Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.018749

ARTICLE

A Lightweight and Robust User Authentication Protocol with User Anonymity for IoT-Based Healthcare

Chien-Ming Chen[1,*], Shuangshuang Liu[1], Shehzad Ashraf Chaudhry[2], Yeh-Cheng Chen[3] and Muhammad Asghar khan[4]

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China

[2]Department of Computer Engineering, Istanbul Gelisim University, Istanbul, 34310, Turkey

[3]Department of Computer Science, University of California, Davis, CA, 95616, USA

[4]Department of Electrical Engineering, Hamdard University, Islamabad, 44000, Pakistan

[*]Corresponding Author: Chien-Ming Chen. Email: chienmingchen@ieee.org

Received: 14 August 2021 Accepted: 09 October 2021

ABSTRACT

With the rise of the Internet of Things (IoT), the word "intelligent medical care" has increasingly become a major vision. Intelligent medicine adopts the most advanced IoT technology to realize the interaction between patients and people, medical institutions, and medical equipment. However, with the openness of network transmission, the security and privacy of information transmission have become a major problem. Recently, Masud et al. proposed a lightweight anonymous user authentication protocol for IoT medical treatment, claiming that their method can resist various attacks. However, through analysis of the protocol, we observed that their protocol cannot effectively resist privileged internal attacks, sensor node capture attacks, and stolen authentication attacks, and their protocol does not have perfect forward security. Therefore, we propose a new protocol to resolve the security vulnerabilities in Masud's protocol and remove some redundant parameters, so as to make the protocol more compact and secure. In addition, we evaluate the security and performance of the new protocol and prove that the overall performance of the new protocol is better than that of other related protocols.

KEYWORDS

IoT; intelligent medical; user authentication

1 Introduction

In the traditional Internet, most of the information exchange and communication took place between computers, where computers operations were manual operations; the traditional Internet realizes the information exchange and communication between people in a certain sense. Now, however, we have forayed into the era of the Internet of Things (IoT) [1,2]. The applicability of the new system goes beyond realizing the mutual exchange of information and communication between people, between people and objects, and between objects. The IoT has a wide range of uses, including intelligent transportation, intelligent fire protection, intelligent home, intelligent power grid, intelligent medical, and other aspects. In short, it facilitates the use of the latest IT technology in all walks of life. Specifically, IoT technology embeds sensors into the power grid, buildings, and other objects [3–6]. The construction industry is using IoT technology ubiquitously. Architecture is the foundation of a city; the progress of technology promotes the intelligent development of architecture, and intelligent architecture is rapidly gaining people's attention. The current smart building methods incorporate power lighting and fire monitoring. Sensors are installed on equipment for sensing, transmission, and remote monitoring, which not only saves considerable time but also energy. Among the many applications of the IoT, smart medicine is one of the most promising applications for the future.

The emergence of IoT technology promotes the further development of medical information technology. IoT technology has great potential in the field of medicine and health [7–9]. It can better realize diagnoses and facilitate intelligent management of things. Furthermore, it realizes digital processing and sharing of resource information, equipment information, drug information, and personnel information. The use of intelligent medicine is prominent in two fields: digital hospitals and medical wearables. The digital hospital includes a hospital information system, medical image storage system, transmission system, and doctor workstation. Their function is to realize the collection, storage, processing, and transmission of patient information. Digital hospitals enable zero-distance contact with patients. Doctors can conduct long-distance consultation, intelligent medical support resource sharing, and cross-regional optimal allocation. In addition, digital medicine can also monitor the vital signs of patients by deploying sensor nodes, which will automatically send an alarm in case of emergency, which reduces the nursing cost of seriously ill patients. The digital hospital also includes a clinical decision-making system, implying that doctors can analyze patients' symptoms while helping formulate the best and effective treatment plan. In addition, digital medicine provides a remote visitation system. When visitors visit patients, they directly do so through the remote visitation system, which can effectively avoid the direct contact between patients and visitors, eliminate the spread of disease, and shorten the recovery process of patients.

Medical wearable technology [10–12] is the deployment of sensor nodes around the patient, through the sensor nodes collecting information and parameters of the user's patient and the surrounding environment, sending it through the network to the cloud, and then processing to the user. The digital hospital is an improvement over the traditional hospital; it realizes the digital equipment's access to electronic medical records and the management of equipment. However, with the introduction of the medical system of the IoT, introducing sensor nodes around the patients to collect information and then transmitting it to remote medical staff is made possible, ensuring the safety of the medical staff. However, the introduction of IoT is bound to involve the transmission of information on the network channel. Due to the universality and openness of the transmission channel, privacy and security of transmitted information have become the main concern of the IoT medical systems.

Fig. 1 shows the architecture of communication between three entities in the IoT-based healthcare environment: the doctor (user), IoT devices, and a gateway. All the IoT devices around the patients collect real-time patients' information and then transmit them to a gateway. An authenticated doctor can access the gateway to obtain effective information from those IoT devices. This means that a gateway can authenticate the identity of doctors. In other words, a gateway is a medium for doctors to communicate with sensors.

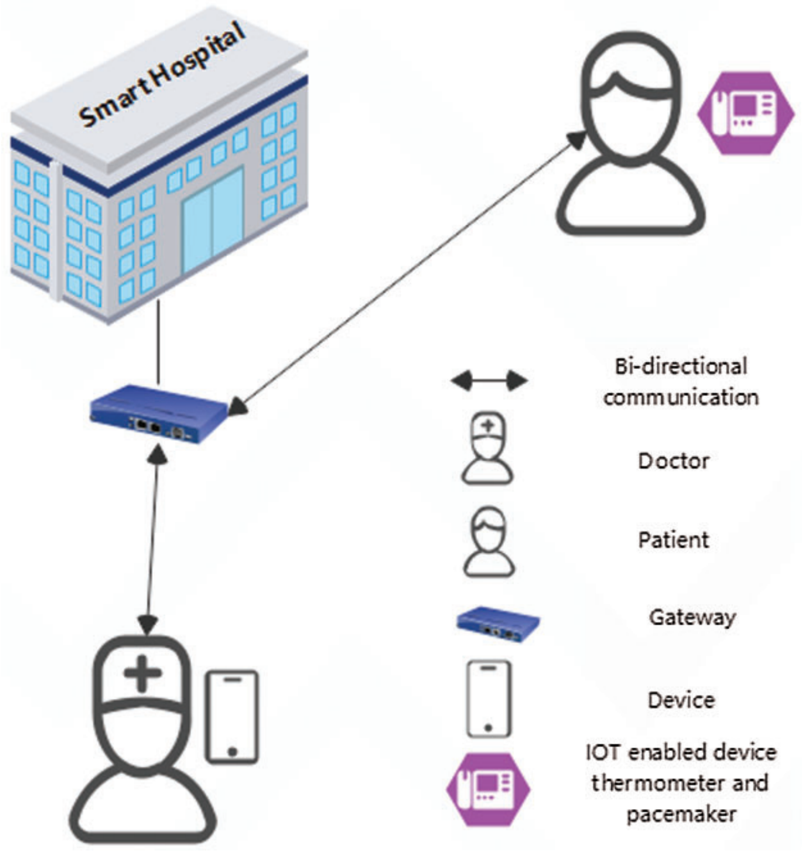

Smart Hospital
Bi-directional communication
Doctor
Patient
Gateway
Device
IOT enabled device thermometer and pacemaker

Figure 1: System model

In 2012, Chen et al. [13] proposed an efficient and secure dynamic identity authentication protocol for telemedicine information systems, which dynamically authenticates the user's identity to achieve user anonymity. However, Cao et al. [14] found that the protocol can track users through offline identity guessing attacks. When the user loses possession of a smart card, there is no guarantee of security as Chen's protocol is also vulnerable to offline password guessing attacks. Therefore, Cao et al. [14] proposed an improved password authentication protocol based on the smart card. In 2015, He et al. [15] proposed a two-factor authentication scheme for wireless medical sensors, which allows medical personnel to access patient information using wireless sensor medical devices. In 2016, Li et al. [16] proposed a network-based electronic medical authentication scheme, which also uses the user's password and smart card for two-factor authentication. He et al. [17] proposed an authentication protocol that is more suitable for the configuration of telemedicine information systems with low power consumption mobile devices. Wei et al. [18] found that this protocol cannot effectively resist password attacks; they proposed an improved authentication protocol for telemedicine information systems and proved that the protocol meets the security requirements of two-factor authentication. In 2018, Wu et al. [19] proposed a lightweight two-factor medical authentication scheme, and they claimed that their protocol had perfect security; however, after analysis, it was found that their protocol could not effectively resist perfect forward security. Therefore, based on the two-factor authentication protocol, Wazid et al. [20] proposed a three-factor network authentication key scheme, which introduced biological information based on the previous authentication password and smart card. The map area of biological information is mainly completed by a biological extractor. In 2019, Sharma et al. [21] proposed a lightweight user authentication protocol, but Canetti et al. [22] found that their protocol could not effectively resist privilege insider attacks. Recently, Masud et al. [23] proposed a protocol for the security of the IoT medical system. The paper mentioned that their protocol is a lightweight anonymous user authentication protocol. The protocol only uses hash primitives to encrypt the information, which reduces the burden of the processor while resisting replaying attacks, man-in-the-middle attacks, anonymity, and untraceability. However, we find that the protocol mentioned in this paper cannot effectively resist internal privilege attacks, sensor node capture attacks, or stolen verification attacks, and it cannot provide perfect forward security.

In this paper, we first demonstrate that Masud et al.'s protocol [23] is insecure against various kinds of attacks. We then propose a lightweight and robust user authentication protocol for IoT-based healthcare with user anonymity. In our design, we only use a single hash function and successive XOR operations; thus, the proposed protocol retains better performance. Additionally, the proposed protocol has perfect forward security and can effectively resist internal privilege, stolen verification, and sensor node capture attacks. In addition, we delete some redundant parameters in Masud et al.'s protocol [23] to make the entire protocol more concise. Furthermore, we compare the proposed protocol with other related protocols in terms of communication and computation cost. The results show that our design has better performance. Also, we use the real-or-random (ROR) model [24] to further prove that the proposed protocol is indeed secure.

The remainder of this paper is organized as follows. In Section 2, we briefly review Masud et al.'s protocol [23] Section 3 demonstrates that Masud et al.'s protocol [23] is vulnerable to privilege internal attacks, stolen verification attacks, and sensor node capture attacks. The proposed protocol is described in Section 4. Section 5 and Section 6 provide security and performance analyses and comparisons. Finally, Section 7 concludes the paper.

2 Review of Masud et al.'s Protocol

In this section, we briefly describe the protocol [23], which consists of three phases: user registration phase, sensor node registration phase, and login and mutual authentication phase. In the first two phases, user and sensor registration is conducted through the gateway.

2.1 User Registration Phase

(1) The user first selects an D_{ID} and password PW_D, and then generates a registration request R_{req}. Then, the user transmits the D_{ID}, PW_D, and R_{req} to the gateway through the secure channel. After the gateway receives the registration request from the user, it generates a random gateway private key R_{SG}^1, calculates

$$a = D_{ID} \oplus R_{SG}^1 \oplus PW_D \tag{1}$$

$$tD_{TID} = R_{SG}^1 \oplus D_{ID} \tag{2}$$

and stores the parameter a, R_{SG}^1 and D_{ID} in memory. Finally, the gateway returns the calculated parameter a to the user through the secure channel.

(2) After receiving the parameter from the gateway, the user first calculates the value of the random gateway key R_{SG}^{1} according to the parameter a, and then calculates the value of the pseudo-identity D_{ID} according to the random private key of the gateway. Secondly, the user encapsulates their password, pseudo-identity, and gateway random private key in parameter B. Finally, the values of user parameters R_{SG}^{1}, D_{ID} and B are stored in their own memory. This completes the user's entire registration process.

2.2 Sensor Registration Phase

(1) Firstly, the sensor selects its own identity S_{ID}, generates a random sensor private key R_{SN}^{1}, and then transmits the generated parameter S_{ID} and R_{SN}^{1} to the gateway through the secure channel.

(2) After receiving the parameters S_{ID}, and R_{SN}^{1}, the gateway first generates a random gateway private key R_{SG}^{2}, encapsulates the sensor's identity, random gateway private key, and random sensor private key in the parameter C through XoR operation, and then calculates

$$S_{TID} = R_{SG}^{2} \oplus S_{ID} \tag{3}$$

Finally, the gateway stores the values of sensor identity, random sensor private key, random gateway private key, and sensor pseudo-identity in the memory.

2.3 Login and Mutual Authentication Phase

(1) First, the user enters the password, then calculates

$$Q = h(PW_D \parallel R_{SG}^{1}) \oplus D_{TID} \tag{4}$$

This is to test whether the value of Q is equal to B stored in the user memory. If these values are equal, the user generates a temporary random number N_D^1 and then calculates

$$N_D^{1*} = N_D^1 \oplus PW_D \tag{5}$$

$$K = h(R_{SG}^{1*} \parallel PW_D) \tag{6}$$

Finally, the user transmits parameters N_D^{1*}, D_{TID}, K, and S_{TID} to the gateway via a common channel.

(2) After receiving the parameter transmitted by the user, the gateway calculates

$$N_D^1 = N_D^{1*} \oplus PW_D \tag{7}$$

and verifies the parameter.
After the verification, it calculates

$$K^* = h(R_{SG}^{1} \parallel PW_D) \tag{8}$$

This is done to verify whether it is equal to the parameter value of K. If it is equal, the gateway generates a temporary random number N_G^1 and then calculates

$$G_W^1 = N_G^1 \oplus S_{TID} \tag{9}$$

$$G_W^2 = h(R_{SN}^{1} \parallel R_{SG}^{2}) \tag{10}$$

$$SK_s = SK \oplus R_{SN}^{1} \oplus N_G^1 \tag{11}$$

$$G_W^3 = R_{SG}^3 \oplus R_{SN}^1 \tag{12}$$

Finally, the gateway transmits the parameters G_W^1, G_W^2, D_{TID}, SK_s, and G_W^3 to the sensor through the secure channel.

(3) The sensor receives the parameters G_W^1, G_W^2, D_{TID}, SK_s, and G_W^3 from the gateway and calculates

$$N_G^1 = G_W^1 \oplus S_{TID} \tag{13}$$

and then verifies N_G^1. After verification, it calculates S_N^1 to verify whether S_N^1 is equal to the G_W^2. If it passes verification, the gateway will obtain the session key

$$SK = SK_s \oplus N_G^1 \oplus R_{SN}^1 \tag{14}$$

Next, the gateway generates a random number N_S^1 and calculates

$$S_N^2 = N_S^1 \oplus S_{TID} \tag{15}$$

$$S_N^3 = h(R_{SG}^{2*} \parallel R_{SN}^1 \parallel SK) \tag{16}$$

$$S_N^4 = R_{SG}^2 \oplus R_{SN}^2 \tag{17}$$

Then, the sensor updates its identity

$$S_{TID}^{new} = R_{SG}^3 \oplus S_{ID} \tag{18}$$

Next, the sensor stores the values of R_{SG}^2, R_{SG}^3, and S_{TID}^{new}. Finally, the sensor sends the values of S_N^2, S_N^3, and S_N^4 to the gateway through the secure channel.

(4) The gateway calculates

$$N_S^1 = S_N^2 \oplus S_{TID} \tag{19}$$

Then verifies N_S^1 and then calculates

$$G_W^4 = h(R_{SG}^2 \parallel R_{SN}^1 \parallel SK) \tag{20}$$

This verifies whether G_W^4 is equal to the received value of S_N^3. If yes, the gateway calculates the values of R_{SN}^2 and S_{TID}^{new}, and then stores the values of R_{SN}^2, R_{SG}^3, and S_{TID}^{new} in memory. Next, the gateway generates a random number N_G^2 and calculates the value of u, SK_u, n, and G_W^5 and then updates the user pseudo-identity. Finally, it stores the values of R_{SG}^4 and D_{TID}^{new} and transmits the values of parameters u, SKu, n, and G_W^5 to the user through the common channel.

(5) The user obtains the value of N_G^2 by XOR of the received u and D_{ID} and then verifies the N_G^2. Then, the value of SK and O is computed. Next, the calculated value of O is compared with the value of n. If it is equal, the user continues to calculate the value of R_{SG}^4 and D_{TID}^{new}. Finally, the user stores the values of R_{SG}^4 and D_{TID}^{new} in the memory. At this point, the entire login authentication process is complete.

3 Cryptanalysis of Masud et al.'s Protocol

In this section, we first introduce the attack model used in this paper and then analyze Masud's protocol [23] according to the attack model. The protocol cannot effectively resist privileged insider, sensor node capture, and stolen verification attacks, and there are loopholes in the perfect forward secrecy.

3.1 Threat Model

The attack model briefly describes the capabilities of $\mathcal{A}$, which has been described and discussed in [25,26] earlier. The details are as follows:

1. According to the "Dolev-Yao threat (DY) model" [27] proposed before, $\mathcal{A}$ can intercept and monitor information through the public channel. In addition, the attacker can modify the transmitted information. In other words, the session messages transmitted between the participants in the protocol through the common channel can be obtained and operated by $\mathcal{A}$. Moreover, $\mathcal{A}$ can act as an insider to obtain the information stored in the gateway during the registration phase.
2. Once the sensor is lost and acquired by $\mathcal{A}$, $\mathcal{A}$ can use power analysis [28,29] to operate the sensor. The sensitive information stored in the sensor can easily be obtained by $\mathcal{A}$. In this case, if the attacker has additional capabilities, it is easy to carry out sensor simulation and sensor node capture attacks [30].
3. In most user sensor authentication protocols, users often need to store some parameters in the registration phase for use in the login authentication phase. Usually, this information is stored in the user's smart card or memory. However, the user's smart card is often easy to lose. Once the smart card is obtained by $\mathcal{A}$, the attacker can use some parameter information stored in the smart card and combine it with some other parameters to carry out a series of attack operations.

3.2 Perfect Forward Secrecy

A good protocol must comprise the perfect forward secrecy feature [31,32], which ensures that master key leakage will not lead to session key leakage. Forward secrecy can protect past communication from the threat of key exposure in the future. Even in the case of master key leaks, the historical communication still has good security. However, in Masud's protocol, we found that if $\mathcal{A}$ obtains the value of the sensor's key R_{SN}^1, it can conveniently obtain the session key between the gateway and sensor. The specific process is as follows:

(1) First, $\mathcal{A}$ obtains the key-value R_{SN}^1 generated by the sensor.

(2) Second, $\mathcal{A}$ intercepts the parameters S_{TID} and G_W^1 through the common channel and then calculates

$$N_G^1 = G_W^1 \oplus S_{TID} \tag{21}$$

(3) The session key between the sensor and the gateway

$$SK = SK_s \oplus N_G^1 \oplus R_{SN}^1 \tag{22}$$

$\mathcal{A}$ can obtain the parameter SK_s as it transmits from the gateway to the gateway through the public channel. N_G^1 can also be calculated through the second step while securing R_{SN}^1 in the first step. Therefore, once the sensor key is exposed, the session key is obtained. However, there are some security vulnerabilities in the protocol.

3.3 Privilege Insider Attack

Privileged insider attack refers to a process in which $\mathcal{A}$ or the user information administrator obtains some of the user's basic information and then uses this information to carry out some basic operations, so as to obtain the user key between the medical staff and the sensor node [33].

(1) First, $\mathcal{A}$ can disguise as a privileged insider. In the process of user registration with the gateway, $\mathcal{A}$ can easily obtain the user's registration information D_{ID} and PW_D stored in the memory.

(2) Second, $\mathcal{A}$ intercepts the message u transmitted by the gateway to the user through the common channel and calculates

$$N_G^2 = u \oplus PW_D \tag{23}$$

(3) The session key SK can be obtained.

$$SK = SK_U \oplus N_G^2 \oplus PW_D \tag{24}$$

In the second step, the N_G^2 is calculated. SKU is transmitted to the user through the common channel in the authentication phase, which can also be obtained by $\mathcal{A}$. The PW_D is obtained by $\mathcal{A}$ as an insider. Therefore, $\mathcal{A}$ can obtain the session key between the user and the gateway. To sum up, Masud's protocol cannot effectively resist a privilege insider attack.

3.4 Stolen Verification Attack

A stolen verification attack implies that $\mathcal{A}$ can decode the value of the session key on the premise of acquiring the information stored in the gateway memory [34]. Masud's protocol cannot effectively resist the stolen verification attack; the specific attack process is as follows:

(1) First, $\mathcal{A}$ obtains the parameter R_{SG}^1 stored in the gateway memory during user registration, intercepts the parameter D_{TID} sent by the user to the gateway through the public channel during authentication, and then calculates

$$D_{ID} = R_{SG}^1 \oplus D_{TID} \tag{25}$$

(2) $\mathcal{A}$ obtains the parameter R_{SG}^4 in the gateway memory during authentication and intercepts the parameter G_W^5 transmitted by the common channel and then calculates

$$PW_D = R_{SG}^4 \oplus G_W^5 \tag{26}$$

(3) $\mathcal{A}$ has calculated the values of N_G^2 and PW_D, and the parameter SK_U transmitted through the secure channel. The session key

$$SK = SK_U \oplus N_G^2 \oplus PW_D \tag{27}$$

between the user and the gateway. Therefore, this protocol cannot effectively resist the stolen verification attack.

3.5 Sensor Node Capture Attack

The sensor node capture attack refers to the process in which the session key is leaked after $\mathcal{A}$ obtains the sensors [35]. Through our analysis, we found that Masud's protocol cannot resist sensor node capture attacks.

(1) During the registration of the sensor with the gateway, the sensor stores the identity S_{TID}, R_{SG}^{2*} and key R_{SN}^{1} in its own memory. However, sensors are likely to be acquired by $\mathcal{A}$.

(2) In the mutual authentication stage of gateway and sensor, the user transmits the parameter S_{TID} to the gateway through the common channel, and the gateway transmits the parameter G_W^1 to the sensor through the common channel. Then, $\mathcal{A}$ calculates

$$N_G^1 = G_W^1 \oplus S_{TID} \tag{28}$$

(3) Session key between sensor and gateway

$$SK = SK_s \oplus N_G^1 \oplus R_{SN}^1 \tag{29}$$

Once $\mathcal{A}$ obtains the sensor, $\mathcal{A}$ can obtain the session keys of both parties through a series of operations.

4 Proposed Protocol

We have analyzed Masud's protocol and listed the detailed attack process. A secure protocol must be able to resist some common attacks. We have improved Masud's protocol, and the improved protocol can successfully repair the aforementioned security vulnerabilities. In addition, we deleted some redundant symbols in the original protocol to make the entire protocol more concise. Our protocol consists of four parts: pre-deployment phase, user registration phase, sensor registration phase, and login authentication phase.

4.1 Symbol Table

The symbols used in the protocol are shown in Table 1.

Table 1: Notations used in the proposed protocol

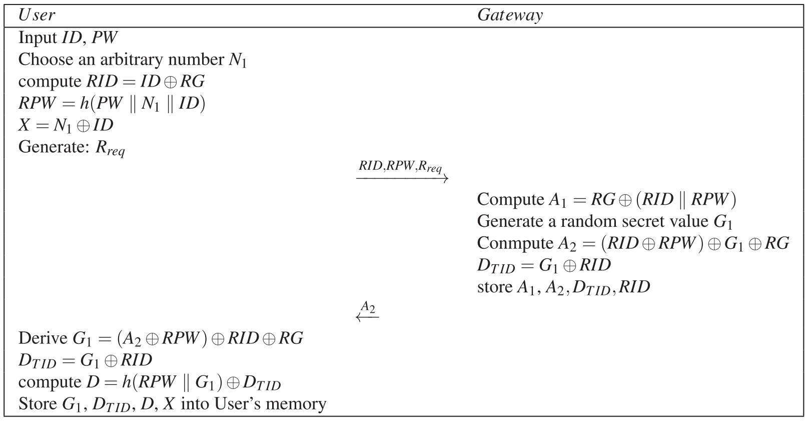
Notations	Descriptions
ID, S_{ID}	User's identity, Sensor node's identity
PW	User's password
R_{req}	User's registration request
RG	Gateway's secret key
G_1,G_2	Gateway's random secret key
S_1	Sensor's random secret key
S_{TID}	Temporary identity of sensor
D_{TID}	Temporary identity of user
N_1	Nonce generated by user
$\Rightarrow$	Private communication channel
$\rightarrow$	Public communication channel
$\mathcal{A}$	The adversary
$\oplus$ $\|$	Bit wise XOR operation, concatenation operator
SK	Session key

4.2 Pre-Deployment Phase

For the pre-deployment phase of users and sensors, the gateway first generates a key-value RG and then sends the key value to the users and sensors through the secure channel in advance.

4.3 User Registration Phase

Fig. 2 illustrates the user registration phase. The detailed steps are as follows:



User

Input ID, PW
Choose an arbitrary number N_1
compute $RID = ID \oplus RG$
$RPW = h(PW \parallel N_1 \parallel ID)$
$X = N_1 \oplus ID$
Generate: R_{req}

$\xrightarrow{RID, RPW, R_{req}}$

Gateway

Compute $A_1 = RG \oplus (RID \parallel RPW)$
Generate a random secret value G_1
Conmpute $A_2 = (RID \oplus RPW) \oplus G_1 \oplus RG$
$D_{TID} = G_1 \oplus RID$
store A_1, A_2, D_{TID}, RID

$\xleftarrow{A_2}$

Derive $G_1 = (A_2 \oplus RPW) \oplus RID \oplus RG$
$D_{TID} = G_1 \oplus RID$
compute $D = h(RPW \parallel G_1) \oplus D_{TID}$
Store G_1, D_{TID}, D, X into User's memory

Figure 2: User registration phase

(1) First, the user selects id ID and password PW, and a random number N_1, calculates

$$RID = ID \oplus RG \tag{30}$$

$$RPW = h(PW \parallel N_1 \parallel ID) \tag{31}$$

$$X = N_1 \oplus ID \tag{32}$$

and generates a request R_req for registration. Finally, the user transmits the information of RID and RPW to the gateway through the secure channel.

(2) After receiving the registration request from the user, the gateway calculates

$$A_1 = RG \oplus (RID \parallel RPW) \tag{33}$$

and then generates a random secret value G_1. It then calculates

$$A_2 = RID \oplus RPW \oplus G_1 \oplus RG \tag{34}$$

$$D_{TID} = G_1 \oplus RID \tag{35}$$

Finally, A_1, A_2, and D_{TID} are stored in the gateway, and A_2 is transmitted to users through a secure channel.

(3) According to the transmitted A_2, calculate

$$G_1 = A_2 \oplus RPW \oplus RID \oplus RG \tag{36}$$

$$D_{TID} = G_1 \oplus RID \tag{37}$$

$$D = h(RPW \parallel G_1) \oplus D_{TID} \tag{38}$$

Finally, G_1, D_{TID}, D and X is stored in the user's memory.

4.4 Sensor Registration Phase

Fig. 3 illustrates the sensor registration phase. The detailed steps are as follows:

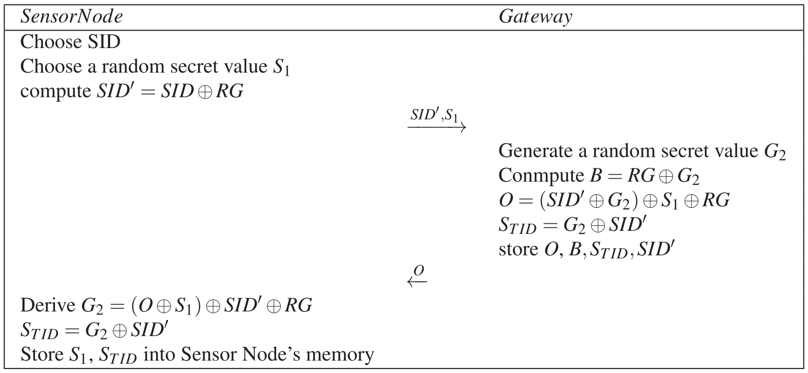

SensorNode
Gateway
Choose SID
Choose a random secret value S_1
compute $SID' = SID \oplus RG$
$\xrightarrow{SID', S_1}$
Generate a random secret value G_2
Conmpute $B = RG \oplus G_2$
$O = (SID' \oplus G_2) \oplus S_1 \oplus RG$
$S_{TID} = G_2 \oplus SID'$
store O, B, S_{TID}, SID'
$\xleftarrow{O}$
Derive $G_2 = (O \oplus S_1) \oplus SID' \oplus RG$
$S_{TID} = G_2 \oplus SID'$
Store S_1, S_{TID} into Sensor Node's memory

Figure 3: Sensor registration phase

(1) First, users select an identity SID for themselves and generate a random key value S_1 to calculate

$$SID' = SID \oplus RG \tag{39}$$

and then the user sends SID', S_1 to the gateway through the secure channel for registration.

(2) After receiving the message from the sensor, the gateway generates a random key value G_2 and encrypts the key value to obtain

$$B = RG \oplus G_2 \tag{40}$$

$$O = SID' \oplus G_2 \oplus S_1 \oplus RG \tag{41}$$

and encrypts the identity of the sensor to obtain the pseudo-identity of the sensor

$$S_{TID} = G_2 \oplus SID' \tag{42}$$

It next stores the parameters O, B, S_{TID} in the gateway memory and then sends the parameter O to the sensor through the secure channel.

(3) After the sensor receives the message, it first extracts the value of the gateway's key G_2,

$$G_2 = SID' \oplus O \oplus S_1 \oplus RG \tag{43}$$

and then calculates the sensor's pseudo-identity

$$S_{TID} = G_2 \oplus SID' \tag{44}$$

Finally, the sensor stores the parameter value S_1 and S_{TID} in the sensor memory.

4.5 Login and Authentication Phase

This section introduces the login and mutual authentication process between the user and the sensor through the gateway in detail as in Fig. 4. The following is the detailed description of login and authentication.

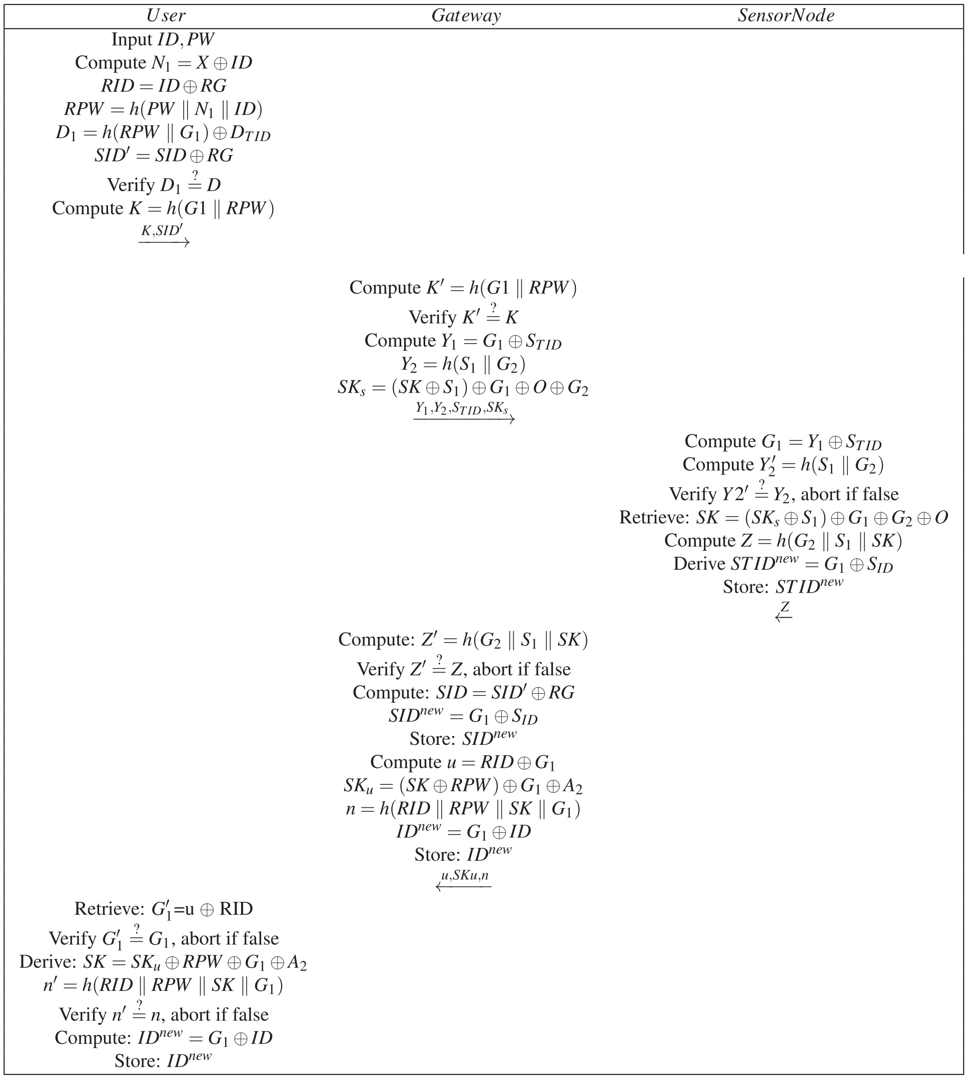

$User$
$Gateway$
$SensorNode$
Input ID, PW
Compute $N_1 = X \oplus ID$
$RID = ID \oplus RG$
$RPW = h(PW \parallel N_1 \parallel ID)$
$D_1 = h(RPW \parallel G_1) \oplus D_{TID}$
$SID' = SID \oplus RG$
Verify $D_1 \overset{?}{=} D$
Compute $K = h(G1 \parallel RPW)$
$\xrightarrow{K, SID'}$
Compute $K' = h(G1 \parallel RPW)$
Verify $K' \overset{?}{=} K$
Compute $Y_1 = G_1 \oplus S_{TID}$
$Y_2 = h(S_1 \parallel G_2)$
$SK_s = (SK \oplus S_1) \oplus G_1 \oplus O \oplus G_2$
$\xrightarrow{Y_1, Y_2, S_{TID}, SK_s}$
Compute $G_1 = Y_1 \oplus S_{TID}$
Compute $Y_2' = h(S_1 \parallel G_2)$
Verify $Y2' \overset{?}{=} Y_2$, abort if false
Retrieve: $SK = (SK_s \oplus S_1) \oplus G_1 \oplus G_2 \oplus O$
Compute $Z = h(G_2 \parallel S_1 \parallel SK)$
Derive $STID^{new} = G_1 \oplus S_{ID}$
Store: $STID^{new}$
$\xleftarrow{Z}$
Compute: $Z' = h(G_2 \parallel S_1 \parallel SK)$
Verify $Z' \overset{?}{=} Z$, abort if false
Compute: $SID = SID' \oplus RG$
$SID^{new} = G_1 \oplus S_{ID}$
Store: SID^{new}
Compute $u = RID \oplus G_1$
$SK_u = (SK \oplus RPW) \oplus G_1 \oplus A_2$
$n = h(RID \parallel RPW \parallel SK \parallel G_1)$
$ID^{new} = G_1 \oplus ID$
Store: ID^{new}
$\xleftarrow{u, SKu, n}$
Retrieve: G_1'=u $\oplus$ RID
Verify $G_1' \overset{?}{=} G_1$, abort if false
Derive: $SK = SK_u \oplus RPW \oplus G_1 \oplus A_2$
$n' = h(RID \parallel RPW \parallel SK \parallel G_1)$
Verify $n' \overset{?}{=} n$, abort if false
Compute: $ID^{new} = G_1 \oplus ID$
Store: ID^{new}

Figure 4: Login and authentication phase

(1) Before logging in, the user first enters the account id ID and password PW, used in registration. Then, the following is calculated:

$$N_1 = X \oplus ID \tag{45}$$

$$RID = ID \oplus RG \tag{46}$$

$$RPW = h(PW \parallel N_1 \parallel ID) \tag{47}$$

$$D_1 = h(RPW \parallel G_1) \oplus D_{TID} \tag{48}$$

$$SID' = SID \oplus RG \tag{49}$$

Subsequently, D_1 is verified to check whether it is equal to the D value previously stored in the user's memory. If it is, it implies that it is a login operation by a legal user. After successful login, the user calculates

$$K = h(G_1 \parallel RPW) \tag{50}$$

Finally, the user sends the parameter K and SID' to the gateway through the common channel.

(2) When the gateway receives the parameters from a legitimate user, it needs to determine whether the message sent has been tampered with by $\boldsymbol{\mathcal{A}}$, so it calculates

$$K' = h(G_1 \parallel RPW) \tag{51}$$

to compare K' with K, and equality implies that it passes verification. Then, the gateway continues to calculate

$$Y_1 = G_1 \oplus S_{TID} \tag{52}$$

$$Y_2 = h(S_1 \parallel G_2) \tag{53}$$

$$SK_s = SK \oplus S_1 \oplus G_1 \oplus G_2 \oplus O \tag{54}$$

SK_s is the operation in which the gateway distributes the key to the sensor. Finally, the gateway sends the parameter Y_1, Y_2, S_{TID}, SK_s to the sensor through the common channel.

(3) The sensor receives the message from the gateway, and first calculates the temporary key value G_1 of the gateway according to the values of S_{TID} and Y_1.

$$G_1 = Y_1 \oplus S_{TID} \tag{55}$$

then calculates

$$Y_2' = h(S_1 \parallel G_2) \tag{56}$$

and compares the S_2 value sent by the gateway with Y_2'. If it is equal, then $\boldsymbol{\mathcal{A}}$ has not tampered the parameters sent by the gateway. Next, the sensor calculates the session key

$$SK = SKs \oplus S_1 \oplus G_1 \oplus G_2 \oplus O \tag{57}$$

$$Z = h(G_2 \parallel S_1 \parallel SK) \tag{58}$$

according to the parameter SK_s sent by the gateway. Finally, the sensor identity is updated, storing the updated sensor parameter value in memory, and the value of parameter Z is sent to the gateway through the common channel.

(4) The gateway receives the parameter from the sensor. First, it checks whether $\boldsymbol{\mathcal{A}}$ intercepted the value of the parameter, calculates

$$Z' = h(G_2 \parallel S_1 \parallel SK) \tag{59}$$

and compares the value of Z' with that of Z. If it is equal, it means it passes verification. Next, the gateway obtains the identity value of the updated sensor through the following operation.

$$SID = SID' \oplus RG \tag{60}$$

$$STID^{new} = G_1 \oplus SID \tag{61}$$

Then, it stores the updated sensor identity value in its memory. Next, the gateway allocates the session key and computes

$$u = RID \oplus G_1 \tag{62}$$

$$SK_u = SK \oplus RPW \oplus G_1 \oplus A_2 \tag{63}$$

and updates the user's identity.

$$n = h(RID \parallel RPW \parallel G_1 \parallel SK) \tag{64}$$

$$ID^{new} = G_1 \oplus ID \tag{65}$$

The gateway stores the updated user's identity parameter in the gateway, and sends the value of parameter u, SKu, n to the user through the common channel.

(5) The user should first check the parameter value sent,

$$G_1' = u \oplus RID \tag{66}$$

If the value of G_1' is equal to the value of G_1 previously stored in the user memory, thus passing the verification. Next, the user calculates the session key

$$SK = SK_u \oplus RPW \oplus G_1 \oplus A_2 \tag{67}$$

between the user and gateway through the value of SK_u sent by the gateway. Then, before updating the user's identity, the following is performed

$$n' = h(RID \parallel RPW \parallel G_1 \parallel SK) \tag{68}$$

which is compared with the received value of n. The parameter is updated if it is equal.

$$ID^{new} = G_1 \oplus ID \tag{69}$$

Finally, the updated identity is stored in the user's memory.

5 Security Analysis

5.1 Formal Proof of the Proposed Protocol

5.1.1 ROR Model

In this section, we use the ROR model [24] to prove the security of the proposed protocol. In the protocol, we define three entities: user, gateway, and sensor node. For this proof, we assume that U_i, G_j, and S_z are the i-th user, the j-th gateway, and the z-th sensor node, respectively, and the parameter $T = \{U_i, G_j, S_z\}$. In the initial stage, $\mathcal{A}$ can perform the following query operations.

Execute (T): By performing this operation, $\mathcal{A}$ can obtain the messages $\{K\}$, $\{Y_1, Y_2, S_{TID}, SK_s\}$, $\{Z\}$, and $\{u, SK_u, n\}$ transmitted by U, G, and S through the common channel.

Send (T, M): By executing this query, $\mathcal{A}$ can transmit information M to T.

CorruptDevice (T): After executing this query, $\mathcal{A}$ can get the information stored in the U, G and S' memory. In addition, $\mathcal{A}$ can also get the long-term key in the protocol and the temporary information generated by the participant.

Hash (*string*): After entering a fixed-length string, $\mathcal{A}$ can get a fixed value after executing the query.

Test (T): In the initial stage, $\mathcal{A}$ tosses a coin O with uniform texture to judge whether the obtained session key is correct. If $O = 1$, the session key obtained is correct. Otherwise, $\mathcal{A}$ obtains a string with the same length as the session key.

Theorem: For the ROR model, if $\mathcal{A}$ performs some basic query operations, the probability that it can break the proposed protocol T in polynomial time is

$$Adv_{\mathcal{A}}^{\mathcal{T}}(\xi) \leq 2max\{D' \cdot q_{send}^{b'}, q_{send}/2^{f}\} + q_{send}/2^{f-2} + 3q_{hash}^{2}/2^{f-1} \tag{70}$$

In the formula, f represents the length of biological information entered by the user in the registration and login stage, and D' and b' represent two constants.

5.1.2 Security Proof

Proof: We defined 6 games $GM0$ to $GM5$ in the specific proof process, and everyone has different game rules. In the proof process, $Succ(GMi)(\eta)$ $(i = 0, 1, 2, 3, 4, 5)$ represents the probability of game success under each rule. The specific proof process is as follows:

$GM0$: In this game, $\mathcal{A}$ does not perform any query operations, so the probability of it breaking protocol T is: $Adv_{\mathcal{A}}^{T}(\eta) = |2Pr[Succ_{\mathcal{A}}^{T}(\eta) - 1]|$.

$GM1$: $GM1$ adds the *Execute* query operation on the basis of $Gm0$. That is, $\mathcal{A}$ can obtain the information $M1$, $M2$, $M3$ and $M4$ transmitted through the common channel. Then, it obtains the session key SK through the *Test* operation, but the random key S_1 of the S and G cannot be obtained, so the probability of success of $GM1$ is equal to that of $GM0$.

$$Pr[Succ_{\mathcal{A}}^{GM_1}] = Pr[Succ_{\mathcal{A}}^{GM_0}].$$

$GM2$: $GM2$ adds the *Send* operation on the basis of $GM1$, that is, $\mathcal{A}$ can send information to participants through the public channel. Under Zipf's law, we can easily obtain

$$|Pr[Succ_{\mathcal{A}}^{GM_2}] - Pr[Succ_{\mathcal{A}}^{GM_1}]| \leq q_{send}/2^{f}.$$

$GM3$: $GM3$ adds *Hash* query on the basis of $GM2$. $\mathcal{A}$ can get specific values through *Hash* operation. According to the birthday paradox, we get

$$|Pr[Succ_{\mathcal{A}}^{GM_3}] - Pr[Succ_{\mathcal{A}}^{GM_2}]| \leq q_{hash}^{2}/2^{f+1}.$$

$GM4$: In this game, we query the *CorruptDevice* to obtain the value of long-term key RG and the value of temporary information N_1 generated by U to verify whether the protocol has perfect forward security and resists temporary information leakage attacks.

Perfect forward secrecy: $\mathcal{A}$ obtains the parameter RG through the *CorruptDevice* operation, but it cannot obtain the user's pseudo identity RID, so the values of parameter A_2 and the user's pseudo password RPW cannot be obtained. Therefore, $\mathcal{A}$ obtains the long-term key, and RG cannot successfully obtain the session key.

Temporary information leakage attack: $\mathcal{A}$ obtains the temporary information N_1 generated by U but cannot obtain the user's identity ID and password PW. Therefore, $\mathcal{A}$ cannot obtain the user's pseudo password RPW. Even if $\mathcal{A}$ obtains the temporary information N_1, it cannot successfully obtain the session key. Therefore, our probability of getting $GM4$ is

$$|Pr[Succ_{\mathcal{A}}^{GM_4}] - Pr[Succ_{\mathcal{A}}^{GM_3}]| \leq q_{hash}^{2}/2^{f+1} + q_{send}/2^{f}.$$

*GM*5: Different from the *GM*4 rule, we query the information stored in the user's memory through *CorruptDevice*, and then prove that the proposed protocol can resist offline password guessing attacks. The probability that it can successfully guess the user password is 1/2, but in Zipf's law, when the number of transmitted bits $q_{send} \le 106$, the probability that $\mathcal{A}$ can successfully guess the user password is greater than 1/2. Therefore, we get

$$|Pr[Succ_{\mathcal{A}}^{GM_5}] - Pr[Succ_{\mathcal{A}}^{GM_4}]| \le max\{D' \cdot q_{send}^{b'}, q_{send}/2^f\}.$$

*GM*6: In *GM*6, in order to verify that the protocol we proposed can successfully resist user simulation attacks, unlike *GM*5, $\mathcal{A}$ queries through *Hash* operation. Therefore, the probability of *GM*6 is

$$|Pr[Succ_{\mathcal{A}}^{GM_6}] - Pr[Succ_{\mathcal{A}}^{GM_5}]| \le q_{hash}^2/2^{f+1}.$$

Because the probabilities of *GM*6 success and failure are equal, $Pr[Succ_{\mathcal{A}}^{GM_6}] = 1/2$.

From the formula calculated above, we can get

$$\begin{aligned} 1/2Adv_{\mathcal{A}}^{\mathcal{T}} &= |Pr[Succ_{\mathcal{A}}^{GM_0}] - 1/2| \\ &= |Pr[Succ_{\mathcal{A}}^{GM_0}] - Pr[Succ_{\mathcal{A}}^{GM_6}]| \\ &= |Pr[Succ_{\mathcal{A}}^{GM_1}] - Pr[Succ_{\mathcal{A}}^{GM_6}]| \\ &\le \sum_{i=0}^{5} |Pr[Succ_{\mathcal{A}}^{GM_{i+1}}] - Pr[Succ_{\mathcal{A}}^{GM_i}]| \\ &= max\{D' \cdot q_{send}^{b'}, q_{send}/2^f\} + q_{send}/2^{f-1} + 3q_{hash}^2/2^f \end{aligned}$$

Then

$$Adv_{\mathcal{A}}^{\mathcal{T}} \le 2max\{D' \cdot q_{send}^{b'}, q_{send}/2^f\} + q_{send}/2^{f-2} + 3q_{hash}^2/2^{f-1}$$

According to the above process, we prove that our proposed protocol can effectively resist user simulation, offline password guessing, and temporary information leakage attacks and has perfect forward security.

5.2 Informal Security Analysis

In this section, we describe how the new protocol can resist several common attacks. The following descriptions further prove the security of our proposed protocol.

5.2.1 Withstands Privileged Insider Attack

In this protocol, we assume that $\mathcal{A}$ disguises itself as a privileged insider. Therefore, $\mathcal{A}$ can obtain the user's pseudo-identity *RID* and pseudo password *RPW*. However, parameter A_2 is obtained after *RG* encryption. $A_2 = RID \oplus RPW \oplus G_1 \oplus RG$, *RG* is the long-term key generated by the gateway, which is only transmitted to users and sensors in the pre-deployment phase, so only users, gateways, and sensors know the key value. As a privileged insider, $\mathcal{A}$ cannot obtain the long-term key value, and thus cannot obtain A_2 and the session key between the user and the gateway.

5.2.2 Withstands Sensor Node Capture Attack

If $\mathcal{A}$ captures the sensor node information S_{TID} and S_1, then, although $\mathcal{A}$ already knows the parameters S_{TID} and S_1, it must know the long-term key RG of the gateway to obtain the parameter O. The calculation of the parameter RG must be participated by G_2 and B. B calculation is obtained by long-term key RG of the gateway, even if the node information of the sensor is captured, it impossible to obtain the public key between the gateway and the sensor. Therefore, our improved protocol can effectively resist the sensor capture attack.

5.2.3 Withstands Stolen Verification Attack

In a stolen verification attack, $\mathcal{A}$ obtains the message in the gateway memory, and $\mathcal{A}$ can obtain the key of both sides of the session. Suppose $\mathcal{A}$ obtains the information A_1, A_2, D_{TID} and RID in the gateway memory. First, parameters Y_1 and S_{TID} are transmitted on the common channel so we can obtain the value of parameter G_1. According to G_1 and D_{TID}, we can obtain RID. However, to calculate RPW, we must know the long-term key RG of a gateway. However, the acquisition of RG must be participated by RPW. Therefore, $\mathcal{A}$ cannot obtain the session key between the user and the gateway effectively. Suppose $\mathcal{A}$ obtains the information O, B, S_{TID} and SID' in the gateway; then, it can get the parameter G_1 according to the obtained information, but the session key is also composed of G_2 and S_1. The sensor's temporary key value S_1 is generated after encryption by the gateway's temporary key G_2 and long-term key RG, and it is impossible to obtain S_1. Therefore, $\mathcal{A}$ cannot successfully obtain the public session key between the gateway and the sensor. In conclusion, our new protocol can successfully resist the stolen authentication attack.

5.2.4 Forward Secrecy

Assuming that $\mathcal{A}$ has obtained the long-term key RG of the gateway, for the public session key between the user and the gateway, $\mathcal{A}$ needs to know the parameter A_2 and the user's pseudo password RPW. However, it does not know the user's pseudo-identity RID, so it cannot obtain the parameters A_2 and RPW, so the session key between the user and the gateway can be effectively protected. Second, for the session key between the sensor and the gateway, even if $\mathcal{A}$ obtains the long-term key RG, the communication between the gateway and the sensor still requires S_1, G_2 and O. G_2 needs to be obtained through the pseudo-identity of the sensor, but the pseudo-identity of the sensor cannot be obtained, and O must be obtained by the participation of the sensor's temporary key value S_1; therefore, $\mathcal{A}$ cannot obtain the session key between the sensor and the gateway.

5.2.5 Provides Anonymity

In the user registration stage, we perform the XOR operation on the user's ID and the long-term key RG of the gateway and then encrypt the user's identity. Subsequently, communication with the gateway occurs through the secure channel. Therefore, it is not easy for $\mathcal{A}$ to obtain the identity of legitimate users, so our protocol protects the identity privacy of users.

5.2.6 Withstands Password Guessing Attack

In the user login phase, the system verifies whether the value of D_1 is equal to the value of D stored in the user memory. $\mathcal{A}$ guesses the identity of a legitimate user if it can successfully guess the user's password. However, the user authentication also needs the participation of the random number N_1 generated by the user in the registration phase, so $\mathcal{A}$ cannot successfully carry out a password guessing attack.

5.2.7 Withstands Temporary Information Leakage Attack

If $\mathcal{A}$ obtains the random N_1 generated by the user in the registration phase but does not know the user's ID and password PW, the user's pseudo password RPW cannot be obtained. However, the session key between the user and the gateway needs the participation of the user's pseudo password RPW. Therefore, $\mathcal{A}$ cannot successfully carry out the temporary information leakage attack.

6 Security and Performance Comparisons

In this section, we analyze the security and performance of the new protocol. We compare the new protocol with other related protocols, mainly by comparing the running time, communication cost, and the ability to resist common attacks to show that our proposed protocol has an advantage in security and performance.

6.1 Security Comparisons

In this part, we compare with other related agreements. Finally, other protocols cannot resist all common attacks, but our new protocol can resist all attacks. At present, common network attacks include A1: Identity anonymity of user device, A2: Identity anonymity of IoT sensor node, A3: privileged-insider attack, A4: off-line password guessing attack, A5: Perfect forward secrecy, A6: man-in-the-middle attack, A7: IoT sensor node impersonation attack, A8: Sensor node capture attack, A9: Stolen verification attack. The comparison results are presented in Table 2. A "*Yes*" implies that the protocol can resist the attack, whereas a "*No*" means that it cannot.

Table 2: Comparisons of security

Protocols	A1	A2	A3	A4	A5	A6	A7	A8	A9
Challa et al. [36]	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*Yes*	*Yes*
Zhou et al. [37]	*Yes*	*Yes*	*Yes*	*No*	*No*	*No*	*No*	*Yes*	*Yes*
Farash et al. [38]	*No*	*No*	*No*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
Sharma et al. [21]	*No*	*No*	*No*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
Turkanovi et al. [39]	*Yes*	*Yes*	*No*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
Wazid et al. [40]	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*Yes*
Masud et al. [23]	*Yes*	*Yes*	*No*	*Yes*	*No*	*Yes*	*Yes*	*No*	*No*
Ours	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*

6.2 Performance Comparisons

For performance analysis, we use the same conditions to analyze the protocols in different environments. In the analysis process, because XOR and join operations take less time, we only analyze according to the non-collision hash function used in the protocol. The time required for the hash function is 0.00089 Ms. In addition, in the communication process, the number of bits required for the non-collision hash function is 256 bits.

First, we compare the communication cost between the protocol proposed in this paper and the related protocols proposed earlier. Here, we only consider the communication cost of the non-collision hash function. The communication cost of our protocol is 1,792 bits, lower than those of Masud et al. [23] (4,096 bits), Wazid et al. [40] (2,304 bits), Turkanovi et al. [39] (5,120 bits), Farash et al. [38] (5,888), Zhou et al. [37] (6,144 bits), and Challa et al. [36] (3,840 bits). This result can be observed in Fig. 5.

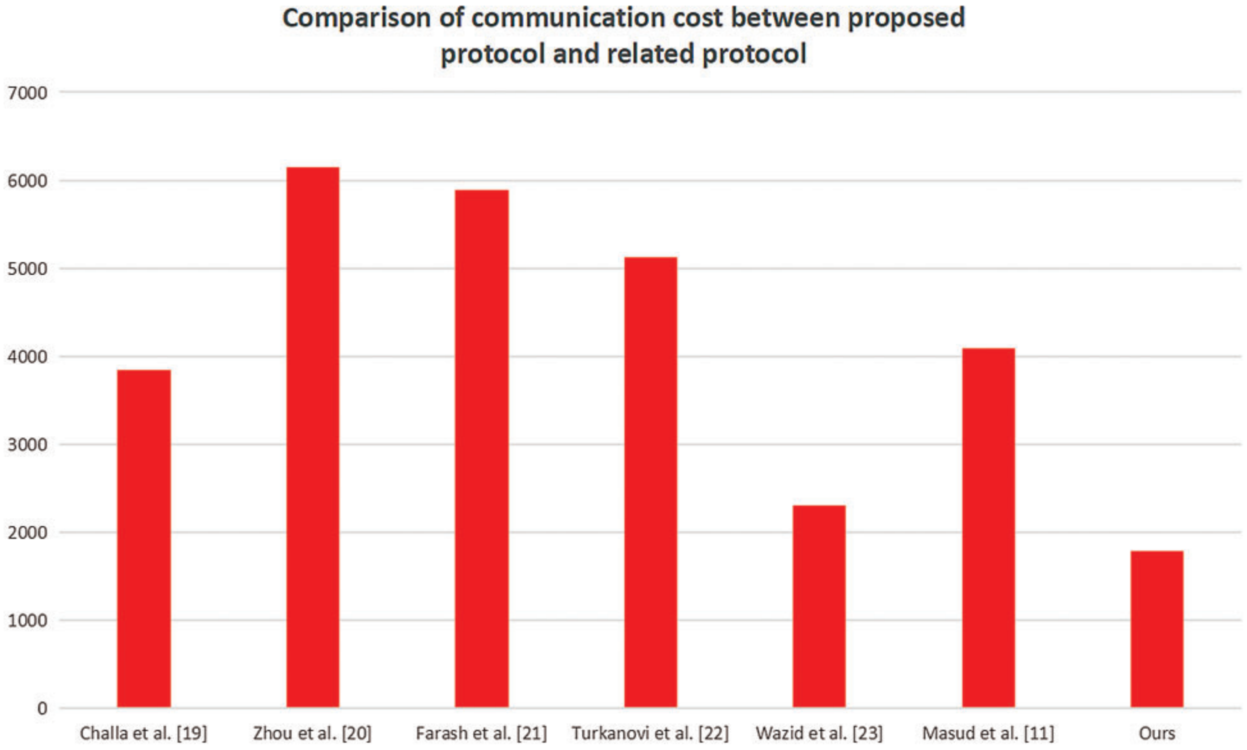


Figure 5: Communication cost

Second, we compare the protocols proposed in this paper with regard to time. Here, we only consider the running time of the non-collision hash function. Table 3 shows the number of hash functions required by the user gateway and sensor nodes during the protocol user registration phase, sensor registration phase, and login authentication phase where *H* represents the hash function. In Table 4, we compare the proposed protocol with those in other related fields. The results show that the time required for our proposed protocol is 0.00623 ms, and for Masud et al. [23], Wazid et al. [40], Turkanovi et al. [39], Sharma et al. [21], Farash et al. [38], Zhou et al. [37], Challa et al. [36], the times are 0.00712, 0.02848, 0.01513, 0.02047, 0.02848, 0.03204, and 0.01068 ms, respectively. It can be seen more intuitively in Fig. 6 that the running cost of the protocol proposed by us is better than those proposed in other relevant papers.

Table 3: The computational cost of the proposed protocol

Phase	User	Gateway	Sensor	Total
User registration	2H	0H	0H	2H
Sensor node registration	0H	0H	1H	1H
Mutual authentication	4H	1H	2H	7H
Total	6H	1H	3H	10H

Table 4: Calculation cost comparison

Protocol	User	Gateway	Sensor	Total	Time cost (ms)
Challa et al. [36]	5H	4H	3H	12H	0.01068
Zhou et al. [37]	10H	7H	19H	36H	0.03204
Farash et al. [38]	11H	14H	7H	32H	0.02848
Sharma et al. [21]	11H	7H	5H	23H	0.02047
Turkanovi et al. [39]	5H	5H	7H	17H	0.01513
Wazid et al. [40]	9H	15H	8H	32H	0.02848
Masud et al. [23]	3H	3H	2H	8H	0.00712
Ours	4H	1H	2H	7H	0.00623

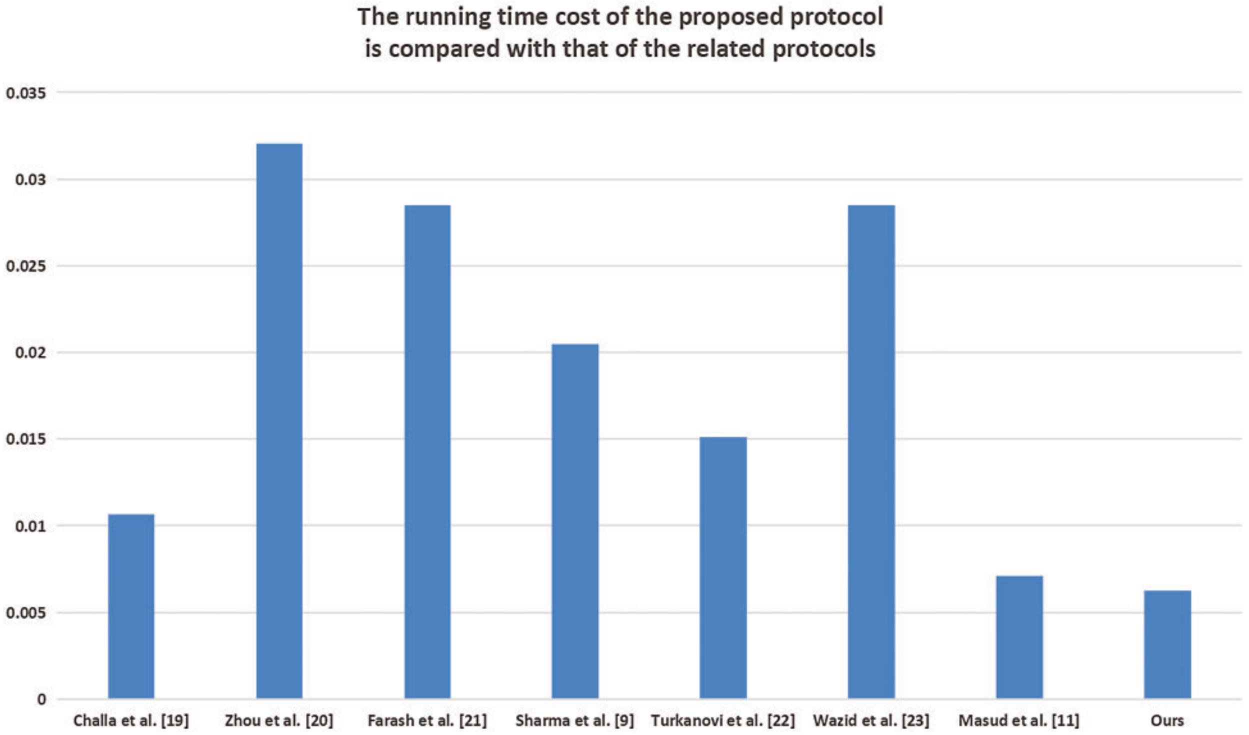


Figure 6: Running time

After comparing our protocol with other related protocols, we can observe that the proposed protocol can effectively resist various attacks, and so we can say that our protocol has perfect security. In addition, our proposed protocol is superior to the existing protocol in terms of communication cost and time running cost. To sum up, the proposed protocol is more suitable for the development of future medical systems and is more convenient and user friendly for future medical staff and patients.

7 Conclusions

This paper improves Masud's authentication protocol for the medical system. The improved protocol not only resists the common attacks that the existing protocol was unable to but also removes the redundant symbols in the original protocol, reducing the communication cost. In addition, it retains the lightweight advantage of the original protocol. The improved protocol still adopts a single hash and bit-by-bit XOR operation, which reduces the running time. The protocol is secure against privileged internal attacks, stolen verification attacks, and sensor node capture attacks, thus presenting perfect forward security. This protocol is more suitable for the future medical environment. It preserves the security in the medical system as well as the user privacy, while additionally enhancing the system performance.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Xiong, H., Huang, X., Yang, M., Wang, L., Yu, S. (2021). Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted Internet of Things. *IEEE Internet of Things Journal.* DOI 10.1109/JIOT.2021.3094323.
2. Hou, Y., Xiong, H., Huang, X., Kumari, S. (2021). Certificate-based parallel key-insulated aggregate signature against fully chosen key attacks for industrial Internet of Things. *IEEE Internet of Things Journal, 8(11),* 8935–8948. DOI 10.1109/JIOT.2021.3056477.
3. Wang, K., Chen, C. M., Liang, Z., Hassan, M. M., Sarné, G. M. et al. (2021). A trusted consensus fusion scheme for decentralized collaborated learning in massive IoT domain. *Information Fusion, 72,* 100–109. DOI 10.1016/j.inffus.2021.02.011.
4. Wang, P., Chen, C. M., Kumari, S., Shojafar, M., Tafazolli, R. et al. (2021). HDMA: Hybrid D2D message authentication scheme for 5G-enabled vanets. *IEEE Transactions on Intelligent Transportation Systems, 22(8),* 5071–5080. DOI 10.1109/TITS.2020.3013928.
5. Wang, X., Liu, Y., Choo, K. K. R. (2020). Fault-tolerant multisubset aggregation scheme for smart grid. *IEEE Transactions on Industrial Informatics, 17(6),* 4065–4072. DOI 10.1109/TII.2020.3014401.
6. Li, C. T., Lee, C. C., Weng, C. Y., Chen, C. M. (2018). Towards secure authenticating of cache in the reader for rfid-based IoT systems. *Peer-to-Peer Networking and Applications, 11(1),* 198–208. DOI 10.1007/s12083-017-0564-6.
7. Wu, T. Y., Wang, T., Lee, Y. Q., Zheng, W., Kumari, S. et al. (2021). Improved authenticated key agreement scheme for fog-driven IoT healthcare system. *Security and Communication Networks, 2021.*
8. Chen, C. M., Li, C. T., Liu, S., Wu, T. Y., Pan, J. S. (2017). A provable secure private data delegation scheme for mountaineering events in emergency system. *IEEE Access, 5,* 3410–3422. DOI 10.1109/ACCESS.2017.2675163.
9. Ayub, M. F., Mahmood, K., Kumari, S., Sangaiah, A. K. (2021). Lightweight authentication protocol for e-health clouds in IoT-based applications through 5G technology. *Digital Communications and Networks, 7(2),* 235–244. DOI 10.1016/j.dcan.2020.06.003.
10. Shahbazi, Z., Byun, Y. C. (2020). Towards a secure thermal-energy aware routing protocol in wireless body area network based on blockchain technology. *Sensors, 20(12),* 3604. DOI 10.3390/s20123604.
11. Benmansour, T., Ahmed, T., Moussaoui, S., Doukha, Z. (2020). Performance analyses of the IEEE 802.15. 6 wireless body area network with heterogeneous traffic. *Journal of Network and Computer Applications, 163,* 102651. DOI 10.1016/j.jnca.2020.102651.

12. Hasan, K., Biswas, K., Ahmed, K., Nafi, N. S., Islam, M. S. (2019). A comprehensive review of wireless body area network. *Journal of Network and Computer Applications, 143,* 178–198. DOI 10.1016/j.jnca.2019.06.016.
13. Chen, H. M., Lo, J. W., Yeh, C. K. (2012). An efficient and secure dynamic ID-based authentication scheme for telecare medical information systems. *Journal of Medical Systems, 36(6),* 3907–3915. DOI 10.1007/s10916-012-9862-y.
14. Cao, T., Zhai, J. (2013). Improved dynamic ID-based authentication scheme for telecare medical information systems. *Journal of Medical Systems, 37(2),* 9912. DOI 10.1007/s10916-012-9912-5.
15. He, D., Kumar, N., Chen, J., Lee, C. C., Chilamkurti, N. et al. (2015). Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimedia Systems, 21(1),* 49–60. DOI 10.1007/s00530-013-0346-9.
16. Li, X., Niu, J., Karuppiah, M., Kumari, S., Wu, F. (2016). Secure and efficient two-factor user authentication scheme with user anonymity for network based e-health care applications. *Journal of Medical Systems, 40(12),* 1–12. DOI 10.1007/s10916-016-0629-8.
17. He, D. B., Chen, J. H., Zhang, R. (2012). A more secure authentication scheme for telecare medicine information systems. *Journal of Medical Systems, 36(3),* 1989–1995. DOI 10.1007/s10916-011-9658-5.
18. Wei, J., Hu, X., Liu, W. (2012). An improved authentication scheme for telecare medicine information systems. *Journal of Medical Systems, 36(6),* 3597–3604. DOI 10.1007/s10916-012-9835-1.
19. Wu, F., Li, X., Sangaiah, A. K., Xu, L., Kumari, S. et al. (2018). A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Generation Computer Systems, 82,* 727–737. DOI 10.1016/j.future.2017.08.042.
20. Wazid, M., Das, A. K., Vasilakos, A. V. (2018). Authenticated key management protocol for cloud-assisted body area sensor networks. *Journal of Network and Computer Applications, 123,* 112–126. DOI 10.1016/j.jnca.2018.09.008.
21. Sharma, G., Kalra, S. (2019). A lightweight user authentication scheme for cloud-IoT based healthcare services. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 43(1),* 619–636. DOI 10.1007/s40998-018-0146-5.
22. Canetti, R., Krawczyk, H. (2002). Universally composable notions of key exchange and secure channels. *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, The Netherlands.
23. Masud, M., Gaba, G. S., Choudhary, K., Hossain, M. S., Alhamid, M. F. et al. (2021). Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. *IEEE Internet of Things Journal*. DOI 10.1109/JIOT.2021.3080461.
24. Abdalla, M., Fouque, P. A., Pointcheval, D. (2005). Password-based authenticated key exchange in the three-party setting. *International Workshop on Public Key Cryptography*, Springer.
25. Wang, D., He, D., Wang, P., Chu, C. H. (2014). Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Transactions on Dependable and Secure Computing, 12(4),* 428–442. DOI 10.1109/TDSC.2014.2355850.
26. Wang, D., Wang, P. (2016). Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE Transactions on Dependable and Secure Computing, 15(4),* 708–722. DOI 10.1109/TDSC.2016.2605087.
27. Dolev, D., Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory, 29(2),* 198–208. DOI 10.1109/TIT.1983.1056650.
28. Kocher, P., Jaffe, J., Jun, B. (1999). Differential power analysis. *Annual International Cryptology Conference*, Springer. California, USA.
29. Ren, Y., Wu, L. (2013). Power analysis attacks on wireless sensor nodes using CPU smart card. *22nd Wireless and Optical Communication Conference*, IEEE, Chongqing, China.
30. Far, H. A. N., Bayat, M., Das, A. K., Fotouhi, M., Pournaghi, S. M. et al. (2021). LAPTAs: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wireless Networks, 27(2),* 1389–1412. DOI 10.1007/s11276-020-02523-9.

31. Li, P., Su, J., Wang, X. (2020). iTLS: Lightweight transport-layer security protocol for IoT with minimal latency and perfect forward secrecy. *IEEE Internet of Things Journal, 7(8),* 6828–6841. DOI 10.1109/JIoT.6488907.

32. Xu, S., Liu, X., Ma, M., Chen, J. (2020). An improved mutual authentication protocol based on perfect forward secrecy for satellite communications. *International Journal of Satellite Communications and Networking, 38(1),* 62–73. DOI 10.1002/sat.1309.

33. Ul Haq, I., Wang, J., Zhu, Y., Maqbool, S. (2020). A survey of authenticated key agreement protocols for multi-server architecture. *Journal of Information Security and Applications, 55,* 102639. DOI 10.1016/j.jisa.2020.102639.

34. Chen, C. M., Ku, W. C. (2002). Stolen-verifier attack on two new strong-password authentication protocols. *IEICE Transactions on Communications, 85(11),* 2519–2521.

35. Xu, G., Wang, F., Zhang, M., Peng, J. (2020). Efficient and provably secure anonymous user authentication scheme for patient monitoring using wireless medical sensor networks. *IEEE Access, 8,* 47282–47294. DOI 10.1109/Access.6287639.

36. Challa, S., Wazid, M., Das, A. K., Kumar, N., Reddy, A. G. et al. (2017). Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access, 5,* 3028–3043. DOI 10.1109/ACCESS.2017.2676119.

37. Zhou, L., Li, X., Yeh, K. H., Su, C., Chiu, W. (2019). Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Generation Computer Systems, 91,* 244–251. DOI 10.1016/j.future.2018.08.038.

38. Farash, M. S., Turkanovi, M., Kumari, S., Hölbl, M. (2016). An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Networks, 36,* 152–176. DOI 10.1016/j.adhoc.2015.05.014.

39. Turkanović, M., Brumen, B., Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks, 20,* 96–112. DOI 10.1016/j.adhoc.2014.03.009.

40. Wazid, M., Das, A. K., Shetty, S., Rodrigues, J. J. P. C., Park, Y. (2019). LDAKM-EIoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment. *Sensors, 19(24),* 5539. DOI 10.3390/s19245539.