the first instance, SGX is the key content of confidential computing, so Balisane and Martin [27] presented a novel authentication method based on SGX technology using a TEE. This method solves the problems related to the wrong SSL certificate and DNS poisoning on the local device. Condé *et al.* [28] combined SGX technology with identity authentication and designed and implemented an authentication module based on SGX. This module uses the enclave of SGX to process the credentials of user notification and checks the credentials according to the password file. It can provide higher security performance with a lower performance overhead. Sun and Xiao [29] proposed a dynamic network authentication using SGX that uses the random number in the network communication to constantly change the key and uses the SGX mechanism to protect the identity certificate.

In addition, hardware-based TEEs are more and more widely used, and TEE-based credential management methods are becoming more and more mature. Kostiainen *et al.* [30]

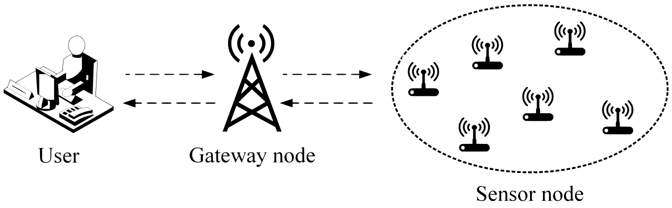


Fig. 1. Network model of WSNs.

designed a server-based credential disabling solution for mobile phones with M-Shield TEE. Kostiainen *et al.* [31] proposed a practical and friendly credential transmission scheme for the lack of a secure user input mechanism on TEE. Kostiainen and Asokan [32] proposed an open architecture for remote provisioning of secure credentials in TEE. Marfario *et al.* [33] pointed out the problem of secure user registration for system-wide TEEs, like ARM TrustZone, and outlined the possible architectural changes in the future. Arfaoui *et al.* [34] built a migration protocol of TEE profiles ensuring its confidentiality and integrity. Shepherd *et al.* [35] proposed a trusted channel protocol based on GlobalPlatform TEE. Shepherd *et al.* [36] designed an efficient protocol for secure remote credential management using TEE. Omori and Yamashita [37] proposed a credential sharing method to improve the availability of equipment in TEE.

III. Preliminaries

A. Network Model

This section introduces the structure of WSNs. Being bound by the resource and environment of the SN, communication energy consumption is the major concern of network designers. According to [38], the communication energy consumption of SNs not only depends on the size of the transmission information but is also closely related to the communication distance. The communication energy consumption is proportional to the distance between two nodes. Therefore, we should avoid having the user directly communicate with the SN in the practical application. Amin and Biswas [39] selected the GWN as the intermediate authentication medium to reduce the energy consumption of the SN and improve communication security. In addition, our research focuses on the case of a single GWN. According to the above analysis, we chose the network structure model as the basis for our research, shown in Fig. 1.

The network model includes three communication entities: 1) user; 2) GWN; and 3) SN. Users can obtain information from SNs through GWN. The GWN is responsible for the information transmission between users and SNs after completing the authentication. The SN collects the environment information and transmits it to GWN. In WSNs, GWNs and SNs remain relatively static. We assume that the computation resources, storage resources, and energy consumption of user and GWN can be replenished in time. Conversely, the computation, storage, communication, and energy consumption of SN are limited and the resources are inconvenient to supplement over time.

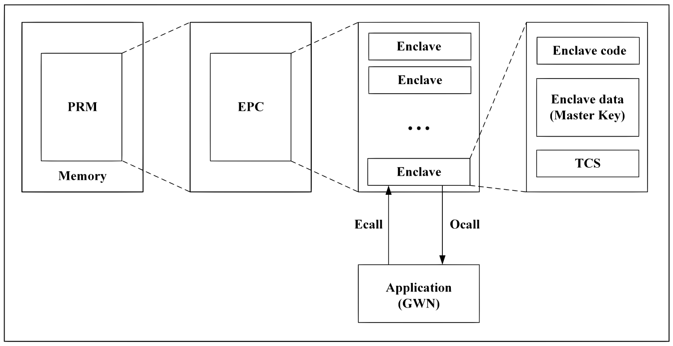


Fig. 2. Architecture of SGX.

B. Intel SGX

Presently, many experts are committed to improving the security of WSNs. However, with the increase in software complexity and attack levels [40], the need for security mechanisms in WSNs is more urgent [41]. A hardware-based, TEE is indispensable.

TPM is a widely used TEE [24], [25], [42] that can provide signature, encryption, and decryption functions. It mainly uses trust chain and integrity measurement technology to ensure the trustworthiness of the operating system and applications. However, this kind of protection is static and cannot protect the security of private data such as a master key in dynamic system memory [43]–[44]. In addition, TPM access is cumbersome and inefficient. Therefore, TPM is not suitable for an authentication scheme based on dynamic credential rotation.

Unlike TPM hardware security technology, the SGX launched by Intel has more advantages. First, the SGX uses memory encryption technology to protect the security of the program's running state, which makes it more difficult to obtain the key information through a memory leak attack. Meanwhile, the SGX reduces the trusted computing base of the system to the CPU, which avoids the harm of more system attacks [45]–[47]. In detail, the core of SGX is a secure container called an enclave, which can store sensitive data and computing code of the application to achieve confidentiality and integrity protection [48]. As shown in Fig. 2, the enclave is stored in a protected physical memory [49] area enclave page cache (EPC). EPC is a part of processor reserved memory (PRM), and the PRM is a part of memory. SGX divides the application into two parts, trusted and untrusted, and these two parts are called by predefined Ecall and Ocall codes. The code and data in the enclave are always encrypted and stored in the memory during operation [50]. Corresponding to the application in this article, GWN uses the master key stored in SGX through Ecall. After SGX completes the confidential computation in the enclave, it returns the computation result through the Ocall. In this way, SGX can effectively provide authentication and master key protection for this scheme. Therefore, we choose SGX to store the GWN master key to enhance the security properties of our scheme for WSNs based on DAC.

C. Threat Model

In this article, we choose the widely accepted Dolev–Yao (DY) threat model [51] as the basis for security analysis. The

TABLE I
NOTATIONS IN THIS ARTICLE

Notation	Definition
U_i, SC, S_j	The registered user, smart card, registered sensor node, respectively.
ID_i, ID_{GWN}, ID_j	The identity of U_i, identity of GWN, identity of S_j, respectively.
PW_i	The password of U_i.
TS	The current timestamp.
K_U, K_{SN}	The master key between U_i and GWN, between GWN and sensor node S_j, respectively.
r_i	The random number generated in user registration phase.
q_i	The validation parameters used in the authentication and key exchange phase.
KEY_{ij}	The temporary encryption key negotiated for the future communication between U_i and S_j.
$h(\cdot), \Vert, \oplus$	The one-way hash function, bitwise concatenation operation, XOR operation, respectively.

DY model allows adversary *A* to intercept, retransmit, modify, delete, or even insert false messages during the communication between two communicating entities through the public channel. In addition, IoT SNs are often deployed in external environments, and can be captured and cracked by the adversary to obtain stored information. Similarly, the adversary can obtain information stored in the smart card by sophisticated power analysis attacks [52]. Especially, the registration phase is carried out under a secure channel by default, and other phases are carried out under a public channel. Finally, this article considers SGX as a TEE in the authentication and key exchange phase and does not consider microarchitecture attacks against SGX, such as foreshadow attacks [53] and software-based fault injection attacks [54].

D. Notation in This Article

The notations used in this article are described in Table I.

IV. Our Proposed Scheme

In this section, we design a secure authentication scheme for WSNs based on DAC and Intel SGX. Our authentication scheme consists of the initialization phase, registration phase, login phase, authentication and key exchange phase, dynamic credentials update phase, and password change phase. In addition, our scheme not only implements DAC rotation but also negotiates a temporary session key KEY_{ij} under the protection of SGX. The session key KEY_{ij} is randomly generated and dynamically changed for each session. It can encrypt the communication data between the user U_i and the SN S_j to improve the security of the communication. The details of our authentication scheme are as follows.

A. Initialization Phase

Before WSNs are deployed and used, the WSNs need to write some basic computation functions and preshared parameters to smart cards, GWNs, and SNs in advance. This phase is executed in a secure offline environment and the detailed steps are as follows.

1) The WSNs write $\{h(\cdot), \oplus, \Vert\}$ into smart cards, GWNs, and SNs.
2) The WSNs allocate ID_{SC} and a random number RE_i to smart card and store the registration table $\{ID_{SC}, RE_i\}$ in the GWN.

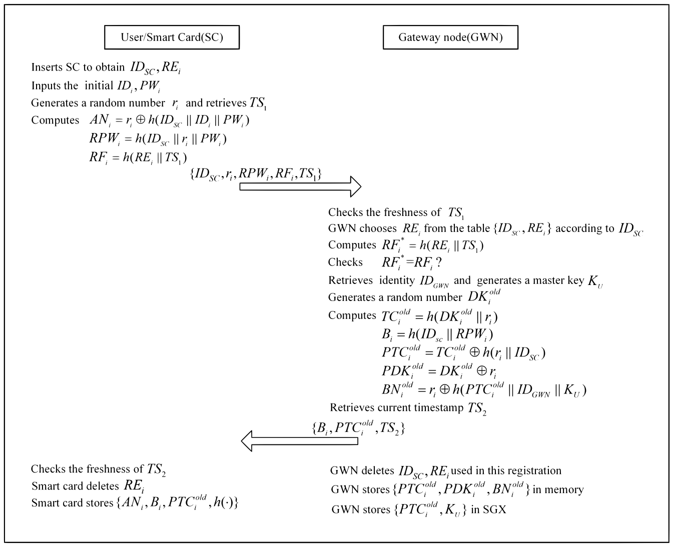

User/Smart Card(SC)
Gateway node(GWN)
Inserts SC to obtain ID_{SC}, RE_i
Inputs the initial ID_i, PW_i
Generates a random number r_i and retrieves TS_1
Computes $AN_i = r_i \oplus h(ID_{SC} \Vert ID_i \Vert PW_i)$
$RPW_i = h(ID_{SC} \Vert r_i \Vert PW_i)$
$RF_i = h(RE_i \Vert TS_1)$
$\{ID_{SC}, r_i, RPW_i, RF_i, TS_1\}$
Checks the freshness of TS_1
GWN chooses RE_i from the table $\{ID_{SC}, RE_i\}$ according to ID_{SC}
Computes $RF_i^* = h(RE_i \Vert TS_1)$
Checks $RF_i^* = RF_i$?
Retrieves identity ID_{GWN} and generates a master key K_U
Generates a random number DK_i^{old}
Computes $TC_i^{old} = h(DK_i^{old} \Vert r_i)$
$B_i = h(ID_{sc} \Vert RPW_i)$
$PTC_i^{old} = TC_i^{old} \oplus h(r_i \Vert ID_{SC})$
$PDK_i^{old} = DK_i^{old} \oplus r_i$
$BN_i^{old} = r_i \oplus h(PTC_i^{old} \Vert ID_{GWN} \Vert K_U)$
Retrieves current timestamp TS_2
$\{B_i, PTC_i^{old}, TS_2\}$
Checks the freshness of TS_2
Smart card deletes RE_i
Smart card stores $\{AN_i, B_i, PTC_i^{old}, h(\cdot)\}$
GWN deletes ID_{SC}, RE_i used in this registration
GWN stores $\{PTC_i^{old}, PDK_i^{old}, BN_i^{old}\}$ in memory
GWN stores $\{PTC_i^{old}, K_U\}$ in SGX

Fig. 3. User registration phase.

B. Registration Phase

The registration phase consists of the user registration phase and the SN registration phase, and is executed under the secure channel by default. In addition, in order to balance the performance and security of confidential computing, the master key which stored in SGX uses a widely used key length 1024 bits. We describe the registration phase for the user in Fig. 3 and the detailed processes are as follows.

1) At the beginning of registration, the user U_i inserts their SC to obtain the identity ID_{SC} and RE_i. Then, the user U_i inputs their identification ID_i and PW_i. Similarly, the SC generates a random number r_i and retrieves the timestamp TS_1. Upon obtaining these parameters, SC computes $AN_i = r_i \oplus h(ID_{SC} \Vert ID_i \Vert PW_i)$, $RPW_i = h(ID_{SC} \Vert r_i \Vert PW_i)$, and $RF_i = h(RE_i \Vert TS_1)$.
2) The SC transmits $\{ID_{SC}, r_i, RPW_i, RF_i, TS_1\}$ to GWN.
3) In the first place, the GWN checks the timestamp TS_1. If TS_1 has already expired, the GWN rejects this registration. On the contrary, the GWN chooses RE_i from the registration table $\{ID_{SC}, RE_i\}$ according to ID_{SC}. Then, GWN computes $RF_i^* = h(RE_i \Vert TS_1)$ and checks $RF_i^* = RF_i$? If $RF_i^* \neq RF_i$, the GWN rejects this registration. Otherwise, the GWN accepts this registration request and retrieves the ID_{GWN}. Then, GWN generates a 1024 bits master key K_U and generates a random

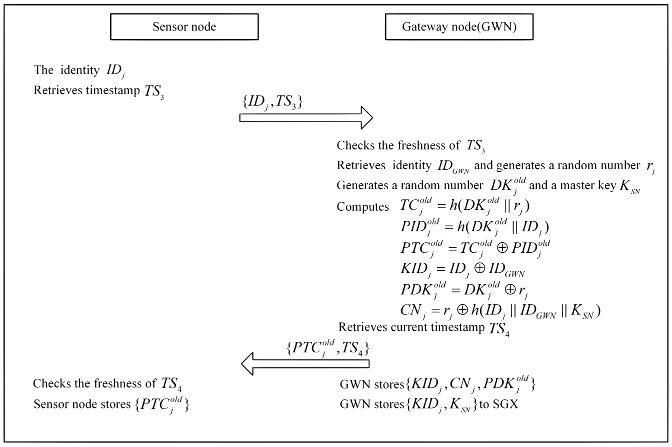

Sensor node
Gateway node(GWN)
The identity ID_j
Retrieves timestamp TS_3
$\{ID_j, TS_3\}$
Checks the freshness of TS_3
Retrieves identity ID_{GWN} and generates a random number r_j
Generates a random number DK_j^{old} and a master key K_{SN}
Computes $TC_j^{old} = h(DK_j^{old} \| r_j)$
$PID_j^{old} = h(DK_j^{old} \| ID_j)$
$PTC_j^{old} = TC_j^{old} \oplus PID_j^{old}$
$KID_j = ID_j \oplus ID_{GWN}$
$PDK_j^{old} = DK_j^{old} \oplus r_j$
$CN_j = r_j \oplus h(ID_j \| ID_{GWN} \| K_{SN})$
Retrieves current timestamp TS_4
$\{PTC_j^{old}, TS_4\}$
Checks the freshness of TS_4
Sensor node stores $\{PTC_j^{old}\}$
GWN stores $\{KID_j, CN_j, PDK_j^{old}\}$
GWN stores $\{KID_j, K_{SN}\}$ to SGX

Fig. 4. SN registration phase.

number $DK_i{}^{\text{old}}$. After getting these parameters, GWN computes $TC_i^{\text{old}} = h(DK_i^{\text{old}} \| r_i)$, $B_i = h(ID_{sc} \| RPW_i)$, $PTC_i^{\text{old}} = TC_i^{\text{old}} \oplus h(r_i \| ID_{SC})$, $PDK_i^{\text{old}} = DK_i^{\text{old}} \oplus r_i$, and $BN_i^{\text{old}} = r_i \oplus h(PTC_i^{\text{old}} \| ID_{\text{GWN}} \| K_U)$. GWN retrieves the timestamp TS_2 after completing calculations.

4) The GWN deletes $\{ID_{SC}, RE_i\}$ used in this registration. Then, the GWN stores $\{PTC_i^{\text{old}}, PDK_i^{\text{old}}, BN_i^{\text{old}}\}$ in the memory and stores PTC_i^{old}, K_U in SGX. Then, the GWN transmits $\{B_i, PTC_i^{\text{old}}, TS_2\}$ to SC.
5) After SC receiving the parameters from GWN, the SC checks the freshness of TS_2. If the timestamp is overtime, the SC rejects this authentication. Otherwise, SC deletes RE_i and stores the secret parameters $\{AN_i, B_i, PTC_i^{\text{old}}, h(\cdot)\}$.

Equally, the SN registration phase is described in Fig. 4 and the detailed processes are as follows.

1) The SN S_j obtains identity ID_j and retrieves the current timestamp TS_3. Then, the SN S_j transmits $\{ID_j, TS_3\}$ to the GWN.
2) Upon receiving the parameters from S_j, the GWN checks the freshness of TS_3. After verification, the GWN retrieves the identifier ID_{GWN} and generates three random number r_j, DK_j^{old}, and K_{SN}, which the length of K_{SN} is 1024 bits. Then, the GWN computes $TC_j^{\text{old}} = h(DK_j^{\text{old}} \| r_j)$, $PID_j^{\text{old}} = h(DK_j^{\text{old}} \| ID_j)$, $PTC_j^{\text{old}} = TC_j^{\text{old}} \oplus PID_j^{\text{old}}$, $KID_j = ID_j \oplus ID_{\text{GWN}}$, $PDK_j^{\text{old}} = DK_j^{\text{old}} \oplus r_j$ and $CN_j = r_j \oplus h(ID_j \| ID_{\text{GWN}} \| K_{SN})$.
3) The GWN stores $\{KID_j, PDK_j^{\text{old}}, CN_j\}$ in the memory and stores $\{KID_j, K_{SN}\}$ in SGX.
4) The GWN retrieves current timestamp TS_4 and transmits $\{PTC_j^{\text{old}}, TS_4\}$ to S_j.
5) Upon receiving the parameters, S_j checks the timestamp TS_4. After passing the verification, S_j stores $\{PTC_j^{\text{old}}\}$.

C. Login Phase

Before the user U_i communicate with GWN and SN S_j, it is important to verify the legitimacy of U_i. We described the detail of the login phase in Fig. 5.

1) First, the user U_i inputs their ID_i and PW_i after inserts SC. The SC obtains the stored information and computes $r_i = AN_i \oplus h(ID_{SC} \| ID_i \| PW_i)$, $RPW_i = h(ID_{SC} \| r_i \| PW_i)$, and $B_i^* = h(ID_{SC} \| RPW_i)$.
2) The SC checks $B_i^* = B_i$? If it is not equal, the SC refuses the user login request. If it is equal, the validity of U_i is verified.

D. Authentication and Key Exchange Phase

In this phase, U_i, GWN, and S_j complete mutual authentication. Correspondingly, U_i and S_j, respectively, generate a session key KEY_{ij} for communication. The details of this phase are shown in Fig. 5.

1) After user U_i completes login, U_i inputs ID_j of the target SN and generates a random nonce N_i. In addition, SC retrieves current timestamp TS_1 and computes $TC_i^{\text{old}} = PTC_i^{\text{old}} \oplus h(r_i \| ID_{SC})$, $q_1 = h(TC_i^{\text{old}} \| ID_j \| N_i \| r_i)$, $PKS_i = N_i \oplus h(TC_i^{\text{old}} \| r_i \| TS_1)$, $PID_j = ID_j \oplus h(TC_i^{\text{old}} \| TS_1 \| N_i)$.
2) The SC sends the message $m_1 = \{q_1, PKS_i, PID_j, PTC_i^{\text{old}}, TS_1\}$ to GWN.
3) Upon receiving parameters m_1 from SC, the GWN checks the validity of TS_1. If TS_1 has already overtime, GWN refuses this authentication request. On the contrary, GWN chooses $\{PDK_i^{\text{old}}, BN_i^{\text{old}}\}$ from the table $\{PTC_i^{\text{old}}, PDK_i^{\text{old}}, BN_i^{\text{old}}\}$ according to PTC_i^{old} and sends $ID_{\text{GWN}}, BN_i^{\text{old}}, PTC_i^{\text{old}}$ to the security interface of SGX. The interface selects K_U according to PTC_i^{old} and computes $r_i = BN_i^{\text{old}} \oplus h(PTC_i^{\text{old}} \| ID_{\text{GWN}} \| K_U)$. The GWN computes $DK_i^{\text{old}} = PDK_i^{\text{old}} \oplus r_i$, $TC_i^{\text{old}} = h(DK_i^{\text{old}} \| r_i)$, $N_i = PKS_i \oplus h(TC_i^{\text{old}} \| r_i \| TS_1)$, $ID_j = PID_j \oplus h(TC_i^{\text{old}} \| TS_1 \| N_i)$, $KID_j = ID_j \oplus ID_{\text{GWN}}$, and $q_1^* = h(TC_i^{\text{old}} \| ID_j \| N_i \| r_i)$. Then, the GWN checks $q_1^* = q_1$? If $q_1^* \neq q_1$, the GWN terminates this request and transmits a reject information to SC. If $q_1^* = q_1$, GWN accepts this authentication request.
4) After GWN accepts the authentication request, GWN chooses $\{PDK_j^{\text{old}}, CN_j\}$ from the table $\{KID_j, PDK_j^{\text{old}}, CN_j\}$ according to KID_j and sends $\{KID_j, CN_j\}$ to the security interface of SGX. The interface selects K_{SN} according to KID_j and computes $r_j = CN_j \oplus h(ID_j \| ID_{\text{GWN}} \| K_{SN})$. Then, the GWN computes $DK_j^{\text{old}} = PDK_j^{\text{old}} \oplus r_j$, $TC_j^{\text{old}} = h(DK_j^{\text{old}} \| r_j)$, $PID_j^{\text{old}} = h(DK_j^{\text{old}} \| ID_j)$, $q_2 = h((TC_j^{\text{old}} \oplus N_i) \| ID_j)$. In the same way, GWN retrieves the timestamp TS_2 and computes $PKS_N = N_i \oplus h(TC_j^{\text{old}} \| ID_j \| TS_2)$.
5) The GWN sends the message $m_2 = \{q_2, PKS_N, PID_j^{\text{old}}, TS_2\}$ to S_j.
6) Upon receiving message m_2, S_j checks the timestamp TS_2. If TS_2 is not fresh, S_j rejects the request from GWN. Otherwise, S_j calculates $TC_j^{\text{old}} = PTC_j^{\text{old}} \oplus PID_j^{\text{old}}$, $N_i = PKS_N \oplus h(TC_j^{\text{old}} \| ID_j \| TS_2)$ and $q_2^* = h((TC_j^{\text{old}} \oplus N_i) \| ID_j)$. Then, S_j checks $q_2^* = q_2$? If $q_2^* \neq q_2$, S_j rejects the request from GWN. If $q_2^* = q_2$, the legitimacy of the GWN is ensured. Moreover, S_j generates a new random number K_j and computes $q_3 = h(N_i \| K_j \| (TC_j^{\text{old}} \oplus ID_j))$. After computation, S_j retrieves the current timestamp

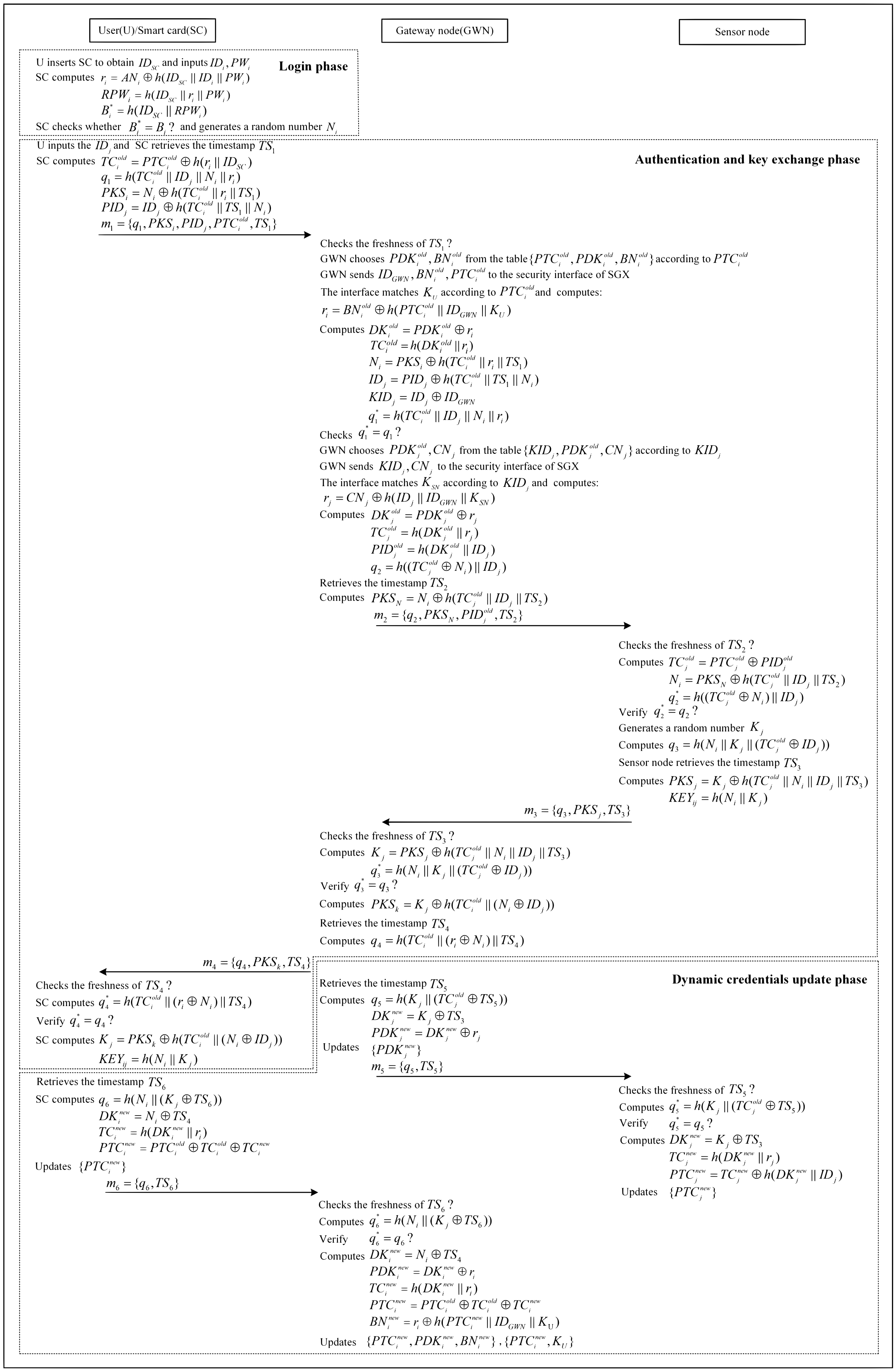

User(U)/Smart card(SC)
Gateway node(GWN)
Sensor node
Login phase
U inserts SC to obtain ID_{SC} and inputs ID_i, PW_i
SC computes $r_i = AN_i \oplus h(ID_{SC} \| ID_i \| PW_i)$
$RPW_i = h(ID_{SC} \| r_i \| PW_i)$
$B_i^* = h(ID_{SC} \| RPW_i)$
SC checks whether $B_i^* = B_i$? and generates a random number N_i
Authentication and key exchange phase
U inputs the ID_j and SC retrieves the timestamp TS_1
SC computes $TC_i^{old} = PTC_i^{old} \oplus h(r_i \| ID_{SC})$
$q_1 = h(TC_i^{old} \| ID_j \| N_i \| r_i)$
$PKS_i = N_i \oplus h(TC_i^{old} \| r_i \| TS_1)$
$PID_j = ID_j \oplus h(TC_i^{old} \| TS_1 \| N_i)$
$m_1 = \{q_1, PKS_i, PID_j, PTC_i^{old}, TS_1\}$
Checks the freshness of TS_1?
GWN chooses PDK_i^{old}, BN_i^{old} from the table $\{PTC_i^{old}, PDK_i^{old}, BN_i^{old}\}$ according to PTC_i^{old}
GWN sends $ID_{GWN}, BN_i^{old}, PTC_i^{old}$ to the security interface of SGX
The interface matches K_U according to PTC_i^{old} and computes:
$r_i = BN_i^{old} \oplus h(PTC_i^{old} \| ID_{GWN} \| K_U)$
Computes $DK_i^{old} = PDK_i^{old} \oplus r_i$
$TC_i^{old} = h(DK_i^{old} \| r_i)$
$N_i = PKS_i \oplus h(TC_i^{old} \| r_i \| TS_1)$
$ID_j = PID_j \oplus h(TC_i^{old} \| TS_1 \| N_i)$
$KID_j = ID_j \oplus ID_{GWN}$
$q_1^* = h(TC_i^{old} \| ID_j \| N_i \| r_i)$
Checks $q_1^* = q_1$?
GWN chooses PDK_j^{old}, CN_j from the table $\{KID_j, PDK_j^{old}, CN_j\}$ according to KID_j
GWN sends KID_j, CN_j to the security interface of SGX
The interface matches K_{SN} according to KID_j and computes:
$r_j = CN_j \oplus h(ID_j \| ID_{GWN} \| K_{SN})$
Computes $DK_j^{old} = PDK_j^{old} \oplus r_j$
$TC_j^{old} = h(DK_j^{old} \| r_j)$
$PID_j^{old} = h(DK_j^{old} \| ID_j)$
$q_2 = h((TC_j^{old} \oplus N_i) \| ID_j)$
Retrieves the timestamp TS_2
Computes $PKS_N = N_i \oplus h(TC_j^{old} \| ID_j \| TS_2)$
$m_2 = \{q_2, PKS_N, PID_j^{old}, TS_2\}$
Checks the freshness of TS_2?
Computes $TC_j^{old} = PTC_j^{old} \oplus PID_j^{old}$
$N_i = PKS_N \oplus h(TC_j^{old} \| ID_j \| TS_2)$
$q_2^* = h((TC_j^{old} \oplus N_i) \| ID_j)$
Verify $q_2^* = q_2$?
Generates a random number K_j
Computes $q_3 = h(N_i \| K_j \| (TC_j^{old} \oplus ID_j))$
Sensor node retrieves the timestamp TS_3
Computes $PKS_j = K_j \oplus h(TC_j^{old} \| N_i \| ID_j \| TS_3)$
$KEY_{ij} = h(N_i \| K_j)$
$m_3 = \{q_3, PKS_j, TS_3\}$
Checks the freshness of TS_3?
Computes $K_j = PKS_j \oplus h(TC_j^{old} \| N_i \| ID_j \| TS_3)$
$q_3^* = h(N_i \| K_j \| (TC_j^{old} \oplus ID_j))$
Verify $q_3^* = q_3$?
Computes $PKS_k = K_j \oplus h(TC_i^{old} \| (N_i \oplus ID_j))$
Retrieves the timestamp TS_4
Computes $q_4 = h(TC_i^{old} \| (r_i \oplus N_i) \| TS_4)$
$m_4 = \{q_4, PKS_k, TS_4\}$
Checks the freshness of TS_4?
SC computes $q_4^* = h(TC_i^{old} \| (r_i \oplus N_i) \| TS_4)$
Verify $q_4^* = q_4$?
SC computes $K_j = PKS_k \oplus h(TC_i^{old} \| (N_i \oplus ID_j))$
$KEY_{ij} = h(N_i \| K_j)$
Dynamic credentials update phase
Retrieves the timestamp TS_5
Computes $q_5 = h(K_j \| (TC_j^{old} \oplus TS_5))$
$DK_j^{new} = K_j \oplus TS_3$
$PDK_j^{new} = DK_j^{new} \oplus r_j$
Updates $\{PDK_j^{new}\}$
$m_5 = \{q_5, TS_5\}$
Checks the freshness of TS_5?
Computes $q_5^* = h(K_j \| (TC_j^{old} \oplus TS_5))$
Verify $q_5^* = q_5$?
Computes $DK_j^{new} = K_j \oplus TS_3$
$TC_j^{new} = h(DK_j^{new} \| r_j)$
$PTC_j^{new} = TC_j^{new} \oplus h(DK_j^{new} \| ID_j)$
Updates $\{PTC_j^{new}\}$
Retrieves the timestamp TS_6
SC computes $q_6 = h(N_i \| (K_j \oplus TS_6))$
$DK_i^{new} = N_i \oplus TS_4$
$TC_i^{new} = h(DK_i^{new} \| r_i)$
$PTC_i^{new} = PTC_i^{old} \oplus TC_i^{old} \oplus TC_i^{new}$
Updates $\{PTC_i^{new}\}$
$m_6 = \{q_6, TS_6\}$
Checks the freshness of TS_6?
Computes $q_6^* = h(N_i \| (K_j \oplus TS_6))$
Verify $q_6^* = q_6$?
Computes $DK_i^{new} = N_i \oplus TS_4$
$PDK_i^{new} = DK_i^{new} \oplus r_i$
$TC_i^{new} = h(DK_i^{new} \| r_i)$
$PTC_i^{new} = PTC_i^{old} \oplus TC_i^{old} \oplus TC_i^{new}$
$BN_i^{new} = r_i \oplus h(PTC_i^{new} \| ID_{GWN} \| K_U)$
Updates $\{PTC_i^{new}, PDK_i^{new}, BN_i^{new}\}, \{PTC_i^{new}, K_U\}$

Fig. 5. Login, authentication and key exchange, and dynamic credentials update phases.

TS_3 and computes $PKS_j = K_j \oplus h(TC_j^{\text{old}} \| N_i \| ID_j \| TS_3)$ and $KEY_{ij} = h(N_i \| K_j)$.

7) S_j transmits the parameters $m_3 = \{q_3, PKS_j, TS_3\}$ to GWN.

8) After receiving the parameters m_3, GWN checks the freshness of TS_3. Then, GWN computes $K_j = PKS_j \oplus h(TC_j^{\text{old}} \| N_i \| ID_j \| TS_3)$ and $q_3^* = h(N_i \| K_j \| (TC_j^{\text{old}} \oplus ID_j))$ to obtain the new random number K_j from S_j. GWN

judges $q_3^* = q_3$? If $q_3^* = q_3$, GWN retrieves current timestamp TS_4 and computes $PKS_k = K_j \oplus h(TC_i^{\text{old}} \| (N_i \oplus ID_j))$, $q_4 = h(TC_i^{\text{old}} \| (r_i \oplus N_i) \| TS_4)$.

9) The GWN sends the message $m_4 = \{q_4, PKS_k, TS_4\}$ to U_i.

10) Similarly, U_i checks the validity of TS_4. If TS_4 has already overtime, U_i refuses this request. Otherwise, SC computes $q_4^* = h(TC_i^{\text{old}} \| (r_i \oplus N_i) \| TS_4)$. Then, SC checks $q_4^* = q_4$? After passing this judgment, SC computes $K_j = PKS_k \oplus h(TC_i^{\text{old}} \| (N_i \oplus ID_j))$ and $KEY_{ij} = h(N_i \| K_j)$.

E. Dynamic Credentials Update Phase

After the session key KEY_{ij} successfully calculated between U_i and S_j, the credentials in authentication are updated dynamically. In Fig. 5, the detailed processes of dynamic credentials update phase are as follows.

1) The GWN retrieves the timestamp TS_5 and computes $q_5 = h(K_j \| (TC_j^{\text{old}} \oplus TS_5))$, $DK_j^{\text{new}} = K_j \oplus TS_3$, $PDK_j^{\text{new}} = DK_j^{\text{new}} \oplus r_j$. After calculation, updates information $\{PDK_j^{\text{new}}\}$ and sends the parameters $m_5 = \{q_5, TS_5\}$ to S_j.

2) Upon receiving m_5 from GWN, the SN S_j checks the timestamp TS_5. If TS_5 is not fresh, S_j rejects the update request from GWN. Instead, S_j computes $q_5^* = h(K_j \| (TC_j^{\text{old}} \oplus TS_5))$. If $q_5^* = q_5$, S_j calculates $DK_j^{\text{new}} = K_j \oplus TS_3$, $TC_j^{\text{new}} = h(DK_j^{\text{new}} \| r_j)$ and $PTC_j^{\text{new}} = TC_j^{\text{new}} \oplus h(DK_j^{\text{new}} \| ID_j)$. Then, S_j update $\{PTC_j^{\text{new}}\}$ in memory.

3) Meanwhile, SC retrieves timestamp TS_6 and computes $q_6 = h(N_i \| (K_j \oplus TS_6))$, $DK_i^{\text{new}} = N_i \oplus TS_4$, $TC_i^{\text{new}} = h(DK_i^{\text{new}} \| r_i)$, $PTC_i^{\text{new}} = PTC_i^{\text{old}} \oplus TC_i^{\text{old}} \oplus TC_i^{\text{new}}$. Finally, SC updates $\{PTC_i^{\text{new}}\}$ and send parameters $m_6 = \{q_6, TS_6\}$ to GWN.

4) After receiving m_6, GWN checks the freshness of TS_6. GWN computes $q_6^* = h(N_i \| (K_j \oplus TS_6))$ and verifies $q_6^* = q_6$? After passing the verification, GWN computes $DK_i^{\text{new}} = N_i \oplus TS_4$, $PDK_i^{\text{new}} = DK_i^{\text{new}} \oplus r_i$, $TC_i^{\text{new}} = h(DK_i^{\text{new}} \| r_i)$, $PTC_i^{\text{new}} = PTC_i^{\text{old}} \oplus TC_i^{\text{old}} \oplus TC_i^{\text{new}}$ and $BN_i^{\text{new}} = r_i \oplus h(PTC_i^{\text{new}} \| ID_{\text{GWN}} \| K_{\text{U}})$. Then, GWN updates $\{PTC_i^{\text{new}}, PDK_i^{\text{new}}, BN_i^{\text{new}}\}$ and $\{PTC_i^{\text{new}}, K_U\}$.

F. Password Change Phase

In this section, we describe the password change phase as follows.

1) In the first place, the user U_i inserts SC and inputs ID_i, PW_i. Then, SC reads the stores information in memory and computes $r_i = AN_i \oplus h(ID_{SC} \| ID_i \| PW_i)$, $RPW_i = h(ID_{SC} \| r_i \| PW_i)$, $B_i^* = h(ID_{SC} \| RPW_i)$.

2) The SC checks $B_i^* = B_i$? If it is equal, U_i enters new password PW_i^{new}. Otherwise, SC rejects this password change request.

3) The SC calculates $AN_i^{\text{new}} = r_i \oplus h(ID_{SC} \| ID_i \| PW_i^{\text{new}})$, $RPW_i^{\text{new}} = h(ID_{SC} \| r_i \| PW_i^{\text{new}})$ and $B_i^{\text{new}} = h(ID_{SC} \| RPW_i^{\text{new}})$. The parameters $\{AN_i, B_i\}$ are replaced with new parameters $\{AN_i^{\text{new}}, B_i^{\text{new}}\}$.

TABLE II
NOTATIONS OF BAN LOGIC

Notation	Definition
P, Q	Related principal part.
X, Y	Related formula and rule.
K, KEY	Secret key.
$P\| \equiv X$	P believes X.
$P \triangleleft X$	P receives the message contain X.
$P\| \sim Q$	P sends the message contain X.
$P \Rightarrow X$	P controls X.
$\#(X)$	X is fresh.
(X, Y)	X and Y as a group.
$< X >_Y$	X is combined with Y.
$\{X\}_Y$	X is encrypted with Y.
$P \xleftrightarrow{K} Q$	P and Q share a secret key K.
$P \overset{X}{\leftrightarrows} Q$	X is only known by P and Q.

V. SECURITY ANALYSIS

In this section, we use formal security analysis [55], [56] and informal security analysis to analyze our authentication scheme.

A. Formal Security Analysis

1) BAN Logic: We use widely used tool BAN logic [57], [58] to analyze the security of our authentication scheme. The notations of BAN logic in this article are defined in Table II.

The reference rule of BAN logic is defined as follows.

1) *The Receiving Rule:* $([P \triangleleft (X, Y)]/P \triangleleft X)$, $[(P \triangleleft < X >_Y)/P \triangleleft X]$.
Rule Interpretation: If P can receive message contains (X, Y), then P can receive message contain X. If P can receive message contains $< X >_Y$, then P can receive message contain X.

2) *The Freshness-Propagation Rule:* $([P| \equiv \#(X)]/[P| \equiv \#(X, Y)])$.
Rule Interpretation: If X is fresh and P believes X, we can get P believes (X, Y) is fresh.

3) *The Nonce-Verification Rule:* $([P| \equiv \#(X), P| \equiv Q| \sim X]/[P| \equiv Q| \equiv X])$.
Rule Interpretation: If P believes X is fresh and P believes Q sent a message contain X, we can get P believes that Q believes X.

4) *The Jurisdiction Rule:* $([P| \equiv Q \quad \Rightarrow X, P| \equiv Q| \equiv X]/[P| \equiv X])$.
Rule Interpretation: If P believes that Q controls X and P believes that Q believes X, we can get P believes X.

5) *The Message-Meaning Rule:* $([P| \equiv P \xleftrightarrow{K} Q, P \triangleleft \{X\}_K]/[P| \equiv Q| \sim X])$,
$([P| \equiv P \overset{X}{\leftrightarrows} Q, P \triangleleft < X >_K]/[P| \equiv Q| \sim X])$.
Rule Interpretation: If P believes that P and Q share a secret key K and P can receive message contain $\{X\}_K$, then P believes that Q sent message contain X. If P believes that X is only known by P and Q, as well as P can receive message contain $< X >_Y$. We can get P believes that Q sent message contain X.

6) *The Session Key Rule:* $([P| \equiv \#(X), P| \equiv Q| \equiv X]/[P| \equiv P \xleftrightarrow{K} Q])$.

Rule Interpretation: If P believes X is fresh and P believes that Q believes X, we can get P believes that P and Q share a secret key K.

Our proposed scheme needs to satisfy the following goals:

$G1$: $U_i| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$

$G2$: $U_i| \equiv S_j| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$

$G3$: $S_j| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$

$G4$: $S_j| \equiv U_i| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$.

To analyze our scheme with BAN logic, we abstract our scheme into a standard form

$M1$: $U_i \rightarrow GWN$:$\{U_i \overset{N_i}{\longleftrightarrow} S_j, TS_1\}_{TC_i}$

$M2$: $GWN \rightarrow S_j$:$\{U_i \overset{N_i}{\longleftrightarrow} S_j, TS_2\}_{TC_j}$

$M3$: $S_j \rightarrow GWN$:$\{U_i \overset{K_j}{\longleftrightarrow} S_j, TS_3\}_{TC_j}$

$M4$: $GWN \rightarrow U_i$:$\{U_i \overset{K_j}{\longleftrightarrow} S_j, TS_4\}_{TC_i}$

$M5$: $GWN \rightarrow S_j$:$\{GWN \overset{K_j}{\longleftrightarrow} S_j, TS_5\}_{TC_j}$

$M6$: $U_i \rightarrow GWN$:$\{U_i \overset{N_i}{\longleftrightarrow} GWN, TS_6\}_{TC_i}$.

We propose the following hypothesis to verify the security of our scheme:

$H1$: $U_i| \equiv \#(TS_1, TS_4, TS_6)$

$H2$: $GWN| \equiv \#(TS_1, TS_2, TS_3, TS_5)$

$H3$: $S_j| \equiv \#(TS_2, TS_3, TS_5)$

$H4$: $U_i| \equiv U_i \overset{TC_i}{\longleftrightarrow} GWN$

$H5$: $GWN| \equiv GWN \overset{TC_j}{\longleftrightarrow} S_j$

$H6$: $S_j| \equiv S_j \overset{TC_j}{\longleftrightarrow} GWN$

$H7$: $GWN| \equiv GWN \overset{TC_i}{\longleftrightarrow} U_i$.

The proof process based on BAN logic is as follows.

According to $M1$, we can get

$S1$: $GWN \triangleleft \{U_i \overset{N_i}{\longleftrightarrow} S_j, TS_1\}_{TC_i}$.

According to $S1$, $H7$, and the message-meaning rule. We can obtain

$S2$: $GWN| \equiv U_i| \sim U_i \overset{N_i}{\longleftrightarrow} S_j$.

According to $S2$, $H2$, the freshness-propagation rule, and nonce-verification rule. We can get

$S3$: $GWN| \equiv U_i| \equiv U_i \overset{N_i}{\longleftrightarrow} S_j$.

According to $M2$, we can obtain

$S4$: $S_j \triangleleft \{U_i \overset{N_i}{\longleftrightarrow} S_j, TS_2\}_{TC_j}$.

According to $S4$, $H6$, and the message-meaning rule. We can get

$S5$: $S_j| \equiv GWN| \sim U_i \overset{N_i}{\longleftrightarrow} S_j$.

According to $S5$, $H3$, the freshness-propagation rule, and the nonce-verification rule. We can obtain

$S6$: $S_j| \equiv GWN| \equiv U_i \overset{N_i}{\longleftrightarrow} S_j$.

In our proposed scheme, GWN acts as a transmission role and does not negotiate the session key. According to $S3$, $S6$, and $KEY_{ij} = h(N_i \| K_j)$, we can get $G4$

$S7$: $S_j| \equiv U_i| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$.

According to $S7$, $H3$, the freshness-propagation rule, and the session key rule, we can obtain $G3$

$S8$: $S_j| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$.

According to $M3$, we can get

$S9$: $GWN \triangleleft \{U_i \overset{K_j}{\longleftrightarrow} S_j, TS_3\}_{TC_j}$.

According to $S9$, $H5$, and the message-meaning rule. We can obtain

$S10$: $GWN| \equiv S_j| \sim U_i \overset{K_j}{\longleftrightarrow} S_j$.

According to $S10$, $H2$, the freshness-propagation rule, and the nonce-verification rule. We can get

$S11$: $GWN| \equiv S_j| \equiv U_i \overset{K_j}{\longleftrightarrow} S_j$.

According to $M4$, we can obtain

$S12$: $U_i \triangleleft \{U_i \overset{K_j}{\longleftrightarrow} S_j, TS_4\}_{TC_i}$.

According to $S12$, $H4$, and the message-meaning rule. We can get

$S13$: $U_i| \equiv GWN| \sim U_i \overset{K_j}{\longleftrightarrow} S_j$.

According to $S13$, $H1$, the freshness-propagation rule, and the nonce-verification rule. We can obtain

$S14$: $U_i| \equiv GWN| \equiv U_i \overset{K_j}{\longleftrightarrow} S_j$.

In our proposed scheme, GWN acts as a transmission role and does not negotiate the session key. According to $S11$, $S14$, and $KEY_{ij} = h(N_i \| K_j)$, we can get $G2$

$S15$: $U_i| \equiv S_j| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$.

According to $S15$, $H1$, the freshness-propagation rule, and the session key rule, we can obtain $G1$

$S16$: $U_i| \equiv U_i \overset{KEY}{\longleftrightarrow} S_j$.

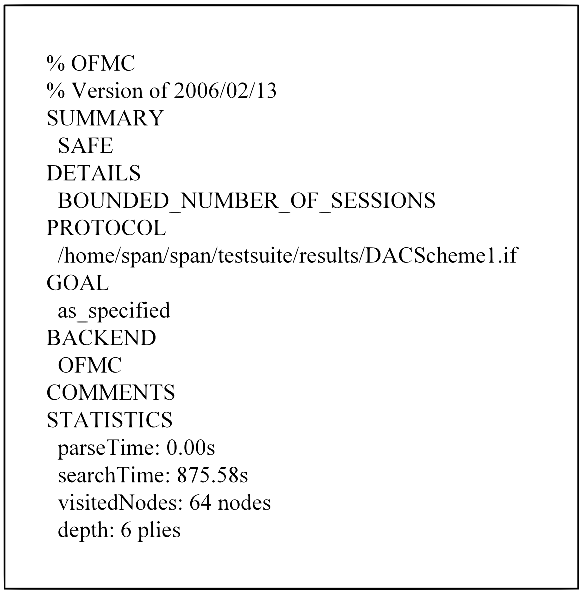
Therefore, we can conclude that our authentication schemes could satisfy Goals 1–4 through the BAN logic, which means our scheme can achieve secure mutual authentication between the user, GWN, and SN.

2) AVISPA Simulation: AVISPA is a widely used security simulation tool [39], [59], it supports OFMC, CL-AtSe, SATMC, and TA4SP four different approaches [60], [61]. In this article, we using the simulation software SPAN which is based on AVISPA to simulate our scheme. The SPAN is installed in the Linux system of Ubuntu 10.10 (32 bits) and the corresponding Linux environment is implemented on the virtual machine Oracle VM VirtualBox (6.0.24). We use the OFMC and CL-AtSe methods of AVISPA to check the security of our proposed authentication scheme. The simulation results are shown in Figs. 6 and 7. From the summary of the simulation, we can conclude that our authentication scheme is secure under the OFMC model and the CL-AtSe model.

B. Informal Security Analysis

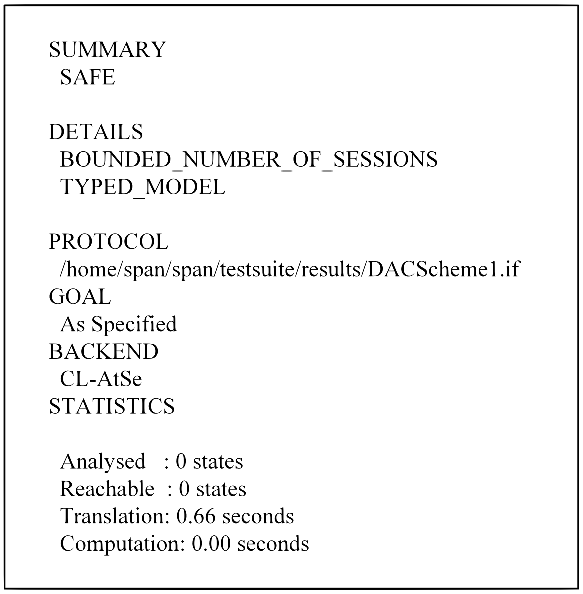
In this section, we use informal security analysis to estimate the security of our authentication scheme. Moreover, we present the security of our scheme compare to those existing schemes [18], [19], [62]–[64]. The analysis detailed processes are as follows.

1) Dynamic Authentication Credential: A traditional authentication scheme only implements identity authentication and session key agreement, where the credential parameters are statically stored in memory. The user and SN do not know if a credential parameter has been leaked, and the security risk is huge in this situation [65]. Moreover, most authentication schemes with DAC do not consider the risk of data packet loss for credential rotation update. Therefore, we design a dynamic credentials update phase in our scheme. After the session key KEY_{ij} is negotiated, the authentication information $\{PTC_i^{\text{old}}\}$,



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/DACScheme1.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 875.58s
  visitedNodes: 64 nodes
  depth: 6 plies
```

Fig. 6. Simulation results for OFMC.



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/DACScheme1.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.66 seconds
  Computation: 0.00 seconds
```

Fig. 7. Simulation results for CL-AtSe.

$\{PTC_i^{\text{old}}, PDK_i^{\text{old}}, BN_i^{\text{old}}, PDK_j^{\text{old}}\}$ and $\{PTC_j^{\text{old}}\}$ stored in the SC, GWN, and SN is dynamically updated. At the same time, the update confirmation messages m_5 and m_6 are added in our scheme. The voucher will be updated after the update confirmation is completed. This design can effectively avoid the impact of packet loss on the availability of the authentication scheme.

2) Authentication Table Leakage Attack and Privileged User Attack: Relevant research shows that most of the attacks in WSNs come from inside. Therefore, the privileged user attack and authentication table leakage attack are the main threat to WSNs. In a practical application of WSNs, the authentication information table must be stored in the GWN to realize the authentication and communication between multiple users and multiple SNs. If the administrator of the GWN becomes an adversary or the authentication table is obtained by the adversary, the adversary can guess the credential information and other sensitive information according to the authentication table. In our scheme, we take advantage of SGX to protect the authentication information. The adversary cannot get the master key and cannot guess other sensitive information, even if the GWN authentication table is leaked. Therefore, our authentication scheme can effectively resist privileged user attacks.

3) Online Guessing Attack: We assume that the adversary can obtain $m_1 = \{q_1, PKS_i, PID_j, PTC_i^{\text{old}}, TS_1\}$, $m_2 = \{q_2, PKS_N, PID_j^{\text{old}}, TS_2\}$, $m_3 = \{q_3, PKS_j, TS_3\}$, and $m_4 = \{q_4, PKS_k, TS_4\}$ of the authentication and key exchange phase by attacking the public communication channel. The adversary launched online guessing attacks by intercepting the information [66] as follows.

1) In order to compute the session key $KEY_{ij} = h(N_i\|K_j)$, the adversary can obtain $PKS_i = N_i \oplus h(TC_i^{\text{old}}\|r_i\|TS_1)$ from m_1, obtain $PKS_N = N_i \oplus h(TC_j^{\text{old}}\|ID_j\|TS_2)$ from m_2, obtain $PKS_j = K_j \oplus h(TC_j^{\text{old}}\|N_i\|ID_j\|TS_3)$ from m_3, and obtain $PKS_k = K_j \oplus h(TC_i^{\text{old}}\|(N_i \oplus ID_j))$ from m_4.
2) If the adversary wants to guess N_i and K_j, the adversary must obtain TC_i^{old}, TC_j^{old}, and r_i. However, there is no transmission of TC_i, TC_j in our scheme and r_i is encrypted with other parameters.

The adversary cannot guess two unknown parameters at the same time. Therefore, we can conclude that the adversary cannot guess the session key online and our scheme can resist online guessing attacks.

4) Tracking Attack: Many papers consider user tracking attacks, but in practical applications of WSNs, the dynamic rotation of SN core parameters is also necessary. Therefore, it is important to prevent tracking for SNs. We assume that an adversary can intercept the communication information $m_1 = \{q_1, PKS_i, PID_j, PTC_i^{\text{old}}, TS_1\}$, $m_2 = \{q_2, PKS_N, PID_j^{\text{old}}, TS_2\}$, $m_3 = \{q_3, PKS_j, TS_3\}$, and $m_4 = \{q_4, PKS_k, TS_4\}$. However, we add random numbers to the calculation so that these parameters $\{PKS_i, PID_j, PTC_i{}^{\text{old}}, PKS_N, PID_j{}^{\text{old}}, PKS_j, PKS_k\}$ are dynamic in each authentication process. Therefore, all parameters transmitted over the common communication channel are dynamic. The adversary cannot launch tracking attacks based on these parameters. According to the preceding analysis, our scheme can effectively resist tracking attacks.

5) D-DOS Attack: In the authentication and key exchange phase, we presume that the adversary can intercept the request message $m_2 = \{q_2, PKS_N, PID_j^{\text{old}}, TS_2\}$ and launch D-DOS attacks to the SN. Upon receiving the message from D-DOS attacks, the SN checks the freshness of these messages. If the timestamp is not fresh, the SN rejects this request without other computes. Our scheme verifies the request message, which saves the computing resources of the SN. Therefore, our scheme can effectively resist D-DOS attacks.

6) Replay Attack: We add a timestamp to every verification request. In addition, after the session key KEY_{ij} is successfully negotiated, the parameters of U_i, GWN, and S_j are updated dynamically. We assume that the adversary intercepted the messages $m_2 = \{q_2, PKS_N, PID_j^{\text{old}}, TS_2\}$ and

$m_3 = \{q_3, PKS_j, TS_3\}$ and launched replay attacks to the GWN, S_j. The SN S_j computes q_2^* and the GWN computes q_3^*. However, the parameters of q_2^* and q_3^* were already updated after the session key was negotiated. Therefore, the SN S_j verifies $q_2^* \neq q_2$, and the GWN verifies $q_3^* \neq q_3$. We conclude that our scheme can withstand replay attacks.

7) Stolen Smart Card Attack and Offline Guessing Attack: Should an adversary obtain the information $\{AN_i, B_i, PTC_i^{\text{old}}, h(\cdot)\}$ stored inside the smart card, that adversary would find that these messages are all ciphertext data encrypted by a one-way hash function. The smart card does not directly store the ID_i and PW_i. Therefore, the adversary cannot get relevant information and our scheme can resist stolen smart card attacks. Moreover, the adversary cannot offline guessing ID_i or PW_i from the information $\{AN_i, B_i, PTC_i^{\text{old}}, h(\cdot)\}$ that is stored in a smart card. We assume that the adversary can obtain the $AN_i = r_i \oplus h(ID_{SC} \| ID_i \| PW_i)$, $RPW_i = h(ID_{SC} \| r_i \| PW_i)$, $PTC_i^{\text{old}} = TC_i^{\text{old}} \oplus h(r_i \| ID_{SC})$. However, the key information $\{ID_i, PW_i, r_i\}$ was protected by the one-way hash function. Therefore, our scheme can resist offline password guessing attacks.

8) Gateway Bypass Attack: In the authentication and key exchange phase, U_i cannot directly authenticate with the SN and GWN initiates the authentication request to the target SN. Moreover, we assume that the adversary can intercept $m_1 = \{q_1, PKS_i, PID_j, PTC_i^{\text{old}}, TS_1\}$, modify m_1 and directly send a new requests to SN. However, the adversary cannot obtain TC_j and the SN cannot successfully verify q_1. Therefore, the adversary cannot bypass the GWN to establish the path with the SN and our scheme can effectively resist the gateway bypass attack.

9) Man-in-the-Middle Attack: According to the threat model, the adversary can launch the man-in-the-middle attack by intercepting information $m_1 = \{q_1, PKS_i, PID_j, PTC_i^{\text{old}}, TS_1\}$, $m_2 = \{q_2, PKS_N, PID_j^{\text{old}}, TS_2\}$, $m_3 = \{q_3, PKS_j, TS_3\}$, and $m_4 = \{q_4, PKS_k, TS_4\}$. However, the information in m_1, m_2, m_3, m_4 is encrypted by the hash function except timestamp. Moreover, we add the timestamp to the calculation and the m_1, m_2, m_3, m_4 of each transmission is dynamic. Once the adversary tampers or delays the intercepted information, it will lead to authentication failure. Therefore, our scheme can effectively resist the man-in-the-middle attack.

10) Forward Secrecy: In our scheme, the user U_i and SN S_j will generate a one-time session key *KEY* after authentication. At the same time, the scheme dynamically updates the temporary credential TC_i, TC_j. Therefore, the session key changes dynamically and randomly in each authentication process. It is impossible for adversary to obtain the session key of the previously transmitted data during this session key agreement. Therefore, our scheme can satisfy forward secrecy.

11) Sensor Node Impersonation Attack and Sybils Attack: In a practical application environment, the adversary can capture deployed SNs to obtain PTC_j in memory. The adversary can impersonate one or more legitimate SNs and receive authentication information from GWN. However, due to the nature of hash function, the adversary cannot obtain the TC_j from $PTC_j^{\text{old}} = TC_j^{\text{old}} \oplus PID_j^{\text{old}}$ and $PID_j^{\text{old}} = h(DK_j^{\text{old}} \| ID_j)$. Therefore, the authentication message from GWN cannot be parsed. In addition, the PTC_j within the legitimate SNs will be updated after each authentication. Therefore, our scheme can effectively resist the SN impersonation attack and sybils attack.

TABLE III
SECURITY AND FUNCTIONALITY COMPARISON WITH OTHER SCHEMES

Approach	[18]	[19]	[62]	[63]	[64]	Ours
A1	N	N	N	N	N	Y
A2	N	N	N	Y	N	Y
A3	Y	N	Y	Y	Y	Y
A4	Y	N	Y	N	N	Y
A5	Y	N	N	Y	Y	Y
A6	Y	N	N	Y	Y	Y
A7	Y	Y	Y	Y	Y	Y
A8	Y	Y	Y	Y	Y	Y
A9	Y	N	N	Y	Y	Y
A10	Y	Y	Y	Y	Y	Y
A11	Y	N	Y	Y	N	Y

A1: Dynamic authentication credential; A2: Authentication table leakage attack and privileged user attack; A3: Online guessing attack; A4: Tracking attack; A5: D-DOS attack; A6: Replay attack; A7: Stolen smart card attack and offline guessing attack; A8: Gateway bypass attack; A9: Man-in-the-middle attack; A10: Forward secrecy; A11: Sensor node impersonation attack and sybils attack;

12) Security Comparison: In this section, we compared the security of our scheme with other schemes [18], [19], [62]–[64]. In Table III, *Y* represents that the approach supports this security properties, and *N* represents that the approach does not support this security properties. *A*1–*A*11 correspond to the abbreviations of 11 kinds of security properties, respectively. According to the comparison results in Table III, we can find that the schemes of [19] and [63] are vulnerable to tracking attacks, and [19] and [62] are vulnerable to replay attacks. The schemes in [18], [19], and [62]–[64] all cannot realize DAC. Therefore, we can conclude that our scheme provides more security than existing schemes through the above analysis.

VI. Performance Analysis

We estimate the performance of our scheme in computation overheads, communication overheads, storage overheads, and SNs energy consumption [48]. We have also compared with the existing scheme [18], [19], [62]–[64] under the same standard. The detail of comparison is described as follows.

A. Analysis Basis

In this section, we define a unified calculation standard for performance analysis as follows. First, we have adopted the performance evaluation model used by Shim [67] as the performance analysis model of the authentication scheme of WSNs. This model also uses the MICAz hardware platform as the analysis basis, and the parameters of MICAz are shown in Table IV.

According to the work of Shim [67] and the model parameters in Table IV, the energy consumption required to perform

TABLE IV
PARAMETERS VALUE OF MICAz

Parameters	Value
Power level	$3.0V$
Current drawn in active mode	$8.0mA$
Receiving current drawn mode	$19.7mA$
Transmitting current drawn mode	$17.4mA$
Date rate	$250kbps$
Operating system	$TinyOS$
Memory	$128KB$

TABLE V
ENERGY CONSUMPTION FOR OPERATION

Operation	Energy in MICAz
Computation/1ms	$3.0V \times 8.0mA \times 1ms = 0.024mJ$
Transmit/1bit	$3.0V \times 17.4mA \times \frac{1bit}{250000b/s} \approx 0.00021mJ$
Receive/1bit	$3.0V \times 19.7mA \times \frac{1bit}{250000b/s} \approx 0.00024mJ$

TABLE VI
REFERENCE OF COMPUTATION TIME

Notation	Definition	User(ms)	GWN&SN(ms)
T_m	The point multiplication	20.23	2450
T_R	The rep operation	20.23	2450
T_S	Symmetric encryption	0.12	3.5
T_H	The hash function	0.03	8

calculations, transmit data, and receive data is shown in Table V.

A lightweight authentication scheme usually uses point multiplication, rep operation, symmetric encryption/decryption, a hash function, and XOR to complete the corresponding operations. The computation cost of the XOR operation usually be ignored, and the rep operation cost is equal to point multiplication [68]. Referring to the works of [69], we assume that the function running time of a user is equivalent to the function running time in an iPhone 6S with 2-GB RAM. We presume that the function running time of the GWN and the SN is equivalent to the function running time in the MICAz hardware platform. The running times of several computing operations are shown in Table VI.

Finally, we assume that the one-way hash function, a random number, a secret number, and an identity are 160 bits, a timestamp is 32 bits, the output of elliptic curve operation is 320 bits, and the size of ciphertext after symmetric encryption is equal to the total size of plaintext before encryption with reference to the work of [63].

B. Computation Overheads

According to the analysis basis, we compare the computation overheads of [18], [19], and [62]–[64] in the login phase, authentication and key exchange phase, and dynamic credentials update phase. The comparison results are detailed in Table VII. It can be concluded from Table VII that we have smaller computation. In addition, our scheme has better security than [18], [19], and [62]–[64] according to Table III.

C. Communication Overheads

According to the analysis basis, we compare the communication overheads with other existing schemes [18], [19], [62]–[64]. According to Table VIII, we can conclude that the communication overheads of our scheme are much smaller than [18], [19], and [62]–[64].

D. Storage Overhead

Referring to the parameter length defined in the above communication overheads, we compare our storage overheads with other existing schemes [18], [19], [62]–[64]. In Table IX, we can find that the storage overheads of our scheme are much better than [18], [19], and [62]–[64].

E. Energy Consumption of Sensor Node

In WSNs, SNs are usually deployed in an unattended environment, and the energy, communication, and storage resources of SNs cannot be replenished. In contrast, smart cards and gateways are easier to expand and supplement than SNs. Thus, we usually think that smart cards and GWN have no energy consumption limitations.

Based on the analysis basis, we analyze our scheme and existing schemes from the energy consumption of SNs. This analysis mainly includes node-computing energy consumption and node-communicating energy consumption. The comparison results are shown in Table X.

From the above comparison, we can find that the SN energy consumption of our scheme is at an intermediate level. This is because we sacrifice part of the energy consumption for the DAC phase, and our scheme provides higher security, according to Table III.

F. NS3 Simulation

We use the widely accepted simulator tool NS-3 [70]–[72] to measure the network performance of our authentication scheme in different scenarios, including end-to-end delay and network throughput. As the computation overhead is a necessary part of the authentication scheme, it must be considered in the simulation. Unfortunately, many researchers ignore adding the computation time into the simulation. In this article, we add computation overheads as a node calculation delay in the sending information interval based on [73] and [74], which makes the simulation experiment closer to the communication of WSNs. We run six simulations using one GWN and a different number of users and SNs. The SNs are distributed radially along a ring (inner radius 20 m, outer radius 80 m), which is centered on the GWN. Users are allowed to move randomly in a square area with a side length of 150 m centered on the GWN. All nodes communicate through the 2.4-GHz IEEE 802.11a Wi-Fi standard. Other parameter settings are shown in Table XI.

1) End-to-End Delay: We show the end-to-end delay result in Fig. 8, which plots the end-to-end delay comparison curve of our authentication scheme and [18], [19], [62]–[64] schemes in six simulation scenarios. We can easily observe that when the information exchanged for authentication continues to increase, the end-to-end delay will also increase.

TABLE VII
COMPARISON OF COMPUTATION OVERHEADS

Approach	User	GWN	SN	Overheads(ms)
[18]	$11T_H + T_R$	$17T_H$	$5T_H$	196.56
[19]	$11T_H$	$11T_H$	$6T_H$	130.33
[62]	$11T_H$	$15T_H$	$6T_H$	160.33
[63]	$13T_H + T_R + 2T_m$	$10T_H$	$4T_H + 2T_m$	5073.08
[64]	$11T_H + T_R$	$12T_H$	$6T_H$	164.56
Ours	$12T_H$	$18T_H$	$8T_H$	208.36

TABLE VIII
COMPARISON OF COMMUNICATION OVERHEADS

Approach	User(bits)		GWN(bits)		SN(bits)		Total
	Trans	Receive	Trans	Receive	Trans	Receive	(bits)
[18]	1024	672	1344	1376	352	672	2720
[19]	960	800	1120	1760	1600	1120	3680
[62]	960	800	1440	1280	320	640	2720
[63]	1152	864	1728	1824	672	864	3552
[64]	672	512	1024	1184	512	512	2208
Ours	864	352	1056	1216	352	704	2272

TABLE IX
COMPARISON OF STORAGE OVERHEADS

Approach	SC(bits)	GWN(bits)	SN(bits)	Total(bits)
[18]	640	480	480	1600
[19]	960	480	160	1600
[62]	800	480	480	1760
[63]	1280	160	320	1760
[64]	480	160	320	960
Ours	480	960	160	1600

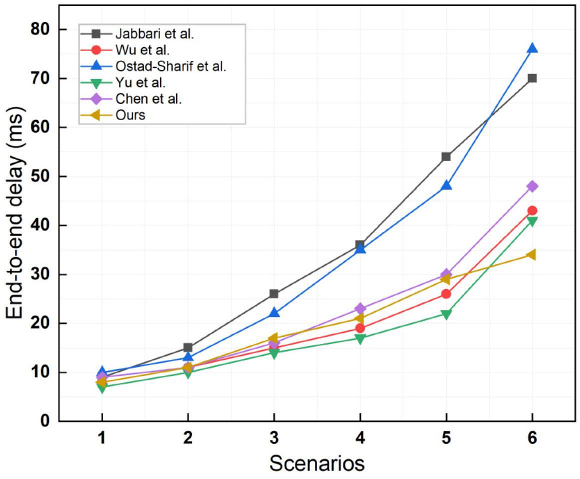


Fig. 8. End-to-end delay.

2) Network Throughput: In this section, the throughput comparison curve between our scheme and [18], [19], [62]–[64] schemes are drawn. From Fig. 9, we can see that the amount of information exchanged for authentication continues to increase, the throughput of each scheme decreases.

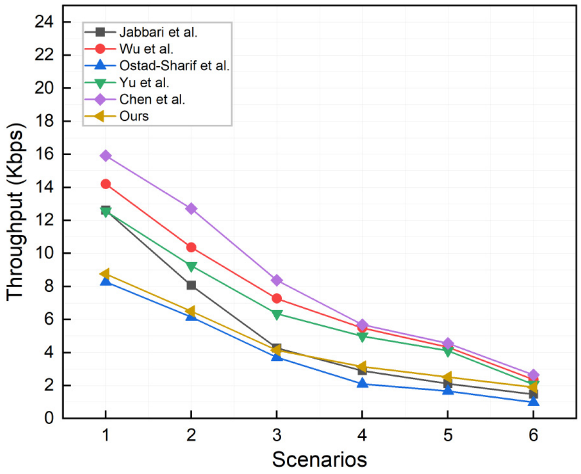


Fig. 9. Throughput.

VII. CONCLUSION

In this article, we design a secure authentication scheme for WSNs based on DAC and Intel SGX. Our scheme realizes the DAC rotation after the session key is negotiated and adds the confirmation action of rotation, which solves the asynchronous update problem caused by packet loss and guarantees the next successful login. Meanwhile, we use SGX to store the master key to protect the authentication table and resist both the privileged user attack and the authentication table leakage attack. Furthermore, we compare our scheme with some of the advanced lightweight authentication schemes in terms of security analysis, computation overheads, communication overheads, storage overheads, and energy consumption of the SN. The comparison results show that our scheme has achieved better progress than others. Although the performance of our scheme is not the best in terms of computing overheads and energy consumption of the SN, other schemes cannot implement DAC rotation and cannot resist an authentication table

TABLE X
ENERGY CONSUMPTION OF SN

Approach	Computation		Trans		Receive		Total
	(ms)	(mJ)	(bits)	(mJ)	(bits)	(mJ)	(mJ)
[18]	40	0.96	352	0.074	512	0.123	1.157
[19]	48	1.152	1600	0.336	1120	0.269	1.757
[62]	40	0.96	320	0.067	480	0.115	1.142
[63]	4932	118.368	672	0.141	864	0.207	118.716
[64]	48	1,152	512	0.108	512	0.123	1.383
Ours	64	1.536	352	0.074	864	0.207	1.817

TABLE XI
PARAMETERS OF OUR NS3 SIMULATION

Parameters	Description	
Platform	NS3(3.27) / Ubuntu 16.04.7 LTS	
Scenarios	No. of users	No. of sensor nodes
1	3	10
2	4	10
3	3	20
4	4	20
5	3	30
6	4	30
Mobility	random (0-3 m/s)	
Simulation time	1600 sec	

leakage attack. In addition, we use the widely accepted simulation tool NS3 to compare our scheme with other advanced schemes, which verifies the practicability of our scheme. The analysis results show that our scheme has better security and is more efficient and suitable for WSNs. In the future, we will continue to use SGX to achieve more secure and efficient computing. We will also apply blockchain and zero trust architecture to the authentication scheme of WSNs and we plan to study and design the post-quantum cryptographic schemes.

ACKNOWLEDGMENT

The authors are grateful to the reviewers and the Associate Editor for their invaluable feedback.

REFERENCES

[1] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7702–7712, Oct. 2019, doi: 10.1109/JIOT.2019.2901840.

[2] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Netw.*, vol. 25, no. 8, pp. 4737–4750, 2019.

[3] H. Fu, G. Manogaran, K. Wu, M. Cao, S. Jiang, and A. Yang, "Intelligent decision-making of online shopping behavior based on Internet of Things," *Int. J. Inf. Manag.*, vol. 50, pp. 515–525, Feb. 2020.

[4] D. Wang, P. Wang, and C. Wang, "Efficient multi-factor user authentication protocol with forward secrecy for real-time data access in WSNs," *ACM Trans. Cyber Phys. Syst.*, vol. 4, no. 3, pp. 1–26, 2020.

[5] S. A. Chaudhry, H. Naqvi, M. S. Farash, T. Shon, and M. Sher, "An improved and robust biometrics-based three factor authentication scheme for multiserver environments," *J. Supercomput.*, vol. 74, pp. 3504–3520, Dec. 2015.

[6] A. Jabbari and J. Bagherzadeh, "A revised key agreement protocol based on chaotic maps," *Nonlinear Dyn.*, vol. 78, no. 1, pp. 669–680, 2014.

[7] M. Ma, D. He, S. Fan, and D. Feng, "Certificateless searchable public key encryption scheme secure against keyword guessing attacks for smart healthcare," *J. Inf. Security Appl.*, vol. 50, Feb. 2020, Art. no. 102429.

[8] W. Liu, X. Wang, and W. Peng, "Secure remote multi-factor authentication scheme based on chaotic map zero-knowledge proof for crowdsourcing Internet of Things," *IEEE Access*, vol. 8, pp. 8754–8767, 2020, doi: 10.1109/ACCESS.2019.2962912.

[9] X. Liu, R. Zhang, and M. Zhao, "A robust authentication scheme with dynamic password for wireless body area networks," *Comput. Netw.*, vol. 161, pp. 220–234, Oct. 2019.

[10] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.

[11] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'" *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[12] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Improved two-factor user authentication in wireless sensor networks," in *Proc. IEEE 6th Int. Conf. Wireless Mobile Comput. Netw. Commun.*, Niagara Falls, ON, Canada, 2010, pp. 600–606, doi: 10.1109/WIMOB.2010.5645004.

[13] J. Kim, D. Lee, W. Jeon, Y. Lee, and D. Won, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6443–6462, 2014.

[14] J. Li, Y. Ding, Z. Xiong, and S. Liu, "An improved two-factor mutual authentication scheme with key agreement in wireless sensor networks," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 11, pp. 5556–5573, 2017.

[15] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, "Secure authentication protocol for wireless sensor networks in vehicular communications," *Sensors*, vol. 18, no. 10, p. 3191, Sep. 2018, doi: 10.3390/s18103191.

[16] M. J. Sadri and M. R. Asaar, "A lightweight anonymous two-factor authentication protocol for wireless sensor networks in Internet of Vehicles," *Int. J. Commun. Syst.*, vol. 33, no. 14, p. e4511, 2020.

[17] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[18] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, and D. Abbasinezhad-Mood, "Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," *Future Gener. Comput. Syst.*, vol. 100, pp. 882–892, Nov. 2019.

[19] C.-T. Chen, C.-C. Lee, and I.-C. Lin, "Efficient and secure three-party mutual authentication key agreement protocol for WSNs in IoT environments," *PLoS ONE*, vol. 15, no. 4, 2020, Art. no. e0232277. [Online]. Available: https://doi.org/10.1371/journal.pone.0232277

[20] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2016.

[21] R. Amin, S. K. H. Islam, N. Kumar, and K.-K. R. Choo, "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 104, pp. 133–144, Feb. 2018.

[22] C.-C. Chang, W.-Y. Hsueh, and T.-F. Cheng, "A dynamic user authentication and key agreement scheme for heterogeneous wireless sensor networks," *Wireless Pers. Commun.*, vol. 89, no. 2, pp. 447–465, 2016.

[23] Z. Yang, J. Lai, Y. Sun, and J. Zhou, "A novel authenticated key agreement protocol with dynamic credential for WSNs," *ACM Trans. Sens. Netw.*, vol. 15, no. 2, pp. 1–27, 2019.

[24] S. Agrawal, M. L. Das, and J. Lopez, "Detection of node capture attack in wireless sensor networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 238–247, Mar. 2019.

[25] D. Fu and X. Peng, "TPM-based remote attestation for wireless sensor networks," *Tsinghua Sci. Technol.*, vol. 21, no. 3, pp. 312–321, Jun. 2016.

[26] H. Tan, W. Hu, and S. Jha, "A remote attestation protocol with trusted platform modules (TPMs) in wireless sensor networks," *Security Commun. Netw.*, vol. 8, no. 13, pp. 2171–2188, 2015.

[27] R. A. Balisane and A. Martin, "Trusted execution environment-based authentication gauge (TEEBAG)," in *Proc. New Security Paradigms Workshop (NSPW)*, New York, NY, USA, 2016, pp. 61–67. [Online]. Available: https://doi.org/10.1145/3011883.3011892

[28] R. C. R. Condé, C. A. Maziero, and N. C. Will, "Using Intel SGX to protect authentication credentials in an untrusted operating system," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Natal, Brazil, 2018, pp. 00158–00163.

[29] H. Sun and S. Xiao, "DNA-X: Dynamic network authentication using SGX," presented at the Proc. 2nd Int. Conf. Cryptogr. Security Privacy, Guiyang, China, 2018, pp. 110–115. [Online]. Available: https://doi.org/10.1145/3199478.3199508

[30] K. Kostiainen, N. Asokan, and J.-E. Ekberg, "Credential disabling from trusted execution environments," in *Information Security Technology for Applications (NordSec)* (Lecture Notes in Computer Science, 7127), T. Aura, K. Järvinen, and K. Nyberg, Eds. Heidelberg, Germany: Springer, 2010. [Online]. Available: https://doi.org/10.1007/978-3-642-27937-9_12

[31] K. Kostiainen, N. Asokan, and A. Afanasyeva, "Towards user-friendly credential transfer on open credential platforms," in *Proc. 9th Int. Conf. Appl. Cryptogr. Netw. Security*, Nerja, Spain, Jun. 2011, pp. 395–412.

[32] K. Kostiainen and N. Asokan, "Credential life cycle management in open credential platforms," in *Proc. ACM Workshop Scalable Trust. Comput.*, 2011, pp. 65–70.

[33] C. Marfario, N. Karapanos, C. Soriente, K. Kostiainen, and S. Capkun, "Secure enrollment and practical migration for mobile trusted execution environments," in *Proc. ACM Workshop Security Privacy Smartphones Mobile Devices*, 2013, pp. 93–98.

[34] G. Arfaoui, S. Gharout, J.-F, Lalande, and J. Traoré, "Practical and privacy-preserving TEE migration," in *Information Security Theory and Practice (IFIP WISTP)* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2015.

[35] C. Shepherd, R. N. Akram, and K. Markantonakis, "Establishing mutually trusted channels for remote sensing devices with trusted execution environments," in *Proc. 12th Int. Conf. Availability Rel. Security (ARES)*, 2017, pp. 1–10.

[36] C. Shepherd, R. N. Akram, and K. Markantonakis, "Remote credential management with mutual attestation for trusted execution environments," in *Information Security Theory and Practice (IFIP WISTP)* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2018.

[37] Y. Omori and T. Yamashita, "Extended inter-device digital rights sharing and transfer based on device-owner equality verification using homomorphic encryption," *IEICE Trans. Inf. Syst.*, vol. E103.D, no. 6, pp. 1339–1354, 2020.

[38] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 316–323, 2013.

[39] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.

[40] Q. Tian *et al.*, "New security mechanisms of high-reliability IoT communication based on radio frequency fingerprint," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7980–7987, Oct. 2019, doi: 10.1109/JIOT.2019.2913627.

[41] W. Liu, X. Wang, and W. Peng, "NCZKP based privacy-preserving authentication scheme for the untrusted gateway node smart home environment," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Rennes, France, 2020, pp. 1–6.

[42] J. Furtak and J. Chudzikiewicz, "Securing transmissions between nodes of WSN using TPM," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Lodz, Poland, 2015, pp. 1059–1068, doi: 10.15439/2015F144.

[43] H. G. Zhang, W. B. Han, X. J. Lai, D. D. Lin, J. F. Ma, and J. H. Li, "Survey on cyberspace security," *Sci. China Inf. Sci.*, vol. 58, no. 11, pp. 1–43, 2015.

[44] L. Wu, J. Wang, K.-K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 319–330, 2019, doi: 10.1109/TIFS.2018.2850299.

[45] P. Jain *et al.*, "OpenSGX: An open platform for SGX research," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, vol. 16, 2016, pp. 21–24, doi: 10.14722/ndss.2016.23011.

[46] Y. Ding *et al.*, "POSTER: Rust SGX SDK: Towards memory safety in Intel SGX enclave," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017 pp. 2491–2493.

[47] Juan *et al.*, "Survey on key technology development and application in trusted computing," *China Commun.*, vol. 13, no. 11, pp. 70–90, Nov. 2016.

[48] V. Costan, I. Lebedev, and S. Devadas, "Secure processors part II: Intel SGX security analysis and mit sanctum architecture," *Found. Trends Electron. Design Autom.*, vol. 11, no. 3, pp. 249–361, 2017.

[49] G. Neiger, A. Santoni, F. Leung, D. Rodgers, and R. Uhlig, "Intel virtualization technology: Hardware support for efficient processor virtualization," *Intel Technol. J.*, vol. 10, no. 3, pp. 167–177, 2006.

[50] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. D. Cuvillo, "Using innovative instructions to create trustworthy software solutions," in *Proc. 2nd Int. Workshop Hardw. Archit. Support Security Privacy*, 2013, p. 11.

[51] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1133–1146, 1 Nov./Dec. 2020, doi: 10.1109/TDSC.2018.2857811.

[52] 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, doi: 10.1109/TC.2002.1004593.

[53] J. V. Bulck *et al.*, "Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution," in *Proc. 27th USENIX Conf. Security Symp.*, 2018, pp. 991–1008.

[54] K. Murdock, D. Oswald, F. D. Garcia, J. V. Bulck, D. Gruss, and F. Piessens, "Plundervolt: Software-based fault injection attacks against Intel SGX," in *Proc. IEEE Symp. Security Privacy (SP)*, 2020, pp. 1466–1482.

[55] M. Alotaibi, "An enhanced symmetric cryptosystem and biometric-based anonymous user authentication and session key establishment scheme for WSN," *IEEE Access*, vol. 6, pp. 70072–70087, 2018, doi: 10.1109/ACCESS.2018.2880225.

[56] J. Lee, S. Yu, M. Kim, Y. Park, and A. K. Das, "On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks," *IEEE Access*, vol. 8, pp. 107046–107062, 2020, doi: 10.1109/ACCESS.2020.3000790.

[57] F. G. Darbandeh and M. Safkhani, "A new lightweight user authentication and key agreement scheme for WSN," *Wireless Pers. Commun.*, vol. 114, no. 1, pp. 3247–3269, 2020.

[58] P. Soni, A. K. Pal, and S. H. Islam, "An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system," *Comput. Methods Programs Biomed.*, vol. 182, Dec. 2019, Art. no. 105054.

[59] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, "Secure authentication protocol for wireless sensor networks in vehicular communications," *Sensors*, vol. 18, no. 10, p. 3191, Sep. 2018, doi: 10.3390/s18103191.

[60] A. Armando *et al.*, "The AVISPA tool for the automated validation of Internet security protocols and applications," in *Proc. 17th Int. Conf. Comput.-Aided Verif.*, 2005, pp. 281–285.

[61] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[62] F. Wu *et al.*, "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 727–737, May 2017.

[63] A. Jabbari and J. B. Mohasefi, "User-sensor mutual authenticated key establishment scheme for critical applications in wireless sensor networks," *Wireless Netw.*, no. 27, pp. 227–248, Aug. 2020.

[64] S. J. Yu and Y. Park, "SLUA-WSN: Secure and lightweight three-factor-based user authentication protocol for wireless sensor networks," *Sensors*, vol. 20, no. 15, p. 4143, 2020.

[65] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *J. Ambient Intell. Humanized Comput.*, vol. 10, pp. 3133–3142, Aug. 2019. [Online]. Available: https://doi.org/10.1007/s12652-018-1029-3

[66] X. Liu, R. Zhang, and Q. Liu, "A temporal credential-based mutual authentication with multiple-password scheme for wireless sensor networks," *PLoS ONE*, vol. 12, no. 1, Jan. 2017, Art. no. e0170657, doi: 10.1371/journal.pone.0170657.

[67] K.-A. Shim, "BASIS: A practical multi-user broadcast authentication scheme in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 1545–1554, 2017, doi: 10.1109/TIFS.2017.2668062.

[68] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 9, no. 1, pp. 223–244, 2016.

[69] C.-M. Chen, B. Xiang, T.-Y. Wu, and K.-H. Wang, "An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks," *Appl. Sci.*, vol. 8, no. 7, p. 1074, 2018.

[70] J. K. Jain, "A coherent approach for dynamic cluster-based routing and coverage hole detection and recovery in bi-layered WSN-IoT," *Wireless Pers. Commun.*, vol. 114, no. 1, pp. 519–543, 2020.

[71] S. K. Swain and P. K. Nanda, "Priority based adaptive rate control in wireless sensor networks: A difference of differential approach," *IEEE Access*, vol. 7, pp. 112435–112447, 2019, doi: 10.1109/ACCESS.2019.2935025.

[72] N. Almansour and S. Alahmadi, "Secure ad hoc on-demand distance vector routing protocol in WSN," in *Proc. 1st Int. Conf. Comput. Appl. Inf. Security (ICCAIS)*, Riyadh, Saudi Arabia, 2018, pp. 1–4.

[73] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. P. C. Rodrigues, and Y. Park, "Physically secure lightweight anonymous user authentication protocol for Internet of Things using physically unclonable functions," *IEEE Access*, vol. 7, pp. 85627–85644, 2019, doi: 10.1109/ACCESS.2019.2926578.

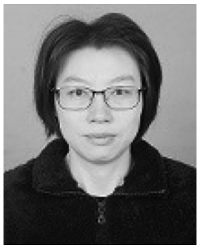
[74] S. Banerjee *et al.*, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8739–8752, Oct. 2019, doi: 10.1109/JIOT.2019.2923373.

Xin Liu received the B.S. degree in communication engineering and the M.Sc. and Ph.D. degrees in computer application technology from Lanzhou University, Lanzhou, China, in 2011, 2014, and 2019, respectively.

He is currently a University Lecturer with the School of Information Science and Engineering, Lanzhou University. His research interests include authentication scheme, blockchain, wireless sensor network security, zero trust architecture, and confidential computing.

Zhenbin Guo received the B.S. degree from Harbin Institute of Technology, Harbin, China, in 2016. He is currently pursuing the M.Sc. degree with Lanzhou University, Lanzhou, China.

His research interests include wireless sensor network security and confidential computing.



Jun Ma received the B.S. degree in computer science and technology from Northwest Normal University, Lanzhou, China, in 2004, and the M.Sc. and Ph.D. degrees in computer application technology from Lanzhou University, Lanzhou, in 2008 and 2019, respectively.

She is currently a University Engineer with the School of Information Science and Engineering, Lanzhou University. Her research interests include pattern recognition, natural language processing, recommendation systems, deep learning, and cheminformatics.

Yuchen Song received the M.Sc. degree in computer application technology from Lanzhou University, Lanzhou, China, in 2018.

He is currently an Engineer with the Information Technology Laboratory, Communication and Information Center, State Administration of Work Safety, Beijing, China. His research interests include machine learning, reverse engineering, pattern recognition, authentication, and wireless sensor networks.