Xie *et al. J Wireless Com Network* (2021) 2021:119
https://doi.org/10.1186/s13638-021-02000-7

EURASIP Journal on Wireless
Communications and Networking

RESEARCH Open Access

A secure and privacy-preserving authentication protocol for wireless sensor networks in smart city

Qi Xie*, Keheng Li, Xiao Tan, Lidong Han, Wen Tang and Bin Hu

*Correspondence:
qixie68@126.com
Key Laboratory of Cryptography
of Zhejiang Province,
Hangzhou Normal University,
Hangzhou 311121, China

Abstract

Smart city can improve the efficiency of managing assets and resources, optimize urban services and improve the quality of citizens' life. Wireless sensor networks (WSNs) can solve many problems in smart city, such as smart transportation, smart healthcare and smart energy. However, security and privacy are the biggest challenges for WSN. Recently, Banerjee et al. proposed a security-enhanced authentication and key agreement scheme for WSN, but their scheme cannot resist offline password guessing attack, impersonation attack, and does not achieve session key secrecy, identity unlinkability, and perfect forward secrecy. In order to fix these flaws, a secure and privacy-preserving authentication protocol for WSN in smart city is proposed. We prove the security of the proposed protocol by using applied pi calculus-based formal verification tool ProVerif and show that it has high computational efficiency by comparison with some related schemes.

Keywords: Wireless sensor networks, Authentication, Anonymity, Smart city

1 Introduction

Smart city means to make use of information and communication technology, such as artificial intelligence, Internet of Things and cloud computing, to sense, analyze and integrate the key information of urban operation core system, so as to make intelligent response to various needs including people's livelihood, environmental protection, public safety, urban services, industrial and commercial activities, to create a better city life for mankind and build sustainable communities. Wireless sensor networks (WSNs) are widely used in smart city, such as environmental monitoring, health care, smart grids and surveillance [1–4]. Through Internet of Things devices, users can access any sensor node in WSN. Therefore, the security of wireless sensor networks is getting more and more attention. Authentication is the first step to ensure the correct transmission of information and the security of WSN. A legitimate user can access a legitimate sensor with user anonymity and receive information from the sensors. Therefore, security and privacy are the biggest challenges for WSN, and many protocols are proposed in the last ten years [5–16]. However, these protocols have one or more weaknesses.

In 2004, Watro et al. [9] proposed an authentication protocol for wireless sensor networks based on public key encryption. In order to strengthen the security of the protocol, Das [10] proposed a two-factor authentication protocol using password and smartcard. Khan and Alghathbar [11] proposed a protocol with better performance than Das's protocol. However, the password update phase in their scheme is faulty. Later, Yeh et al. [12] proposed a mutual authentication scheme based on elliptical curve cryptosystem, but it has a higher computation cost. Xue et al. [13] proposed a temporal-credential-based protocol in WSN. However, their scheme cannot resist many attacks, such as stolen smart card attack and impersonation attack. Later, Gope et al. [14] proposed a lightweight two-factor protocol for WSN, but Luo et al. [15] pointed out that their protocol exists several drawbacks and proposed an improved scheme. However, the improved scheme is still insecure. Recently, Turkanović et al. [16] proposed an authentication and key agreement scheme for wireless sensor networks, but Banerjee et al. [17] found that Turkanvic et al.'s scheme cannot resist identity theft attack and eavesdropping attack, and then, Banerjee et al. proposed an improved scheme based on the biometric and smart card.

Banerjee et al. [17] claimed that their scheme can resist various attacks. However, in this paper, we find that their scheme has some weaknesses, it cannot resist offline password guessing attack and impersonation attack and does not achieve session key secrecy, identity unlinkability and perfect forward secrecy. Therefore, we propose a new scheme to overcome the weaknesses of Banerjee et al.'s scheme.

The rest of the paper is structured as follows: Sections 2 and 3 introduce methods and preliminaries. Sections 4 and 5 review the Banerjee et al.'s scheme and present the attacks on their scheme. The proposed scheme, security analysis, results and discussion are given in Sects. 6, 7 and 8. Section 9 is conclusions.

2 Methods

The authentication model for WSN consists of users, sensor nodes and gateway nodes. Sensor nodes collect data from their environment, and users can access and receive data from sensor nodes. Gateway nodes are responsible for authentication between users and sensor nodes. In order to prevent unauthorized users from accessing data stored in sensors nodes, before users access sensor nodes, users and sensors nodes should authenticate each other with the help of gateway nodes and establish session keys to encrypt data transmitted between users and sensors nodes.

The threat assumptions of this model are as follows [18]:

- The adversary can be a user, any registered user can act as an adversary.
- The adversary can intercept or eavesdrop on all communication messages in a public channel, thereby capturing any exchanged messages between a user and gateway or sensor.
- The adversary has the ability to eavesdrop, intercept, modify, or delete the transmitted message.
- The adversary has the ability to obtain all information stored in users' smart cards by using the side channel attack [19].
- An external adversary can also register, login and receive his smart card.

According to above threat assumptions, the proposed protocol for WSN should meet the following security and privacy criteria:

- Mutual authentication and key agreement: user and sensor node should authenticated each other with the help of gateway node and establish session key.
- Anonymity and unlinkability: the protocol protects the user's real identity and the adversary cannot trace the user's activities.
- Password friendly: the user can update and change his/her password freely.
- No password guessing attacks: the protocol can protect the user's password from guessing attack and ensure the adversary cannot verify whether the password is right or not.
- No smart stolen/lost attacks: even if the smart card is lost or stolen, the adversary can obtain all information stored in it, but the adversary cannot attack the protocol successfully.
- Perfect forward secrecy: even if an adversary can compromise long secret keys, he or she still cannot compute the session keys.
- Known session key security: even if an adversary knows session key, the protocol still safety.
- No replay attack: the protocol prevents the adversary from replaying the transmission information to attack the protocol successfully.
- No various known attacks: the protocol can resist various known attacks, such as forgery attacks, impersonation attacks and man-in-the-middle attacks.

3 Preliminaries

In this section, we introduce the elliptic curve cryptosystem, the fuzzy extractor and some notations, which will be used in our protocol.

3.1 Elliptic curve cryptosystem

The elliptic curve cryptosystem (ECC) is widely used to design password-based authentication protocols, which are created by Miller [20] and Koblitz [21], respectively. ECC uses the following formula:

$$y^2 = x^3 + ax + b \bmod p, \quad a, b \in F_p$$

The above equation is ECC on F_p. The following conditions must be met in order to ensure safety:

$$4a^3 + 27b^2 \neq 0$$

We choose P as a base point on F_p, then $xP = \overbrace{P + \cdots + P}^{x}$, xyP is a Diffie–Hellman value based on ECC.

3.2 Fuzzy extractor

It is very difficult for users to lose and steal their biological information. In many protocols, the users' biometric will be taken as an important factor. There is a slight difference in each extraction of biological information, which can be corrected by using fuzzy extraction. The fuzzy extractor consists of two procedures (Gen, Rep) [22, 23]:

$$(\alpha, \beta) = Gen(B), \quad \alpha = Rep\left(B^*, \beta\right)$$

where B is the biometric, and B^* is closed to B. Gen function returns a string $\alpha \in \{0,1\}^k$ and a coadjutant string $\beta \in \{0,1\}^*$. For each biometric B, Gen function outputs a key α and a help data β. For each biometric B^*, Rep function recovers a key α with the help data β.

3.3 Notations

The notations used in the paper are shown in Table 1.

4 Brife review of Banerjee et al.'s scheme

The Banerjee et al.'s scheme [17] has six phases: pre-deployment phase, registration phase, login phase, authentication and key agreement phase, password change phase and dynamic node addition phase. We omit the last two phases.

4.1 Pre-deployment

In this phase, the administrator uses the setup server to establish the environment. The setup server chooses identity SID_j for each sensor node S_j and provides a key $GWNPS_j$ shared with the gateway node GWN. The GWN is also provided with a secret key S_g and stores $\{SID_j,\ GWNPS_j,\ S_g\}$.

Table 1 Notations

Notations	Descriptions
U_i	ith user
USC	The user's smart card
S_j	jth sensor node
GWN	Gateway node
S_g	Secret key of the gateway node
PK_g	Public key of the gateway node
$GWNPS_j$	Secret key of the gateway node shared with the sensor node
$GWNPU_i$	Secret key of the gateway node shared with the user
UID_i	User's identity
$UPWD_i$	User's password
SID_j	Sensor node's identity
BIO_i	User's biological information
T_x	Current timestamp
ΔT	Allowed transmission delay
SK	Shared session key
$E_x()/D_x()$	Encryption or decryption function using x
$h()$	One-way hash function
$BH()$	Bio-hash operation
$\oplus$	Performing XOR operation
$\|\|$	Concatenation operation

4.2 Registration phase

4.2.1 User registration phase

The user U_i chooses his identity UID_i, password $UPWD_i$ and a random number r_i and then calculates $MID_i = h(UID_i||r_i)$ and $MPWD_i = h(UPWD_i||r_i)$. U_i sends $\{MID_i, MPWD_i\}$ to the gateway node GWN through secure channel.

After receiving $\{MID_i, MPWD_i\}$, GWN selects secret key $GWNPU_i$ and calculates $MXIP_i = h(MID_i||MPWD_i) \oplus GWNPU_i$ and $X_i = h(MID_i||S_g)$, and stores $\{MXIP_i, X_i, h()\}$ into the smart card USC and issues it to U_i securely.

After receiving USC, U_i calculates $GWNPU_i = h(MID_i||MPWD_i) \oplus MXIP_i$,$img_x = BH(r_i \oplus BIO_i)$,$V_i = h(GWNPU_i||img_x)$ and $A_i = h(UID_i||UPWD_i) \oplus r_i$. The user appends $\{BH(), V_i, A_i\}$ to the USC.

4.2.2 Sensors registration phase

The sensor node S_j chooses a random number r_j. S_j calculates $MX_j = h(SID_j||r_j||GWNPS_j)$ and $MY_j = r_j \oplus GWNPS_j$. S_j sends $\{SID_j, MX_j, MY_j\}$ to GWN through secure channel.

After receiving $\{SID_j, MX_j, MY_j\}$, GWN calculates $r_j = MY_j \oplus GWNPS_j$ and verifies $MX_j = h(SID_j||r_j||GWNPS_j)$. And then calculates $P_j = h(MX_j||S_g)$. The GWN stores P_j securely in its memory and issues P_j to S_j. S_j stores P_j securely.

4.3 Login phase

The user U_i inputs UID_i, $UPWD_i$ and BIO_i. USC calculates $r_i = h(UID_i||UPWD_i) \oplus A_i$, $img_x = BH(r_i \oplus BIO_i)$, $MID_i = h(UID_i||r_i)$, $MPWD_i = h(UPWD_i||r_i)$ and $GWNPU_i = h(MID_i||MPWD_i) \oplus MXIP_i$. And then USC verifies $V_i ?= h(GWNPU_i||img_x)$. If not, the user U_i re-does it. Otherwise, the smart card USC chooses a random number r_1 and calculates $M_1 = h(X_i||GWNPU_i||r_1)$ and $M_2 = r_1 \oplus X_i$. U_i sends the request message $\{MID_i, M_1, M_2\}$ to sensor node S_j.

4.4 Authentication and key agreement phase

After receiving $\{MID_i, M_1, M_2\}$, S_j chooses a random number r_2 and calculates $M_3 = P_j \oplus r_2$ and $M_4 = h(GWNPS_j||M_2||r_2)$. S_j sends $\{SID_j, MID_i, M_1, M_2, M_3, M_4\}$ to GWN.

After receiving $\{SID_j, MID_i, M_1, M_2, M_3, M_4\}$,$GWN$ calculates $X_i^* = h(MID_i||S_g)$, $r_1^* = M_2 \oplus X_i$, and $r_2^* = M_3 \oplus P_j$.GWN verifies $M_1 ?= h(X_i||GWNPU_i||r_1^*)$ and $M_4 = h(GWNPS_j||M_2||r_2^*)$. If both are equal, the GWN authenticates the user U_i and the sensor node S_j The GWN chooses a random number r_3 and calculates $M_5 = r_3 \oplus h(GWNPS_j \oplus r_2^*)$, $M_6 = h(X_i^*||GWNPS_j||r_1^*||r_2^*||r_3)$, $P_1 = r_1 \oplus P_j$ and $P_2 = X_i^* \oplus h(P_j||r_2^*||r_1^*)$. And then sends $\{M_5, M_6, P_1, P_2\}$ to S_j.

After receiving $\{M_5, M_6, P_1, P_2\}$,S_j calculates $r_1^* = P_1 \oplus P_j$, $X_i^* = P_2 \oplus h(P_j||r_2||r_1^*)$, $r_3^* = M_5 \oplus h(GWNPS_j \oplus r_2)$ and verifies $M_6 ?= h(X_i^*||GWNPS_j||r_1^*||r_2||r_3^*)$. If it is equal, the S_j authenticates the GWN and then calculates the session key $SK = h(X_i^*||P_j||r_3^*||r_2||r_1^*)$. S_j calculate $M_7 = X_i^* \oplus r_2$, $M_8 = (P_j||r_3^*) \oplus r_2$ and $M_9 = h(r_1^*||r_2)$. S_j sends $\{M_7, M_8, M_9\}$ to U_i.

After receiving $\{M_7, M_8, M_9\}$,U_i calculates $r_2^* = X_i^* \oplus M_7$,$(P_j||r_3^*) = M_8 \oplus r_2^*$ and then verifies $M_9 ?= h(r_1||r_2^*)$. If the result is equal, U_i authenticates S_j and then calculates the session key $SK = h(X_i||P_j||r_3^*||r_2^*||r_1)$.

5 Security flaws of Banerjee et al.'s scheme

Though Banerjee et al. claimed that their scheme can resist various attacks, in this section, we show that their scheme has some security flaws.

5.1 Identity linkability

In Banerjee et al.'s scheme, because $MID_i = h(UID_i||r_i)$ transmitted in public channel and unchanged in each session, the scheme exists the user identity linkability. Further, the adversary may get the user's real identity according to the user's behavior information. That is, the user's anonymity may be broken.

5.2 Offline password guessing attacks

If an adversary can obtain $\{MXIP_i, X_i, h(), BH(), V_i, A_i\}$ stored in user's smart card, and $\{MID_i, M_1, M_2\}$ from the public channel, then he can guess the user's password $UPWD$ and computes $\mathrm{r}_\mathrm{i}' = \mathrm{h}(\mathrm{UID}_\mathrm{i}||UPWD) \oplus \mathrm{A}_\mathrm{i}$,$MID_i' = h(UID_i||r_i')$, $MPWD_i' = h(UPWD||r_i')$,$GWNPU_i' = h(MID_i'||MPWD_i') \oplus MXIP_i$, $r_1' = M_2 \oplus X_i$ and verifies whether $M_1 = h(X_i||GWNPU_i'||r_1')$ or not. If yes, the guessed password is correct. Otherwise, the adversary does it again till to find the correct password.

This attack may success; the reason is that the user's identity UID_i may easily to be known (e.g., an insider attacker, like the user's colleague) or it is often publicly available, and the password dictionary size is very restricted. Even if the adversary needs to guess UID_i and $UPWD_i$ simultaneously, the time complexity of the above attacking procedure is $\mathrm{O}(|D_{id}| * |D_{pw}| * (T_h))$, where T_h is the running time for hash operation and can guess the correct identity and password quickly [24].

If an adversary can get the user's password by offline password guessing attack, then he can know $GWNPU_i$ and X_i, and he can launch impersonation attack.

5.3 No perfect forward secrecy

Because the session key is $SK = h(X_i||P_j||r_3||r_2||r_1)$, if an adversary can obtain the secret key S_g of GWN, then he can compute $X_i = h(MID_i||S_g)$, $r_1 = M_2 \oplus X_i$, $P_j = r_1 \oplus P_1$, $r_2 = X_i \oplus M_7$, $(P_j||r_3) = M_8 \oplus r_2$. That is, the adversary can compute the session key, since he can get $\{MID_i, M_2, P_1, M_7, M_8\}$ from public channels.

5.4 No session key secrecy

If a legal user U_l have pass through the authentication of the sensor node S_j, then U_l can know P_j. After that, when other user U_i wants to pass through the authentication of the sensor node S_j, U_l can get the authentication messages $\{MID_i, M_2, P_1, M_7, M_8\}$ from public channels, so U_l can compute $r_1 = P_j \oplus P_1$, $X_i = M_2 \oplus r_1$, $r_2 = X_i \oplus M_7$, $(P_j||r_3) = M_8 \oplus r_2$; therefore, U_l can compute the session key $SK = h(X_i||P_j||r_3||r_2||r_1)$ shared between the U_i and S_j.

5.5 Impersonation attack

If a legal user U_l have pass through the authentication of the sensor node S_j, and obtains P_j, then he can impersonate the S_j. When other user U_i wants to login onto the sensor node S_j, U_l sends $\{MID_i, M_1, M_2\}$ to S_j, S_j sends $\{SID_j, MID_i, M_1, M_2, M_3, M_4\}$

to GWN. When GWN responses the message $\{M_5, M_6, P_1, P_2\}$, U_l intercepts it, and chooses a random number r_3', and computes $r_1^* = P_1 \oplus P_j$,$r_2^* = M_3 \oplus P_j$, $X_i^* = P_2 \oplus h(P_j||r_2||r_1^*)$, the session key $SK = h(X_i^*||P_j||r_3'||r_2||r_1^*)$, S_j calculate $M_7 = X_i^* \oplus r_2*$, $M_8 = (P_j||r_3') \oplus r_2*$ and $M_9 = h(r_1^*||r_2*)$. U_l sends $\{M_7, M_8, M_9\}$ to U_l. Obviously, U_l can compute and verify the correction of the session key SK. However, U_l shares the session key SK with U_l, not the sensor node S_j.

6 Proposed scheme

There are three entities of our proposed scheme: the user U_i, the sensor node S_j and the gateway node GWN. The user and the sensor node can authenticate each other and establish a session key with the help of the gateway node. Our protocol has four phases: initialization phase, registration phase, authentication and key agreement phase and password change phase.

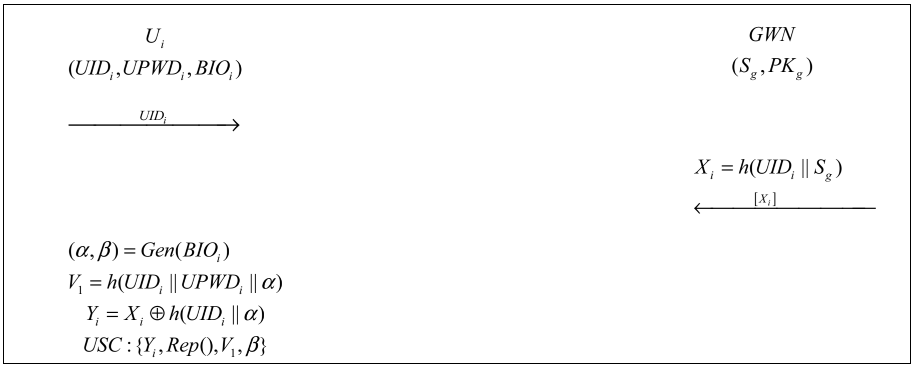
6.1 Initialization phase

The administrator provides an identity SID_j and a secret key $GWNPS_j$ (shared with GWN) for the sensor node S_j and chooses a prime number P and an additive group $G1$, the GWN's long secret key $S_g \in Z_p$ and computes the GWN's public key $PK_g = S_gP$, where $P \in G1$.

6.2 Registration phase

The registration phase is run through the secure channel as shown in Algorithm 1. The user U_i firstly chooses its identity UID_i and password $UPWD_i$ and sends $\{UID_i\}$ to GWN. After receiving $\{UID_i\}$, GWN calculates $X_i = h(UID_i||S_g)$. GWN enters $\{X_i\}$ into the smart card and sends it to U_i through the secure channel.

U_i imprints the biological information BIO_i and calculates $(\alpha, \beta) = Gen(BIO_i)$, $V_1 = h(UID_i||UPWD_i||\alpha)$ and $Y_i = X_i \oplus h(UID_i||\alpha)$.$U_i$ stores $\{Y_i, Rep(), V_1, \beta\}$ in the smart card USC.



Algorithm 1: User-gateway registration phase

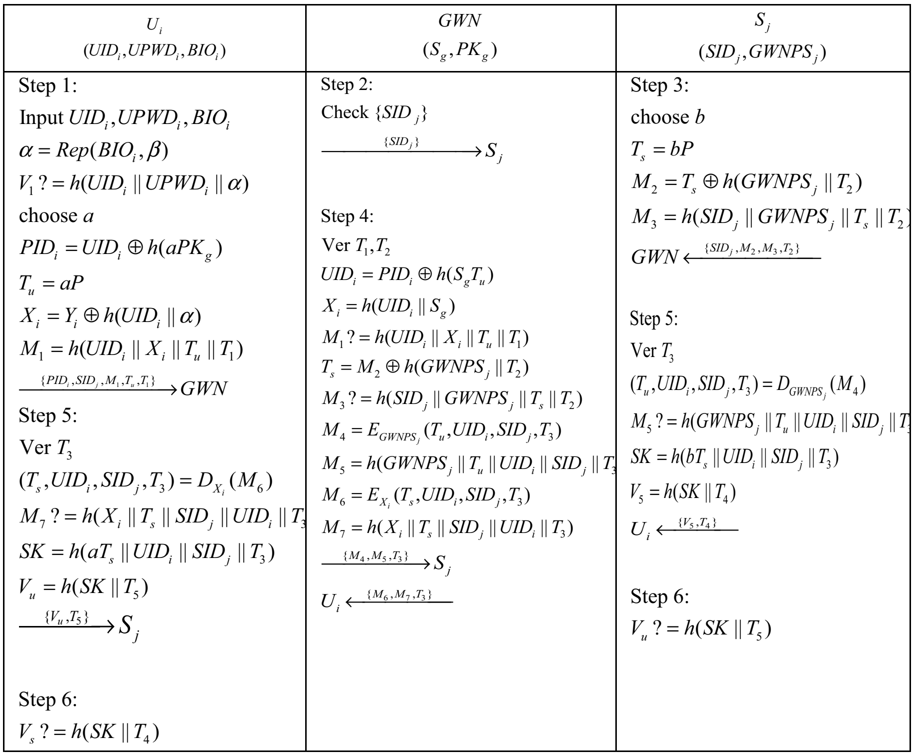
6.3 Authentication and key agreement phase

The user, the sensor and the gateway authenticate with each other, and the user and the sensor negotiate session key as shown in Algorithm 2.

Step 1 U_i inserts smart card and inputs identity UID_i, password $UPWD_i$ and biological information BIO_i. The smart card USC calculates $\alpha = Rep(BIO_i, \beta)$ and verifies whether $V_1 ? = h(UID_i || UPWD_i || \alpha)$ or not. If not, the user re-does it. Otherwise, USC chooses a random numbers a and calculates the user's temporary identity $PID_i = UID_i \oplus h(aPK_g)$, $T_u = aP$, $X_i = Y_i \oplus h(UID_i || \alpha)$, and $M_1 = h(UID_i || X_i || T_u || T_1)$ where T_1 is the current timestamp. U_i sends $\{PID_i, SID_j, M_1, T_u, T_1\}$ to GWN.

Step 2 After receiving $\{PID_i, SID_j, M_1, T_u, T_1\}$, GWN checks the validity of T_1 and SID_j and forwards $\{SID_j\}$ to the sensor node S_j.

Step 3 After receiving $\{SID_j\}$, S_j chooses a random number b and calculates $T_s = bP$, $M_2 = T_s \oplus h(GWNPS_j || T_2)$ and $M_3 = h(SID_j || GWNPS_j || T_s || T_2)$ where T_2 is the current timestamp. S_j sends $\{SID_j, M_3, M_4, T_2\}$ to the GWN.



U_i $(UID_i, UPWD_i, BIO_i)$	GWN (S_g, PK_g)	S_j $(SID_j, GWNPS_j)$
Step 1: Input $UID_i, UPWD_i, BIO_i$ $\alpha = Rep(BIO_i, \beta)$ $V_1 ? = h(UID_i \| UPWD_i \| \alpha)$ choose a $PID_i = UID_i \oplus h(aPK_g)$ $T_u = aP$ $X_i = Y_i \oplus h(UID_i \| \alpha)$ $M_1 = h(UID_i \| X_i \| T_u \| T_1)$ $\xrightarrow{\{PID_i, SID_j, M_1, T_u, T_1\}} GWN$ Step 5: Ver T_3 $(T_s, UID_i, SID_j, T_3) = D_{X_i}(M_6)$ $M_7 ? = h(X_i \| T_s \| SID_j \| UID_i \| T_3$ $SK = h(aT_s \| UID_i \| SID_j \| T_3)$ $V_u = h(SK \| T_5)$ $\xrightarrow{\{V_u, T_5\}} S_j$ Step 6: $V_s ? = h(SK \| T_4)$	Step 2: Check $\{SID_j\}$ $\xrightarrow{\{SID_j\}} S_j$ Step 4: Ver T_1, T_2 $UID_i = PID_i \oplus h(S_g T_u)$ $X_i = h(UID_i \| S_g)$ $M_1 ? = h(UID_i \| X_i \| T_u \| T_1)$ $T_s = M_2 \oplus h(GWNPS_j \| T_2)$ $M_3 ? = h(SID_j \| GWNPS_j \| T_s \| T_2)$ $M_4 = E_{GWNPS_j}(T_u, UID_i, SID_j, T_3)$ $M_5 = h(GWNPS_j \| T_u \| UID_i \| SID_j \| T_3$ $M_6 = E_{X_i}(T_s, UID_i, SID_j, T_3)$ $M_7 = h(X_i \| T_s \| SID_j \| UID_i \| T_3)$ $\xrightarrow{\{M_4, M_5, T_3\}} S_j$ $U_i \xleftarrow{\{M_6, M_7, T_3\}}$	Step 3: choose b $T_s = bP$ $M_2 = T_s \oplus h(GWNPS_j \| T_2)$ $M_3 = h(SID_j \| GWNPS_j \| T_s \| T_2$ $GWN \xleftarrow{\{SID_j, M_2, M_3, T_2\}}$ Step 5: Ver T_3 $(T_u, UID_i, SID_j, T_3) = D_{GWNPS_j}(M_4)$ $M_5 ? = h(GWNPS_j \| T_u \| UID_i \| SID_j \| T$ $SK = h(bT_s \| UID_i \| SID_j \| T_3)$ $V_5 = h(SK \| T_4)$ $U_i \xleftarrow{\{V_5, T_4\}}$ Step 6: $V_u ? = h(SK \| T_5)$

Algorithm 2: Authentication and key agreement phase

Step 4 After received the authentication message, GWN first verifies the timestamp T_2. If $T - T_2 \leq \Delta T$ is false where T is the current timestamp, GWN refuses the authentication request. Otherwise, GWN calculates $UID_i = PID_i \oplus h(S_g T_u)$, $X_i = h(UID_i || S_g)$ and verifies $M_1 ? = h(UID_i || X_i || T_u || T_g || T_1)$. If the result is false, GWN terminates the protocol. Otherwise, GWN authenticates the user successfully. After that, GWN calculates $T_s = M_2 \oplus h(GWNPS_j || T_2)$ and verifies $M_3 ? = h(SID_j || GWNPS_j || T_s || T_2)$. If the result is false, GWN also terminates the protocol. Otherwise, GWN authenticates the sensor node S_j successfully. Then, GWN computes $M_4 = E_{GWNPS_j}(T_u, UID_i, SID_j, T_3)$ and $M_6 = E_{X_i}(T_s, UID_i, SID_j, T_3)$ and calculates $M_5 = h(GWNPS_j || T_u || UID_i || SID_j || T_3)$

and $M_7 = h(X_i||T_s||SID_j||UID_i||T_3)$, where T_3 is the current time stamp. GWN sends $\{M_4, M_5, T_3\}$ to S_j and $\{M_6, M_7, T_3\}$ to U_i.

Step 5 After receiving the message $\{M_4, M_5, T_3\}$, S_j firstly verifies T_3. If $T' - T_3 \leq \Delta T$ is false where T' is the current timestamp, S_j terminates the protocol. Otherwise, S_j computes $(T_u, UID_i, SID_j, T_3) = D_{GWNPS_j}(M_4)$ and verifies $M_5? = h(GWNPS_j||T_u||UID_i||SID_j||T_3)$. If the result is equal, S_j authenticates GWN successfully. And then, S_j calculates the session key $SK = h(bT_u||UID_i||SID_j||T_3)$ and $V_s = h(SK||T_4)$ where T_4 is the current timestamp. S_j sends $\{V_s, T_4\}$ to U_i.

At the same time, when U_i receives $\{M_6, M_7, T_3\}$ from GWN, U_i firstly verifies T_3. If $T'' - T_3 \leq \Delta T$ is false where T'' is the current timestamp, U_i terminates the protocol. Otherwise, U_i decrypts $(T_s, UID_i, SID_j, T_3) = D_{X_i}(M_6)$ and verifies $M_7? = h(X_i||T_s||SID_j||UID_i||T_3)$. If the result is false, U_i also terminates the protocol. Otherwise, U_i authenticates GWN successfully and calculates the session key $SK = h(aT_s||UID_i||SID_j||T_3)$ and $V_u = h(SK||T_5)$, where T_5 is the current timestamp. Then, U_i sends $\{V_u, T_5\}$ to S_j.

Step 6. After receiving $\{V_s, T_4\}$ and $\{V_u, T_5\}$, U_i and S_j verifies the freshness of T_4 and T_5, and verifies the correctness of V_s and V_u, respectively. After verification of correctness, U_i and S_j share the session key SK.

6.4 Password change phase

If the user wants to change or update his passwords, U_i inserts USC in card reader and inputs identity UID_i, password $UPWD_i^{old}$ and biological information BIO_i. Next, the smart card calculates $\alpha = Rep(BIO_i, \beta)$ and verifies $V = h(UID_i||UPWD_i^{old}||\alpha)$. If the result is false, the smart card does not recognize him as a legitimate user. Otherwise, the user inputs new password $UPWD_i^{new}$. The smart card calculates $V^{new} = h(UID_i||UPWD_i^{new}||\alpha)$ and replaces V with V^{new}.

7 Security analysis

In the section, we analyze the security of the proposed protocol by using formal and informal security analysis.

7.1 Formal security analysis

We simulate our scheme using ProVerif simulation tool [25], which is widely used for proving the authentication and session key secrecy. We defined three public channels ch1, ch2, ch3. The details of variables and functions are shown in Table 2. The details of process of user authentication are shown in Table 3. The process of sensor authentication is shown in Table 4. The simulation process of gateway is shown in Table 5.

We use the ProVerif to prove our authentication phase. First, the smart card authenticates the user successfully. Second, the gateway node authenticates the user successfully. Third, the sensor node authenticates the gateway node successfully. Fourth, the gateway node authenticates the sensor node successfully. Finally, the user authenticates the gateway node and the sensor node successfully. That is, users, sensors and gateway achieve mutual authentication each other. If the scheme successfully goes on,

Table 2 Variables and functions

```
free ch1:channel.
free ch2:channel.
free ch3:channel.
type user.
type sensor.
type GWN.
free ui:user.
free sj:sensor.
free gwn:GWN.
free UIDi:bitstring.
free UPWDi:bitstring.
free BIOi:bitstring.
free Sg:bitstring[private].
free PKg:bitstring.
free GWNPSj:bitstring[private].
free SIDj:bitstring.
free SKus:bitstring[private].
free SKsu:bitstring[private].
(*fun hash*)
fun hash(bitstring):bitstring.
(*encrpt and dencrpt*)
fun E(bitstring,bitstring):bitstring.
fun D(bitstring,bitstring):bitstring.
(*BioHash*)
fun gen(bitstring):bitstring.
fun Rep(bitstring,bitstring):bitstring.
(*XOR operation*)
fun XOR(bitstring,bitstring):bitstring.
equation forall m:bitstring,n:bitstring;XOR(XOR(m,n),n)=m.
(*Diffie-Hellman fun*)
fun G(bitstring):bitstring.
fun GK(bitstring,bitstring):bitstring.
(*concatenation operation*)
fun concat(bitstring,bitstring):bitstring.
```

Table 3 User simulation process

```
(*processuser*)
let processuser=
  new bi:bitstring;
  new V:bitstring;
  let ai=Rep(BIOi,bi) in
  let V'=hash(concat(concat(UIDi,UPWDi),ai)) in
  if V'=V then
  event SmartcardAccept(ui);
  new T1:bitstring;
  new a:bitstring;
  new Yi:bitstring;
  let Tu=G(a) in
  let Xi=XOR(Yi,hash(concat(UIDi,ai))) in
  let PIDi=XOR(UIDi,hash(GK(a,PKg))) in
  let M1=hash(concat(UIDi,concat(Xi,concat(Tu,T1)))) in
  out(ch1,(PIDi,SIDj,M1,Tu,T1));
  in(ch1, (M6:bitstring,M7:bitstring,T3:bitstring));
  let (Ts:bitstring,UIDi:bitstring,SIDj:bitstring,T3:bitstring)=D(Xi,M6) in
  let M7'=hash(concat(Xi,concat(Ts,concat(SIDj,concat(UIDi,T3))))) in
  let SKus=hash(concat(GK(a,Ts),concat(UIDi,concat(SIDj,T3))))in
  new T5:bitstring;
  let Vu=hash(concat(SKus,T5)) in
  out(ch3,(Vu,T5));
  in(ch3,(Vs:bitstring,T4:bitstring));
  let Vs'=hash(concat(SKus,T4)) in
  event Sksuccessful(ui,sj)
```

Table 4 Sensor node simulation process

```
(*processsensornode*)
let processsensornode =
  in(ch2,SIDj:bitstring);
  new b:bitstring;
  new T2:bitstring;
  let Ts = G(b)in
  let M2 = XOR(Ts,hash(concat(GWNPSj,T2))) in
  let M3 = hash(concat(SIDj,concat(GWNPSj,concat(Ts,T2)))) in
  out(ch2,(SIDj,M2,M3,T2));
  in(ch2,(M4:bitstring,M5:bitstring,T3:bitstring));
  let (Tu:bitstring,UIDi:bitstring,SIDj:bitstring,T3:bitstring) = D(GWNPSj,M4) in
  let M5’= hash(concat(GWNPSj,concat(Tu,concat(UIDi,concat(SIDj,T3))))) in
  if M5 = M5’ then
  event sensorAccept(gwn);
  let SKsu = hash(concat(GK(b,Ts),concat(UIDi,concat(SIDj,T3)))) in
  new T4:bitstring;
  let Vs = hash(concat(SKsu,T4))in
  out(ch3,(Vs,T4));
  in(ch3,(Vu:bitstring,T5:bitstring));
  let Vu’= hash(concat(SKsu,T5))in
  if Vu’=Vu then
  event Sksuccessful(ui,sj)
```

Table 5 Gateway node simulation process

```
(*processgateway*)
let processgateway =
  in(ch1,(SIDj:bitstring,PIDi:bitstring,M1:bitstring,Tu:bitstring,T1:bitstring));
  in(ch2,(SIDj:bitstring,M2:bitstring,M3:bitstring,T2:bitstring));
  let UIDi = XOR(PIDi,hash(GK(Sg,Tu)))in
  let Xi = hash(concat(UIDi,Sg)) in
  let M1’= hash(concat(UIDi,concat(Xi,concat(Tu,T1))))in
  if M1’= M1 then
  event gwnAcceptU(ui);
  let Ts = XOR(M2,hash(concat(GWNPSj,T2)))in
  let M3’= hash(concat(SIDj,concat(GWNPSj,concat(Ts,T2)))) in
  if M3’= M3 then
  event gwnAcceptS(sj);
  new T3:bitstring;
  let M4 = E(GWNPSj,concat(Tu,concat(UIDi,concat(SIDj,T3))))in
  let M5 = hash(concat(GWNPSj,concat(Tu,concat(UIDi,concat(SIDj,T3)))))in
  let M6 = E(Xi,concat(Ts,concat(UIDi,concat(SIDj,T3))))in
  let M7 = hash(concat(Xi,concat(Ts,concat(SIDj,concat(UIDi,T3)))))in
  out(ch2,(M4,M5,T3));
  out(ch1,(M6,M7,T3))
```

the user and sensor node will negotiate the same session key. Therefore, our scheme has six time points, and their code is represented in the ProVerif as follows:

event SmartcardAccept(user) means the user logins the smart card successfully.
event gwnAcceptU(user) means the gateway node authenticates the user successfully.
event gwnAcceptS(sensor) means the gateway node authenticates the sensor node successfully.
event sensorAccept(GWN) means the sensor node authenticates the gateway node successfully.

event userAccept(sensor,GWN) means the user authenticates the gateway node and the sensor node successfully.

event Sksuccessful(user,sensor) means user and sensor node get the same session key.

The authentication order of our protocol is as follows:

```
query Ui:user; inj-event(gwnAcceptU(Ui))==>inj-event(SmartcardAccept(Ui)).
query Ui:user,Gateway:GWN; inj-event(sensorAccept(Gateway))==>inj-event (gwnAcceptU(Ui)).
query Sj:sensor,Gateway:GWN; inj-event(gwnAcceptS(Sj))==>inj-event (sensorAccept(Gateway)).
query Ui:user,Sj:sensor,Gateway:GWN; inj-event(userAccept(Sj,Gateway))==>inj-event (gwnAcceptS(Sj)).
```

Our protocol must also protect the session keys (SKus and SKsu). The code is:

```
query attacker(SKsu).
query attacker(SKus).
```

Finally, we use the following code to start the verification:

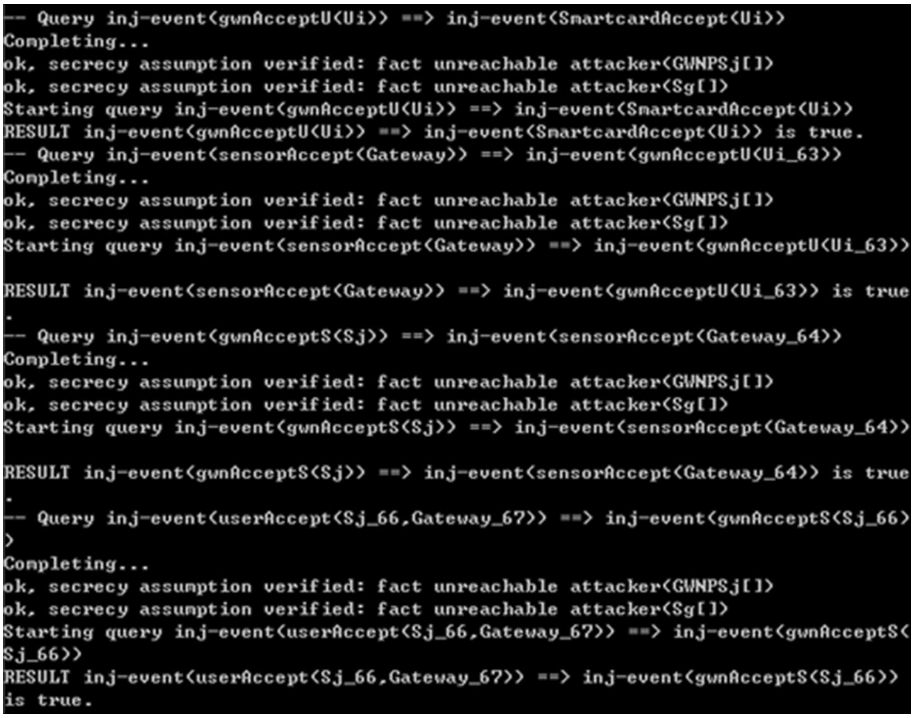
```
process
(*constant computed*)
    let Xi=hash(concat(Sg,UIDi)) in
    let (ai:bitstring,bi:bitstring)=gen(BIOi) in
    let V1=hash(concat(UIDi,concat(UPWDi,ai))) in
    let Yi=XOR(Xi,hash(concat(UIDi,ai))) in
!processuser |!processsensornode|!processgateway
```

The simulation authentication result is shown in Fig. 1. The simulation result of the session key is shown in Fig. 2. The result shows that our scheme achieves mutual authentication and the session key security.

7.2 Informal security analysis

7.2.1 Anonymity and unlinkability

In our scheme, the user's real identity is contained in those parameters $< PID_i, M_4, M_5, M_6, M_7 >$.$PID_i$ is protected by Diffie–Hellman problem. And the M_4 and M_6 are encrypted by X_i and $GWNPS_j$, respectively. The rest of the parameters $< M_6, M_8 >$ are protected by the hash function. So, the adversary cannot know the user's real identity. Our scheme meets the need of anonymity. The PID_i changes in each session because of the use of the random number and Diffie–Hellman value. So, our scheme is also unlinkability.

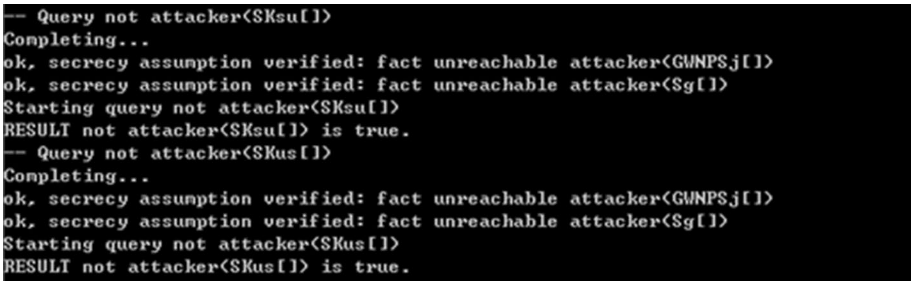


```
-- Query inj-event(gwnAcceptU(Ui)) ==> inj-event(SmartcardAccept(Ui))
Completing...
ok, secrecy assumption verified: fact unreachable attacker(GWNPSj[])
ok, secrecy assumption verified: fact unreachable attacker(Sg[])
Starting query inj-event(gwnAcceptU(Ui)) ==> inj-event(SmartcardAccept(Ui))
RESULT inj-event(gwnAcceptU(Ui)) ==> inj-event(SmartcardAccept(Ui)) is true.
-- Query inj-event(sensorAccept(Gateway)) ==> inj-event(gwnAcceptU(Ui_63))
Completing...
ok, secrecy assumption verified: fact unreachable attacker(GWNPSj[])
ok, secrecy assumption verified: fact unreachable attacker(Sg[])
Starting query inj-event(sensorAccept(Gateway)) ==> inj-event(gwnAcceptU(Ui_63))

RESULT inj-event(sensorAccept(Gateway)) ==> inj-event(gwnAcceptU(Ui_63)) is true
.
-- Query inj-event(gwnAcceptS(Sj)) ==> inj-event(sensorAccept(Gateway_64))
Completing...
ok, secrecy assumption verified: fact unreachable attacker(GWNPSj[])
ok, secrecy assumption verified: fact unreachable attacker(Sg[])
Starting query inj-event(gwnAcceptS(Sj)) ==> inj-event(sensorAccept(Gateway_64))

RESULT inj-event(gwnAcceptS(Sj)) ==> inj-event(sensorAccept(Gateway_64)) is true
.
-- Query inj-event(userAccept(Sj_66,Gateway_67)) ==> inj-event(gwnAcceptS(Sj_66)
)
Completing...
ok, secrecy assumption verified: fact unreachable attacker(GWNPSj[])
ok, secrecy assumption verified: fact unreachable attacker(Sg[])
Starting query inj-event(userAccept(Sj_66,Gateway_67)) ==> inj-event(gwnAcceptS(
Sj_66))
RESULT inj-event(userAccept(Sj_66,Gateway_67)) ==> inj-event(gwnAcceptS(Sj_66))
is true.
```

Fig. 1 Simulation result of the authentication



```
-- Query not attacker(SKsu[])
Completing...
ok, secrecy assumption verified: fact unreachable attacker(GWNPSj[])
ok, secrecy assumption verified: fact unreachable attacker(Sg[])
Starting query not attacker(SKsu[])
RESULT not attacker(SKsu[]) is true.
-- Query not attacker(SKus[])
Completing...
ok, secrecy assumption verified: fact unreachable attacker(GWNPSj[])
ok, secrecy assumption verified: fact unreachable attacker(Sg[])
Starting query not attacker(SKus[])
RESULT not attacker(SKus[]) is true.
```

Fig. 2 Simulation result of the session key

7.2.2 Offline password guessing attacks

Assume that an adversary knows the parameters $\{Y_i, Rep(), V_1, \beta\}$ stored in the smart card and all messages transmitted in all public channels, but he cannot guess the true password. Since the user's password is protected by the bio-information and the hash function, the adversary cannot verify the parameter V_1. On the other hand, we assume that an adversary knows the legal user's password and all messages transmitted through the public channel, but do not know the parameters stored in the smart card. However, the adversary cannot login the protocol because he cannot obtain the bio-information. So, our scheme can resist the offline password guessing attacks.

7.2.3 Replay attack

There are two ways to prevent replay attacks: adding timestamps and random numbers. Our scheme uses time stamps to prevent replay attack. In every session, the timestamps are different and the entity checks the fresh of the timestamps.

7.2.4 Impersonation attack and man-in-the-middle attack

In our scheme, the gateway node authenticates the user by the parameter M_1. The user authenticates the gateway node by the parameter M_7. If an adversary wants to impersonate the legal user, he must know those parameter $< T_u, X_i, T_s >$ which X_i is pre-shared with the gateway node. However, X_i is contained in Y_i stored in the smart card securely. Similarly, an adversary cannot impersonate the sensor node because of $GWNPS_j$. If an adversary captures a sensor node, he cannot know the others' key parameters. So he cannot impersonate other sensor node and the user. Therefore, our scheme resists impersonation attack, and the man-in-the-middle is also invalid.

7.2.5 Perfect forward secrecy

Assume that an adversary knows the user's password $UPWD_i$, the sensor node's secret key $GWNPS_j$. Since the session key is $SK = h(abP||UID_i||SID_j||T_3)$, an adversary cannot compute abP due to Elliptic Curve Diffie–Hellman problem (ECDHP). So he cannot compute the session key.

7.2.6 Known session key security

In our scheme, the adversary cannot compute the session key because of ECDHP. The session key is also different in each session due to two random numbers a, b. So, if the adversary knows a session key, he cannot know the before and the future session keys.

7.2.7 Sensors capture attack

If an adversary can capture a sensor node S_j, then he can obtain the secret key $GWNPS_j$ shared with GWN. However, each sensor node has a different secret key shared with GWN, so the adversary cannot impersonate another sensor node to pass through the authentication with GWN. On the other hand, even if the adversary can know the secret key $GWNPS_j$ of a sensor node S_j, he cannot know the user's $X_i = h(UID_i||S_g)$ from the session run. Therefore, the proposed scheme is secure even if the sensor node is captured.

8 Results and discussion

In this section, we will discuss security and performance comparison with some related schemes, such as Fan et al. [1], Yeh et al. [12], Luo et al. [15], Banerjee et al. [17], Choi et al. [26], Park et al. [27] and Challa et al.'s schemes [28].

T_m means the time of the point multiplicative operation in ECC, T_{Rep} means the running time to performance *Rep* which is equal to T_m [29], T_s means the time in symmetric encryption or decryption, T_h means the time of hash operation, and T is the time or searching the identity in verification table which is related to the number of users. The running time is shown in Table 6 [30].

As shown in Tables 7 and 8, we can know that the proposed scheme achieves both security and computational efficiency.

Table 6 Notations of time symbols

Symbol	Meaning	Time (ms)
T_m	Time of the point multiplicative operation	2.226
T_{Rep}	Time of fuzzy extractor	2.226
T_s	Time in symmetric encryption or decryption	0.0046
T_h	Time of hash operation	0.0023
T	The time for searching the identity in verification table	$O(n)$, n is the number of users

Table 7 Computation cost comparison

References	User (ms)	Gateway (ms)	Sensor (ms)	Total	Total (ms)
[1]	$12T_h + T_{Rep} + 2T_m$	$10T_h$	$3T_h + 2T_m$	$25T_h + T_{Rep} + 4T_m$	$=11.19$
[12]	$1T_h + 2T_m$	$4T_h + 4T_m$	$3T_h + 2T_m$	$8T_h + 8T_m$	$=17.83$
[15]	$8T_h + T_{Rep}$	$11T_h + 2T$	$5T_h + T$	$24T_h + T_{Rep} + 3T$	$=2.2812$
[17]	$9T_h + T_{Rep}$	$6T_h$	$6T_h$	$20T_h + T_{Rep}$	$=2.27$
[26]	$10T_h + T_{Rep} + 2T_m + T_s$	$10T_h + 2T_s$	$6T_h + 2T_m + T_s$	$26T_h + T_{Rep} + 4T_m + 5T_s$	$=16.19$
[27]	$10T_h + T_{Rep} + 2T_m$	$11T_h+T$	$4T_h + 2T_m$	$25T_h + T_{Rep} + 4T_m+T$	$=11.19 + O(n)$
[28]	$5T_h + T_{Rep} + 5T_m$	$4T_h + 5T_m+T$	$3T_h + 4T_m$	$12T_h + T_{Rep} + 14T_m+T$	$=33.42 + O(n)$
Ours	$8T_h + T_{Rep} + 3T_m + T_s$	$7T_h + T_m + 2T_s$	$5T_h + 2T_m + T_s$	$20T_h + T_{Rep} + 6T_m + 4T_s$	$=15.646$

Table 8 Comparison of security features

Security features	[1]	[12]	[15]	[17]	[26]	[27]	[28]	ours
Impersonation attack	√	×	√	×	×	√	√	√
Anonymity and unlinkability	×	×	√	×	×	×	×	√
Verification table stolen attack	√	√	√	√	√	×	×	√
Password guessing attack	×	×	√	×	√	√	√	√
Known session key security	√	×	√	×	√	√	√	√
Replay attack	√	×	√	×	√	√	√	√
Man-in-the-middle attack	√	×	√	×	√	√	√	√
Perfect forward secrecy	√	×	×	×	√	√	√	√
Session key security	√	√	×	√	√	√	√	√
Sensor capture attack	√	√	×	√	√	√	√	√

9 Conclusions

In this paper, we have shown that the recently proposed Banerjee et al.'s protocol cannot resist offline password guessing attack, impersonation attack, and does not achieve session key secrecy, identity unlinkability and perfect forward secrecy. Then, we proposed a secure and privacy-preserving protocol to fix their security flaws. According to the formal security proof and performance comparison with some related schemes, we can know that our protocol achieves both security and computational efficiency and can be used to the smart city. In the future, we will design more secure authentication protocols for smart city applications, such as smart transportation and smart healthcare.

Abbreviations

WSNs: Wireless sensor networks; USC: The user's smart card; GWN: Gateway node.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61702152 and 61702153, URL http://www.nsfc.gov.cn/).

Authors' contributions

QX, BH and KL have analyzed the security of Banerjee et al.'s scheme and have modeled and designed the protocol. KL and WT wrote the paper. XT and LH proved the authentication and security of the proposed scheme. BH and KL verified the authentication, security and anonymity of the proposed scheme in the latest version 1.95 of ProVerif. QX, BH and KL revised the manuscript. All authors read and approved the final manuscript.

Funding

This research was supported by the National Key R&D Program of China (Grant No. 2017YFB0802000, URL http://www.most.gov.cn/).

Availability of data and materials

The dataset used and analyzed during the current study is available in the manuscript.

Declarations

Competing interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Received: 30 December 2020 Accepted: 26 April 2021

References

1. W. Fan, X. Lili, K. Saru, L. Xiong, An improved and provably secure three-factor user authentication scheme for wireless sensor networks. Peer Peer Net. Appl. **11**, 1–20 (2018)
2. V.R. Maneesha, R. Prabha, H. Thirugnanam, R.A. Devidas, R.K. Pathinarupothi, Achieving sustainability through smart city applications: protocols, systems and solutions using IoT and wireless sensor network. CSI Trans ICT **8**, 1–18 (2020)
3. Q. Xie, L. Hwang, Security enhancement of an anonymous roaming authentication scheme with two-factor security in smart city. Neurocomputing **347**, 131–138 (2019)
4. T. Gaber, S. Abdelwahab, M. Elhoseny, A.E. Hassanien, Trust-based secure clustering in WSN-based intelligent transportation systems. Comput. Netw. **146**, 151–158 (2018)
5. S.A. Chaudhry, K. Yahya, F. Al-Turjman, M.H. Yang, A secure and reliable device access control scheme for IoT based sensor cloud systems. IEEE Access **8**, 139244–139254 (2020)
6. S.A. Chaudhry, M.S. Farash, N. Kumar, M.H. Alsharif, PFLUA-DIoT: a pairing free lightweight and unlinkable user access control scheme for distributed IoT environment. IEEE Syst. J. **PP**, 1–8 (2020)
7. S.A. Chaudhry, Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid." Int. J. Electr. Energy Syst. **125**, 106529 (2021)
8. A. Irshad, M. Usman, S.A. Chaudhry, H. Naqvi, M. Shafiq, A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework. IEEE Trans. Ind. Appl. **56**(4), 4425–4435 (2020)
9. R. Watro, D. Kong, S. Cuti, C. Gardiner, P. Kruus, P.K. Tiny, Securing sensor networks with public key technology, in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks* (ACM, 2004), pp. 59–64
10. M.L. Das, Two-factor user authentication in wireless sensor networks. IEEE Trans. Wireless Commun. **8**(3), 1086–1090 (2009)
11. M.K. Khan, K. Alghathbar, Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks.' Sensors **10**(3), 2450–2459 (2010)
12. 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), 4767–4779 (2011)
13. 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), 316–323 (2013)
14. P. Gope, T. Hwang, A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. IEEE Trans. Industr. Electron. **63**(11), 7124–7132 (2016)
15. H. Luo, G. Wen, J. Su, Lightweight three factor scheme for real-time data access in wireless sensor networks. Wirel. Netw. **26**(2), 955–970 (2020)
16. 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**, 96–112 (2014)
17. S. Banerjee, C. Chunka, S. Sen, R.S. Goswami, An enhanced and secure biometric based user authentication scheme in wireless sensor networks using smart cards. Wirel. Pers. Commun. **107**, 1–28 (2019)
18. L. Yanrong, L. Lixiang, Y. Xing, Y. Yixian, Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS ONE **10**(5), e0126323 (2015)

19. 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), 541–552 (2002)
20. V.S. Miller, Use of elliptic curves in cryptography, in Conference on the theory and application of cryptographic techniques (Springer, Berlin, 1985), pp. 417–426
21. N. Koblitz, Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
22. A.K. Das, A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. Int. J. Commun Syst **30**(1), 1–25 (2017)
23. Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data, in International Conference on the Theory and Applications of Cryptographic Techniques (Springer, Berlin, 2004), pp. 523–540
24. W. Ding, Q. Gu, H. Cheng, W. Ping, The request for better measurement: a comparative evaluation of two-factor authentication schemes, in Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (2016), pp. 475–486
25. B. Blanchet, An efficient cryptographic protocol verifier based on prolog rules, in 14th IEEE Computer Security Foundations Workshop (CSFW-14) (2014), pp. 82–96
26. Y. Choi, Y. Lee, D. Won, Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. Int. J. Distrib. Sens. Netw. **4**, 1–16 (2016)
27. Y. Park, Y. Park, Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors **16**(12), 2123 (2016)
28. S. Challa, M. Wazid, A.K. Das, N. Kumar, R.A. Goutham, E.J. Yoon, K.Y. Yoo, Secure signature-based authenticated key establishment scheme for future IoT applications. IEEE Access **5**, 3028–3043 (2015)
29. L. Xu, F. Wu, Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care. J. Med. Syst. **39**(10), 1–9 (2015)
30. H.H. Kilinc, T. Yanik, A survey of sip authentication and key agreement schemes. IEEE Commun. Surv. Tutor. **16**(2), 1005–1023 (2014)

Publisher's Note