Future Generation Computer Systems 82 (2018) 727–737

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks

Fan Wu [a], Xiong Li [b,*], Arun Kumar Sangaiah [c], Lili Xu [d], Saru Kumari [e], Liuxi Wu [a], Jian Shen [f]

[a] Department of Computer Science and Engineering, Xiamen Institute of Technology, Xiamen 361021, China
[b] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China
[c] School of Computer Science and Engineering, VIT University, Vellore-632014, Tamil Nadu, India
[d] School of Information Science and Technology, Xiamen University, Xiamen 361005, China
[e] Department of Mathematics, Chaudhary Charan Singh University, Meerut, 250005, Uttar Pradesh, India
[f] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

## HIGHLIGHTS

- A lightweight two-factor authentication scheme using WMSNs away from being tracked is presented.
- We use the famous tool Proverif to prove that our scheme is secure against the common attacks.
- The informal analysis and performance comparison with recent schemes show that ours is the best.
- The simulation with NS-3 shows that our scheme is applicable for practice.

## ARTICLE INFO

*Article history:*
Received 31 March 2017
Received in revised form 8 July 2017
Accepted 23 August 2017
Available online 6 September 2017

*Keywords:*
Wireless medical sensor network
Mutual authentication
Proverif tool
User anonymity
Personalized healthcare system

## ABSTRACT

Wireless Sensor Network (WSN) is a very important part of Internet of Things (IoT), especially in e-healthcare applications. Among them, wireless medical sensor networks (WMSNs) have been used in the personalized healthcare systems (PHSs). In recent years, professionals use their mobile devices to access the data collected from sensors which are placed in or on patients' bodies. Due to the danger of wireless transmission circumstance, the security of the data which are collected by the sensors and also transmitted to the doctors faces challenges. In the past decade, many authentication schemes for WMSNs are proposed. However, security disadvantages have been found in such schemes. To overcome the historical security problems, we propose a robust and lightweight authentication scheme for WMSNs, which meets the common security requirements, and keeps away user tracking from attackers. The popular tool Proverif is employed to express that our scheme resists the simulated attacks. Also, the informal security analysis is demonstrated. With the comparison to several very recent schemes and simulation by NS-3, the proposed scheme is suitable for PHSs.



## 1. Introduction

To meet the requirements of exploring the global circumstance, the notion Internet of Things (IoT) is proposed and applied over the whole world. Such conception means that a system contains computing devices, digital objects, animals, plants and persons. Every entity in the system owns a unique identity and an address, so the data of the concerned targets can be transmitted via the networks. Among the applications, wireless sensor network (WSN) is important. WSN contains a set of technologies which profoundly affect the current industries, agriculture, military, medical care and so forth. With many customized sensors gathering signals from the aimed object, people can obtain the timely situation and make decisions.

Nowadays, social IoT (SIoT) becomes a hot topic among researchers. It means that some objects in IoT build social relationships with other objects, which are about human. These relationships are based on the objects' movements, profile and functions provided for people, such as locating someone in an emergency call, checking the suitable bus and time for going out, etc. Wireless Medical Sensor Network (WMSN) is an important kind of

* Corresponding author.
*E-mail address:* lixiong@hnust.edu.cn (X. Li).

http://dx.doi.org/10.1016/j.future.2017.08.042
0167-739X/

application of the WSN for personalized healthcare systems (PHSs) in e-healthcare scope. To get various sorts of information from the patient in personalized healthcare systems, heterogeneous sensors in WMSN which can collect different kinds of data show their advantages in many fields. Many bio-sensors of different kinds are placed in or on the patient's body and data like heart and breathing rates, blood pressure and movement are gathered by them. Those data are transferred in the wireless channel, which is known to be very insecure.

The concrete architecture of WMSN in personalized healthcare systems is demonstrated in Fig. 1. There are three kinds of participants in a classic WMSN: the professional, or the user, such as the doctor or the nurse, who needs the data of target patient; the sensors, which gather the special data from the aimed patient and have weak energy and computation ability; the gateway (*GW*), which has strong calculation power, and much more resource and energy for communication, is a critical and secure intermediary between the user and the sensors. Different patients use different sensor suites. In China, a famous doctor often needs to diagnose patients distributed in different locations. So different WMSNs should be accessed. Generally, if the professional makes a request for data from some sensor, he/she uses his/her mobile device to contact the sensor set via *GW* first and obtains the data from the special sensor. In recent years, some authentication schemes have employed the way that the sensor contacts with the user directly [1–3]. That way will make the energy in the weak sensor wasted seriously and decrease the life of sensor [4,5]. How to guarantee the security of data transmission between the three entities is an emergent question. To avoid the various attacks from the adversary, information encryption is an important technology to provide the secrecy of entities [6–9]. Moreover, mutual authentication and user anonymity [5,10–19] are two basic required features. However, researchers now consider that a fixed pseudo-identity for user may lead to be tracked by the adversary and try to design schemes where the user employs different pseudo-identities in different sessions. This is stronger than user anonymity. And many researchers proposed their authentication schemes for WMSNs to satisfy the aims [2,19–22].

### 1.1. Related work

In 2009, Das [23] presented a two-factor authentication scheme for WSNs. In such scheme, the user needs both his/her own password and a mobile device (e.g., a smart card) storing data related to his/her own information, to access the remote server. But soon researchers [24–26] showed that many weaknesses existed in Das' scheme, such as vulnerability to the insider attack and the impersonation attack. Enhanced schemes are proposed in the above three papers. In 2011, Kumar et al. [27] pointed out that weaknesses like lack of mutual authentication and key agreement existed in [25,26]. Also in 2011, Yeh et al. [28] considered that the scheme in [24] had weaknesses containing susceptibility to the insider attack and was devoid of password change. In 2012, Kumar et al. [29] put forward a new two-factor authentication scheme for WMSNs and claimed that it could withstand known attacks. But He et al. [1] showed that the scheme in [29] was vulnerable to the off-line guessing attack and the insider attack. Moreover, the property user anonymity cannot be kept in [29], either. In 2013, Xue et al. [30] showed an authentication scheme for WSNs, with temporal credential including the hash results of user data. But unluckily, Jiang et al. [20] gave the disadvantages of scheme in [30], including vulnerability to the off-line password guessing attack and tracking attack. But papers [21,31] showed that Jiang et al.'s scheme had weaknesses like susceptibility to the desynchronization attack and user forgery attack. In 2014, Turkanović et al. [32] proposed a new lightweight two-factor authentication scheme for WSNs. In the scheme there are only hash functions and exclusive-or computations. Soon in 2015, Farash et al. [33] and Amin-Biswas [14] considered that the scheme in [32] could not resist the identity guessing attack, the off-line password guessing attack and the user forgery attack. As the illustration in [14], both the schemes in [32,33] employed the style that the user contacted the sensor directly. But the way is not suitable for applications due to sensor energy waste. And they presented a new multi-gateway-based authentication scheme. In 2015, Li et al. [34] and Wu et al. [2] pointed out that the disadvantages like the sensor capture attack, the de-synchronization attack and the off-line guessing attack could be completed on the scheme in [1], respectively. In 2016, Wu et al. [35] claimed that the scheme in [14] could not resist attacks such as sensor capture attack and tracking attack.

In 2016, Amin et al. [4] considered that the schemes in [1,2,29] also had a critical weakness that the user could contact the sensor directly at last in the authentication. And they proposed a novel two-factor authentication scheme for WMSNs. In fact, Amin et al.'s scheme [4] still has weaknesses, such as susceptibility to the off-line guessing attack and the de-synchronization attack. Moreover, Kumari and Om [36] and Amin et al. [4] proposed their two-factor authentication schemes for WSNs, respectively. Both the two schemes are lightweight. And unlike the most of the above schemes, the session key are constructed by all the three entities. Unfortunately, both of them still have weaknesses. The scheme in [36] cannot withstand the off-line guessing attack and the user tracking attack. Also, destitution of proper encryption makes the session key disclosure. In the same year, Kumari et al. [37] showed that the temporal-credential-based authentication schemes proposed by Li et al. [38] and He et al. [17] were both insecure, e.g., vulnerable to off-line password guessing attack. Unfortunately, their scheme is under the off-line guessing attack and has relatively heavy computation burden since chaotic map is employed in all three entities in the session. Furthermore, Gope and Hwang [39] proposed a lightweight authentication scheme for real-time WSN data access. They used a suit of backup mechanism to resist the denial-of-service attack. In 2017, Srinivas et al. [3] deemed that the scheme in [2] has weaknesses such as insider attack and off-line password guessing attack. However, it is unlucky that both the schemes in [3,39] are not secure since off-line guessing attacks could affect them, and they are impractical if running. The latter problem is the most important point.

### 1.2. Motivation and contributions

From the above demonstration, it is an urgent task to propose a lightweight and secure mutual authentication scheme for WMSNs. Our paper meets this requirement. Moreover, the contributions of our paper are below:

1. A novel and lightweight two-factor authentication scheme for WMSNs resistant from being tracked by the attacker is presented.
2. We use the famous tool Proverif [40] to prove that our scheme is secure against various common attacks.
3. The informal analysis and performance comparison with some very recent schemes of the same sort show that ours is the best.
4. The simulation with NS-3 shows that our scheme is applicable for practice.

### 1.3. Organization of the paper

The remainder of our paper is constructed as follows. The preliminary knowledge is in Section 2. Our scheme and the relative Proverif code are illustrated in Sections 3 and 4, respectively. After that we give the informal analysis and performance comparison in Section 5. Then the tool NS-3 is used to make a simulation in Section 6. Finally, the conclusion appears in Section 7.

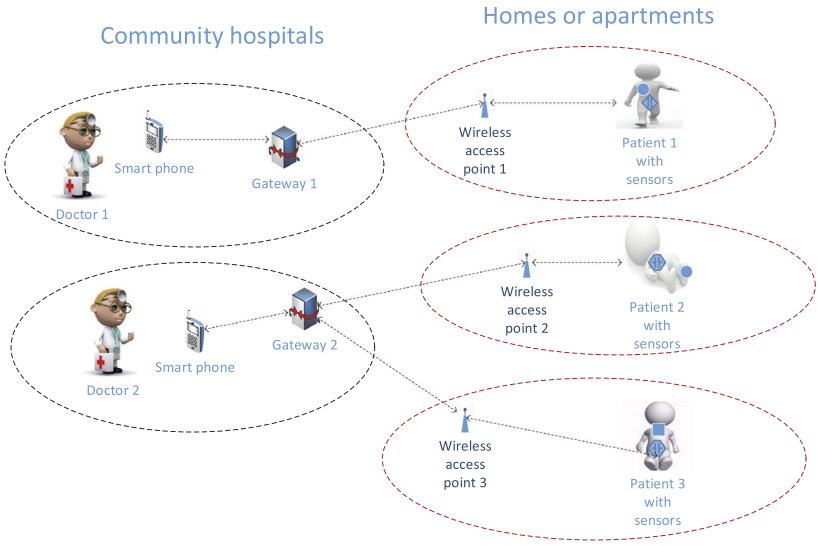


**Fig. 1.** Architecture of WMSNs.

**Table 1**
Notations.

| Symbol | Meaning |
|---|---|
| $U_i, ID_i, PW_i$ | The $i$th user with his/her identity and password |
| $GW_j, GID_j, G_j$ | The $j$th gateway with its identity and secret key |
| $N_l$ | The number of the sensor set |
| $SN_k, SID_k$ | The $k$th sensor with its identity |
| $SG_k$ | The shared secret key between $SN_k$ and the corresponding gateway in the WMSN |
| $sk_u, sk_g, sk_s$ | The session keys formed by the user, the gateway and the sensor |
| $M_i (i = 1, 2...)$ | messages in authentication phase |
| $\mathcal{A}$ | The adversary |
| $h(.)$ | The hash function |
| $\oplus$ | The exclusive-or operation |
| $\|$ | The concatenation operation |

## 2. Preliminaries

### 2.1. Notations

Here we illustrate some notations used throughout our paper in Table 1.

### 2.2. Premises for informal security analysis

To analyze the security properties, we give some assumptions. We only focus on the abilities of $\mathcal{A}$ but not on how to do them.

**Assumption 1.** Based on [41], we suppose that the data in the user's mobile device can be retrieved by $\mathcal{A}$. This is for the accident of mobile device loss.

**Assumption 2.** In papers [4,36], researchers give a hypothesis that $\mathcal{A}$ can guess either the user's identity or the user's password in polynomial time, but not the both. Moreover, there is no explanation for that in the papers. That does not make sense. In fact, some earlier papers [1,2,21,42] have employed the suitable hypothesis that the adversary $\mathcal{A}$ can guess the two strings simultaneously in polynomial time. And we use this rational premise. However, $\mathcal{A}$ cannot guess the hash results, random numbers and secret keys since they have the security length. Also, $\mathcal{A}$ cannot find the collisions of hash results in polynomial time.

**Assumption 3.** Two channels exist in the scheme: a secure, or a private channel in which messages in the registration phase are transmitted, and an insecure, or a public channel in which messages in authentication and password change phases are transmitted. $\mathcal{A}$ can control the public channel under the two-factor authentication environment [21,22,43], but cannot get any data from the private channel. Two-factor authentication means that the user can use his password and his mobile device (such as smart card, smart phone, etc.) to protect the privacy. In [44,45], the attacker has unlimited ability in the authentication. This premise is fit for the formal proof, and puts the concentration on the probability for $\mathcal{A}$ guessing the judging bit in the aim session, with enhancing the attacker's ability step by step, such as the Section 6 of [43]. But in the informal analysis, the parlance "two-factor authentication scheme is secure even if one of the factor is cracked" is the conventional rule for the discussion of security properties. One question appears: when the factor mobile device can be cracked by $\mathcal{A}$? In fact, we can see that the operation steps including first eavesdropping messages in public channel and then retrieving data from the mobile device exist in all the analysis of off-line guessing attacks for schemes in Section 1.1. That is just the implication of the conventional rule. On the opposite, if $\mathcal{A}$ gets the data from the mobile device first and then eavesdrops the authentication messages, the security based on two factors only depends on the password and the mobile device loses its effect, and "two-factor" turns to be "only one factor". It denotes that the attacker breaks one factor without any burden and gains some relative data about the remaining factor. In other words, $\mathcal{A}$ gets more information under such condition than under the password—only architecture. "two-factor" is meaningless under that condition. So we use the view in [43] and emphasize "two-factor authentication environment"

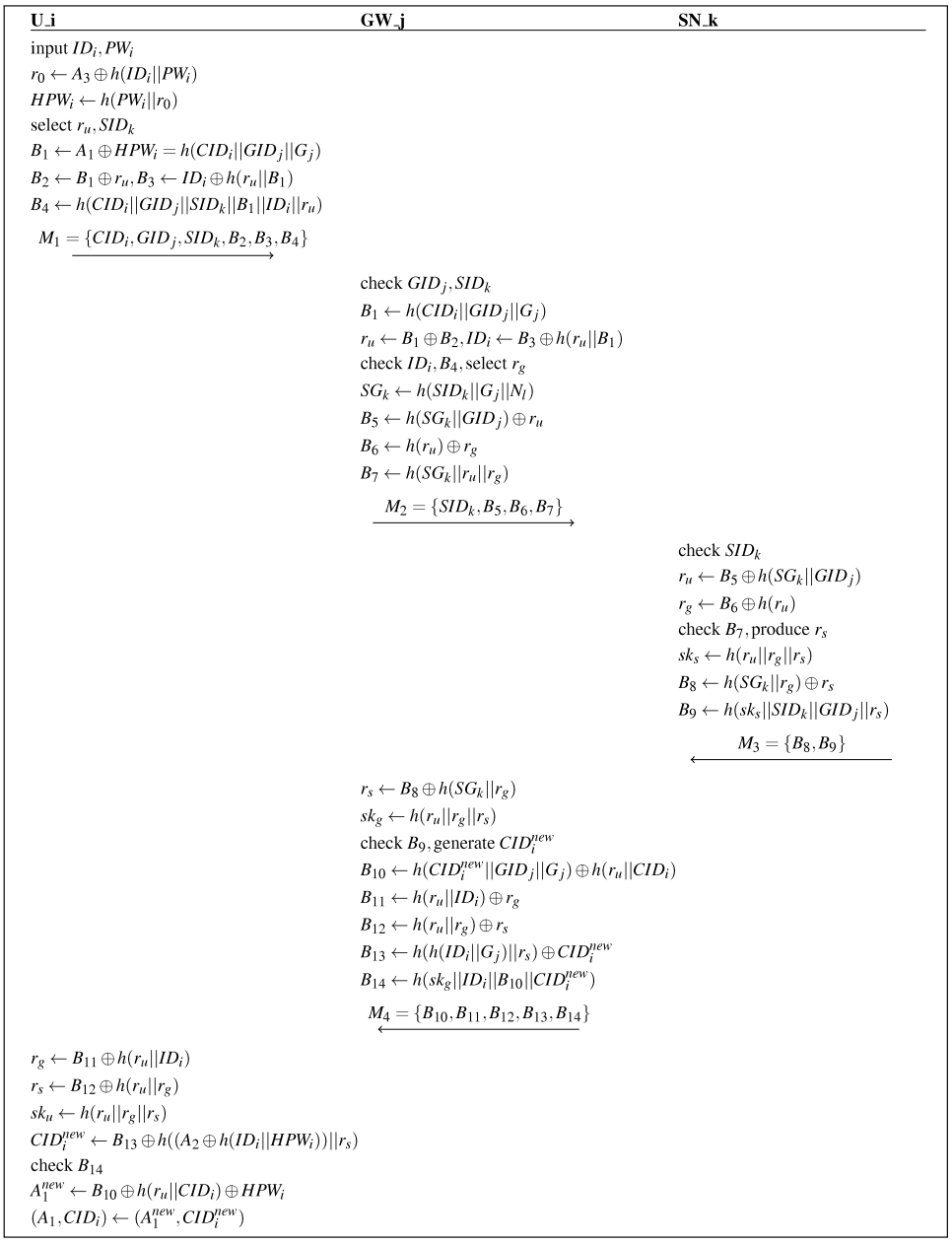


**Fig. 2.** Authentication phase of our scheme.

as the case that password and the mobile device are both valid for authentication and such time period is fit for $\mathcal{A}$'s control in the public channel.

**Assumption 4.** Sensor capture attack means that $\mathcal{A}$ can compromise some sensors but not the one which communicates with $U_i$. The focus of this attack is about the relation of secret keys in different sensors. In fact, researchers have already used this attack, and this point is mentioned in the review paper [46].

## 3. Proposed scheme

Our scheme is divided into four phases: initialization, registration, authentication and password change. The messages in the registration phase are transmitted via a secure channel, while the insecure channel is used in the last two phases. And the authentication phase is illustrated in Fig. 2.

### 3.1. Initialization

Each $GW_j$ has an identity $GID_j$ and its own secret key $G_j$, different from other gateways. A gateway may contact some different WMSNs. Finally, $GW_j$ selects a collision-resistant cryptographic hash function $h(.)$ which is used in the WMSN.

### 3.2. Registration

Here we divide this phase into two subphases:

#### 3.2.1. Sensor registration

Each sensor node $SN_k$ has an identity $SID_k$. Generally $SN_k$ belongs to a special network region. Suppose that $SN_k$ is in the sensor set numbered $N_l$ contacting with the gateway $GW_j$. Here we use the number $N_l$ to tell apart different sensor suites. The common secret key between $GW_j$ and $SN_k$ is $SG_k = h(SID_k\|G_j\|N_l)$. Then the information $\{SID_k, SG_k, GID_j\}$ is injected to $SN_k$ and the sensor is

placed to the suitable position to work. $GW_j$ also stores $(SID_k, N_l)$ in its database.

If a sensor node is broken and a new one should be changed, the above operations can be done and after that the old one can be replaced.

#### 3.2.2. User registration

$U_i$ has his/her own mobile device to store the information about registration in different WMSNs. Here we give an example that $U_i$ registers on a gateway $GW_j$. If $U_i$ registers on other gateways, similar steps will happen.

- Step 1: $U_i$ selects $ID_i$ and $PW_i$ and a nonce $r_0$, computes $HPW_i = h(PW_i \parallel r_0)$ and sends $\{ID_i, HPW_i\}$ to $GW_j$ via a secure channel.
- Step 2: $GW_j$ checks if $ID_i$ exists in the database. If so, it sends a denial notification to $U_i$. Otherwise, $GW_j$ stores $ID_i$ in database, selects a pseudo-identity $CID_i$ and computes $A_1 = h(CID_i \| GID_j \| G_j) \oplus HPW_i$ and $A_2 = h(ID_i \parallel G_j) \oplus h(ID_i \parallel HPW_i)$. Then it sends $\{A_1, A_2, CID_i, GID_j\}$ to $U_i$ via a private channel.
- Step 3: $U_i$ stores $(A_1, A_2, A_3 = h(ID_i \parallel PW_i) \oplus r_0, CID_i, GID_j)$ into his/her mobile device.

### 3.3. Authentication

- Step 1: First $U_i$ selects the WMSN which he/she wants to access, and inputs $ID_i$ and $PW_i$. The mobile device computes $r_0 = A_3 \oplus h(ID_i \parallel PW_i)$ and $HPW_i = h(PW_i \parallel r_0)$. Then it selects a random number $r_u$ and the required sensor $SID_k$, and calculates $B_1 = A_1 \oplus HPW_i = h(CID_i \| GID_j \| G_j)$, $B_2 = B_1 \oplus r_u$, $B_3 = ID_i \oplus h(r_u \parallel B_1)$ and $B_4 = h(CID_i \parallel GID_j \parallel SID_k \parallel B_1 \parallel ID_i \parallel r_u)$. Finally the message $M_1 = \{CID_i, GID_j, SID_k, B_2, B_3, B_4\}$ is sent to $GW_j$.
- Step 2: $GW_j$ first checks if $GID_j$ is right and $SID_k$ is in the WMSN. If so, $GW_j$ computes $B_1 = h(CID_i \| GID_j \| G_j)$, $r_u = B_1 \oplus B_2$ and $ID_i = B_3 \oplus h(r_u \parallel B_1)$ and checks if $ID_i$ is in the database and $B_4 \stackrel{?}{=} h(CID_i \parallel GID_j \parallel SID_k \parallel B_1 \parallel ID_i \parallel r_u)$. If either of the two verifications does not pass three times in a short time span, the user account $ID_i$ will be locked. Otherwise, $GW_j$ searches $(SID_j, N_l)$ from the database, selects a nonce $r_g$, and computes $SG_k = h(SID_k \| G_j \| N_l)$, $B_5 = h(SG_k \parallel GID_j) \oplus r_u$, $B_6 = h(r_u) \oplus r_g$ and $B_7 = h(SG_k \| r_u \| r_g)$. Then the message $M_2 = \{SID_k, B_5, B_6, B_7\}$ is sent to $SN_k$.
- Step 3: $SN_k$ first checks if $SID_k$ is correct. If so, it calculates $r_u = B_5 \oplus h(SG_k \parallel GID_j)$ and $r_g = B_6 \oplus h(r_u)$, and verifies $B_7 \stackrel{?}{=} h(SG_k \| r_u \| r_g)$. If the equation is right, $SN_k$ produces $r_s$ and computes $sk_s = h(r_u \| r_g \| r_s)$, $B_8 = h(SG_k \parallel r_g) \oplus r_s$ and $B_9 = h(sk_s \parallel SID_k \parallel GID_j \parallel r_s)$. Then $M_3 = \{B_8, B_9\}$ is sent to $GW_j$.
- Step 4: $GW_j$ computes $r_s = B_8 \oplus h(SG_k \parallel r_g)$ and $sk_g = h(r_u \| r_g \| r_s)$, and judges $B_9 \stackrel{?}{=} h(sk_g \parallel SID_k \parallel GID_j \parallel r_s)$. If so, a novel pseudo-identity $CID_i^{new}$ is generated by $GW_j$ and the following data are calculated: $B_{10} = h(CID_i^{new} \parallel GID_j \parallel G_j) \oplus h(r_u \parallel CID_i)$, $B_{11} = h(r_u \parallel ID_i) \oplus r_g$, $B_{12} = h(r_u \parallel r_g) \oplus r_s$, $B_{13} = h(h(ID_i \parallel G_j) \parallel r_s) \oplus CID_i^{new}$ and $B_{14} = h(sk_g \parallel ID_i \parallel B_{10} \parallel CID_i^{new})$. Then $M_4 = \{B_{10}, B_{11}, B_{12}, B_{13}, B_{14}\}$ is sent to $U_i$.
- Step 5: When receiving $M_4$, the mobile device calculates $r_g = B_{11} \oplus h(r_u \parallel ID_i)$, $r_s = B_{12} \oplus h(r_u \parallel r_g)$, $sk_u = h(r_u \| r_g \| r_s)$ and $CID_i^{new} = B_{13} \oplus h((A_2 \oplus h(ID_i \| HPW_i)) \| r_s)$, and checks $B_{14} \stackrel{?}{=} h(sk_u \parallel ID_i \parallel B_{10} \parallel CID_i^{new})$. If so, the mobile device computes $A_1^{new} = B_{10} \oplus h(r_u \parallel CID_i) \oplus HPW_i$ and replaces $(A_1, CID_i)$ with $(A_1^{new}, CID_i^{new})$.

```
(*–channels–*)
free ch: channel.
free sch0: channel [private].
free sch1: channel [private].

(*–session keys–*)
free sku: bitstring [private].
free sks: bitstring [private].
free skg: bitstring [private].

(*–GW's secret key–*)
free Gj:bitstring [private].

(*–constants–*)
free IDi:bitstring [private].
free PWi:bitstring [private].
const SIDk:bitstring.
const GIDj:bitstring.
const Nl:bitstring.
table d(bitstring).
```

```
(*–functions–*)
fun h(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.

(*–equations–*)
equation forall m:bitstring,n:bitstring;
xor(xor(m,n),n)=m.

(*–queries–*)
query attacker(sku).
query attacker(sks).
query attacker(skg).
query id:bitstring; inj-event(UserAuth(id))
 ==> inj-event(UserStart(id)).

(*—event–*)
event UserStart(bitstring).
event UserAuth(bitstring).
```

**Fig. 3.** Premises for the code.

### 3.4. Password change

If $U_i$ wants to change his/her password, he/she must contact the corresponding gateway to finish the authentication.

- Step 1: $U_i$ selects the required $N_j$, and inputs $ID_i$ and $PW_i$. The mobile device computes $r_0 = A_3 \oplus h(ID_i \parallel PW_i)$ and $HPW_i = h(PW_i \parallel r_0)$, chooses a nonce $r_u$, and calculates $B_1, B_2, B_3$ and $B_{15} = h(CID_i \| GID_j \| B_1 \| ID_i \| r_u)$. Finally it sends $M_5 = \{CID_i, GID_j, B_2, B_3, B_{15}\}$ with a request of changing password to $GW_j$.
- Step 2: $GW_j$ computes $B_1$ and $ID_i$ and searches $ID_i$ in database. If it exists, $GW_j$ computes $r_u$ and checks $B_{15} \stackrel{?}{=} h(CID_i \| GID_j \| B_1 \| ID_i \| r_u)$. $GW_j$ selects a new pseudo-identity $CID_i^{new}$ for $U_i$, and computes $B_{16} = h(CID_i^{new} \| GID_j \| G_j) \oplus h(r_u \parallel CID_i)$, $B_{17} = h(h(ID_i \parallel G_j) \parallel r_u) \oplus CID_i^{new}$ and $B_{18} = h(ID_i \| CID_i \| CID_i^{new} \| B_1 \| B_{16})$. Finally it sends $M_6 = \{B_{16}, B_{17}, B_{18}\}$ to $U_i$ with a permission.
- Step 3: After $U_i$ receives $M_6$, the mobile device calculates $CID_i^{new} = B_{17} \oplus h((A_2 \oplus h(ID_i \| HPW_i)) \| r_u)$ and checks $B_{18} \stackrel{?}{=} h(ID_i \| CID_i \| CID_i^{new} \| B_1 \| B_{16})$. If so, $U_i$ inputs a new password $PW_i^{new}$ and the mobile device generates a new nonce $r_0^{new}$, and computes $HPW_i^{new} = h(PW_i^{new} \parallel r_0^{new})$, $A_1^{new2} = B_{16} \oplus h(r_u \parallel CID_i) \oplus HPW_i^{new}$, $A_2^{new} = A_2 \oplus h(ID_i \parallel HPW_i) \oplus h(ID_i \parallel HPW_i^{new})$ and $A_3 = h(ID_i \parallel PW_i^{new}) \oplus r_0^{new}$. Finally the mobile device replaces $(A_1, A_2, A_3, CID_i)$ with $(A_1^{new2}, A_2^{new}, A_3^{new}, CID_i^{new})$.

## 4. Formal verification

Proverif is a famous tool to verify the cryptographic protocols. It can check the security properties containing authentication, equivalences between processes and secrecy. The modeled attacker can govern the public channel and control all data in it. We make the code for our scheme to test the security. First the premises including public and private channels, session keys, secret key of the gateway, constants, functions, equations, queries and events are illustrated in Fig. 3. We explain some of the commands which may make people confused. ch is the public channel and all authentication messages go through it. sch0 and sch1 are private channels, which are for user registration and sensor registration, respectively. h, xor and con denote for hash function, x-or exclusive

```
let User=
new r0:bitstring;
let HPWi=h(con(PWi,r0)) in
out(sch0,(IDi,HPWi));
in(sch0,(A1:bitstring,A2:bitstring,CIDi:bitstring));
let A3 = xor(h(con(IDi,PWi)),r0) in

--------------------------------------------------
!
(
event UserStart(IDi);
new uru:bitstring;
let ur0 = xor(A3,h(con(IDi,PWi))) in
let uHPWi = h(con(PWi,ur0)) in
let uB1 = xor(A1,uHPWi) in
let uB2 = xor(uB1,uru) in
let uB3 = xor(IDi,h(con(uru,uB1))) in
let uB4 = h(con(con(con(con(con(CIDi,GIDj),SIDk),uB1),IDi),uru)) in
let M1 = (CIDi,uB2,uB3,uB4) in
out(ch,M1);
in (ch,(uB10:bitstring,uB11:bitstring,uB12:bitstring,uB13:bitstring,uB14:bitstring));
let urg= xor(uB11,h(con(uru,IDi))) in
let urs = xor(uB12,h(con(uru,urg))) in
let sku = h(con(con(uru,urg),urs)) in
let CIDinew = xor(uB13,h(con(xor(A2,h(con(IDi,uHPWi))),urs))) in
if uB14 = h((con(con(sku,IDi),uB10),CIDinew)) then
let A1new = xor(xor(uB10,h(con(uru,CIDi))),uHPWi) in
let A1 = A1new in
let CIDi = CIDinew in
0
).
```

```
let Sensor =
in(sch1,SGk:bitstring);

--------------------------------------------------
!
(
in(ch,(sB5:bitstring,sB6:bitstring,sB7:bitstring));
new srs:bitstring;
let sru = xor(sB5,h(con(SGk,GIDj))) in
let srg = xor(sB6,h(sru)) in
if sB7 = h(con(con(SGk,sru),srg)) then
let sks = h(con(con(sru,srg),srs)) in
let B8 = xor(h(con(SGk,srg)),srs) in
let B9 = h(con(con(con(sks,SIDk),GIDj),srs)) in
let M3 = (B8,B9) in
out(ch,M3);
0
).
```

```
let GWReg1 =
in(sch0,(gIDi:bitstring,gHPWi:bitstring)); new gCIDi:bitstring;
let gA1 = xor(h(con(con(gCIDi,GIDj),Gj)),gHPWi) in
let gA2 = xor(h(con(gIDi,Gj)),h(con(gIDi,gHPWi))) in
insert d(gIDi);
out (sch0,(gA1,gA2,gCIDi)).

--------------------------------------------------
let GWReg2 =
let gSGk = h(con(con(SIDk,Gj),Nl)) in
out(sch1,gSGk).

--------------------------------------------------
let GWAuth =
in (ch,(gCIDi2:bitstring,gB2:bitstring,gB3:bitstring,gB4:bitstring));
new grg: bitstring;
let gB1 = h(con(con(gCIDi2,GIDj),Gj)) in
let gru = xor(gB1,gB2) in
let gIDi2 = xor(gB3,h(con(gru,gB1))) in
get d(=gIDi2) in
let gru = xor(gB3,gIDi2) in
if gB4 = h(con(con(con(con(gCIDi2,GIDj),SIDk),gB1),gIDi2),gru)) then
event UserAuth(gIDi2);
let gSGk = h(con(con(SIDk,Gj),Nl)) in
let gB5 = xor(h(con(gSGk,GIDj)),gru) in
let gB6 = xor(h(gru),grg) in
let gB7 = h(con(con(gSGk,gru),grg)) in
let M2= (gB5,gB6,gB7) in
out(ch,M2);
in (ch,(gB8:bitstring,gB9:bitstring));
new gCIDinew:bitstring;
let grs = xor(gB8,h(con(gSGk,grg))) in
let skg = h(con(con(gru,grg),grs)) in
if gB9 = h((con(con(skg,SIDk),GIDj),grs)) then
let gB10 = xor(h(con(con(gCIDinew,GIDj),Gj)),h(con(gru,gCIDi2))) in
let gB11 = xor(h(con(gru,gIDi2)),grg) in
let gB12 = xor(h(con(gru,grg)),grs) in
let gB13 = xor(h(con(h(con(gIDi2,Gj)),grs)),gCIDinew) in
let gB14 = h(con(con(con(skg,gIDi2),gB10), gCIDinew)) in
let M4 = (gB10,gB11,gB12,gB13,gB14) in
out(ch,M4).

--------------------------------------------------
let GW = GWReg1 | GWReg2 | GWAuth.
```

**Fig. 4.** Code of our scheme with Proverif.

and string concatenation, respectively. The first three queries are to test the security of the session keys and the last is for the order of the two events below it.

The processes of the user, the sensor and the gateway are demonstrated in Fig. 4. We explain them simply below:

- In part of "Process of user", line 2 to 6 in first blank denote the process of user part in Section 3.2.2. The lines in the second blank are for the authentication phase in Section 3.3. In the registration part, all variables are same as the scheme. And in the authentication part, all received and calculated variables have a prefix "u-", except A1 and CIDinew, which are for the consistency of replacement.
- In part of "Process of sensor", line 2 of the first blank is the sensor registration in Section 3.2.1. The lines in the second blank are for the sensor part in Section 3.3. In the registration part, all variables are same as the scheme. And in the authentication part, all received and calculated variables have a prefix "s-".
- In part of "Process of gateway", the parts "GWReg1", "GWReg2" and "GWAuth" are for the user registration, sensor registration and authentication operations of gateway, respectively. All received and calculated variables have a prefix "g-". The command in the last blank is the whole definition of gateway, which consists of the above three parts.

```
RESULT inj-event(UserAuth(id))
==> inj-event(UserStart(id)) is true.
RESULT not attacker(skg[]) is true.
RESULT not attacker(sks[]) is true.
RESULT not attacker(sku[]) is true.
```

**Fig. 5.** Results of the code.

Furthermore, the command *process* !*User*|!*GW*|!*Sensor* is the drive of the entire program. The results are demonstrated in Fig. 5.

The first result shows that the two events *UserStart*(*id*) and *UserAuth*(*id*) can be executed in a normal order. The rest three results demonstrate that the session keys are secure against the various attacks simulated by Proverif.

## 5. Security and performance comparison

We analyze ten security properties below, and compare our scheme with five very recent schemes [3,4,36,37,39]. Readers can check the concrete papers for the analysis below. The comparison is in Table 2. In this table, the symbol ✓ denotes that the scheme meets the property. On the contrary, the symbol × is used.

### 5.1. Resistance to the insider attack

This attack means that the password of user may be guessed or obtained via someway. Here $U_i$ submits $HPW_i = h(PW_i \| r_0)$ to $GW_j$ in registration. The administrator of $GW_j$ cannot guess $PW_i$ since it is protected by a hash function with a random number $r_0$ mixed.

### 5.2. Resistance to the off-line guessing attack

According to Section 2.2, if $\mathcal{A}$ obtains the messages $M_1^{old}$, $M_2^{old}$, $M_3^{old}$, $M_4^{old}$ in the last session from the public channel and $(A_1, A_2, A_3, CID_i)$ from $U_i$'s mobile device, he/she can guess $(\overline{ID}, \overline{PW})$ and calculate $\overline{r_0} = A_3 \oplus h(\overline{ID} \| \overline{PW})$ and $\overline{HPW} = h(\overline{PW} \| \overline{r_0})$. The following two equations can be applied: $B_{10}^{old} = A_1 \oplus h(r_u^{old} \| CID_i^{old}) \oplus \overline{HPW}$ and $B_3^{old} = \overline{ID} \oplus h(r_u^{old} \| B_1^{old})$. We can see $r_u^{old}$ appears in the above two equations, but the calculation of $r_u^{old}$ requires $A_1^{old}$: $A_1^{old} \oplus \overline{HPW} \oplus r_u = B_2^{old}$. Unfortunately, this element diminished and $\mathcal{A}$ cannot finish the attack.

Moreover, we should mention that schemes in [3,4,36,37,39] cannot resist this attack. Here we use the symbols in the above schemes below. First we suppose that $\mathcal{A}$ gets the old session messages and then the information in $U_i$'s smart card, and guesses $(\overline{ID}, \overline{PW})$.

- In [36], $\mathcal{A}$ calculates $\overline{b} = HB_i \oplus h(\overline{PW} \| \overline{ID})$, $\overline{CID} = h(\overline{ID} \| \overline{b})$, $\overline{HPW} = h(\overline{PW} \| \overline{b})$, $\overline{B_1^{old}} = A_3 \oplus h(\overline{CID} \| \overline{HPW})$ and $\overline{r_u} \| \overline{T_1} = B_3^{old} \oplus \overline{B_1^{old}}$, and checks $B_2^{old} \stackrel{?}{=} h(\overline{T_1} \| \overline{r_u} \| \overline{B_1^{old}} \| GID_j)$ until he/she obtains the correct pair $(ID_i, PW_i)$.
- In [4], $\mathcal{A}$ can calculate $\overline{HPW} = h(\overline{ID} \oplus \overline{PW})$ and $\overline{R_i} = A_i \oplus \overline{HPW}$ and checks $CID_i^{old} \stackrel{?}{=} \overline{ID} \oplus h(TID_i^{old} \| \overline{R_i} \| T_1^{old})$ until the right pair $(ID_i, PW_i)$ appears.
- In [39], $\mathcal{A}$ can calculate $\overline{K_{ug}} = K_{ug}^* \oplus h(h(\overline{ID}) \oplus h(\overline{PSW}))$ (Here $PSW$ is the password, or $PW$ we have defined) and $\overline{N_u} = N_x^{old} \oplus \overline{K_{ug}}$, and checks $AID^{old} \stackrel{?}{=} h(\overline{ID} \| \overline{K_{ug}} \| \overline{N_u} \| Ts_{ug}^{old})$ until the right pair $(ID_i, PW_i)$ appears.
- In [3], $\mathcal{A}$ can calculate $\overline{r_i} = E_2 \oplus h(\overline{ID} \| \overline{PW})$ and $HPW_i = h(\overline{ID} \| \overline{PW} \| \overline{r_i})$. Then he decrypts $C_{ig}^{old}$ and gets $\overline{HID_i}$, $\overline{r_g}$, $\overline{ID_{SN_j}}$, $\overline{SK_{SN_j}}$ and $\overline{ID_g}$. $\mathcal{A}$ first checks $\overline{HID_i} \stackrel{?}{=} h(\overline{ID} \oplus \overline{r_i})$, and computes $\overline{A_1} = h(C_{ig} \| \overline{r_g} \| \overline{HPW_i} \| \overline{ID_{SN_j}} \| \overline{SK_{SN_j}})$ and $\overline{A_2} = h(\overline{ID_g} \| \overline{A_1} \| \overline{ID_{SN_j}} \| \overline{HPW_i} \| C_{ig}^{old})$. Finally $\mathcal{A}$ can encrypt $\overline{A_2}$ with $\overline{A_1}$ and check if the result equals to $CID_i^{old}$. $\mathcal{A}$ could repeat the process until correct pair $(ID_i, PW_i)$ appears.
- In [37], $\mathcal{A}$ can calculate $Q_i = h(\overline{ID} \| \overline{PW} \| r_i)$, $R_i = I_1^{old} \oplus D1$, and $\overline{I_2} = Q_i \oplus R_i$. Then he decrypts $I_4$ with $\overline{I_2}$ and gets $\overline{ID_i}$, $\overline{TE_i}$, $\overline{TC_i}$, $\overline{h(Q_i)}$, $\overline{T_3}$ and $\overline{I_3}$. $\mathcal{A}$ can check $\overline{ID_i} \stackrel{?}{=} \overline{ID}$, $\overline{h(Q_i)} \stackrel{?}{=} h(\overline{Q_i})$ and $\overline{TC_i} \stackrel{?}{=} TC_i$ until he gets the correct $(ID_i, PW_i)$ pair.

### 5.3. Resistance to the user forgery attack

If $\mathcal{A}$ wants to forge $M_1$, he/she must calculate legal elements including $B_2$, $B_3$ and $B_4$. But any of them needs the secret key $G_j$. Moreover, only three trials can be made, or $ID_i$ will be frozen. So it is impossible for $\mathcal{A}$ to succeed in this attack.

### 5.4. Resistance to the sensor capture attack

If $\mathcal{A}$ hijacks some sensors, other than $SN_k$, which communicates with $U_i$, he/she could not forge $M_3$ since $SG_k$ is used to construct $B_8$. Any other sensor's shared key with $GW_j$ has no relation with $SG_k$. So even other sensors are captured, $\mathcal{A}$ could not do this attack successfully.

### 5.5. Resistance to the gateway forgery attack

This attacks means that $\mathcal{A}$ has the ability to forge the message $M_2$ or $M_4$. We illustrate them below. To forge the message $M_2$, $SG_k$ is a critical element required to calculate $B_5$ and $B_7$. Moreover, both $B_{10}$ and $B_{14}$ in $M_4$ need $G_j$. Unluckily, according to Section 2.2, the two secret strings are hard to get by $\mathcal{A}$.

### 5.6. Resistance to the de-synchronization attack

Two cases can lead to this attack: the first is that there is at least one consistent string stored in both the gateway and the user, and the second is that there is no checking mechanism in the password change phase. If the attack blocks some message between the gateway and the user, the first case for attack will appear. The second is that if the user inputs a wrong old password in the password change phase, this may make the authenticated data from $U_i$ be different from $GW_j$'s calculation. Our scheme can avoid the two cases perfectly, but in [36], if the last confirmation message is blocked, or lost due to time delay, the gateway will not replace the pair $(TID_i, D_i)$, and such data will be inconsistent between $GW_j$ and $U_i$.

### 5.7. Resistance to the user tracking attack

In each session, $U_i$ uses a different temporary pseudo-identity generated by the gateway in $M_1$. $\mathcal{A}$ cannot track any user by any same element in $M_1$. But in [36], every time $r_2$ is sent to the gateway without change. Though researchers claim that the pseudo-identity varies in different session, $\mathcal{A}$ can track $r_2$ as a pseudo-identity in sessions.

### 5.8. Resistance to the session key disclosure attack

It is clear that any of the random numbers $r_u$, $r_s$ or $r_g$ cannot be calculated by $\mathcal{A}$, and then the session key cannot be computed or leaked. Moreover, we should mention that in [36], the scheme has design flaws and the session key may be leaked. The main problem is that researchers have not considered the lengths of the hash results and random numbers. Generally speaking, the two kinds of data are supposed to be secure against guessing attack by $\mathcal{A}$. So they should reach the secure length. We can consider

**Table 2**
Comparison of security properties and performance.

| Properties | [36] | [4] | [39] | [3] | [37] | Ours |
|---|---|---|---|---|---|---|
| Resistance to the insider attack | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Resistance to the off-line guessing attack | × | × | × | × | × | ✓ |
| Resistance to the user forgery attack | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Resistance to the sensor capture attack | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Resistance to the gateway forgery attack | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Resistance to the de-synchronization attack | ✓ | × | ✓ | ✓ | ✓ | ✓ |
| Resistance to the user tracking attack | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| Resistance to session key disclosure | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| Session key constructed by three parties | ✓ | ✓ | × | × | × | ✓ |
| Mutual authentication | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Table 3**
Performance comparison.

| Properties | [36] | [4] | [39] | [3] | [37] | Ours |
|---|---|---|---|---|---|---|
| Time for User ($\mu s$) | $10T_h$ $= 0.328$ | $12T_h$ $= 0.3936$ | $7T_h$ $= 0.2296$ | $2T_s + 8T_h$ $= 43.2294$ | $2T_c + 2T_s + 4T_h$ $= 254127.0982$ | $11T_h$ $= 0.3608$ |
| Time for gateway ($\mu s$) | $8T_h$ $= 0.2624$ | $19T_h$ $= 0.6232$ | $9T_h$ $= 0.2952$ | $T_s + 4T_h$ $= 21.6147$ | $2T_s + 6T_h$ $= 43.1638$ | $17T_h$ $= 0.5576$ |
| Time for sensor ($\mu s$) | $6T_h$ $= 0.1968$ | $6T_h$ $= 0.1968$ | $3T_h$ $= 0.984$ | $2T_s + 4T_h$ $= 43.0982$ | $2T_c + 3T_h$ $= 254084.0984$ | $6T_h$ $= 0.1968$ |
| Communication cost (bits) | 3040 | 2720 | 2400 | 5344 | 5248 | 2720 |
| Practicality | Yes | Yes | No | No | Yes | Yes |

that they are as long as each other. So in [36], when $\mathcal{A}$ gets all messages in the public channel, he/she can get $r_u$ from $B_3$ since $B_3 = (r_u \parallel T_1) \oplus h(A_1\|r_s\|G_j)$ and only $T_1$ can be masked by the hash results. Similarly, $r_u$ can be obtained from $B_5$ and $r_s$ can be got from $B_8$. Then $\mathcal{A}$ can compute $A_1 = B_6 \oplus h(SID_k\|h(r_g)\|r_u)$ and the session key can be calculated with $h(A_1 \parallel r_u \parallel r_g \parallel r_s)$.

### 5.9. Session key constructed by three parties

All can see that the session key in our scheme is built on the random numbers produced by the three participants, respectively. So the proposed scheme fits for this character. However, in [39], the session key is generated only by the gateway node. It is not flexible enough. Also, in [3,37], only the user and the sensor construct the session key.

### 5.10. Mutual authentication

We can divide the case as follows: $GW_j$ checks $ID_i$ and $B_4$ to verify $U_i$, and $B_9$ to verify $SN_k$; $SN_k$ checks $SID_k$ and $B_7$ to authenticate $GW_j$ directly and $U_i$ indirectly; $U_i$ checks $B_{14}$ to verify $GW_j$ directly and $SN_k$ indirectly. So either pair of parties reaches mutual authentication.

### 5.11. Performance comparison

The relative content is shown in Table 3. We compare the time cost of authentication phase of the five recent schemes. $T_c$, $T_s$ and $T_h$ denote the time for one chaotic map function, one symmetric encryption/decryption and one hash function, respectively. Here Sha1 is used to test $T_h$. We use the test platform in [43], and get $T_c = 127\,042$ $\mu$s, $T_s = 21.4835$ $\mu$s and $T_h = 0.0328$ $\mu$s. We suppose that the random numbers, timestamps, hash results and identities of gateway and sensors in all schemes reach the length 160 bits, which is secure. And the chaotic map result is 1024 bits long, since the modular prime should be a secure one. Here we should mention that in [36], the basis of the communication cost is ridiculous. They give a hypothesis that the identities of entities, timestamps and random numbers are 32 bits, or only four Bytes. It is hard to register an identity less than six characters nowadays. And since DES is known to be insecure [47], even 56 bits cannot be considered as a secure length. We can see that our scheme costs higher than [36,39] for the user and gateway time, but better than the other three. For the sensor time cost, our scheme takes more time than [39], equals to [4,36], and is better than [3,37]. Moreover, our scheme costs more than [39] for the communication, and same as [4], but less than others.

Above all, our scheme is lightweight and secure and it is suitable to put into practice. But the two schemes in [3,39] are not practical completely. In [39], the de-synchronization attack is easy to happen, since the packet loss phenomenon is usual in wireless circumstance. Even if no attacker blocks the packets, once the loss of packet sent from sensor to gateway or from gateway to user occurs, the de-synchronization appears. It means that the WSN system could be paralytic without any attacker when data transmission turns to be more. Although some steps which prevent such attack, it is troublesome for the recovery, especially between the sensor and the gateway. Moreover, in [3], when user registers, the gateway only sends the critical string $C_{ig}$ including one sensor information $ID_{SN_j}$. It means that the user could only access one fixed sensor. And the architecture has energy waste problem for sensors, which we have demonstrated in Section 1.1.

## 6. Simulation with NS-3

NS-3 [48] is a hot tool which is for doing discrete-event network simulations. It is an open source platform. Models such as TCP/UDP protocols, 6LoPAN module and WiFi module, etc., are supported to the user for completing experiments. NS-3 includes many libraries providing core and margin functions in many different sorts of networks. Users can work with C++/Python tools to get the results.

### 6.1. Simulation circumstance

The basic parameter values for simulation are demonstrated in Table 4. Here the sensors are disposed in a rectangle. The distance between two sensors is 20 m, whatever in row or in column. The quantity of sensors increases with the step 20, and reaches 120 at last. In order to simplify the topology, one user, one gateway and the mentioned sensors are included in the simulation. Moreover, according to the scheme and 802.11ah standard which is used in WSN [49], we employ part of the values in [50] to complete our

**Table 4**
Parameters for simulation.

| Parameter | Value |
|---|---|
| Operating system | Ubuntu 16.10 |
| Area of distributed sensors | 400 × 100 $m^2$ |
| Distance between the neighbor sensors | 20 m |
| Bandwidth | 2 Mbps |
| Simulation time | 1800 s |

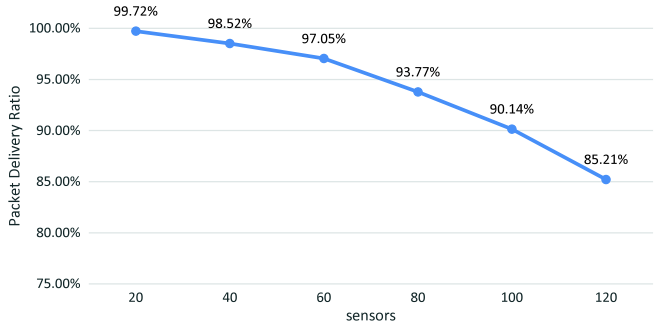


**Fig. 6.** Packet delivery ratio.

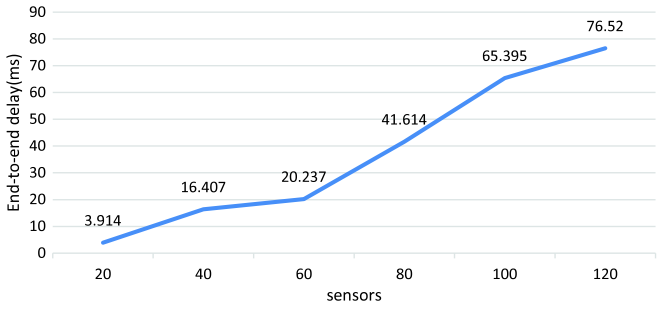


**Fig. 7.** End-to-end delay.

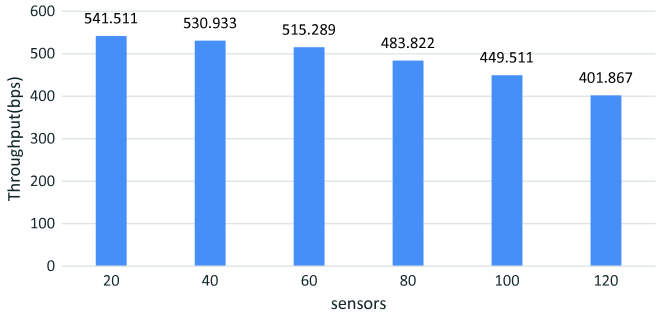


**Fig. 8.** Throughput.

code. The messages $M_1$, $M_2$, $M_3$ and $M_4$ have 960, 640, 320 and 800 bits, respectively. We make the user send packet every 5 s in each scenario, considering the sensor selection operations. Then we test three indexes including packet delivery ratio (PDR), end-to-end delay (E2ED) and throughput (TH). The results are in Figs. 6, 7 and 8, respectively.

### 6.2. Effect on packet delivery ratio

PDR is an important index to test the performance of network communication. It is the ratio of received number to sent number for all packets. From Fig. 6, we see that the PDR becomes lower when the sensors turns to be more, and the scope is relatively high, more than 85% until the situation of 120 sensors. The reason is that more congestion occurs if more nodes are involved in the simulation. And in NS-3, packets sent to wireless circumstance are with energy. The energy is spoiled when the packet is transmitted in channel according to the error model. According to the energy threshold set in the recipient node in the model, the packet may be lost when transmission distance is so far.

### 6.3. Effect on end-to-end delay

E2ED is the average time of packet transmission from source to destination. Or the formula $E2ED = \frac{\sum_{i=1}^{n}(Tr_i - Ts_i)}{n}$ can be used. $i$ is the number of packet. $n$ is the total number of received packets. $Tr_i$ is the received time of $i$th packet while $Ts_i$ is the sent time. From Fig. 7, we see that E2ED increases when number of sensors increases. Since more sensors are increasing step by step, the total distance and more congestion rise. So E2ED turns to be longer when such variation appears.

### 6.4. Effect on throughput

TH is an important index which describes the transmitted bits in one second, and can be denoted as $TH = \frac{\sum(Nr_i \times L_i)}{T}$. $Nr_i$ is the number of received packets belonging to the $i$th sort. $L_i$ is the length of such packet. $T$ is the total time. The result is bit per second (bps). From Fig. 8, the variation is same as PDR. The reason is that based on the fixed total number of packets, it is no doubt that less data are exchanged with increasing number of sensors and decreasing PDR.

## 7. Conclusion

In this paper, we propose a novel two-factor authentication scheme to provide security in WMSNs for PHSs. Mutual authentication exists in either two participants protects the data transferred in the hazard wireless circumstance away from various of adversaries. We can see the protocol achieves security requirements with low time and communication cost. That is suitable for data transmission for PHSs. Verification code with Proverif and informal analysis demonstrate that the scheme withstands various sorts of attacks and keeps security characters. Also, the performance and simulation with NS-3 denote that our scheme has high efficiency for practice. The above illustration shows that the new protocol can be applied in the e-healthcare for PHSs.

## Acknowledgments

The authors thank the anonymous reviewers for their valuable comments. This research is supported by University Distinguished Young Research Talent Training Program of Fujian Province (Year 2016), Fujian Education and Scientific Research Program for Young and Middle-aged Teachers under Grant No. JAT160633, and Teaching team construction project at Xiamen Institute of Technology under grant no. TD2017006. Dr. Xiong Li is supported by National Natural Science Foundation of China under Grant No. 61300220 & 61772194, and the Scientific Research Fund of Hunan Provincial Education Department under Grant no. 16B089. Dr. Saru Kumari is sponsored by the University Grants Commission, India through UGC-BSR Start-up grant under Grant no. 3(A)(60)31.

## References


[1] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, S.-S. Yeo, Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks, Multimedia Syst. 21 (1) (2015) 49–60. http://dx.doi.org/10.1007/s00530-013-0346-9.

[2] F. Wu, L. Xu, S. Kumari, X. Li, An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks, Multimedia Syst. 23 (2) (2017) 195–205.

[3] J. Srinivas, D. Mishra, S. Mukhopadhyay, A mutual authentication framework for wireless medical sensor networks, J. Med. Syst. 41 (5) (2017) 80.

[4] R. Amin, S.H. Islam, G. Biswas, M.K. Khan, N. Kumar, A robust and anonymous patient monitoring system using wireless medical sensor networks, Future Gener. Comput. Syst. (2016). http://dx.doi.org/10.1016/j.future.2016.05.032.

[5] R. Amin, S.H. Islam, G. Biswas, M.K. Khan, L. Leng, N. Kumar, Design of anonymity preserving three-factor authenticated key exchange protocol for wireless sensor network, Comput. Netw. 101 (2016) 42–62.

[6] C. Yuan, X. Sun, R. Lv, Fingerprint liveness detection based on multi-scale lpq and pca, China Commun. 13 (7) (2016) 60–65.

[7] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, K. Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing, IEEE Trans. Inf. Forensics Secur. 11 (11) (2016) 2594–2608.

[8] Z. Fu, K. Ren, J. Shu, X. Sun, F. Huang, Enabling personalized search over encrypted outsourced data with efficiency improvement, IEEE Trans. Parallel Distrib. Syst. 27 (9) (2016) 2546–2559.

[9] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. 98 (1) (2015) 190–200.

[10] X. Li, J. Ma, W. Wang, Y. Xiong, J. Zhang, A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments, Math. Comput. Modelling 58 (1) (2013) 85–95.

[11] X. Li, J. Niu, M.K. Khan, J. Liao, An enhanced smart card based remote user password authentication scheme, J. Netw. Comput. Appl. 36 (5) (2013) 1365–1371.

[12] F. Wu, L. Xu, S. Kumari, X. Li, A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks, Comput. Electr. Eng. 45 (2015) 274–285.

[13] F. Wu, L. Xu, S. Kumari, X. Li, A. Alelaiwi, A new authenticated key agreement scheme based on smart cards providing user anonymity with formal proof, Secur. Commun. Netw. 8 (18) (2015) 3847–3863.

[14] R. Amin, G. Biswas, A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks, Ad Hoc Netw. 36 (2016) 58–80.

[15] T. Ma, J. Zhou, M. Tang, Y. Tian, A. Al-dhelaan, M. Al-rodhaan, S. Lee, Social network and tag sources based augmenting collaborative recommender system, IEICE Trans. Inf. Syst. 98 (4) (2015) 902–910.

[16] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, A. Alelaiwi, Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy, Nonlinear Dynam. 83 (2015) 2085–2101.

[17] D. He, N. Kumar, N. Chilamkurti, A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks, Inform. Sci. 321 (2015) 263–277.

[18] D. He, D. Wang, Robust biometrics-based authentication scheme for multi-server environment, IEEE Syst. J. 9 (3) (2015) 816–823.

[19] Q. Jiang, M.K. Khan, X. Lu, J. Ma, D. He, A privacy preserving three-factor authentication protocol for e-Health clouds, J. Supercomput. 72 (10) (2016) 3826–3849.

[20] Q. Jiang, J. Ma, X. Lu, Y. Tian, An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks, Peer-To-Peer Netw. Appl. 8 (6) (2015) 1070–1081. http://dx.doi.org/10.1007/s12083-014-0285-z.

[21] F. Wu, L. Xu, S. Kumari, X. Li, A new and secure authentication scheme for wireless sensor networks with formal proof, Peer-To-Peer Netw. Appl. 10 (1) (2017) 16–30.

[22] F. Wu, L. Xu, S. Kumari, X. Li, A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security, J. Ambient Intell. Hum. Comput. 8 (1) (2017) 101–116.

[23] M.L. Das, Two-factor user authentication in wireless sensor networks, IEEE Trans. Wirel. Commun. 8 (3) (2009) 1086–1090.

[24] T.-H. Chen, W.-K. Shih, A robust mutual authentication protocol for wireless sensor networks, ETRI J. 32 (5) (2010) 704–712.

[25] D. He, Y. Gao, S. Chan, C. Chen, J. Bu, An enhanced two-factor user authentication scheme in wireless sensor networks, Ad Hoc Sensor Wirel. Netw. 10 (4) (2010) 361–371.

[26] M.K. Khan, K. Alghathbar, Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks, Sensors 10 (3) (2010) 2450–2459.

[27] P. Kumar, H.-J. Lee, Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks, in: Wireless Advanced (WiAd), 2011, IEEE, 2011, pp. 241–245.

[28] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, H.-W. Wei, A secured authentication protocol for wireless sensor networks using elliptic curves cryptography, Sensors 11 (5) (2011) 4767–4779.

[29] P. Kumar, S.-G. Lee, H.-J. Lee, E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks, Sensors 12 (2) (2012) 1625–1647.

[30] K. Xue, C. Ma, P. Hong, R. Ding, A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks, J. Netw. Comput. Appl. 36 (1) (2013) 316–323.

[31] A.K. Das, A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks, Peer-To-Peer Netw. Appl. 9 (1) (2016) 223–244.

[32] M. Turkanović, B. Brumen, M. Hölbl, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion, Ad Hoc Netw. 20 (2014) 96–112.

[33] M.S. Farash, M. Turkanović, S. Kumari, M. Hölbl, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment, Ad Hoc Netw. 36, Part 1 (2016) 152–176.

[34] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, M.K. Khan, A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity, Secur. Commun. Netw. 9 (15) (2016) 2643–2655.

[35] F. Wu, L. Xu, S. Kumari, X. Li, J. Shen, K.-K.R. Choo, M. Wazid, A.K. Das, An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in iot deployment, J. Netw. Comput. Appl. 8 (1) (2017) 101–116.

[36] S. Kumari, H. Om, Authentication protocol for wireless sensor networks applications like safety monitoring in coal mines, Comput. Netw. 104 (2016) 137–154.

[37] S. Kumari, X. Li, F. Wu, A.K. Das, H. Arshad, M.K. Khan, A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps, Future Gener. Comput. Syst. 63 (2016) 56–75.

[38] C.-T. Li, C.-Y. Weng, C.-C. Lee, An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks, Sensors 13 (8) (2013) 9589–9603.

[39] P. Gope, T. Hwang, A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks, IEEE Trans. Ind. Electron. 63 (11) (2016) 7124–7132.

[40] B. Blanchet, B. Smyth, ProVerif 1.90: Automatic cryptographic protocol verifier, user manual and tutorial, 2015. URL: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf.

[41] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.

[42] F. Wu, L. Xu, An improved and provable self-certified digital signature scheme with message recovery, Int. J. Commun. Syst. 28 (2) (2015) 344–357.

[43] F. Wu, L. Xu, S. Kumari, X. Li, A.K. Das, M.K. Khan, M. Karuppiah, R. Baliyan, A novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks, Secur. Commun. Netw. 9 (16) (2016) 3527–3542.

[44] D. Wang, P. Wang, On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions, Comput. Netw. 73 (2014) 41–57.

[45] D. Wang, D. He, P. Wang, C.-H. Chu, Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment, IEEE Trans. Dependable Secure Comput. 12 (4) (2015) 428–442.

[46] S. Kumari, M.K. Khan, M. Atiquzzaman, User authentication schemes for wireless sensor networks: A review, Ad Hoc Netw. 27 (2015) 159–194.

[47] W. Stallings, Cryptography and Network Security: Principles and Practice, International Edition: Principles and Practice, Pearson Higher Ed, 2014.

[48] nsnam.org, NS-3.26, 2017. https://www.nsnam.org/ns-3-26/.

[49] Y. Seok, Ieee 802.11ah (wi-fi in 900 mhz license-exempt band) for iot application, 2017. http://www.standardsuniversity.org/e-magazine/august-2016-volume-6/ieee-802-11ah-wi-fi-900-mhz-license-exempt-band-iot-application/ ..

[50] L. Tian, S. Deronne, S. Latré, J. Famaey, Implementation and validation of an ieee 802.11 ah module for ns-3, in: Proceedings of the Workshop on Ns-3, ACM, 2016, pp. 49–56.


**Fan Wu** received the Bachelor of Engineering degree in Computer Science from Shandong University, Jinan, China, in 2003 and received Master of Engineering degree in Computer Software and Theory from Xiamen University, Xiamen, China, in 2008. Now, he is an associate professor in Xiamen Institute of Technology. His current research interests include information security, internet protocols, and network management.

**Xiong Li** now is an associate professor at School of Computer Science and Engineering of the Hunan University of Science and Technology (HNUST), China. He received his masters degree in mathematics and cryptography from Shaanxi Normal University (SNNU), China in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), China in 2012. He has published more than 50 referred journal papers in his research interests, which include cryptography, information security, cloud computing security etc. He has served on TPC member of several international conferences on information security and reviewer for more than 30 ISI indexed journals. He is a winner of Journal of Network and Computer Applications 2015 best research paper award.

**Arun Kumar Sangaiah** has received his Doctor of Philosophy (Ph.D.) degree in Computer Science and Engineering from the VIT University, Vellore, India. He is presently working as an Associate Professor in School of Computer Science and Engineering, VIT University, India. His area of interest includes software engineering, computational intelligence, wireless networks, bio-informatics, and embedded systems. He has authored more than 100 publications in different journals and conference of national and international repute. Moreover, he has carried out number of funded research projects for Indian government agencies. Also, he was registered a one Indian patent in the area of Computational Intelligence. Besides, Dr. Sangaiah is responsible for Editorial Board Member/Associate Editor of various international journals.

**Lili Xu** received the Bachelor degree in Computer Science from Shandong University, Jinan, China in 2004, and received Master degree in Computer Architecture from Xiamen University, Xiamen, China in 2011. Now she is an engineer in Xiamen University. Her current research interests include information processing, and information security.

**Saru Kumari** is currently an Assistant Professor with the Department of Mathematics, C.C.S. University, Meerut, U.P, India. She received Ph.D. degree in Mathematics in 2012 from C.C.S. University, Meerut, Uttar Pradesh, India. She has published 45 papers in international journals and conferences including 30 research publications in SCI indexed journals. Her current research interests include Information Security, Digital Authentication and Security of Wireless Sensor Networks.

**Liuxi Wu** received his master degree in Computer Software and Theory from Xiamen University. Now he is a lecturer in Xiamen Institute of Technology. His current research interests include big data dealing, information system management and network management.

**Jian Shen** received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2007 and the M.E. and Ph.D. degrees in Computer Science from Chosun University, Gwangju, South Korea, in 2009 and 2012, respectively. Since late 2012, he has been a professor in the School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China. His research interests include information security, network security, mobile computing and networking, and ad hoc networks and systems.