Received: 4 January 2023 | Revised: 9 May 2023 | Accepted: 28 June 2023

DOI: 10.1002/spy2.335

RESEARCH ARTICLE

WILEY

A robust and secure user authentication scheme based on multifactor and multi-gateway in IoT enabled sensor networks

Ravi Kumar[1] | Samayveer Singh[1] | Deepti Singh[2] | Mohit Kumar[3] | Sukhpal Singh Gill[4]

[1]Department of Computer Science & Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India

[2]Department of Computer Science & Engineering, NSUT, New Delhi, India

[3]Department of Information Technology, Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India

[4]School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Correspondence
Samayveer Singh, Department of Computer Science & Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India.
Email: samays@nitj.ac.in

Abstract

The Internet of Things (IoT) enabled wireless sensor networks (WSNs) are used to collect data from various nodes in hostile environments, but it is essential to authenticate legitimate nodes to prevent malicious attacks. Traditional authentication schemes may be vulnerable to attack, necessitating the development of more secure authentication techniques. To address this problem, we propose a novel authentication technique for multi-gateway IoT-enabled WSNs that achieves effective and secure data communication. The proposed scheme is based on biological information, hash, and XOR operations, which reduce computational costs. Passwords, biometric information, and session keys are updated securely to maintain forward and backward secrecy during communication. We provide an informal security analysis of the proposed scheme and a formal security analysis of the session key using the real-or-random (ROR) model. We also use the AVISPA simulation toolkit to verify the proposed user authentication scheme. Experimental results demonstrate that our proposed scheme is more effective than previous multi-factor multi-gateway authentication protocols.

KEYWORDS

IoT, multi gateway, multifactor, secure communication, session key, user authentication

1 | INTRODUCTION

Digital transformation has become increasingly important in bridging the physical and cyber worlds, creating computer-based technologies for individuals and organizations. In particular, Internet of Things (IoT) enabled wireless sensor networks (WSNs) have gained popularity as low-cost, cable-free solutions in a wide range of applications. The IoT enabled WSNs are ideal for monitoring environments where running wires or cables is impractical or too expensive, such as oceans, volcanoes, forests, and battlefields. These networks consist of geographically scattered sensor nodes that monitor and record surrounding conditions, transmitting the data to a centralized point. The IoT enabled WSNs can monitor temperature, sound, pollution levels, humidity, and wind, among other things.[1] They have seen widespread application in various industries, providing convenience for day-to-day work and living activities such as real-time video streaming, mobile shopping, and healthcare monitoring. As the number of users of these IoT applications continues to grow, gateways must be able to handle a significant amount of traffic.[2]

IoT enabled WSNs are a essential part of technological infrastructure, providing a source of data for associated applications. These networks typically consist of user nodes, sensor nodes, and gateway nodes, each of which plays a crucial role in the network architecture.[3] When a user needs to access the data, the user node, sensor node, and gateway node connect securely over a wireless channel to share information. However, due to limited resources on sensor nodes, gateway nodes serve as a medium between user nodes and sensor nodes, providing higher processing and communication capacity.[4] In traditional networks, a single gateway is often used, creating bottleneck problems that can negatively impact network performance. To avoid these issues, multi-gateway architectures are becoming increasingly popular, allowing for more efficient and effective data communication.[5]

In order to ensure the effectiveness of the communication system, it is important to avoid a single point of failure in the gateway, as this could render the entire system ineffective.[6] Moreover, with the advent of the future 5G mobile network and its increased standards for energy efficiency, the single gateway design may no longer be efficient enough to meet the demands of the network.[6] As a result, transitioning to a multi-gateway environment is imperative. During a continuing conversation, there are various security breaches that can occur, including eavesdropping, unauthorized message modification, replay attacks, and masquerading.[7,8] To prevent these issues, it is necessary for the communicating parties to engage in mutual authentication.[9] Additionally, the user and gateway must agree on a session key to maintain message confidentiality. If an attacker is able to link multiple instances of the user or determine the user's true identity through traffic analysis or tracing attacks, then the user's privacy could be compromised.[10,11] Therefore, it is essential to ensure that successive sessions are unlinkable and anonymous. Given the vulnerability of IoT-enabled WSNs to a wide range of malicious attacks, it is critical to implement highly efficient security solutions.[2,3] However, the limited power, computing, and storage resources of SNs make it challenging to utilize complex security solutions.

This work proposes a novel authentication technique for multi-gateway IoT enabled WSNs to achieve secure and effective data communication. The proposed method utilizes biological data, in combination with XOR and hash operations, to reduce the amount of computing needed. A formal security analysis of the session key is provided, along with an informal security study of the proposed method. Performance analysis of the suggested technique demonstrates that it is more efficient than existing multi-factor, multi-gateway authentication protocols.

1.1 | Motivation

Authentication is a vital aspect of network security, as it helps organizations to ensure that only authorized users can access their resources and protect their sensitive data from being compromised or stolen. The importance of authentication has become more significant in recent years, with the increase in cyber-attacks and security breaches that have targeted various organizations worldwide. Many authentication schemes have been proposed over the years, but none of them are entirely free from security attacks. As attackers continue to evolve their tactics and methods to bypass security measures, there is a growing need for developing more robust and secure authentication schemes to protect against these threats. To evaluate the security of authentication methods, several criteria can be employed. Formal security analysis is one such criterion, which involves analyzing the security of the authentication method mathematically. Security features, communication cost, and computing overhead are other criteria that can be used to evaluate the effectiveness of authentication methods. Furthermore, the adversary model and system architecture can play a significant role in designing and evaluating authentication methods. The adversary model helps identify the possible threats and attacks that an authentication method may encounter, while the system architecture outlines the components and interactions of the authentication system. By considering the adversary model and system architecture, authentication schemes can be designed to better protect against potential attacks and ensure secure communication.

1.2 | Our contributions

In IoT-enabled WSNs, user identification is a important aspect for privileged access control and personalized services. However, the constrained resources of the sensors, such as processing speed, storage capacity, bandwidth, and connectivity options, make it challenging to achieve a secure and authentic network. To address this issue, a multifactor multi-gateway authentication strategy for IoT-enabled WSNs is proposed in this study. The proposed scheme utilizes multiple gateways to ensure reliable communication while maintaining a high level of security. The main contributions of this work include:

- This work proposes a multifactor authentication scheme to ensure robust and secure communication in multi-gateway based IoT enabled WSNs.
- The scheme includes updating password and biometric information in a secure manner, ensuring forward and backward secrecy even if the database is compromised.
- The session key is updated dynamically using adaptive privacy preservation to maintain the session key confidentiality among users, base station, and sensor nodes.
- The proposed authentication scheme provides robustness against gateway impersonation attacks with low computation and communication costs as compared to existing schemes.
- The security analysis of the multifactor and multi-gateway based authentication scheme for IoT enabled WSNs is performed through both formal and informal security analysis techniques.
- The proposed scheme is effective in mitigating security threats and ensures reliable and secure communication in IoT enabled WSNs with multiple gateways.

The rest of the paper is structured as follows: Section 2 presents the related work. Section 3 discusses the system model and proposed methodology is discussed in the Section 4. The security analysis is given in Section 5 and performance analysis of the proposed and existing papers discusses in Section 6. Finally, the paper is concluded in Section 7 with the future work.

2 | RELATED WORK

In this section, a literature survey of existing authentication techniques in IoT enabled WSNs is discussed. The majority of authentication techniques focus on the session key generation for communication parties. The work presented in Reference [12] discusses a two-factor user authentication scheme using a hash function. However, the limitations of this scheme are that denial of service, node compromise attacks, and the problem of updating passwords are possible. Another work[13] discusses an Elliptic Curve Cryptosystem (ECC) and key agreement-based scheme that removes the drawbacks of[12] by providing protection against privileged insider attacks, offline password guessing attacks, forgery and impersonation attacks, and smart card loss attacks. The scheme is efficient in terms of computation and communication overheads. Next, a work presented in Reference [14] discusses the MAC function and symmetric key cryptosystem which uses OTP (one-time-password) instead of a login password. This scheme reduces communication overhead to improve better security. However, it is vulnerable to replay attacks, which can be exploited to gain unauthorized access. Further, the authors in Reference [15] describe a method that can control replay attacks, smart card breach attacks, stolen smart card attacks, and stolen verifier attacks. However, it is vulnerable to offline password guessing assaults, impersonation attacks, and session key secrecy is not completed.

Xie et al.[16] propose an ECC and Fuzzy extractor-based scheme that can resist offline password guessing attacks and impersonation attacks but does not achieve session key secrecy or perfect forward secrecy. Other works[17,18] have also proposed various methods based on authentication schemes in IoT-enabled WSNs. Another approach is discussed in Reference [19] where a 3-factor authentication scheme is proposed that provides anonymity, forward secrecy, and resistance to various attacks. However, it suffers from vulnerabilities to stolen-verifier attack, de-synchronization attack, and denial of service attack, and does not provide perfect forward secrecy. Singh et al.[18] proposed a scheme that considers 3-FAS and ECC-based fuzzy extractor algorithms, which improves computational cost and security while resisting stolen verification attacks and providing perfect forward secrecy. However, this approach is also limited in terms of the security attacks considered.

Kalid et al.[19] propose an authentication method for IoT enabled WSNs based on 3-FAS using the Rabin cryptosystem. The scheme presented in Reference [12] is based on two-factor user authentication using a hash function, which has limitations such as vulnerability to denial-of-service attacks, node compromise attacks, and difficulty in password updating. However, Reference [19] introduces a scheme based on ECC and key agreement that addresses the drawbacks of the scheme in Reference [12]. The proposed scheme resists attacks from privileged insiders, offline password guessing attacks, forgery and impersonation attacks, smart card loss attacks, and other similar attacks.[16] On the other hand, Reference [20] presents an authentication technique based on ECC and Fuzzy extractor, which can resist offline password guessing attacks and impersonation attacks, but it fails to provide complete session key secrecy and perfect forward secrecy. Moreover, the proposed technique can only handle specific security threats.[16] Several other papers, such as

References [12,21,22], also discuss authentication schemes based on 3FAS in IoT enabled WSNs. While these schemes provide anonymity, forward secrecy, and resistance to multiple attacks, they are vulnerable to stolen-verifier attacks and de-synchronization attacks. Singh et al.[18] proposed a system that uses a one-time password (OTP) instead of a login password, which reduces communication overhead and provides improved security. Additionally, the paper presents a technique based on the combination of 3-FAS and ECC with a fuzzy extractor algorithm, which improves computational cost and security while resisting stolen verification attacks and providing perfect forward secrecy. However, only limited security attacks were considered in this approach.

In Reference [23], a scheme is proposed that satisfies the requirement of perfect forward secrecy, which means that a compromised key in the future cannot be used to decrypt previous communications. The scheme can operate without user participation, but it has a significant drawback: if an adversary gains control of a lost or stolen device along with the necessary decryption information, they can access the encrypted data. Park et al.[24] developed a user anonymity-preserving authentication technique, but subsequent research[25] showed vulnerabilities to attacks such as password guessing, privileged insider, and known session-specific transient information. Furthermore, the scheme does not guarantee user privacy, and the gateway node's private key is not secure. The papers[26–28] provide an in-depth analysis of various authentication schemes used in a range of applications, including smart home communication, intelligent transportation systems, and map street view. These authentication schemes play a crucial role in ensuring secure and reliable communication between different entities.

The proposed method uses various symbols to represent different aspects of the authentication process. A list of symbols and their descriptions is provided below:

Symbols	**Description**
U_i	ith user node
GW_j	$j$thgateway node
SN_k	$k$thsensor node
RC	Registration center
ID_u	Identity of user
ID_{sn_k}	Identity of sensor node
PWD_u	Password of user
Bio_u	User′s biometric informtion
K_u	Secret key of user
K_{GW_j}	Secret key of $j$thgateway
R_u	Biometric secret key of user
hd_u	Helper data used for generating user′s biometric secret key
$Rep(.)$	Reproduction function
$h(.)$	One way hash function
N_u	Nonce generated by user
N_{SN_k}	Nonce generated by $k$thsensor node
SK_{GW-U}	Shared secret key between gateway and user
SK_{GW-SN}	Shared secret key between gateway and sensor node
SK_{RC-SN}	Shared secret key between registration center and sensor node
PK_{GW_j}	Public key of jth gateway node
PK_{U_i}	Public key of $i$thuser
P	Initial point of eleptic curve
I_i	Instance of messages where $i = 1, 2 \ldots$.

SCN_i	Unique smart card number
TS_i	Time stamp where $i = 1, 2 \ldots$
TS_c	Current time stamp
T_d	Transmission delay
P_{id}^{new}	New pseudo identity
P_{id}	Pseudo identity
RN_u	Random number of user
RN_{GW_j}	Random number of jth gateway node
RN_{SN}	Random number of sensor node
$K_{SS}^{SN}, K_{SS}^{GW}, K_{SS}^{U}$	Secure session key generated by the sensor node, gateway node and user
K_{SS}	Secure session key

3 | SYSTEM MODEL

This section presents the proposed secure and effective user authentication system model for multi-gateway IoT enabled WSNs. The model consists of four entities: a centralized registration center (RC), a user (U_i), gateway $\left(GW_j\right)$ and a target sensor node (SN_k). The user first registers with the RC, followed by gateways will register with the RC. When any user sends the message then RC will authenticate the user by checking their identity in the database. After that RC will allow the user for connecting the required GW_j. The SN_k nodes collect the data from the target nodes then forward it to the gateways and then further gateway sends it to the RC. RC will verify the data by considering it is coming from the authorized gateway. If it is legitimate then it will forward to the user. RC is considered as the reliable authority for the U_i and GW_j, using multifactor authentication for entity identification as shown in Figure 1. To keep the design as lightweight as possible, we use fuzzy extractors and novel security assumptions tailored for multi-gateway IoT-enabled WSNs. These are described in detail below.

3.1 | Fuzzy extractor

To keep the design as light as possible, a fuzzy extractor is utilized for the secure authentication process. Moreover, this system model uses the hash function, ex-or operation, and leaves other expensive encryption structures. To maintain security and secrecy, two main functions namely F_zE_z.Gen(.) and F_zE_z. Rec(.) are used for secret key production and

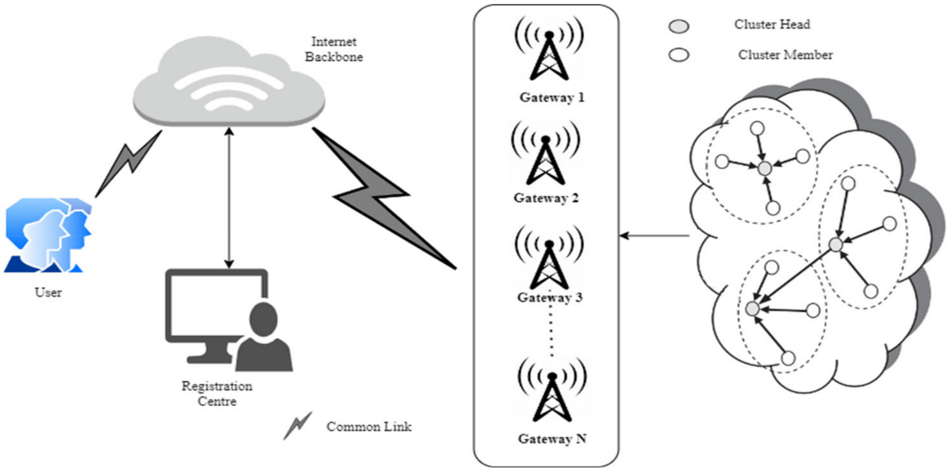


FIGURE 1 The architecture of multi-gateway in IoT enabled WSN.

a re-production function, respectively which are combined to design a lightweight fuzzy extractor.[29,30] These functions are applied to a given input bit string like s, $(K, hd) = F_zE_z\ Gen(s)$, where, $F_zF_z.Gen(.)$ as a probabilistic technique for producing a key K and helper data hd. where $K = FE.Rec\ \left(s', hd\right)$ is the hamming distance between s and s' is no more than d. $F_zE_z.Rec(.)$ is a deterministic function that takes R' as a noisy input and hd as a helper data.

3.2 | Threat model

The following fundamental assumptions are considered under the threat model for examining authentication schemes:

- Attackers have the ability to access and control public channels, making them susceptible to eavesdropping, interception, modification, and replay.
- Any authorized participant, including users, sensors, or base stations, may act as an attacker.
- It is relatively easy for an attacker to guess a low-entropy password or identity individually. However, simultaneously guessing both secret parameters such as password and identity would be computationally infeasible within polynomial time.
- Adversaries are unable to access the private keys, random numbers, or hash values of communication participants.
- An attacker can copy or steal one's smart card (SC) and use power analysis to reveal the engraved data on the stolen or copied SC.
- Attackers may gain access to user credentials if they are not sent securely.
- In practice, it is assumed that the attacker has knowledge of the authentication protocol used in the system.

4 | PROPOSED METHODOLOGY

In this section, a multifactor and multi-gateway based authentication scheme for IoT enabled WSNs is proposed to maintain secrecy and helps to provide a secure network. It consists of nine stages: the user registration phase, gateway registration phase, sensor registration phase, login and authentication phase, password updating phase, biometric updating phase, phase of revoking, session key updating phase, and new node addition phase. While the registration phase employs a private channel, the login and authentication phases use a public channel. The flow of information for the registration, login, and authentication phases is illustrated in Figure 2.

4.1 | Pseudo code for proposed scheme

In this section, the pseudo code for node authentication, integrity function using hash function, and login and authentication phase for proposed scheme is given as follows:

```
Node authentication
Step 1: Wireless sensor nodes join the network for communication purposes.
Step 2: A trusted authority (TA) is designated as the node with higher resources.
Step 3: The TA carries out the following tasks:
        a. Registering sensor nodes
        b. Certifying mobile nodes for communication
        c. Tracing node behavior
```

```
Integrity function using hash function
Inputs: $M_I$= input message, $K$= key, $N$= a nonce
    Output: $M_E$= encrypted message of $M_I$
```

1. $E = Stream\ Cipher(K, N)$ → N vector is encrypted with key (K)

2. $H = Hash\,(K, M_I)$ → K, M_I is input message

3. $MAC = E \bigoplus H$ → Message Authentication Code

4. $M_J = M_I \parallel MAC$ → This MAC store at the file end

5. $K' = K \bigoplus E$ → Key K modified by encrypted nonce N

6. $M_E = Stream\ Cipher\left(K', M_I\right)$ → M_E is the encrypted message of M_E

Login and authentication phase for proposed scheme

Step 1: The user (U_i) enters their credentials for login.
Step 2: The gateway $\left(GW_j\right)$ verifies the user's credentials with its database for authentication and checks the message's freshness and integrity.

```
    **If ($U_i$ is not valid)**
        then $GW_j$ discards the request and send a request for registration to $U_i$.
    **Else**
        $GW_j$ passes the user's message to the target sensor node ($SN_k$) and $SN_k$ authenticates $U_i$ and $GW_j$.
        **If ($U_i$ and $GW_j$ are valid)**
            Then $SN_j$ generate a new session key, update transmission key, and passes it to $GW_j$ while checking the message integrity and freshness.
            Now $GW_j$ authenticates $SN_k$ and verifies the session key and updates the transmission key. After that, it sends response to $U_i$ while checking the message freshness and integrity.
            After receiving the response, $U_i$ authenticates $GW_j$, $SN_k$ and verifies session key while checking message freshness and integrity.
            Now $U_i$, $GW_j$, and $SN_k$ are ready to communicate with each other using the secure session key $K_{SS} = K_{SS}^{U} = K_{SS}^{GW} = K_{ss}^{SN}$ that was established.
        **Else**
            $SN_k$ discards the authentication request of $U_i$ and $GW_j$.
```

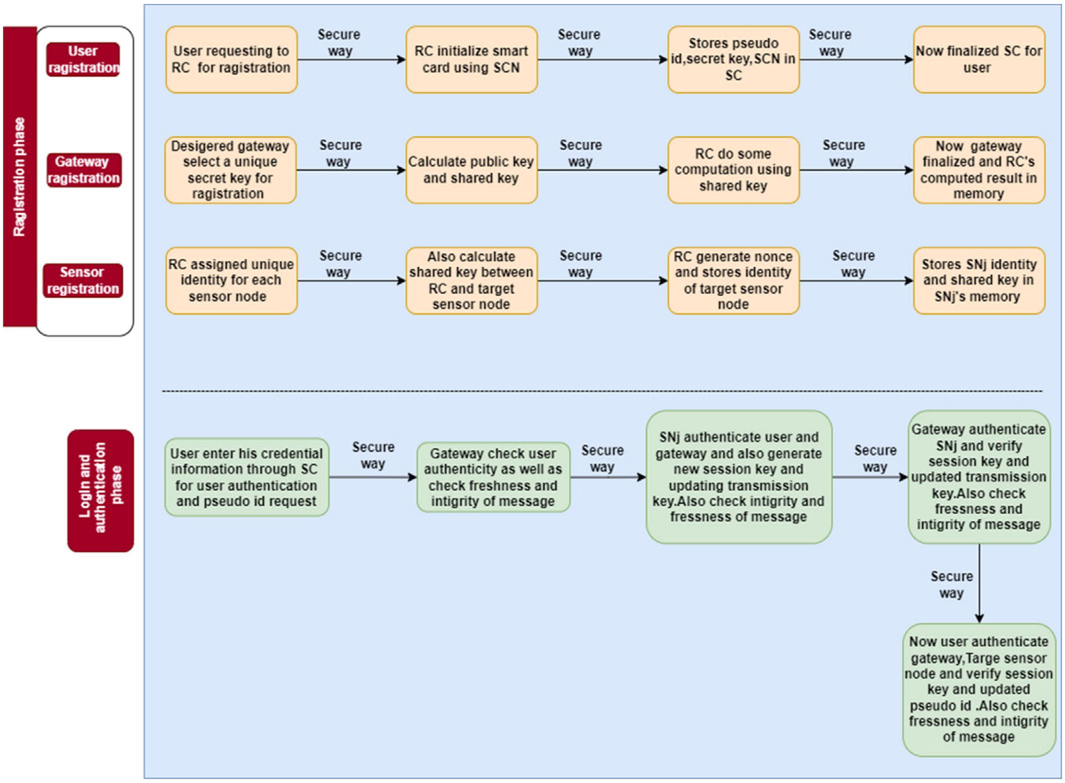

Ragistration phase
User ragistration
Gateway ragistration
Sensor registration
User requesting to RC for ragistration
Secure way
RC initialize smart card using SCN
Secure way
Stores pseudo id,secret key,SCN in SC
Secure way
Now finalized SC for user
Desigered gateway select a unique secret key for ragistration
Secure way
Calculate public key and shared key
Secure way
RC do some computation using shared key
Secure way
Now gateway finalized and RC's computed result in memory
RC assigned unique identity for each sensor node
Secure way
Also calculate shared key between RC and target sensor node
Secure way
RC generate nonce and stores identity of target sensor node
Secure way
Stores SNj identity and shared key in SNj's memory
Login and authentication phase
User enter his credential information through SC for user authentication and pseudo id request
Secure way
Gateway check user authenticity as well as check freshness and intigrity of message
Secure way
SNj authenticate user and gateway and also generate new session key and updating transmission key.Also check intigrity and fressness of message
Secure way
Gateway authenticate SNj and verify session key and updated transmission key.Also check fressness and intigrity of message
Secure way
Now user authenticate gateway,Targe sensor node and verify session key and updated pseudo id .Also check fressness and intigrity of message

FIGURE 2 Flow of information of the registration, login and authentication phase.

4.2 | User registration phase

During the user registration phase, the registration of user (U_i) with gateway $\left(GW_j\right)$ is carried out, following the below steps which involve registering with the registration center (RC). A summary of the registration process is provided in Figure 3, while the detailed steps and their corresponding explanations are presented below:

Step 1: The user U_i enter ID_u, password $PW_{u,}$ and biometric information Bio_u into the user's device. After that device chooses a nonce N_u and a secret key K_u for the user. Then, the device calculates $Gen\,(K_u, Bio_u) = (R_u, hd_u)$, $I_1 = h\,(ID_u \parallel PWD_u \parallel R_u \parallel N_u)$ where *Gen* is used to make the user's biometric secret key R_u and corresponding helper data hd_u using the user's biometric data Bio_u and secret key K_u. Now, $\{ID_u, I_1, N_u\}$ are sent to RC in a safe channel.

Step 2: RC picks an SC number SCN_i after getting U_i's registration information. Then, RC does the computation: $I_2 = h\,(ID_u \parallel I_1)$, $P_{id} = SCN \bigoplus ID_u \bigoplus I_1$, $SK_{GW-U} = h\,(K_{RC} \parallel ID_u \parallel N_u)$, and $I_3 = SK_{GW-U} \bigoplus I_2$ where P_{id} is U_i's pseudo-identity and SK_{GW-U} is the shared secret key that *GW*, RC, and U_i know. K_{RC} is the secret key that is only known by *RC* while N_u is the random nonce created by the user. After *RC* has done all of the calculations, it stores $\{ID_u, P_{id}, N_u\}$ in its database and puts $\{I_2, P_{id}, I_3\}$ into the SC that goes with it. The SC is then sent to U_i through a safe channel.

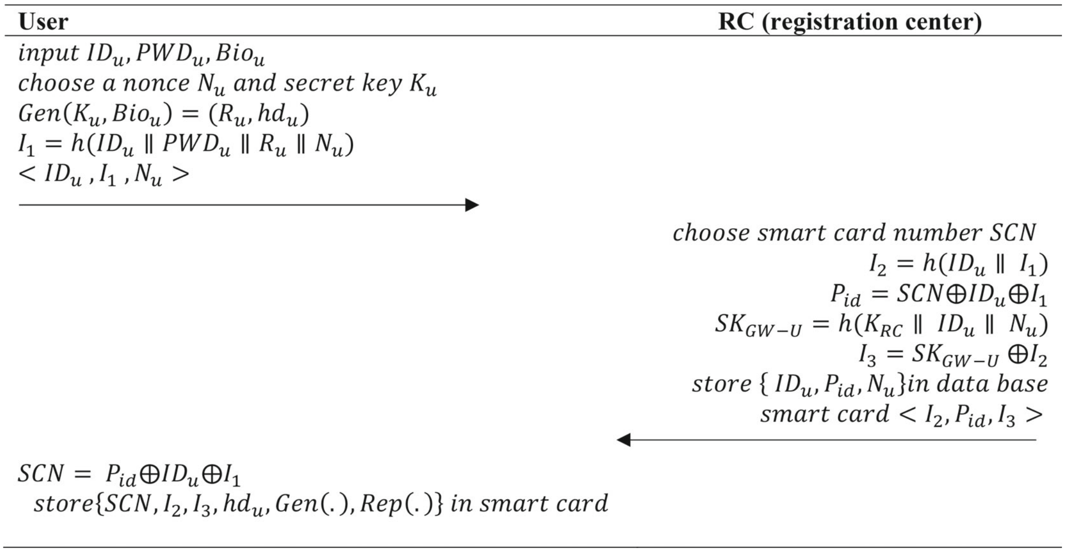
Step 3: After U_i gets the smart card (SC), he figures out $SCN = P_{id} \bigoplus ID_u \bigoplus I_1$ and *store* $\{SCN, I_2, I_3, hd_u, Gen(.), Rep(.)\}$ In SC.

4.3 | Gateway registration phase

Before offering its services, each GW_j has to register with RC. The steps in the GW_j registration phase are as follows. Figure 4 shows a summary of how to register a GW_j. GW_j chooses its true identity, ID_{GW_j}, and a secret key K_{GW_j}. Then, GW_j compute $PK_{GW_j} = K_{GW_j}.P$ and $SK_{GW-U} = K_{GW_j}$. where PK_{U_i} is public key of user and PK_{GW_j} is public key of gateway. Finally, GW_j sends $\left\{ID_{GW_j}, SK_{GW-U}\ \right\}$ to RC. After that RC computes $C = h\left(ID_{GW_j} \parallel SK_{GW-U}\right).K_{RC}$ and it goes to GW_j. Now gateway stores computed value C in his memory.

4.4 | Sensor registration phase

Here is a description of the sensor registration phase. First, RC picks a unique ID_{sn_k} and a random number N_{RC} for the newly added sensor node SN_k. Then, RC figures out the shared secret key, which is $SK_{RC-SN} = ID_{sn_k} \bigoplus N_{RC}$. Last but not least, RC stores $\left\{ID_{sn_k}N_{RC}\right\}$ in its database and $\left\{ID_{sn_k}, SK_{RC-SN}\right\}$ in $SN_k's$ memory before SN_k is put in a special area.



User	**RC (registration center)**
input ID_u, PWD_u, Bio_u *choose a nonce* N_u *and secret key* K_u $Gen(K_u, Bio_u) = (R_u, hd_u)$ $I_1 = h(ID_u \parallel PWD_u \parallel R_u \parallel N_u)$ $< ID_u\ , I_1\ , N_u >$ $\longrightarrow$	
	choose smart card number SCN $I_2 = h(ID_u \parallel\ I_1)$ $P_{id} = SCN \oplus ID_u \oplus I_1$ $SK_{GW-U} = h(K_{RC} \parallel\ ID_u \parallel\ N_u)$ $I_3 = SK_{GW-U} \oplus I_2$ *store* $\{\ ID_u, P_{id}, N_u\}$ *in data base* *smart card* $< I_2, P_{id}, I_3 >$ $\longleftarrow$
$SCN = \ P_{id} \oplus ID_u \oplus I_1$ *store*$\{SCN, I_2, I_3, hd_u, Gen(.), Rep(.)\}$ *in smart card*	

FIGURE 3 User Registration between user and registration centre.

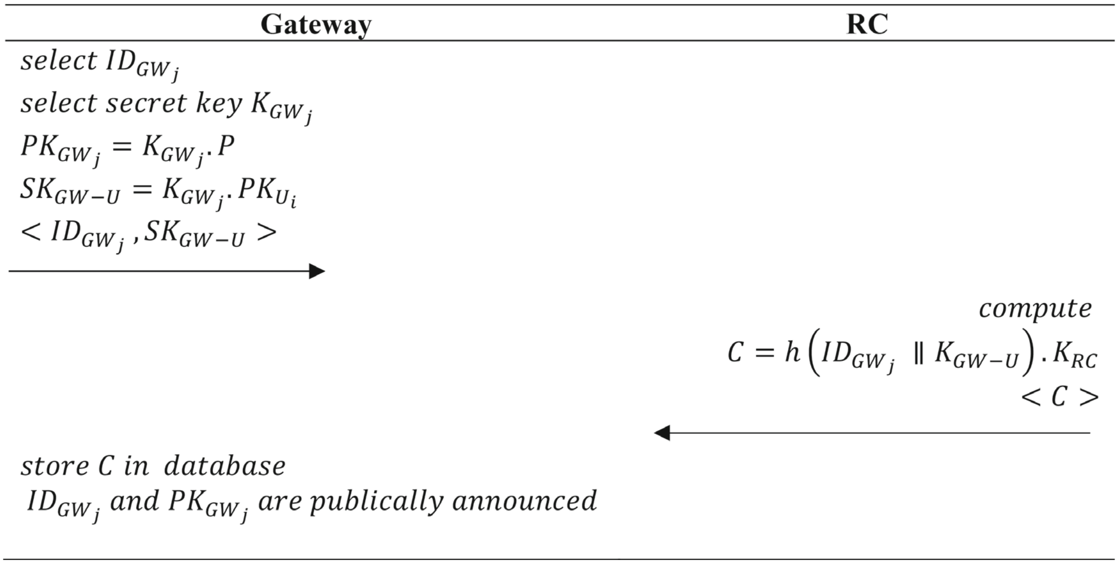

Gateway
RC
$select\ ID_{GW_j}$
$select\ secret\ key\ K_{GW_j}$
$PK_{GW_j} = K_{GW_j}.P$
$SK_{GW-U} = K_{GW_j}.PK_{U_i}$
$< ID_{GW_j}, SK_{GW-U} >$
$compute$
$C = h\left(ID_{GW_j} \parallel K_{GW-U}\right).K_{RC}$
$< C >$
$store\ C\ in\ database$
$ID_{GW_j}\ and\ PK_{GW_j}\ are\ publically\ announced$

FIGURE 4 Gateway registration between gateway and registration center.

4.5 | Login and authentication phase

Whenever U_i needs to connect with a specific SN_k then the following steps will take place as shown in Figure 5. First, U_i enter his smart card SC into the user device (card reader) and enters ID'_u, PWD'_u, Bio'_u and secret key K_u.

Step 1: *Rep*(.) used as a recovery process using adjacent biometrics of user $\left(Bio'_u\right)$ hd_u and secret key K_u. Now card reader computes $R_u = Rep\left(Bio'_u \parallel K_u \parallel hd_u\right)$, $I'_1 = h\left(PWD'_u \parallel R_u\right)$, $I'_2 = h\left(ID'_u \parallel I'_1\right)$. Now check if $I_2 = I'_2$ then device computed $SK_{GW-U} = I_3 \bigoplus h\left(I_1 \parallel ID_u\right)$ and $P_{id} = SCN \bigoplus ID_u \bigoplus I_1$ otherwise the session is terminated and U_i rejected, here P_{id} is pseudo-identity. Now U_i generate a new random number $RN_u = P_{id}^{new} \bigoplus P_{id}$ where P_{id}^{new} is a new pseudo-identity. Now user device compute $m_1 = ID_u \bigoplus h\left(P_{id} \parallel SK_{GW-U}\right)$, $m_2 = RN_u \bigoplus h\left(SK_{GW-U} \parallel ID_u \parallel TS_1\right)$, $m_3 = ID_{sn_j} \bigoplus h\left(SK_{GW-U} \parallel RN_u \parallel P_{id} \parallel TS_1\right), m_4 = h\left(P_{id} \parallel ID_u \parallel ID_{sn_k} \parallel TS_1\right)$. Finally U_i send this authentication request information $INFO\_1 :< m_1, m_2, m_3, m_4, P_{id}, TS_1 >$ to the selected gateway GW_j via a secure communication channel.

Step 2: When GW_j receiving Authentication information $INFO\_1 :< m_1, m_2, m_3, m_4, P_{id}, TS_1 >$ then GW_j first check if $|TS_c - TS_1| \leq T_d$ is false then the authentication request will be discarded, otherwise GW_j fetch ID_u and N_u from the database and compute $SK_{GW-U} = h(K_{GW_j} \parallel ID_u \parallel N_u)$, $ID'_u = m_1 \bigoplus h\left(P_{id} \parallel SK_{GW-U}\right)$ and check if $ID_u = ID'_u$ false than U_i discarded and GW_j treat as an unauthorized user, otherwise U_i is authenticated by GW_j. After that GW_j calculate $RN_u = m_2 \bigoplus h\left(SK_{GW-U} \parallel ID_u \parallel TS_1\right)$, $ID_{sn_k} = m_3 \bigoplus h\left(SK_{GW-U} \parallel RN_u \parallel P_{id} \parallel TS_1\right)$ and check if $m_4 = h\left(P_{id} \parallel ID_u \parallel ID_{sn_k} \parallel RN_u \parallel TS_1\right)$ is false then GW_j rejects authentication request of U_i else U_i is authenticated by GW_j. After that GW_j fetch N_u from the database and compute $SK_{GW-SN} = ID_{sn_k} \bigoplus N_{SN_k}$, $m_5 = RN_u \bigoplus h(SK_{GW-SN} \parallel ID_{sn_k} \parallel TS_2)$, $m_6 = RN_{GW_j} \bigoplus h\left(RN_u \parallel SK_{GW-SN} \parallel TS_2\right)$, $m_7 = h(ID_{sn_k} \parallel P_{id} \parallel RN_u \parallel RN_{GW_j} \parallel TS_2)$ where RN_{GW} is a random number generated by GW_j and TS_2 is a time stamp. Now GW_j send this authentication information $INFO\_2 :< m_5, m_6, m_7, P_{id}, TS_2 >$ to the targeted SN_j via secure way.

Step 3: When SN_j receive authentication request information $INFO\_2 :< m_5, m_6, m_7, P_{id}, TS_2 >$ then first SN_k check the freshness of the timestamp if $|TS_c - TS_2| \leq T_d$ false then the authentication request is discarded otherwise SN_j calculate $RN_u = m_5 \bigoplus h\left(SK_{Gw-SN} \parallel ID_{sn_k} \parallel TS_2\right)$, $RN_{GW_j} = m_6 \bigoplus h\left(RN_u \parallel SK_{GW-SN} \parallel TS_2\right)$ and check if $m_7 = h\left(ID_{sn_k} \parallel P_{id} \parallel RN_u \parallel RN_{GW_j} \parallel TS_2\right)$ false then SN_k treat as unauthorized GW_j and SN_k discard authentication request of GW_j, otherwise GW_j successfully authenticated by SN_k and further, compute $K_{ss}^{SN} = h\left(\left(ID_{sn_k} \parallel P_{id}\right) \parallel RN_u \parallel RN_{GW_j} \parallel RN_{SN}\right)$, $m_8 = RN_{SN} \bigoplus h\left(RN_u \parallel RN_{GW_j}\right)$, $m_9 = h\left(ID_{sn_k} \parallel RN_{GW_j} \parallel RN_{SN} \parallel K_{ss}^{SN} \parallel TS_3\right)$, $SK_{GW-SN}^{new} = SK_{GW-SN} \bigoplus RN_{GW_j}$ where RN_{SN} is a random number generated by SN. Now SK_{GW-SN} replace with SK_{GW-SN}^{new} in memory. Finally SN_k send authentication response information $INFO_3 :< m_8, m_9, TS_3 >$ to the GW_j via a secure channel.

User **Gateway** **Sensor node**

For login, U_i insert smart card

U_i input ID'_u, PWD'_u, Bio'_u and secret key K_u

$R_u = Rep(Bio'_u \parallel K_u \parallel hd_u)$

$I'_1 = h(PWD'_u \parallel R_u \parallel N_u \parallel ID'_u)$

$I'_2 = h(ID'_u \parallel I'_1)$

check $I_2 = I'_2$

$SK_{GW-U} = I_3 \oplus h(I_1 \parallel ID_u)$

$P_{id} = SCN_i \oplus ID_u \oplus I_1$

generate time stamp TS_1

select a new pseudo id P_{id}^{new}

$RN_u = P_{id}^{new} \oplus P_{id}$

$m_{1=}ID_u \oplus h(P_{id} \parallel SK_{GW-U})$

$m_2 = RN_u \oplus h(SK_{GW-U} \parallel ID_u \parallel TS_1)$

$m_3 = ID_{sn_k} \oplus h(SK_{GW-U} \parallel RN_u \parallel P_{id} \parallel TS_1)$

$m_4 = h(P_{id} \parallel ID_u \parallel ID_{sn_k} \parallel TS_1)$

$INFO\_1: < m_1, m_2, m_3, m_4, P_{id}, TS_1 >$

$\longrightarrow$

if $|TS_c - TS_1| \le T_d$, then

fetch ID_u and N_u in database(from P_{id})

$SK_{GW-U} = h(K_{GW_j} \parallel ID_u \parallel N_u)$

$ID'_u = m_1 \oplus h(P_{id} \parallel SK_{GW-U})$

check if $ID_u = ID'_u$

U_i is authenticated by GW_j

$RN_u = m_2 \oplus h((SK_{GW-U} \parallel ID_u \parallel TS_1)$

$ID_{sn_k} = m_3 \oplus h(SK_{GW-U} \parallel RN_u \parallel P_{id} \parallel TS_1)$

check if $m_4 = h\left(P_{id} \parallel ID_u \parallel ID_{sn_k} \parallel RN_u \parallel TS_1\right)$ then,

U_i is authenticated by GW_j

Now, fetch N_u in database $\left(from\ ID_{sn_k}\right)$

$SK_{GW-SN} = ID_{sn_k} \oplus N_{SN_k}$

generate RN_{GW} and timestamp TS_2

$m_5 = RN_u \oplus h(SK_{GW-SN} \parallel ID_{sn_k} \parallel TS_2)$

$m_6 = RN_{GW_j} \oplus h(RN_u \parallel SK_{GW-SN} \parallel TS_2)$

$m_7 = h(ID_{sn_k} \parallel P_{id} \parallel RN_u \parallel RN_{GW_j} \parallel TS_2)$

$INFO\_2: < m_5, m_6, m_7, P_{id}, TS_2 >$

$\longrightarrow$

if $|TS_c - TS_2| \le T_d$, then

$RN_u = m_5 \oplus h(SK_{Gw-SN} \parallel ID_{sn_k} \parallel TS_2)$

$RN_{GW_j} = m_6 \oplus h(RN_u \parallel SK_{GW-SN} \parallel TS_2)$

check if $m_7 = h\left(ID_{sn_k} \parallel P_{id} \parallel RN_u \parallel RN_{GW_j} \parallel TS_2\right)$

GW_j is authenticate by SN_k

then,

generate RN_{SN} and timestamp TS_3

$K_{SS}^{SN} = h\left(\left(ID_{sn_k} \parallel P_{id}\right) \parallel RN_u \parallel RN_{GW_j} \parallel RN_{SN}\right)$

FIGURE 5 Continued on next page.

$$m_8 = RN_{SN} \oplus h(RN_u \parallel RN_{GW_j})$$
$$m_9 = h(ID_{sn_k} \parallel RN_{GW_j} \parallel RN_{SN} \parallel K_{SS}^{SN} \parallel TS_3)$$
$$SK_{GW-SN}^{new} = SK_{GW-SN} \oplus RN_{GW_j}$$
$$replace\ SK_{GW-SN}\ with\ SK_{GW-SN}^{new}\ in\ memory$$
$$INFO\_3: < m_8, m_9, TS_3 >$$

←

$$if\ |TS_c - TS_3| \leq T_d\ ,then$$
$$RN_{SN} = m_8 \oplus h\left(RN_u \parallel RN_{GW_j}\right)$$
$$K_{SS}^{GW} = h((P_{id} \oplus ID_{sn_k}) \parallel RN_u \parallel RN_{GW_j} \parallel RN_{SN})$$
$$check\ if\ m_9 = h(ID_{sn_k} \parallel RN_{GW_j} \parallel K_{SS}^{GW} \parallel RN_{SN} \parallel TS_3)\ then\ ,$$
$$SN_k\ is\ authenticated\ by\ GW_j$$
$$\text{Now, } generate\ timestamp\ TS_4$$
$$P_{id}^{new} = P_{id} \oplus RN_u$$
$$N_{SN_k}^{new} = N_{SN_k} \oplus RN_{GW_j}$$
$$m_{10} = RN_{GW_j} \oplus h(P_{id}^{new} \parallel SK_{GW-U} \parallel TS_4)$$
$$m_{11} = h(P_{id}^{new} \parallel K_{SS}^{GW} \parallel RN_{GW_j} \parallel RN_{SN} \parallel TS_4)$$
$$update < ID_u\ , P_{id}^{new} > in\ database$$
$$update < ID_{sn_k}, N_{SN_k}^{new} > in\ database$$
$$INFO\_4: < m_8, m_{10}, m_{11}, TS_4 >$$

←

$$if\ |TS_c - TS_4| \leq T_d\ ,then$$
$$RN_{GW} = m_{10} \oplus h(P_{id}^{new} \parallel SK_{GW-U} \parallel TS_4)$$
$$RN_{SN} = m_8 \oplus h(RN_u \parallel RN_{GW_j})$$
$$K_{SS}^{U} = h((P_{id} \oplus ID_{sn_k}) \parallel RN_u \parallel RN_{GW_j} \parallel RN_{SN} \parallel TS_4)$$
$$check\ if\ \ m_{11} = h(P_{id}^{new} \parallel K_{SS}^{U} \parallel RN_{GW_j} \parallel RN_{SN} \parallel TS_4)$$
$$GW_j\ is\ authenticated\ by\ user\ U_i$$
$$Now\ compute\ ,$$
$$SCN^{new} = SCN \oplus RN_u$$
$$replace\ SCN\ with\ SCN^{new}\ in\ smart\ card$$

User, gateway, sensor are agree on session key

$$K_{SS} = K_{SS}^{U} = K_{SS}^{GW} = K_{ss}^{SN}$$

FIGURE 5 Login and authentication phase for proposed scheme.

Step 4: When GW_j receive authentication response information $INFO\_3 :< m_8, m_9, TS_3 >$ then first check if $|TS_c - TS_3| \leq T_d$ false, then GW_j discard authentication response, otherwise, calculate $RN_{SN} = m_8 \bigoplus h\left(RN_u \parallel RN_{GW_j}\right)$, $K_{SS}^{GW} = h\left(P_{id} \bigoplus ID_{sn_k}\right) \parallel RN_u \parallel RN_{GW_j} \parallel RN_{SN}$ and check if $m_9 = h\left(ID_{sn_k} \parallel RN_{GW_j} \parallel K_{SS}^{GW} \parallel RN_{SN} \parallel TS_3\right)$ is false then discard the authentication response, otherwise SN_k is authenticated by GW_j and after that GW_j calculate $P_{id}^{new} = P_{id} \bigoplus RN_u$, $N_{SN_k}^{new} = N_{SN_k} \bigoplus RN_{GW_j}$, $m_{10} = RN_{GW_j} \bigoplus h\left(P_{id}^{new} \parallel SK_{GW-U} \parallel TS_4\right)$, $m_{11} = h\left(P_{id}^{new} \parallel K_{SS}^{GW} \parallel RN_{GW_j} \parallel RN_{SN} \parallel TS_4\right)$. Now GW_j update $\left\{ID_u, P_{id}^{new}\right\}$, $\left\{ID_{sn_k}, N_{SN_k}^{new}\right\}$ in the database of GW_j. Now finally GW_j send authentication response information $INFO4 :< m_8, m_{10}, m_{11}, TS_4 >$ to the U_i via a secure channel.

Step 5: When U_i receive authentication response information $INFO\_4 :< m_8, m_{10}, m_{11}, TS_4 >$ then first check the freshness of the timestamp if $|TS_c - TS_4| \leq T_d$ false then U_i discard authentication response otherwise, compute

$RN_{GW} = m_{10} \bigoplus h\left(P_{id}^{new} \parallel SK_{GW-U} \parallel TS_4\right)$, $RN_{SN} = m_8 \bigoplus h\left(RN_u \parallel RN_{GW_j}\right)$, $K_{SS}^{U} = h(\left(P_{id} \bigoplus ID_{sn_k}\right) \parallel RN_u \parallel RN_{GW_j} \parallel RN_{SN} \parallel TS_4)$, and check if $m_{11} = h\left(P_{id}^{new} \parallel K_{SS}^{U} \parallel RN_{GW_j} \parallel RN_{SN} \parallel TS_4\right)$ false then U_i treat as an unauthenticated GW_j else U_i treat as an authenticated GW_j. After that U_i compute $SCN^{new} = SCN \bigoplus RN_u$ and finally U_i replace SCN with SCN^{new} in SC.

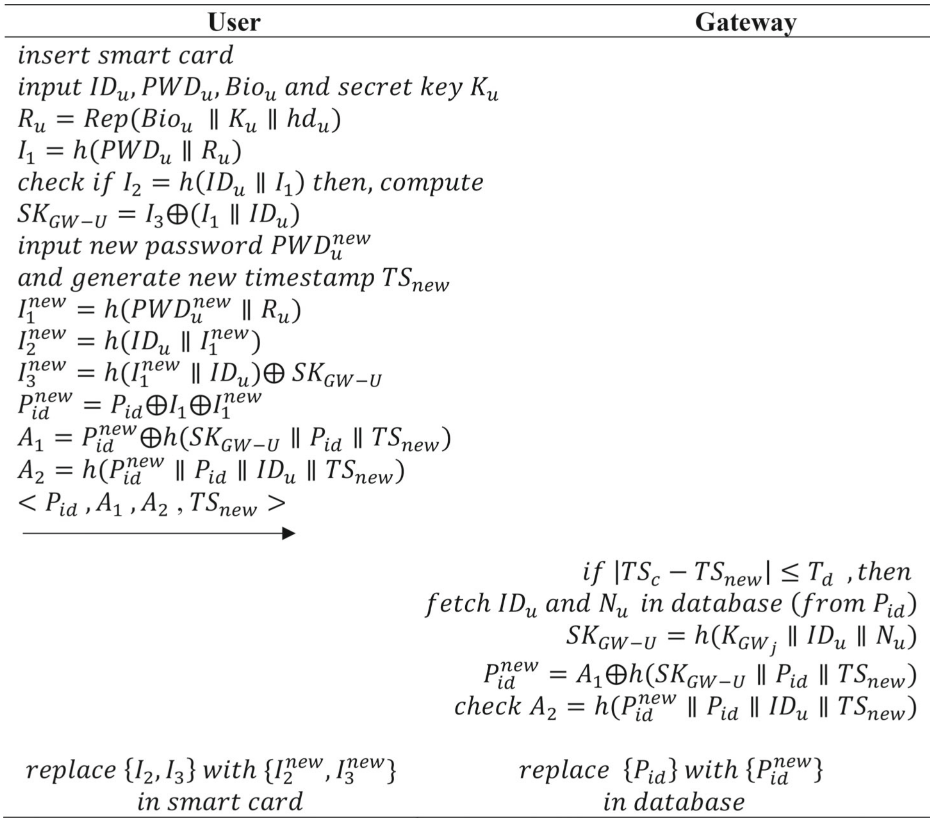
Then, U_i, GW_j, and SN_k will be ready to communicate with each other using the secure session key $K_{SS} = K_{SS}^{U} = K_{SS}^{GW} = K_{ss}^{SN}$ that was established.

4.6 | Password updating phase

The password updating phase is required to maintain secrecy and protect against an information breach attack. Figure 6 depicts the process, while the required steps are described below:

Step 1: First, the user puts his SC into a user device and enters his identity ID_u, password PWD_u, biometric information Bio_u, and secret key K_u. After that $rep(.)$ generate a R_u using Bio_u, K_u and hd_u. After that user device compute $I_1 = h\,(PWD_u \parallel R_u)$ and check if $I_2 = h\,(ID_u \parallel I_1)$ is false then discard the process otherwise user device compute $SK_{GW-U} = I_3 \bigoplus (I_1 \parallel ID_u)$, $I_1^{new} = h\,(PWD_u^{new} \parallel R_u)$, $I_2^{new} = h\left(ID_u \parallel I_1^{new}\right)$, $I_3^{new} = h\left(I_1^{new} \parallel ID_u\right) \bigoplus SK_{GW-U}$, $P_{id}^{new} = P_{id} \bigoplus I_1 \bigoplus I_1^{new}$ where PWD_u^{new} is a new password taken by the user and TS_{new} is a new time stamp. Now U_i calculate $A_1 = P_{id}^{new} \bigoplus h\,(SK_{GW-U} \parallel P_{id} \parallel TS_{new})$, $A_2 = h\left(P_{id}^{new} \parallel P_{id} \parallel ID_u \parallel TS_{new}\right)$ and finally U_i send authentication request information $\{P_{id}, A_1, A_2, TS_{new}\}$ to the gateway GW_j m as shown in Figure 6.

Step 2: When GW_j receive authentication request information $\{P_{id}, A_1, A_2, TS_{new}\}$, first GW_j check the freshness of the timestamp if $|TS_c - TS_{new}| \leq T_d$ false then discard the authentication process and terminate the session otherwise GW_j compute $SK_{GW-U} = h\left(K_{GW_j} \parallel ID_u \parallel N_u\right)$, $P_{id}^{new} = A_1 \bigoplus h\,(SK_{GW-U} \parallel P_{id} \parallel TS_{new})$ and check if $A_2 = h\left(P_{id}^{new} \parallel P_{id} \parallel ID_u \parallel TS_{new}\right)$ false then discard further process and terminate the session otherwise GW_j replace P_{id} with new P_{id}^{new} in database. Finally, the user device replace $\{I_2, I_3\}$ with $\left\{I_2^{new}, I_3^{new}\right\}$ in SC.



User | **Gateway**

insert smart card

input ID_u, PWD_u, Bio_u *and secret key* K_u

$R_u = Rep(Bio_u \parallel K_u \parallel hd_u)$

$I_1 = h(PWD_u \parallel R_u)$

check if $I_2 = h(ID_u \parallel I_1)$ *then, compute*

$SK_{GW-U} = I_3 \oplus (I_1 \parallel ID_u)$

input new password PWD_u^{new}

and generate new timestamp TS_{new}

$I_1^{new} = h(PWD_u^{new} \parallel R_u)$

$I_2^{new} = h(ID_u \parallel I_1^{new})$

$I_3^{new} = h(I_1^{new} \parallel ID_u) \oplus SK_{GW-U}$

$P_{id}^{new} = P_{id} \oplus I_1 \oplus I_1^{new}$

$A_1 = P_{id}^{new} \oplus h(SK_{GW-U} \parallel P_{id} \parallel TS_{new})$

$A_2 = h(P_{id}^{new} \parallel P_{id} \parallel ID_u \parallel TS_{new})$

$< P_{id}, A_1, A_2, TS_{new} >$ $\longrightarrow$

if $|TS_c - TS_{new}| \leq T_d$ *, then*

fetch ID_u *and* N_u *in database (from* P_{id}*)*

$SK_{GW-U} = h(K_{GW_j} \parallel ID_u \parallel N_u)$

$P_{id}^{new} = A_1 \oplus h(SK_{GW-U} \parallel P_{id} \parallel TS_{new})$

check $A_2 = h(P_{id}^{new} \parallel P_{id} \parallel ID_u \parallel TS_{new})$

replace $\{I_2, I_3\}$ *with* $\{I_2^{new}, I_3^{new}\}$ *in smart card* | *replace* $\{P_{id}\}$ *with* $\{P_{id}^{new}\}$ *in database*

FIGURE 6 Password updating phase for the proposed scheme.

4.7 | Biometric updating phase

When U_i needs to change his/her biometric data, U_i must validate his/her ID_u and biometric data to GW_j in a secure manner. After successful validation, *Ui* enters his credential information like id, password, and secret key for making a new secret biometric data. To re-register U_i, *GW* will use the original ID_u, PWD_u, secret key, and new updated biometric data. Registration and offline confirmation are used to update biometric data. If you need more information about reregistering a user, look at Figure 3.

4.8 | Phase of revoking

In the proposed scheme, both U_i and SN are controlled by the gateway GW_j and GW_j has the right to reject their authentication request if GW_j found any valid reasons. Also GW_j will remove their registration record $\{ID_u, P_{id}, N_u\}$ or $\{ID_{sn_k}, N_{SN_k}\}$ from its database. During the authentication of U_i and SN_k, GW_j also check ID_u, ID_{sn_k} and N_u in its database for verifying U_i and SN_k. If the information is wrong, GW_j will stop the process of authentication and deny the request of $U_i{}'$ as well as SN_k's request.

4.9 | Session key updating phase

To maintain secrecy and confidentiality session key updation is required. The following steps are considered for this phase.

Step 1: U_i select new nonce $N_u^{new\_1}$ and generate a new random number $RN_u^{new\_1} = P_{id}^{new} \bigoplus P_{id}$. Now user device calculates $m_{12} = RN_u^{new\_1} \bigoplus K_{SS}$, $m_{13} = h\left(P_{id}^{new} \parallel K_{SS} \parallel ID_u \parallel ID_{sn_k} \parallel TS_5\right)$ and send updated request information $\{m_{12}, m_{13}, TS_5\}$ to GW_j.

Step 2: After receiving an updated request GW_j check the freshness of the timestamp if $|TS_c - TS_5| \leq T_d$ false then discard the request otherwise GW_j computes $RN_u^{new\_1} = m_{12} \bigoplus K_{SS}$, and verify if $m_{13} = h\left(P_{id}^{new} \parallel K_{SS} \parallel ID_u \parallel ID_{sn_k} \parallel RN_u^{new\_1} \parallel TS_5\right)$ is false then discard the updated request otherwise GW_j compute $m_{14} = RN_u^{new\_1} \bigoplus K_{SS}$, $m_{15} = RN_{GW_j}^{new\_1} \bigoplus K_{SS}$, $m_{16} = h(ID_{sn_k} \parallel P_{id}^{new} \parallel RN_u^{new\_1} \parallel RN_{GW}^{new\_1} \parallel TS_6)$ and forward updated request information $\{m_{14}, m_{15}, m_{16}, TS_6\}$ to SN_k.

Step 3: After receiving updated request information $\{m_{14}, m_{15}, m_{16}, TS_6\}$, first SN_k check the freshness of the time stamp if $|TS_c - TS_6| \leq T_d$ false then discard the updated request otherwise computes $RN_u^{new\_1} = m_{14} \bigoplus K_{SS}$, $RN_{GW}^{new_1} = m_{15} \bigoplus K_{SS}$ and check if $m_{16} = h\left(ID_{sn_k} \parallel P_{id}^{new} \parallel RN_u^{new\_1} \parallel RN_{GW_j}^{new\_1} \parallel TS_6\right)$ false then discard the updated request otherwise SN_k generate a *new* session key $K_{NSS}^{SN} = h\left(\left(ID_{sn_k} \parallel P_{id}^{new}\right) \parallel RN_u^{new\_1} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1}\right)$ and compute $m_{17} = RN_{SN}^{new_1} \bigoplus K_{SS}$, $m_{18} = h\left(ID_{sn_k} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1} \parallel K_{NSS}^{SN} \parallel TS_7\right)$ and now SN_k send updated response information $\{m_{17}, m_{18}, TS_7\}$ to GW_j.

Step 4: After receiving updated response information $\{\mathrm{m}_{17}, \mathrm{m}_{18}, \mathrm{TS}_7\}$, GW_j check if $|TS_c - TS_7| \leq T_d$ is false then terminate the session, otherwise compute $RN_{SN}^{new\_1} = m_{17} \bigoplus K_{ss}$, generate $K_{NSS}^{GW} = h\left(\left(P_{id}^{new} \bigoplus ID_{sn_k}\right) \parallel RN_u^{new\_1} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1}\right)$ and verify if $m_{18} = h\left(ID_{sn_k} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1} \parallel K_{NSS}^{SN} \parallel TS_7\right)$ is false then terminate the session otherwise compute $m_{19} = RN_{GW_j}^{new\_1} \bigoplus K_{SS}$, $m_{20} = h(P_{id}^{new\_1} \parallel K_{NSS}^{GW} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1} \parallel TS_8)$ and finally, send GW_j send updated response information $\{m_{17}, m_{19}, m_{20}, TS_8\}$ to U_i.

Step 5: Now after receiving updated response information $\{m_{19}, m_{20}, TS_8\}$, U_i check freshness of TS_8. If TS_8 expired then U_i terminate session else U_i compute $RN_{GW_j}^{new\_1} = m_{19} \bigoplus K_{SS}$, $RN_{SN}^{new\_1} = m_{17} \bigoplus K_{SS}$ and generate a new session key $K_{NSS}^{U} = h\left(\left(P_{id}^{new\_1} \bigoplus ID_{sn_k}\right) \parallel RN_u^{new\_1} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1} \parallel TS_8\right)$. Now U_i check if, $m_{20} = h\left(P_{id}^{new\_1} \parallel K_{NSS}^{U} \parallel RN_{GW_j}^{new\_1} \parallel RN_{SN}^{new\_1} \parallel TS_8\right)$ is false then terminate the session else every participant (U_i, GW_j, SN_k) agree on the session key $K_{NSS} = K_{NSS}^{U} = K_{NSS}^{GW} = K_{NSS}^{SN}$.

4.10 | New node addition phase

Nodes can be added when needed because the IoT enabled WSN is flexible and can grow. This phase will be run when a SN_{new} is added to the IoT enabled WSN. First, just like in the pre-deployment phase, SN_{new} is already set up with an identifier called $ID_{sn_{new}}$ and a shared secret key called $SK_{GW-SN_{new}}$ that is shared with GW_j. SN_{new}'s memory stores $SK_{GW-SN_{new}}$, and GW_j stores $ID_{sn_{new}}$ and $SK_{GW-SN_{new}}$ for SN_{new}. Second, registration is done between GW_j and a SN_{new}. After going through the steps above, SN_{new} is successfully added to the IoT enabled WSN, and a user can connect to it just like any other SN.

The time-line diagram and state transition diagram are given in Figures 7 and 8, respectively.

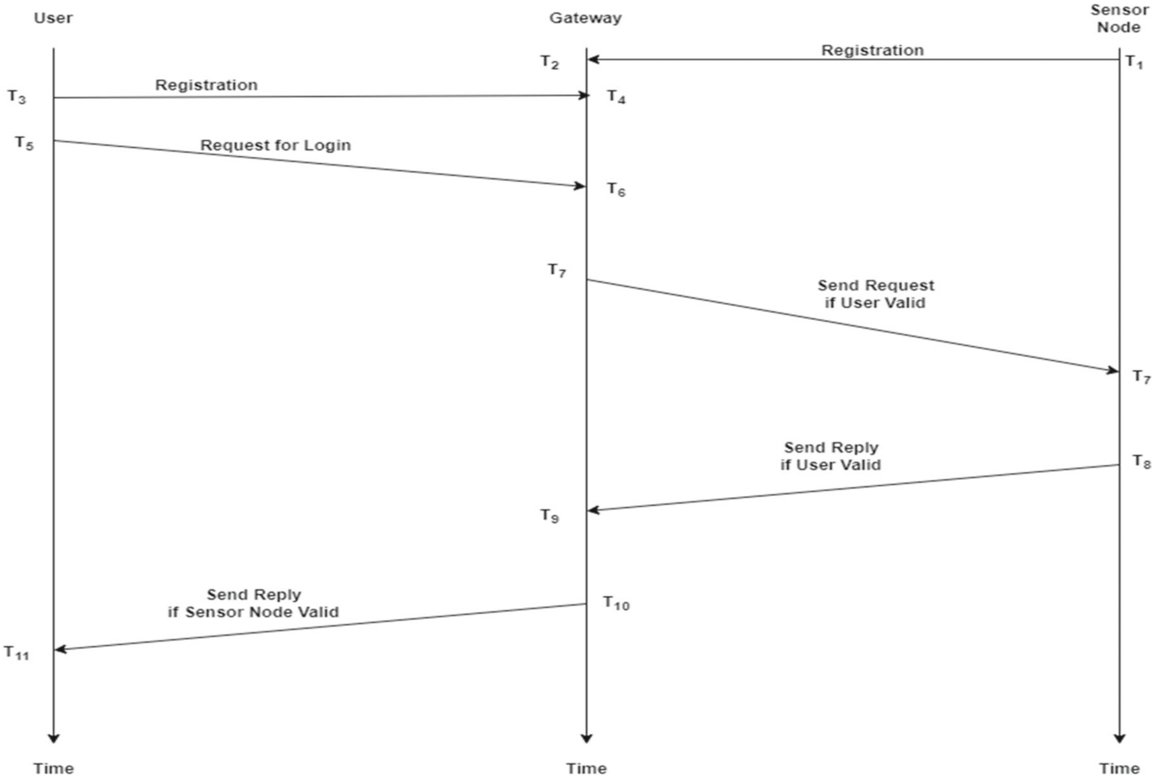


FIGURE 7 Time-line diagram.

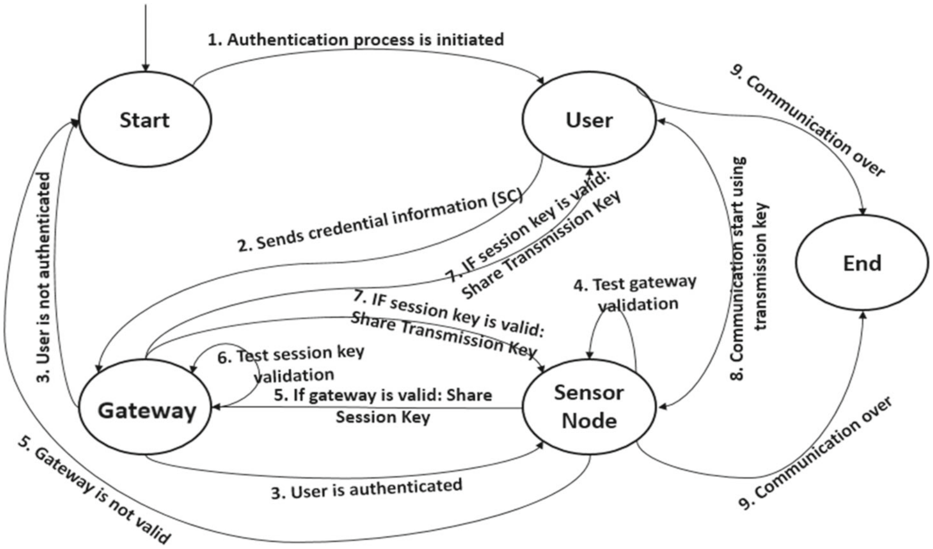

1. Authentication process is initiated
Start
User
End
Gateway
Sensor Node
9. Communication over
2. Sends credential information (SC)
7. IF session key is valid: Share Transmission Key
7. IF session key is valid: Share Transmission Key
4. Test gateway validation
8. Communication start using transmission key
3. User is not authenticated
6. Test session key validation
5. If gateway is valid: Share Session Key
3. User is authenticated
5. Gateway is not valid
9. Communication over

FIGURE 8 State transition diagram.

5 | SECURITY ANALYSIS

This section presents the performance analysis of the proposed scheme. To verify the security analysis we apply ROR[31,32] model to the proposed scheme in a formal way which is described as follows.

5.1 | Formal security analysis using ROR model

This model uses U_i, GW_j and SN_k oracle instances in which the attacker introduces many queries and gets corresponding replies. We begin with a formal adversary model and some formal definitions. Based on the formal adversary model and definition, we build a theorem for our proposed scheme to analyze security formally. When we show that our build theorem is true then security analysis verification is completed. Let I_u^i be the Oracle instance of U_i, I_{GW}^j be the Oracle instance of GW_j, I_{SN}^k be the Oracle instance of SN_k.

Participants: in our scheme total of three participants appear that are user U_i, gateway GW_j and target SN_k are instances, respectively, I_u^i, I_{GW}^j and I_{SN}^k.

Accept state: When an instance $\mathbb{I}$ (which can be I_u^i, I_{GW}^i or I_{SN}^i) has received all of the messages in a session for which it has been assigned a session identifier, it is said to be in the accepted state.

Freshness: I_u^i, I_{GW}^j and I_{SN}^k are said to be "fresh" if they have successfully set up a session key and $\mathbb{A}$ has not asked them to reveal it.

Partnering: I_u^i, I_{GW}^j and I_{SN}^k are partners if all the participants are in an accepted state and also completed their mutual authentication with retaining their identities.

One-way hash function that can handle collisions: Let $h : \{0,1\}^* \rightarrow \{0,1\}^n$ be an irreversible hash function that generates a fixed length n-bit message digest M from an input k, so that $M = h(k)$ where k is variable length. Then, the chance of finding a collision where $h(k_1) = h(k_2)$ is nearly impossible where $k_1 \neq k_2$.

In this model, the attacker can listen, change, delete or insert the messages being sent and received during the communication. In addition, $\mathbb{I} = \{I_u^i, I_{GW}^j, I_{SN}^k\}$ and the attacker will have access to the following inquiries.

1. Execute($\mathbb{I}$) **:** The result of this query, which detects passive listening for protocols, includes a communication transcript for every exchange that took place throughout the authentication protocol between I_u^i, I_{GW}^j and I_{SN}^k.
2. send $\left(\boldsymbol{I_u^i}, \text{start}\right)$: The authentication protocol is being initialized with this query.
3. send($\mathbb{I}$,$\boldsymbol{M}$) **:** This query detects ongoing attacks by filtering and snooping on communications. The attacker sends message M to instance $\mathbb{I}$ and receives a reply from $\mathbb{I}$. coin C
4. Currupt($\mathbb{I}$, λ) : $\mathbb{A}$ is capable of disclosing all the factors except one. The following circumstances will be happened if the attacker executes this query.
   (4.1) If $\lambda = 1$ and $\mathbb{I} = I_u^i$, then $\mathbb{A}$ will be able to access all the parameters stored on the SC as well as I_u^i's password.
   (4.2) If $\lambda = 2$ and $\mathbb{I} = I_u^i$, then $\mathbb{A}$ obtains I_u^i's biometric data in addition to all of the values stored on the SC.
   (4.3) If $\lambda = 3$ and $\mathbb{I} = I_u^i$, then biometric data and I_u^i's password is returned to $\mathbb{A}$.
   (4.4) If $\lambda = 4$ and $\mathbb{I} = I_u^i$, then $\mathbb{A}$ discloses a secret key K_{GW} of I_{GW}^j.
5. Hash($\boldsymbol{S}$) : When the attacker executes, this query then it takes as a variable length input string S to receive as s fixed-length hash value of string S.
6. Reveal($\mathbb{I}$) : $\mathbb{A}$ Executes this query to I (or its partner) to get the K_{SS}.
7. Test($\mathbb{I}$) : If $\mathbb{A}$ does the query, then coin C is turned over. If C = 0, $\mathbb{A}$ is returned by $\mathbb{A}$ random key with the same size as K_{SS}. If C = 1, $\mathbb{A}$ can get the right session key K_{SS}.

Theorem 1. In the ROR model,[33,34] suppose the attacker may be Execute, Send, Hash, Corrupt, and Test queries. Let ADV_A^P be the dominant function for an attacker who can break the security of our proposed scheme in polynomial time t. then,

$$ADV_A^P(t) \leq \frac{3q_H^2}{2^L} + \frac{(q_{SND} + q_{EX})^2}{n} + \frac{2q_{SND}}{2^{L-1}} + 2C'.q_{SND}^{S'} \tag{1}$$

Here q_{SND}: *Send* queries execution time, q_{EX}: *Execute* queries execution time, q_H: *Hash* queries execution time, L: required bit for biometric information, C' and S' are two constants respectively.[33]

Proof. The theorem is shown to be true by using the game sequence $Game_n$, where $n = 0, 1, 2, 3, 4, 5, 6$, which is similar to how[35–37] were shown to be true. Let these game sequence $Game_n$ played $\mathbb{A}$ and the successful probability of $Game_n$ is $P\,[success_n]$ in time t means $\mathbb{A}$ guess the flip coin in $Game_n$ successfully. ■

***Game*₀**: In the ROR model at the beginning, the simulation of $Game_0$ is the same as the real attack. So, the chance that $\mathbb{A}$ will break the proposed scheme which is given as

$$ADV_{A}^{P}(t) = |2P\,[success_0]| - 1 \tag{2}$$

***Game*₁**: This game is almost identical to $Game_0$ with the exception that $Game_1$ has an additional step which is "Execute query." In $Game_1$, $\mathbb{A}$ can only get messages $m_1, m_2, m_3, m_4, P_{id}, TS_1, m_5, m_6, m_7, P_{id}^{new}, TS_2, m_8, m_9, TS_3, m_{10}, m_{11}$, and TS_4. After finishing $Game_1$, $\mathbb{A}$ asks *test* for the session key. But $\mathbb{A}$ does not know about RN_u, RN_{GW}, and RN_{SN}. So probability of $Game_0$ is almost the same as $Game_1$ or

$$P\,[success_1] = P\,[success_0] \tag{3}$$

***Game*₂**: There are many similarities found if we compare $Game_1$ and $Game_2$, except the trials of attackers and this game may be terminated if a collision occurred in a Hash query or transcription. Moreover, the chance of a collision in a transcript and risk of a collision in a hash query is at most $\frac{(q_{SND}+q_{EX})^2}{2n}$ and $\frac{q_H^2}{2^{L+1}}$, respectively. The probability of $Game_2$ is given by

$$P\,[success_2] - P\,[success_1] \leq \frac{q_H^2}{2^{L+1}} + \frac{(q_{SND} + q_{EX})^2}{2n} \tag{4}$$

which is in accordance with the birthday paradox and,[35,38] respectively.

***Game*₃**: This game is the same as $Game_2$ except when $\mathbb{A}$ figures out the value of the verifiers without using the *hash* quires then it will be terminated. So, we can say that the chance of $Game_3$ win is

$$P\,[success_3] - P\,[success_2] \leq \frac{q_{SND}}{2^L} \tag{5}$$

***Game*₄**: This game thinks about the security of the session key in the following two circumstances. In the first scenario, it is necessary to discover the hidden K_{GW} of I_{GW}^{j} to demonstrate that forward secrecy is real. The second instance involves acquiring temporary information in order to determine whether or not it is vulnerable to attack by known session-specific temporary information (KSSTI) attacks. In the first case, $\mathbb{A}$ can hack the session key after getting $Currupt\left(I_{GW}^{j}, \lambda\right)$ to reveal I_{GW}^{j}'s secret K_{GW}. If $\lambda = 1$, $\mathbb{A}$ cannot get the temporary information RN_u, RN_{GW}, and RN_{SN} to build the session key $K_{SS} = h\left(\left(P_{id} \bigoplus ID_{sn_k}\right) \parallel RN_u \parallel RN_{GW} \parallel RN_{SN}\right)$. In the second scenario, $\mathbb{A}$ tries to steal K_{SS} by accessing any random number (RN_u, RN_{GW}, or RN_{SN}). Still, the $\mathbb{A}$ cannot figure out any two of the three numbers from the other one, for guessing out the session key. So, from the above discussion $\mathbb{A}$ can only hack by using either *Send* or *Hash* queries in both cases. Hence the success ratio of $Game_4$ is

$$P\,[success_4] - P\,[success_3] \leq \frac{q_{SND}}{2^L} + \frac{q_H^2}{2^{L+1}} \tag{6}$$

***Game*₅**: It is almost similar to $Game_4$ except that $\mathbb{A}$ tries to use *corrupt* query to launch a replay attack and password guessing attack. If you ask about $corrupt\left(I_u^i, 2\right)$, you cannot ask about $corrupt\left(I_u^i, 1\right)$ or $corrupt\left(I_u^i, 3\right)$ then q_s times sending queries are allowed for guessing the password. According to Reference [35], the chance of getting the password is at most $C'.q_{SND}^{S'}$ in send query, where C' and S' are constants and depend on the length of the password. So, the chance of $Game_5$ happening is

$$P\,[success_5] - P\,[success_4] \leq C'.q_{SND}^{S'} \tag{7}$$

Game₆: This game is different from Game₅ in that case, this game comes to a close when $\mathbb{A}$ starts $h\left(\left(P_{id} \bigoplus ID_{sn_k}\right) \| RN_u \| RN_{GW} \| RN_{SN}\right)$ queries to try to predict the K_{SS}. The main aim of $Game_6$ is to check the proposed scheme's ability to defend against impersonation assaults

$$P\left[success_6\right] - P\left[success_5\right] \le \frac{q_H^2}{2^{L+1}} \tag{8}$$

The theorem is shown to be true for game GM0 using GM6 and the proof is shown below.

$$\begin{aligned} ADV_A^P(t) &= \left|2P\left[success_0\right]\right| - 1 = 2P\left[Success_6\right] - 1 + 2(P\left[success_0\right] - \\ & P\left[success_6\right]) \le 2 \times \frac{1}{2} - 1 + 2\sum_{n=1}^{6}\left(P\left[success_n\right] - P\left[success_{n-1}\right]\right) \\ &= 2 \times \left[\frac{q_H^2}{2^{L+1}} + \frac{\left(q_{SND}+q_{EX}\right)^2}{2n} + \frac{q_{SND}}{2^L} + \frac{q_{SND}}{2^L} + \frac{q_H^2}{2^{L+1}} + C'.q_{SND}^{S'} + \frac{q_H^2}{2^{L+1}}\right] \\ &= \frac{3q_H^2}{2^L} + \frac{\left(q_{SND}+q_{EX}\right)^2}{n} + \frac{2q_{SND}}{2^{L-1}} + 2\,C'.q_{SND}^{S'} \end{aligned} \tag{9}$$

5.2 | Informal security analysis

The proposed technique is looked at informally, and the security of the proposed technique against known attacks is presented in this analysis phase. The multi-gateway platform and SNs must be able to resist these attacks, such as man-in-the-middle attacks and stolen verifier attacks. Table 1 shows how the proposed technique compares to other schemes in terms of how secure it is.

5.2.1 | Support for multiple gateways

According to the information provided above, U_i just has to register with RC once and may access a wide range of services hosted on several gateways. The U_i just has to remember one password for authentication. Therefore, the suggested framework is appropriate for a multi-gateway setup.

5.2.2 | Data integrity

To secure the identity and password before transmission in the suggested approach, the one-way hash function $h(.)$ is applied, which alters the message to make it impossible $m_4 = h\left(P_{id} \parallel ID_u \parallel ID_{sn_k} \parallel TS_1\right)$. The data is further associated with a newly generated nonce $N_u \in Z_n^*$. As a result, it ensures data integrity since it is very difficult to modify the message in our system.

5.2.3 | Defend against key compromise attacks and ensure perfect forward secrecy

The session key in our protocol is specified as $K_{SS} = h((P_{id}, ID_{SN_k}) \| \; RN_u \; \| \; RN_{GW} \; \| \; RN_{SN})$, where RN_u, RN_{GW}, and RN_{SN} are sent in an encrypted format. Theorem 1 states that the adversary $\mathbb{A}$ will not be able to recreate the session key K_{SS} even if $\mathbb{A}$ access secret K_{GW}. Hence our proposed scheme achieves forward secrecy perfectly and is also protected against key compromise attacks.

5.2.4 | Prevent against offline password/biometric key guessing attacks

Assume that throughout the authentication procedure, as well as the password updating of the proposed scheme, $\mathbb{A}$ intercepts all untrusted communications. Because there is no plaintext version of PWD_u or R_u in the above communications,

TABLE 1 Comparison of security attributes of the proposed scheme with other related existing schemes.

Security features	42	43	44	41	20	Proposed scheme
Perfect forward secrecy	√	√	√	X	√	√
Prevent offline password/biometric key guessing	X	X	X	√	X	√
Prevent attack by a trusted insider	X	√	X	√	√	√
Prevent attacks to capture a sensor node	X	X	X	X	X	√
Prevent clone card attack	X	√	√	X	X	√
Mutual authentication	√	√	√	√	√	√
Prevent against smart card stolen attack	X	√	√	√	√	√
Un-traceability	X	X	X	√	√	√
Anonymity	√	√	√	√	√	√
Prevent impersonation attack	√	√	√	X	√	√
Prevent KSSIT attack	X	√	X	X	√	√
Adaptive protection of privacy	X	X	X	X	X	√
Support session key update	X	X	X	X	X	√
Support password update	√	√	√	√	√	√
Support biometric Update	X	X	X	X	X	√
Support multi gateway	X	X	X	X	X	√
Support multifactor	X	X	X	X	X	√
Support data integrity	√	X	√	√	√	√
Support revocation	X	√	X	√	√	√
Prevent replay attack	√	√	√	X	√	√
Prevent DoS attack	X	√	√	X	X	√

so $\mathbb{A}$ is unable to excess the user's password PWD_u and biometric secret key R_u. As a result, the proposed scheme can overcome attempts to guess an offline password or biometric key.

5.2.5 | Prevent attacks by a trusted insider

Let us say that adversary $\mathbb{A}$ gets U_i's registration information (ID_u, P_{id}, and N_u) from a particular gateway GW_j. Here, $\mathbb{A}$ cannot get the shared secret key SK_{GW-U} without finding a secret key K_{GW_j} (only GW_j know). Moreover, GW_j does not access any relevant information about R_u from Bio_u, and $I_1' = h\left(PWD_u' \parallel R_u\right)$, which is hidden by R_u. Due to biometric key R_u and irreversible hash function $h(.)$, it is computationally impossible for an insider $\mathbb{A}$ to correctly guess a user's password using a power analysis attack.[39]

5.2.6 | Prevent attacks to capture a sensor node

In the proposed scheme, GW_j stores ID_{SN_k}, N_{SN_k} in its database while SN_k stores information (ID_{SN_k}, SK_{GW-SN}) in his memory. It is clear that N_{GW_j} and $SK_{GW-SN} = ID_{SN_k} \bigoplus N_{SN_k}$ are sent over the secure channel and after each login N_{SN_k} is updated automatically. There are two ways of SN capture attacks that are looked at.[40] First, compromised 1 SN's pre-shared secret key $SK_{GW-SN} = ID_{SN_k} \bigoplus N_{SN_k}$ is not two linked to two other uncompromised SNs because GW_j chose a unique ID_{SN_k}. So, even if shared secretkey SK_{GW-SN} is reveled, this does not affect the transmission keys of other SN that have not been hacked. Second, if $\mathbb{A}$ gets ID_{SN_k} and SK_{GW-SN} from a compromised SN_k, it is hard for an $\mathbb{A}$ to access the previous shared secret key $SK_{GW-SN}^{pre} = SK_{GW-SN} \bigoplus RN_{GW}^{pre}$ of SN_k, since RN_{GW}^{pre} is sent in cipher text. Because of this, it is nearly impossible for an $\mathbb{A}$ to rebuild previous K_{SS} of a compromised SN.

5.2.7 | Prevent replay attacks

Let us say that an $\mathbb{A}$ reads the messages on the public channel during a certain session and then plays back the same messages over time. Because all of these messages contain timestamps, authentication will fail when these timestamps are checked for freshness. So, a replay attack will not work on our proposed protocol.

5.2.8 | Prevent clone card attack

In the proposed scheme, users enter their ID_u, PWD_u, and Bio_u and check the integrity $I_2' = h\left(ID_u' \parallel I_1'\right) = h\left(ID_u' \parallel h\left(PWD_u' \parallel R_u\right)\right)$. If integrity is compromised then the user changes their password and user authenticates the SNs again. If a valid user wants to change his/her biometric data during the password and biometric updating phase, user must confirm his identity to GW_j in a secure manner. So, it becomes very hard for unauthorized users to use the same valid ID_u and different passwords/biometrics for login into the system. Thus, our scheme is safe from a clone card attack.

5.2.9 | Prevent denial-of-service (DoS) attacks

The $\mathbb{A}$ attempts to interrupt the regular session by replaying messages. However, in preventing replay attacks, this does not work because the freshness of the timestamp has to be verified at each authentication step. Let us imagine that after one of the last three phases of the login and authentication procedure, $\mathbb{A}$ attempts to edit the messages. There may be a chance that the parameters of the participant (transmission key, anonymity, etc.) may not be synchronized and tamper with the normal authentication process. So, for protecting this we apply massage integrity at each step of the authentication. Now further SN_k checks m_7 and updates his shared secret key SK_{GW-SN}^{new}. If it is not true, the SN will discard the session. After that GW_j checks m_9 and updates as P_{id}^{new} and $N_{SN_k}^{new}$. If m_9 is not true, GW_j reject the authentication request and terminate the current session. After that U_i checks m_{11} and updates the pseudo id P_{id}^{new} during login and authentication. If m_{11} is not correct then the current session will be terminated. As shown in Figure 5, either U_i, GW_j, or SN_k can roll back the transaction and reach the last value. So that procedure can continue as usual.

In particular, if $\mathbb{A}$ changes the messages (m_8, m_9, TS_3) from SN_k then verification of m_9 will fail and the request during the authentication process will be discarded while the updated shared secret key K_{GW-SN}^{new} of SN_k will be rolled back to its previous value. Similarly, if $\mathbb{A}$ changes the messages $(m_8, m_{10}, m_{11}, TS_4)$ from GW_j then verification of m_{11} will fail and the current session will be rejected, and the updated value of GW_j and SN_k will be reset to their previous values. Finally SC, the database of GW_j, and memory of SN_k are in sync with each other. So, the proposed protocol cannot be taken down by a DoS attack.

5.2.10 | Mutual authentication

In the proposed protocol, the following three steps are used for mutual authentication:

- For user authentication, the gateway GW_j verify m_4 and if we authenticate the user U_i and for SN authentication U_i is authenticated by GW_j verify m_9 to authenticate SN_k.
- For gateway GW_j authentication SN_k verify m_7 to authenticate GW_j directly and also U_i authenticated by SN_k indirectly.
- User U_i Verify m_{11} to authenticate GW_j directly and also authenticate SN_k indirectly.

5.2.11 | Protect against stolen smart card attack

Assume that $\mathbb{A}$ gets SC then $\mathbb{A}$ can access $(SCN_i, I_2, I_3, hd_u, Gen(.), Rep(.))$ From SC's memory using the power analysis attacks,[41] where $I_2 = h(ID_u \parallel I_1)$, $I_3 = SK_{GW-U} \bigoplus I_2, I_1 = h(ID_u \parallel PWD_u \parallel R_u \parallel N_u)$ and $SK_{GW-U} = h(K_{GW_j} \parallel ID_u \parallel N_u)$. Theorem 1 says that it is nearly impossible for $\mathbb{A}$ to get PWD_u from SC. It is very hard for $\mathbb{A}$ to pretend to be the U_i. As we have already talked about, our protocol can prevent someone from stealing an SC.

5.2.12 | Untraceability and anonymity

In an unsecured environment, there is no plaintext identity data that can be used to figure out who someone is during communication. Furthermore, ID_u and ID_{SN_k} are encrypted during the authentication process as $m_1 = ID_u \bigoplus h(P_{id} \parallel SK_{GW-U})$ and $m_3 = ID_{sn_k} \bigoplus h(SK_{GW-U} \parallel RN_u \parallel P_{id} \parallel TS_1)$, respectively. So, the proposed scheme makes sure that each participant stays anonymous. Also, an adversary cannot track $ID_{sn_k} = m_3 \bigoplus h(SK_{GW-U} \parallel RN_u \parallel P_{id} \parallel TS_1)$ because TS_1 is generated immediately, and an adversary cannot track $ID_u = m_1 \bigoplus h(P_{id} \parallel SK_{GW-U}) = P_{id} \bigoplus SCN_i \bigoplus PWD_u$, because P_{id} is changed every time. These encrypted identities in our scheme are made up of special random numbers that are generated by the participants. This makes it impossible to track any of the parties. So, our plan makes sure that everyone stays anonymous and cannot be tracked.

5.2.13 | Prevent impersonation attack

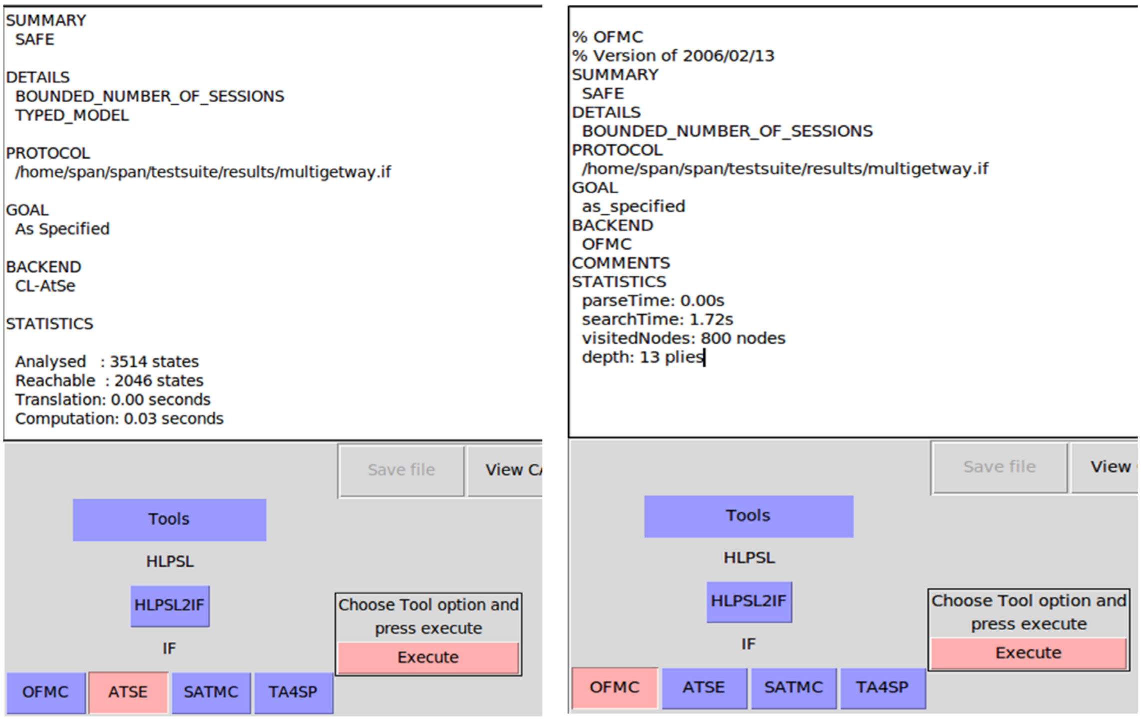
Let us say that $\mathbb{A}$ tries to pretend to be the person who sent the message by intercepting it. There are mainly three possibilities that happen in this attack. First, if $\mathbb{A}$ wants to pretend to be U_i, he or she needs to intercept the messages $m_1, m_2, m_3, m_4, P_{id}$, and TS_1 and figure out that $ID_u = m_1 \bigoplus h(P_{id} \parallel SK_{GW-U})$, $m_2^A = RN_u^A \bigoplus h\left(SK_{GW-U} \parallel ID_u \parallel TS_1^A\right)$, $m_3^A = ID_{sn_k}^A \bigoplus h\left(SK_{GW-U} \parallel RN_u^A \parallel P_{id} \parallel TS_1^A\right)$ and $m_4^A = h\left(P_{id} \parallel ID_u \parallel ID_{sn_k}^A \parallel RN_u^A \parallel TS_1^A\right)$. But it is very hard for the $\mathbb{A}$ to do this without knowing that $SK_{GW-U} = h\left(K_{GW_j} \parallel ID_u \parallel N_u\right) = I_3 \bigoplus h(I_1 \parallel ID_u)$. Second, if $\mathbb{A}$ wants to pretend to be SN_k, he or she must get the transmission key $SK_{GW-SN} = ID_{sn_k} \bigoplus N_{SN_k}$ for the same reason. As discussed about non-traceability and anonymity proposed protocol to make sure that SN_k is anonymous and makes it very difficult for an $\mathbb{A}$ to get ID_{sn_k}. So, pretending to be SN_k is also a hard task for $\mathbb{A}$. Third, because it is hard for adversary $\mathbb{A}$ to figure out how to get transmission keys SK_{GW-U} and SK_{GW-SN}, to be pretended as an authorized entity to GW.

5.2.14 | Prevent known session-specific temporary information (KSSTI) attacks

When an $\mathbb{A}$ can determine the session key using temporary information, such as a random number provided by the protocol participants, a KSSTI attack occurs. The formula for our session key is $K_{SS} = h\left(P_{id} \bigoplus ID_{sn_k}\right) \parallel RN_u \parallel RN_{GW} \parallel RN_{SN}$. Theorem 1 states that the session key cannot be compromised even if the $\mathbb{A}$ receives a nonce, proving that our suggested scheme is resistant to KSSTI attack.

5.2.15 | Adaptive protection of privacy

The proposed method provides adaptive privacy protection for users that goes beyond anonymity and is hard to track. The "login and registration process" and the "password and biometric updating process" are the main parts of the implementation. During each "login and authentication phase," U_i changes his or her pseudo id P_{id}. $\mathbb{A}$ valid user chooses P_{id}^{new} for himself or herself. In the "password and biometric updating phase," pseudo id P_{id} is also updated forwardly. After each time a password or biometric is changed, I_1 is changed to I_1^{new}. The pseudonym $P_{id} = SCN_i \bigoplus ID_u \bigoplus I_1$ in the gateway database and the corresponding $I_2 = h(ID_u \parallel I_1)$ and $I_3 = SK_{GW-U} \bigoplus I_2$ in the user's SC needs to be changed to $P_{id}^{new} = SCN_i \bigoplus ID_u \bigoplus I_1^{new}$, $I_2^{new} = h\left(ID_u \parallel I_1^{new}\right)$, $I_3^{new} = SK_{GW-U} \bigoplus I_2^{new}$. In most of the previous protocols, updating a pseudo id was done passively. However, the proposed protocol allows valid users to take the initiative to update their pseudo id, either by choosing a new pseudo id during each authentication or by updating their passwords and



```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

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

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 3514 states
 Reachable  : 2046 states
 Translation: 0.00 seconds
 Computation: 0.03 seconds
```

(A)

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/multigetway.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 1.72s
 visitedNodes: 800 nodes
 depth: 13 plies
```

(B)

FIGURE 9 Analysis of the simulation results using (A) CL-AtSe and (B) OFMC.

biometrics. So, in addition to being hard to track and anonymity, the proposed protocol also supports adaptive privacy preservation.

5.3 | AVISPA simulation for formal verification

To verify that the proposed protocol is safe, we simulated the proposed scheme under the AVISPA simulation tool with the integrated modules of the Constraint-Logic-based Attack Searcher (CL-AtSe) and the On-the-Fly Model Checker (OFMC). The protocol's defense against SN capture attacks is evaluated by CL-AtSe. CL-AtSe used to determine for an untrusted user might carry SNs capture attack out or not. Additionally, the OFMC confirms the proposed protocol's security against replay attacks. Figure 9 illustrates how the proposed protocol is protected against SN capture attacks and replay attacks. According to the OFMC verification, the search time for visiting 800 nodes was 1.72 s, while the CL-AtSe verification took 0.03 s to compute 3514 states.

6 | PERFORMANCE ANALYSIS

The performance of the proposed and existing schemes is mainly based on parameters viz security analysis, computational cost, and communicational cost. In security features, methods must be free from well-known attacks like replay attacks, impersonation attacks, etc. communication cost is the total number of messages (in bits) are required to communicate among participants in the network whereas computational cost is the time cost and total time required to communicate among participants in IoT enabled WSNs.

6.1 | Comparative security analysis

There are some different security features of the proposed method and other relevant existing schemes shown in Table 1. Here we can observe that the proposed scheme is more secure than other relevant existing schemes.[20,41,42] In Table 1 all relevant schemes[20,41,42] do not prevent attacks to capture node except our scheme. Also, the proposed scheme incorporates the session key updates as well as multi-factor while other relevant schemes do not support these features. Hence, we can say that our proposed scheme provides better security than other relevant existing schemes.[20,41–44]

6.2 | Comparative computational cost

The proposed scheme shows less computational costs in comparison with many relevant existing schemes as shown in Table 2. Here, T_h represent the time taken to execute hash function $h(.)$, T_{ED} indicates the required time for symmetric encryption and decryption and T_{Bio} used to represent the required time for biometric key generation using fuzzy logic which is almost similar to T_m.[41] To calculate the computational costs we referred[41] for initial parametric values like, $T_h = 0.00076$ ms, $T_{ED} = 0.00296$ ms, and $T_{Bio} \approx T_m = 0.27864$ ms. After analysis of Table 2, it is evident that the proposed approach is better than[41,43,44] with respect to the computational cost of the user,[20,43,44] the computational cost of the gateway,[20,43,44] and computational overhead of SNs. Finally, the proposed approach is better than all relevant existing schemes with respect to the total cost.

6.3 | Comparative communicational cost

We provide the communication overhead between the proposed scheme and other relevant existing schemes[20,41–44] as shown in Table 3. The number of messages sent and the amount of bandwidth used during the login and authentication stages determine the communication cost. A point on the ECC requires 320 bits, while the symmetric encryption/decryption requires 128 bits. It is well knowledge that the output of the hash function $h(.)$ has 160 bits. The timestamp is 32 bits, the random value and response value are both 160 bits, and the identity is 160 bits.

Figure 10 displays the communication overhead of our proposed scheme when compared to other relevant existing schemes, namely.[20,41,42] The communication cost during the login and authentication stages is determined by the number of messages exchanged and the amount of bandwidth used. The proposed scheme employs elliptic curve cryptography (ECC), which requires 320 bits to represent a point on the curve. Symmetric encryption and decryption, on the other hand, require 128 bits. The hash function $h(.)$ produces an output of 160 bits, a well-established fact. The various components of our proposed scheme also have specific bit requirements. The timestamp is represented using 32 bits, while both the random value and response value require 160 bits. The identity component of the scheme is also represented using 160 bits. In summary, the communication overhead of our proposed scheme is compared to existing schemes in Figure 10. The communication cost is determined by the number of messages and the bandwidth used. Our scheme uses ECC, which requires 320 bits to represent a point on the curve. Symmetric encryption/decryption requires 128 bits, while the hash function $h(.)$ produces an output of 160 bits. Other components of our scheme are represented using specific bit requirements.

After analysis of Table 3, we reach some conclusions such as

- The communication cost of[20,41–44] schemes is 2720 bits, 1156 bits, 2080 bits, 2528 bits and 2496 bits, respectively, while the proposed scheme's cost required is 2368 bits.

TABLE 2 Comparison of computation cost of the proposed scheme with other related existing schemes.

Schemes	User	Gateway	Sensor nodes	Total cost	Overall time (ms)
42	$11T_h + T_{Bio}$	$17T_h$	$5T_h$	$33T_h + T_{Bio}$	0.303925
43	$10T_h + T_{ED} + T_{Bio}$	$3T_h + 2T_{ED}$	$5T_h + T_{ED}$	$18T_h + 4T_{ED} + T_{Bio}$	0.304287
44	$8T_h + 3T_{ED} + T_{Bio}$	$8T_h + T_{ED}$	$4T_h + 2T_{ED}$	$20T_h + 6T_{ED} + T_{Bio}$	1.965849
41	$13T_h + T_{Bio}$	$15T_h$	$4T_h$	$32T_h + T_{Bio}$	0.303159
20	$10T_h + T_{Bio}$	$10T_h + T_{ED}$	$4T_h + T_{ED}$	$24T_h + 2T_{ED} + T_{Bio}$	0.302957
Proposed scheme	$11T_h + T_{Bio}$	$13T_h$	$6T_h$	$30T_h + T_{Bio}$	0.301627

TABLE 3 Comparison of communication cost of proposed scheme with other related existing schemes.

Schemes	No. of messages	Login phase (bits)	Authentication phase (bits)	Total cost (bits)	Communication mode among participants
42	6	1376	1344	2720	$U_i \to GW_j, GW_j \to U_i, U_i \to GW_j, GW_j \to SN_k, SN_k \to GW_j, GW_j \to U_i$
43	3	128	928	1156	$U_i \to GW_j, GW_j \to SN_k, SN_k \to U_i$
44	4	640	1440	2080	$U_i \to GW_j, GW_j \to SN_k, SN_k \to GW_j, GW_j \to U_i$
41	4	992	1536	2528	$U_i \to GW_j, GW_j \to SN_k, SN_k \to GW_j, GW_j \to U_i$
20	4	1312	1184	2496	$U_i \to GW_j, GW_j \to SN_k, SN_k \to GW_j, GW_j \to U_i$
Proposed scheme	4	832	1536	2368	$U_i \to GW_j, GW_j \to SN_k, SN_k \to GW_j, GW_j \to U_i$

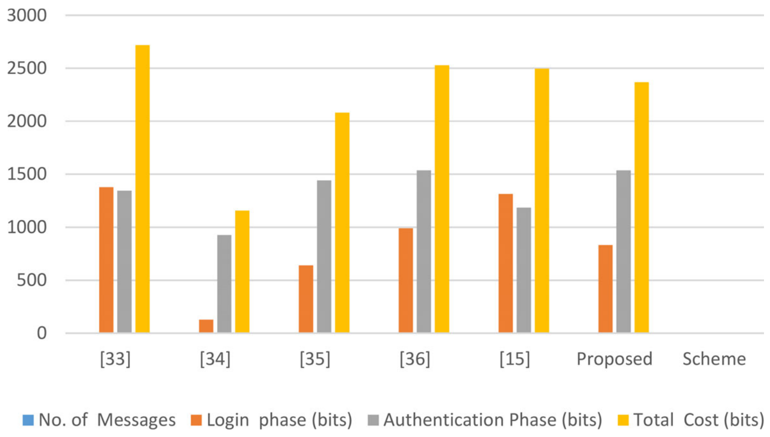


FIGURE 10 Communication cost of proposed scheme with other related existing schemes.

- [43,44] schemes have less communication cost than the proposed scheme but these schemes[43,44] do not prevent various attacks as discussed in Section 6.

As a result, the proposed scheme is far more effective than any of the other related schemes that are already in existence.

7 | CONCLUSIONS AND FUTURE SCOPE

This study proposes a multifactor user authentication scheme to enable robust and secure communication in multi-gateway based IoT enabled WSNs. Password and biometric information are updated securely to ensure both forward and backward secrecy during communication. The session key is also dynamically updated using adaptive privacy preservation techniques to maintain its confidentiality among the user, base station, and sensor nodes. Additionally, the proposed authentication scheme provides robustness against gateway impersonation attacks with lower computation and communication costs compared to existing schemes. Furthermore, formal and informal security analyses were conducted to verify the proposed multifactor and multi-gateway based authentication scheme's security. Moreover, the proposed scheme can be utilized with certain improved capabilities while minimizing security concerns and reducing computational and communication costs. In future work, it is recommended to focus on developing better security and data communication in user authentication and node authentication schemes. It is also suggested to consider less complex authentication schemes while maintaining high-performance and security.

CONFLICT OF INTEREST STATEMENT

All the authors have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Samayveer Singh https://orcid.org/0000-0002-4199-721X
Sukhpal Singh Gill https://orcid.org/0000-0002-3913-0369

REFERENCES

1. Hayajneh R, Doomun GA-M, Mohd BJ. An energyefficient and security aware route selection protocol for wireless sensor networks. *Secur Commun Netw*. 2014;7(11):2015-2038.
2. Gill SS, Xu M, Ottaviani C, et al. AI for next generation computing: emerging trends and future directions. *Internet Things*. 2022;19:100514.
3. Siddiqui F, Beley J, Zeadally S, Brought G. Secure and lightweight communication in heterogeneous IoT environments. *Internet Things*. 2021;14:100093.
4. Lee W-K, Schubert MJW, Ooi B-Y, Ho SJ-Q. Multi-source energy harvesting and storage for floating wireless sensor network nodes with long range communication capability. *IEEE Trans Ind Appl*. 2018;54(3):2606-2615.
5. Rajan AS, Gobriel S, Maciocco C, et al. Understanding the bottlenecks in virtualizing cellular core network functions. In: The 21st IEEE International Workshop on Local and Metropolitan Area Networks, pp. 1–6. IEEE; 2015.
6. Buzzi S, Chih-Lin I, Klein TE, Vincent Poor H, Yang C, Zappone A. A survey of energy-efficient techniques for 5G networks and challenges ahead. *IEEE J Sel Areas Commun*. 2016;34(4):697-709.
7. Dolev D, Yao A. On the security of public key protocols. *IEEE Trans Inf Theory*. 1983;29(2):198-208.
8. IjazAhmad ML, Shahabuddin S, Ylianttila M, Gurtov A. Design principles for 5g security. A Comprehensive Guide to 5G Security, p. 75. 2018.
9. Gupta A, Tripathi M, Muhuri S, Singal G, Kumar N. A secure and lightweight anonymous mutual authentication scheme for wearable devices in medical internet of things. *J Inf Secur Appl*. 2022;68:103259.
10. Panda S, Mondal S, Kumar N. SLAP: a secure and lightweight authentication protocol for machine-to-machine communication in industry 4.0. *Comput Electr Eng*. 2022;98:107669.
11. Chaudhry SA, Irshad A, Yahya K, Kumar N, Alazab M, Zikria YB. Rotating behind privacy: an improved lightweight authentication scheme for cloud-based IoT environment. *ACM Trans Internet Technol (TOIT)*. 2021;21(3):1-19.
12. Das ML. Two-factor user authentication in wireless sensor networks. *IEEE Trans Wirel Commun*. 2009;8(3):1086-1090.
13. Chang C, H. L.-I. T. on Wireless, and U. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. ieeexplore.ieee.org 2015 2015.
14. Singh D, Kumar B, Singh S, Chand S. SMAC-AS: MAC based secure authentication scheme for wireless sensor network. *Wirel Pers Commun*. 2019;107(2):1289-1308.
15. Banerjee S, Chunka C, Sen S, Goswami RS. An enhanced and secure biometric based user authentication scheme in wireless sensor networks using smart cards. *Wirel Pers Commun*. 2019;107(1):243-270.
16. Xie Q, Li K, Tan X, Han L, Tang W, Hu B. A secure and privacy-preserving authentication protocol for wireless sensor networks in smart city. *Eurasip J Wirel Commun Netw*. 2021;1:2021.
17. Singh D, Kumar B, Singh S, Chand S. Evaluating authentication schemes for real-time data in wireless sensor network. *Wirel Pers Commun*. 2020;114(1):629-655.
18. Singh D, Kumar B, Singh S, Chand S. A secure IoT-based mutual authentication for healthcare applications in wireless sensor networks using ECC. *Int J Healthc Inf Syst Inform*. 2021;16(2):21-48.
19. Khalid H, Hashim SJ, Ahmad SMS, Hashim F, Chaudhary MA. Cross-sn: a lightweight authentication scheme for a multi-server platform using IoT-based wireless medical sensor network. *Electron*. 2021;10(7):790.
20. Singh D, Kumar B, Singh S, Chand S, Singh PK. RCBE-AS: Rabin cryptosystem–based efficient authentication scheme for wireless sensor networks. *Pers Ubiquitous Comput*. 2021. https://doi.org/10.1007/s00779-021-01592-7.
21. Shuai M, Yu N, Wang H, Xiong L, Li Y. A lightweight three-factor anonymous authentication scheme with privacy protection for personalized healthcare applications. 33(3):1-18.
22. Dahia G, Jesus L, Pamplona Segundo M. Continuous authentication using biometrics: an advanced review. *Wiley Interdiscip Rev Data Min Knowl Discov*. 2020;10(4):e1365.
23. Mansour MM, Salem FM, Saad EM. A secure mutual authentication scheme with perfect forward-secrecy for wireless sensor networks. *Adv Intell Syst Comput*. 2019;845:446-456.
24. Park YY, Choi Y, Lee K. A study on the design and implementation of facial recognition application system. *Int J Bio-Sci Bio-Technol*. 2014;6(2):1-10.
25. Tai WL, Chang YF, Li WH. An IoT notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks. *J Inf Secur Appl*. 2017;34:133-141.

26. Sirisha Uppuluri GL. Secure user authentication and key agreement scheme for IoT device access control based smart home communications. *Wirel Netw*. 2023;29(3):1333-1354.
27. Mahmood K, Ferzund J, Saleem MA, Shamshad S, Das AK, Park Y. A provably secure mobile user authentication scheme for big data collection in IoT-enabled maritime intelligent transportation system. *IEEE Trans Intell Transp Syst*. 2023;24(2):2411-2421.
28. Mishra KC, Dutta S. A simple and secure user authentication scheme using map street view with usability analysis based on ISO/IEC 25022. *Int J Inf Sec*. 2023;22(2):403-415.
29. Dodis Y, Katz J, Reyzin L, Smith A. Robust fuzzy extractors and authenticated key agreement from close secrets. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4117 LNCS, pp. 232–250, 2006.
30. Dodis Y, Reyzin L, Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics), vol. 3027, pp. 523–540, 2004.
31. Canetti R, Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2045, pp. 453–474, 2001.
32. Canetti R, Krawczyk H. Universally composable notions of key exchange and secure channels. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics), vol. 2332, pp. 337–351, 2002.
33. Wang D, Cheng H, Wang P, Huang X, Jian G. Zipf's law in passwords. *IEEE Trans Inf Forensics Secur*. 2017;12(11):2776-2791.
34. Canetti R, Goldreich O, Halevi S. The random oracle methodology, revisited. *J ACM*. 2004;51(4):557-594.
35. Qiu S, Wang D, Xu G, Kumari S. Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for Mobile lightweight devices. *IEEE Trans Dependable Secur Comput*. 2022;19(2):1338-1351.
36. Sutrala AK, Obaidat MS, Saha S, Das AK, Alazab M, Park Y. Authenticated key agreement scheme with user anonymity and Untraceability for 5G-enabled Softwarized industrial cyber-physical systems. *IEEE Trans Intell Transp Syst*. 2022;23(3):2316-2330.
37. Hussain S, Chaudhry SA, Alomari OA, Alsharif MH, Khan MK, Kumar N. Amassing the security: an ECC-based authentication scheme for internet of drones. *IEEE Syst J*. 2021;15:4431-4438.
38. Abbasinezhad-Mood D, Mazinani SM, Nikooghadam M, Ostad-Sharif A. Efficient provably-secure dynamic ID-based authenticated key agreement scheme with enhanced security provision. *IEEE Trans Dependable Secur Comput*. 2022;19(2):1227-1238.
39. Martinasek Z, Dzurenda P, Malina L. Profiling power analysis attack based on MLP in DPA contest V4.2. Paper presented at: 2016 39th Int Conf Telecommun Signal Process, pp. 223–226; 2016.
40. Far HAN, Bayat M, Das AK, Fotouhi M, Pournaghi SM, Doostari MA. LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wirel Netw*. 2021;27(2):1389-1412.
41. Wu F, Li X, Xu L, Vijayakumar P, Kumar N. A novel three-factor authentication protocol for wireless sensor networks with IoT notion. *IEEE Syst J*. 2021;15(1):1120-1129.
42. Ostad-Sharif A, Arshad H, Nikooghadam M, Abbasinezhad-Mood D. Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme. *Future Gener Comput Syst*. 2019;100:882-892.
43. Chen Y, Ge Y, Wang Y, Zeng Z. An improved three-factor user authentication and key agreement scheme for wireless medical sensor networks. *IEEE Access*. 2019;7:85440-85451.
44. Li X, Peng J, Obaidat MS, Wu F, Khan MK, Chen C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. *IEEE Syst J*. 2020;14(1):39-50.

How to cite this article: Kumar R, Singh S, Singh D, Kumar M, Gill SS. A robust and secure user authentication scheme based on multifactor and multi-gateway in IoT enabled sensor networks. *Security and Privacy*. 2023;e335. doi: 10.1002/spy2.335