Hindawi
Security and Communication Networks
Volume 2021, Article ID 4799223, 12 pages
https://doi.org/10.1155/2021/4799223

Research Article

A Secure and Privacy-Preserving Three-Factor Anonymous Authentication Scheme for Wireless Sensor Networks in Internet of Things

Qi Xie, Zixuan Ding, and Bin Hu

Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China

Correspondence should be addressed to Qi Xie; qixie68@126.com

Received 23 July 2021; Revised 16 August 2021; Accepted 24 August 2021; Published 29 September 2021

Academic Editor: Marimuthu Karuppiah

The Internet of things is playing more and more important role in smart healthcare, smart grids, and smart transportation, and using wireless sensor network (WSN), we can easily obtain and transmit information. However, the data security and users' privacy are the biggest challenges for WSN because sensor nodes have low computing power and low storage capacity and are easy to be captured, and wireless networks are vulnerable. In 2021, Shuai et al. proposed a lightweight three-factor anonymous authentication scheme for WSN. However, we found that their protocol is vulnerable to stolen-verifier attack, modification of messages' attack, and no perfect forward secrecy. Then, a new three-factor anonymous authentication scheme using elliptic curve cryptography (ECC) is proposed. Through informal and formal security analyses, our scheme can resist various known attacks and maintains low computational complexity.

1. Introduction

In recent years, with the rapid development of Internet of things (IoT) technology, wireless sensor networks (WSN) are widely used in medical, military, agriculture, and other fields [1]. A large number of wireless sensor nodes are deployed in the target fields to collect the data in WSN, but sensor nodes have low computing power and low storage capacity and are easy to be captured; on the contrary, compared with the traditional wired network, messages are transmitted through wireless channels, and it may be easily attacked by means of eavesdropping, capture, replay, forgery, and so on. In order to protect the data security and users' privacy, it is very important to design secure and privacy-preserving authentication and key agreement protocol for WSN in IoT.

Many authentication protocols have been proposed in the past ten years; however, these protocols exist one or more security flaws [2]. In 2013, Li et al. proposed a communication scheme in IoT [3], which provides authentication, integrity, nonrepudiation, and confidentiality. However, this scheme is based on bilinear pairing, so it is hard to be deployed in WSN [4]. In 2014, Turkanović et al. [5] proposed a hash function-based authentication scheme for WSN. Farash et al. [6] pointed out that it suffers from impersonation attack, smart card loss attack, and session key disclosure attack; then, Farash et al. designed a new two-factor authentication (2FA) protocol. Amin and Biswas [7] also showed that Turkanović et al.'s scheme [5] suffers from offline password-guessing attacks and impersonation attacks, and Amin et al. proposed a 2FA protocol for multi-gateway WSN. Meanwhile, Amin et al. found that, in Farash et al.'s [6] scheme, there exists some security flaws, such as impersonation attack, smart card loss attack, and offline password-guessing attack.

In order to improve the security of authentication protocol, Diffie–Hellman key agreement algorithm, Chebyshev chaotic map [8], and elliptic curve cryptography (ECC) are used to design secure user authentication and key agreement protocol [9, 10]. In 2009, Das [11] proposed an

authentication protocol based on ECC for WSN, but their scheme suffers from privilege insider attacks and gateway bypass attacks [12]. Later, Kumar et al. [13] proposed an efficient authentication protocol for WSN. He et al. [14] showed that their scheme suffers from offline password-guessing attack and privilege insider attacks. To overcome these security flaws, they proposed an improved authentication scheme for WSN. Unfortunately, Li et al. [15], Wu et al. [16], and Mir et al. [17] pointed out that He et al.'s scheme is still insecure, and it may suffer from offline password-guessing attack and impersonation attack. Therefore, Li et al. [15] proposed a three-factor authentication (3FA) scheme to overcome these flaws because two-factor authentication (2FA) schemes usually suffer from offline password-guessing attacks [18]. Compared with 2FA schemes, 3FA schemes can improve the security because 3FA schemes use biometrics to avoid password-guessing attacks. Yeh et al. [19] and Chang and Hai [20] proposed 3FA schemes for WSN to resist various known attacks, but these schemes suffer from smart card loss attacks, impersonation attacks, and so on. So, Challa et al. [21] proposed the signature-based authentication scheme to achieve security, but the computation cost is high. In 2021, Tanveer et al. [22] proposed a lightweight user authentication and key exchange scheme for smart home, and Xie et al. [23] designed an ECC-based secure and privacy-protected authentication protocol for smart city. Shuai et al. [24] proposed a 3FA scheme for WSN, which uses a bio-hash function to enhance security.

1.1. Motivations and Contributions. In 2021, Shuai et al. [24] proposed a lightweight 3FA anonymous authentication scheme; however, we pointed out that Shuai et al.'s scheme is vulnerable to stolen-verifier attack, modification of messages attack, and no perfect forward security. To solve these problems, we propose a new 3FA scheme based on ECC and Fuzzy Extractor algorithm. We summarize our contributions as follows:

(1) We pointed out that Shuai et al.'s scheme suffers from the stolen-verifier attack, modification of messages attack, and no perfect forward security

(2) A new three-factor authentication scheme based on ECC and fuzzy extractor algorithm used for WSNs is proposed

(3) We use formal verification tool ProVerif [25] which is based on applied pi calculus to prove the security of the proposed scheme

(4) The informal security analysis shows that the proposed scheme can resist various known attacks

(5) We evaluate the computational cost of the proposed scheme with some related schemes; the result shows that the proposed scheme has better performance

1.2. Attack Model. Referring to the Dolev-Yao threat model [26], we present the abilities of an adversary as follows:

(1) U_A has the ability to eavesdrop on all the messages which are transmitted via an open channel

(2) U_A can modify, insert, replay, modify, and reroute the eavesdropped messages

(3) If U_A obtains the smart card of the user U_i, he/she can get all the data kept in the smart card

(4) U_A can obtain all data stored in sensor node if U_A captures a sensor node

(5) U_A maybe an insider attacker

The rest of the paper is as follows. We review the scheme of Shuai et al. in Section 2. Section 3 shows the security analysis of Shuai et al. 's scheme. We propose the new scheme in Section 4. Sections 5 and 6 present the informal and formal security analyses of the proposed scheme. In Section 7, we exhibit the performance analysis between the proposed scheme and some related schemes. Finally, the paper concludes in Section 8.

2. Review the Shuai et al.'s Scheme

Shuai et al.'s scheme [24] consists of three phases: registration phase, login and authentication phase, and password change phase.

2.1. Registration Phase. The registration phase includes user (may be health professional) registration and medical sensor node registration. The user registration phase is as follows:

Step UR1: the user U_i chooses identity ID_i and inputs password PW_i and fingerprint fg_i via the sensor device; the device generates a random number m_i. After that, the device computes $MB_i = BH(m_i \| fg_i)$ and $\mathrm{MPW}_i = h(\mathrm{ID}_i \| \mathrm{PW}_i \| MB_i \| m_i)$ and then sends $\{\mathrm{ID}_i, \mathrm{MPW}_i\}$ and the personal credential to GWN via a private channel.

Step UR2: once the message is received, GWN generates random numbers n_i, r_i, and K_1 and computes $\mathrm{HID}_i = \mathrm{ID}_i \oplus r_i$, $X_i = h(\mathrm{ID}_i \| K_1 \| n_i)$, $Y_i = X_i \oplus \mathrm{MPW}_i$, and $V_i = h(X_i \| \mathrm{MPW}_i)$. GWN stores $\{\mathrm{ID}_i, \mathrm{HID}_i, n_i, K_1\}$ and user's credential in its memory and stores $\{\mathrm{HID}_i, Y_i, V_i, K_1, h(.), \mathrm{BH}(.)\}$ into a smart card; GWN issues the smart card to U_i via a private channel.

Step UR3: once the smart card is received, U_i writes m_i into the smart card. At the end of the user registration phase, the smart card contains $\{\mathrm{HID}_i, Y_i, V_i, K_1, h(.), BH(.), m_i\}$.

The registration phase of sensor node is as follows.

Step SR1: the medical sensor node SN_j chooses identity SID_j and sends it to GWN via a private channel.

Step SR2: on receiving SID_j, GWN first checks the uniqueness of the SID_j; if the SID_j is not unique, it refuses the registration request. Otherwise, GWN generates a random number K_2 and stores $\{\mathrm{SID}_j, K_2\}$ in its memory. Then, GWN transmits K_2 to SN_j via a private channel.

Step SR3: on receiving K_2, SN_j stores K_2.

2.2. Login and Authentication Phase

Step LA1: the user U_i inserts the smart card and enters identity ID_i, password PW_i, and fingerprint fg_i. The smart card computes $MB_i^* = \mathrm{BH}(m_i \| \mathrm{fg}_i)$, $\mathrm{MPW}_i^* = h(\mathrm{ID}_i \| \mathrm{PW}_i \| MB_i^* \| m_i)$, $X_i^* = Y_i \oplus \mathrm{MPW}_i^*$, and $V_i^* = h(X_i^* \| \mathrm{MPW}_i^*)$ and checks if V_i^* and V_i are equal. If not, it terminates the session. Otherwise, proceed to the next step.

Step LA2: if the user U_i is legal, the smart card generates a random number R and current timestamp T_1; U_i selects an identity SID_j of sensor node SN_j; the smart card computes $\mathrm{UG} = h(\mathrm{HID}_i \| X_i \| K_1)$, $M_1 = E_{\mathrm{UG}}(R \| \mathrm{SID}_j)$, and $CK_1 = h(\mathrm{ID}_i \| R \| X_i \| \mathrm{HID}_i \| K_1 \| T_1)$. Then, U_i sends message $\{\mathrm{HID}_i, M_1, CK_1, T_1\}$ to GWN via a public channel.

Step LA3: on receiving the message from U_i, GWN checks the time stamp T_1 first. GWN gets the current time T_1^* and compares with T_1 if $|T_1^* - T_1| > \Delta T$, where ΔT is the predefined threshold value, and GWN terminates the session. Otherwise, according to HID_i, GWN extracts identity ID_i, random number n_i, and K_1 of user U_i from the storage table. Then, GWN computes $X_i = h(\mathrm{ID}_i \| K \| n_i)$, $\mathrm{UG} = h(\mathrm{HID}_i \| X_i \| K_1)$, $(R^* \| \mathrm{SID}_j) = D_{\mathrm{UG}}(M_1)$, and $\mathrm{CK}_1^* = h(\mathrm{ID}_i \| R^* \| X_i \| \mathrm{HID}_i \| K_1 \| T_1)$ and compares CK_1^* with CK_1. If they are not equal, terminate the session. Otherwise, the user U_i is legal. In addition, GWN generates a timestamp T_2 and session key SK and computes $M_2 = (\mathrm{SK} \| \mathrm{ID}_i) \oplus h(K_2 \| \mathrm{SID}_j)$ and $\mathrm{CK}_2 = h(\mathrm{ID}_i \| \mathrm{SID}_j \| \mathrm{SK} \| \mathrm{K}_2 \| T_2)$. Finally, GWN sends the message $\{M_2, \mathrm{CK}_2, T_2\}$ to the sensor node SN_j via an open channel.

Step LA4: on receiving the message $\{M_2, \mathrm{CK}_2, T_2\}$, SN_j gets the current time T_2^* and compares with T_2. If $|T_2^* - T_2| > \Delta T$, terminate the session. Otherwise, SN_j computes $(\mathrm{SK} \| \mathrm{ID}_i) = M_2 \oplus h(K_2 \| \mathrm{SID}_j)$ and $\mathrm{CK}_2^* = h(\mathrm{ID}_i \| \mathrm{SID}_j \| \mathrm{SK} \| \mathrm{K}_2 \| T_2)$. Then, SN_j compares CK_2^* with CK_2. If they are not equal, terminate the session. Otherwise, SN_j generates a timestamp T_3 and computes $\mathrm{CK}_3 = h(\mathrm{SID}_j \| \mathrm{ID}_i \| \mathrm{SK} \| T_3)$. Finally, SN_j updates $K_2 = h(K_2)$ and sends the message $\{\mathrm{CK}_3, T_3\}$ to GWN via an open channel.

Step LA5: on receiving the message $\{\mathrm{CK}_3, T_3\}$, GWN gets the current time T_3^*; if $|T_3^* - T_3| < \Delta T$, compute $\mathrm{CK}_3^* = h(\mathrm{SID}_j \| \mathrm{ID}_j \| \mathrm{SK} \| T_3)$. Then, GWN compares CK_3^* with CK_3. If they are not equal, terminate the session. Otherwise, GWN generates a random number r_i^* and T_4 and computes $\mathrm{HID}_i^* = \mathrm{ID}_i \oplus r_i^*$, $\mathrm{GU} = h(\mathrm{R} \| \mathrm{HID}_i \| X_i \| K_1)$, $M_3 = E_{\mathrm{GU}}(\mathrm{SK} \| \mathrm{HID}_i^* \| \mathrm{SID}_j)$, and $\mathrm{CK}_4 = h(\mathrm{ID}_i \| \mathrm{SK} \| \mathrm{HID}_i \| T_4)$. Then, GWN updates K_1, K_2, and HID_i with $K_1 = h(K_1)$, $K_2 = h(K_2)$, and $\mathrm{HID}_i = \mathrm{HID}_i^*$. Finally, GWN sends the message $\{M_3, \mathrm{CK}_4, T_4\}$ to U_i via an open channel.

Step LA6: on receiving the message $\{M_3, \mathrm{CK}_4, T_4\}$, U_i gets the current time T_4^*; if $|T_4^* - T_4| < \Delta T$, compute $\mathrm{GU} = h(R \| \mathrm{HID}_i \| X_i \| K_1)$, $(\mathrm{SK} \| \mathrm{HID}_i^* \| \mathrm{SID}_j) = D_{\mathrm{GU}}(M_3)$, and $\mathrm{CK}_4^* = h(\mathrm{ID}_i \| \mathrm{SK} \| \mathrm{HID}_i \| T_4)$. Then, U_i compares CK_4^* with CK_4. If they are equal, U_i updates K_1 and HID_i with $K_1 = h(K_1)$ and $\mathrm{HID}_i = \mathrm{HID}_i^*$ and completes the authentication.

2.3. Password Change Phase

Step PC1: the user U_i inserts the smart card and enters identity ID_i, password PW_i, and fingerprint fg_i. The smart card computes $MB_i = BH(m_i \| \mathrm{fg}_i)$, $\mathrm{MPW}_i = h(\mathrm{ID}_i \| \mathrm{PW}_i \| MB_i \| m_i)$, $X_i = Y_i \oplus \mathrm{MPW}_i$, and $V_i^* = h(X_i \| \mathrm{MPW}_i)$ and compares V_i^* with V_i, which is stored in the smart card. If the values are equal, the smart card allows U_i to enter a new password PW_i^*. Otherwise, it rejects the request for password change.

Step PC2: the smart card computes $\mathrm{MPW}_i^* = h(\mathrm{ID}_i \| \mathrm{PW}_i^* \| MB_i \| m_i)$, $Y_i^* = X_i \oplus \mathrm{MPW}_i^* = Y_i \oplus \mathrm{MPW}_i \oplus \mathrm{MPW}_i^*$, and $V_i^* = h(X_i \| \mathrm{MPW}_i^*)$.

Step PC3: finally, the smart card deletes Y_i and V_i and stores Y_i^* and V_i^*.

3. Analysis of the Shuai et al.'s Scheme

In this section, we will show that Shuai et al.'s protocol has some security flaws.

3.1. Modification of Messages/Desynchronization Attack. In Shuai et al.'s scheme, SN_j updates $K_2 = h(K_2)$ and sends the message $\{\mathrm{CK}_3, T_3\}$ to GWN via an open channel in Step LA5. On receiving the message $\{\mathrm{CK}_3, T_3\}$, GWN gets the current time T_3^*; if $|T_3^* - T_3| < \Delta T$, compute $\mathrm{CK}_3^* = h(\mathrm{SID}_j \| \mathrm{ID}_j \| \mathrm{SK} \| T_3)$. If $\mathrm{CK}_3^* = \mathrm{CK}_3$, GWN updates $K_2 = h(K_2)$. Suppose an attacker U_A intercepts or changes information $\{\mathrm{CK}_3, T_3\}$, GWN will not update $K_2 = h(K_2)$ before the session terminated. Therefore, SN_j and GWN store different K_2. The sensor node SN_j is paralyzed.

The same attack method can be used between GWN and the user U_i. If an attacker U_A intercepts or changes information $\{M_3, \mathrm{CK}_4, T_4\}$ between Step LA5 and Step LA6, U_i will not update the value of K_1. However, GWN has updated K_1 already. Later on, U_i cannot pass the authentication of GWN.

3.2. Stolen-Verifier Attack. In their scheme, GWN stores $\{\mathrm{SID}_j, K_2\}$. SID_j is the identity of sensor node SN_j; the random number K_2 is generated by GWN for the sensor node SN_j.

Assuming that SID_j and K_2 of each node is known by the attacker U_A, U_A can eavesdrop on $\{M_2, \mathrm{CK}_2, T_2\}$ via an open channel. By computing $(\mathrm{SK} \| \mathrm{ID}_i) = M_2 \oplus h(K_2 \| \mathrm{SID}_j)$, the attacker U_A gets session key SK and user's identity ID_i.

If attacker U_A knows $\{\mathrm{SID}_j, K_2\}$, he/she can intercept all messages and impersonate any sensor node. After knowing $\{\mathrm{SID}_j, K_2\}$, U_A can forge $\{M_2, \mathrm{CK}_2, T_2\}$ and send the message to the sensor node SN_j, where $M_2 = (\mathrm{SK} \| \mathrm{ID}_i) \oplus h(K_2 \| \mathrm{SID}_j)$ and $\mathrm{CK}_2 = h(\mathrm{ID}_i \| \mathrm{SID}_j \| \mathrm{SK} \| K_2 \| T_2)$. SK, ID_i, and T_2 can be randomly generated by the attacker U_A. The sensor node verifies the message by computing $(\mathrm{SK} \| \mathrm{ID}_i) = M_2 \oplus h(K_2 \| \mathrm{SID}_j)$ and $\mathrm{CK}_2^* = h(\mathrm{ID}_i \| \mathrm{SID}_j \|$

$SK\|K_2\|T_2$) and checks if $CK_2^* = CK_2$. There is no doubt that they are equal. Then, the sensor node updates $K_2 = h(K_2)$ and cannot respond to the legitimate request. Finally, the sensor node is paralyzed.

So, if an attacker U_A can get access to the database, he/she can obtain session key SK, impersonate sensor nodes, or paralyze sensor nodes.

3.3. No Perfect Forward Security. In Shuai et al.'s scheme, if an attacker U_A obtains the secret random number K_2 stored in the sensor node SN_j, he/she can get the current session key SK by computing $(SK\|ID_i) = M_2 \oplus h(K_2\|SID_j)$, where SID_j is the identity of SN_j and M_2 is transmitted via an open channel and can be eavesdropped on by the attacker U_A. The next long-term key K_2^* is updated by $K_2^* = h(K_2)$. It is easy for the attacker U_A to eavesdrop next M_2^* via an open channel; then, the next session key SK^* can be computed by $(SK^*\|ID_i) = M_2^* \oplus h(K_2^*\|SID_j)$. Therefore, the scheme of Shuai et al. cannot provide perfect forward/backward security.

4. Our Proposed Scheme

In this section, we propose a new three-factor anonymous authentication scheme using ECC and fuzzy extractor algorithm. Table 1 shows the notations and intuitive abbreviations mentioned in the proposed scheme.

4.1. System Setup Phase. GWN chooses an elliptic curve $E(GF_q)$ defined over $GF(q)$, where $GF(q)$ is a finite field defined over a large prime number q. P is a generator point on the curve. GWN chooses a secret parameter K_{GWN}. GWN computes public key as $PK_G = K_{GWN} \cdot P$ and publishes $Rep(.)$, $Gen(.)$, $h(.)$, and PK_G, where $Rep(.)$ and $Gen(.)$ are reproduction and generation algorithm of fuzzy extractor algorithm, respectively. $h(.)$ is a hash function.

4.2. User Registration Phase

Step UR1: U_i chooses its ID_i and sends ID_i to GWN via a private channel.

Step UR2: GWN verifies the effectiveness and legitimacy of ID_i; if not, GWN requests U_i to choose a new ID_i. Otherwise, GWN computes $a_i = h(ID_i\|K_{GWN})$. GWN stores the information $\{a_i, PK_G, P\}$ into a smart card (SC) and transmits it to U_i.

Step UR3: U_i inserts the SC into a card reader and enters its ID_i, PW_i, and fingerprint fng_i; the device computes

$$\begin{aligned} (\sigma_i, \tau_i) &= Gen(fng_i), \\ MPW_i &= h(ID_i\|PW_i\|\sigma_i), \\ F_i &= a_i \oplus h(ID_i\|\sigma_i\|PW_i). \end{aligned} \tag{1}$$

Table 1: Notations.

Notations	Description
U_i	i^{th} User
U_A	Adversary
SN_j	j^{th} sensor node
GWN	Gateway node
ID_i	Unique identity of U_i
PW_i	Password of U_i
fng_i	Biometric information of U_i
SID_j	Unique identity of SN_j
SK	Session key
K_{GWN}	GWN's secret parameter
T_1, T_2, T_3, T_4	Timestamp
$h(.)$	Hash function
$\|$	Concatenation
$\oplus$	XOR operation
P	The generator point on the curve
$Rep(.)$, $Gen(\cdot)$	Fuzzy extractor algorithm for reproduction and generation
τ_i	Reproduction parameter of fuzzy extractor algorithm
σ_i	Biometric key of fuzzy extractor algorithm
ΔT	The transmission delay time

Then, U_i updates a_i with F_i. Finally, $\{MPW_i, \tau_i, F_i, PK_G, P\}$ are stored in SC.

4.3. Sensor Node Registration Phase

Step SR1: GWN chooses a unique identity SID_j for sensor node SN_j and computes $b_j = h(SID_j\|K_{GWN})$. Then, GWN sends $\{b_j, SID_j, P\}$ to SN_j via a private channel.

Step SR2: upon receiving $\{b_j, SID_j, P\}$, SN_j stores them into its memory.

4.4. Login and Authentication Phases

Step LA1: U_i inserts the smart card into the device and inputs the identity ID_i^* and the password PW_i^* and enters the fingerprint fng_i^*. Then, the device calculates

$$\begin{aligned} \sigma_i^* &= Rep(fng_i^*, \tau_i), \\ MPW_i^* &= h(ID_i^*\|PW_i^*\|\sigma_i^*). \end{aligned} \tag{2}$$

If $MPW_i^* \neq MPW_i$, SC refuses the login request of U_i. Otherwise, go on.

Step LA2: U_i creates a random number m_i and computes

$$\begin{aligned} a_i^* &= F_i \oplus h(ID_i^*\|\sigma_i^*\|PW_i^*), \\ M_1 &= m_i \cdot P, \\ M_2 &= (ID_i^*\|SID_j) \oplus h(m_i \cdot PK_G\|T_1), \\ M_3 &= h(a_i^*\|M_1\|M_2\|T_1), \end{aligned} \tag{3}$$

where PK_G is the public key of GWN, T_1 is the current timestamp, And U_i sends the message

$\text{MES}_1 = \{M_1, M_2, M_3, T_1\}$ to GWN via a public channel.

Step LA3: on receiving MES_1, GWN first checks the timestamp. GWN creates the current time T_1^*; if $|T_1^* - T_1| > \Delta T$, terminate the session. Otherwise, GWN computes

$$\begin{aligned} \left(\text{ID}_i', \text{SID}_j'\right) &= M_2 \oplus h\left(M_1 \cdot k_G \| T_1\right), \\ a_i' &= h\left(ID_i' \| K_{\text{GWN}}\right), \\ M_3' &= h\left(a_i' \| M_1 \| M_2 \| T_1\right). \end{aligned} \tag{4}$$

If $M_3' \neq M_3$, GWN declines the request. Otherwise, GWN generates the current time T_2 and calculates

$$\begin{aligned} S_i &= h\left(a_i' \| \text{SID}_j' \| T_2\right), \\ b_j' &= h\left(\text{SID}_j' \| K_{\text{GWN}}\right), \\ N_1 &= \left(S_i \| \text{ID}_i'\right) \oplus h\left(b_j' \| \text{SID}_j' \| T_2\right), \\ N_2 &= h\left(M_1 \| \text{ID}_i' \| S_i \| \text{SID}_j' \| T_2\right). \end{aligned} \tag{5}$$

GWN transmits the message $\text{MES}_2 = \{M_1, N_1, N_2, T_2\}$ to SN_j via an open channel.

Step LA4: after obtaining the message $\text{MES}_2 = \{M_1, N_1, N_2, T_2\}$, SN_j checks whether $|T_2^* - T_2| \leq \Delta T$, where T_2^* is the current timestamp. If not, SN_j rejects the session. Otherwise, SN_j computes

$$\begin{aligned} \left(S_i' \| \text{ID}_i''\right) &= N_1 \oplus h\left(b_j \| \text{SID}_j \| T_2\right), \\ N_2' &= h\left(M_1 \| \text{ID}_i'' \| S_i' \| \text{SID}_j \| T_2\right). \end{aligned} \tag{6}$$

If $N_2' \neq N_2$, terminate the session. Otherwise, SN_j generates a random number c_j and the current time T_3 and computes

$$\begin{aligned} N_3 &= c_j \cdot P, \\ \text{SK}_j &= h\left(c_j \cdot M_1 \| \text{SID}_j \| \text{ID}_i'' \| S_i'\right), \\ N_4 &= h\left(S_i' \| \text{SK}_j \| N_3 \| \text{ID}_i'' \| T_3\right). \end{aligned} \tag{7}$$

SN_j sends the message $\text{MES}_3 = \{N_3, N_4, T_2, T_3\}$ to U_i via an open channel.

Step LA5: upon receiving the message $MES_3 = \{N_3, N_4, T_2, T_3\}$, U_i generates the current timestamp T_3^* and ensures that $|T_3^* - T_3| \leq \Delta T$; if it is not, reject the session; otherwise, U_i computes

$$\begin{aligned} S_i^* &= h\left(a_i^* \| \text{SID}_j \| T_2\right), \\ \text{SK}_u &= h\left(m_i \cdot N_3 \| \text{SID}_j \| \text{ID}_i^* \| S_i^*\right), \\ N_4' &= h\left(S_i^* \| \text{SK}_u \| N_3 \| \text{ID}_i^* \| T_3\right). \end{aligned} \tag{8}$$

If $N_4' \neq N_4$, terminate the session. Otherwise, the authentication is completed. Figure 1 demonstrates the steps of the mutual authentication and the key agreement phase.

5. Informal Security Analysis

In this section, we discuss the possible attacks on the proposed scheme.

5.1. Stolen and Hyphen: Verifier Attack. In our proposed scheme, GWN does not store information related to the verification table. Therefore, there is no stolen-verifier attack against our proposed scheme.

5.2. Offline Password Guessing Attack. In our proposed scheme, MES_1, MES_2, MES_3, and MES_4 are transmitted via an open channel; even if an attacker eavesdrops on the communication and obtains these messages, he/she cannot guess the password. Because the password and fingerprint are used in login verification and not transmitted openly. Though an attacker obtains the message $\{\text{MPW}_i, \tau_i, F_i, \text{PK}_G, P\}$ stored in smart card, where $\text{MPW}_i = h(\text{ID}_i \| PW_i \| \sigma_i)$ and $F_i = a_i \oplus h(\text{ID}_i \| \sigma_i \| \text{PW}_i)$, he/she cannot verify whether the guessed password is correct without knowing the biometric key σ_i.

5.3. Replay Attack. Suppose that an adversary U_A impersonates user U_i and intercepts and replays $\text{MES}_1 = \{M_1, M_2, M_3, T_1\}$. The replayed MES_1 cannot pass the GWN's verification process if the timestamp is invalid. Even if a replay of MES_1 worked, and U_A gets MES_3; however, the session key $\text{SK}_u = h(m_i \cdot N_3 \| \text{SID}_j \| \text{ID}_i^* \| S_i^*)$, where $S_i^* = h(a_i^* \| \text{SID}_j \| T_2)$ and m_i is a random number created by U_i. U_A cannot obtain a_i^* or m_i. Therefore, it is useless to replay MES_1.

Suppose that U_A replays GWN's messages or sensor nodes' messages. First, the replayed messages cannot pass the validity verification of the timestamp. In addition, U_i, GWN, and SN_j generate new random numbers in a new session, which are used in the verification and generation of the session key. Therefore, our scheme is resistant to replay attacks.

5.4. Forger Attack and Impersonation Attack. Suppose an attacker impersonates the user U_i and sends $\text{MES}_1 = \{M_1, M_2, M_3, T_1\}$ to GWN, where $M_3 = h(a_i^* \| M_1 \| M_2 \| T_1)$ and $a_i^* = F_i \oplus h(\text{ID}_i^* \| \sigma_i^* \| \text{PW}_i^*)$; if the attacker does not have ID_i^*, PW_i^*, and fng_i^*, he/she cannot forge M_3. In other words, the attacker cannot impersonate a user.

If the attacker tries to impersonate GWN and forge $\text{MES}_2 = \{M_1, N_1, N_2, T_2\}$, where $N_2 = h(M_1 \| S_i \| \text{ID}_i' \| \text{SID}_j' \| T_2)$ and $b_j' = h(\text{SID}_j' \| K_{\text{GWN}})$, the attacker does not know K_{GWN}, so the forged N_2 cannot pass the verification of SN_j.

If the attacker impersonates the sensor node, he/she cannot forge valid $N_4 = h(S_i' \| \text{SK}_j \| N_3 \| ID_i'' \| T_3)$ without knowing SID_j and b_j.

5.5. Smart Card Loss Attack. Suppose the smart card stolen by an attacker U_A; U_A can get $\langle \text{MPW}_i, \tau_i, F_i, \text{PK}_G, P \rangle$, where $\text{MPW}_i = h(\text{ID}_i \| \text{PW}_i \| \sigma_i)$, τ_i is the reproduction parameter of the fuzzy extractor algorithm, $F_i = a_i \oplus h(\text{ID}_i \| \sigma_i \| \text{PW}_i)$, PK_G is the public key of GWN, and P is the base point of the elliptic curve. MPW_i and F_i are protected by the user's biometric information and password. Therefore, an attacker

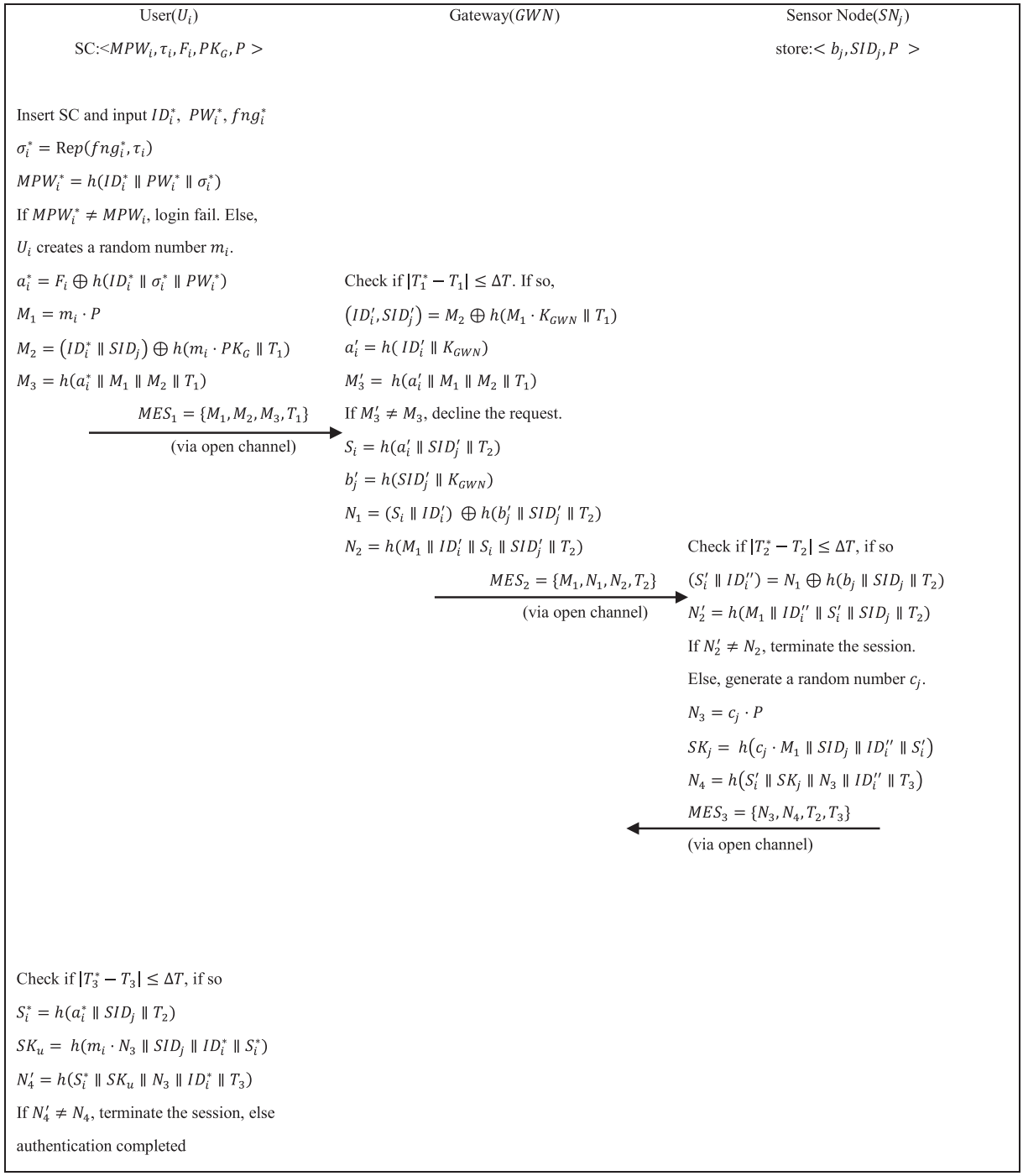

Figure 1: Mutual authentication and key agreement phase.

cannot get any plaintext information or pass through the verification without knowing ID_i, PW_i, and fng_i.

5.6. Sensor Node Capture Attack. In the proposed scheme, each sensor node SN_j stores $\{b_j, \mathrm{SID}_j, P\}$, where $b_j = h(\mathrm{SID}_j \| K_{\mathrm{GWN}})$, SID_j is the identity of the sensor, and P is the base point on the curve. An attacker cannot get K_{GWN} even if he/she captures the sensor. In other words, capturing a sensor node cannot influence other sensor nodes. Therefore, the proposed scheme resists sensor capture attacks.

5.7. Known-Key Attack. The session key $\mathrm{SK}_j = h(c_j \cdot M_1 \| \mathrm{SID}_j \| ID_i'' \| S_i') = \mathrm{SK}_u = h(m_i \cdot N_3 \| \mathrm{SID}_j \| \mathrm{ID}_i^* \| S_i^*)$, where c_j and m_i are random numbers generated in every session, and the CDH problem is intractable. Therefore, even if an attacker gets session keys, he/she cannot solve the CDH problem.

```
(*--The two public channel--*)
free chn1:channel.
free chn2:channel.
(*--The basic type--*)
type User.  (*--type participant.--*)
type Server.
type Sensor.
type key.
type nonce.
type fingerprint.
type timestamp.
(*--The basic variables--*)
free IDi: bitstring[private].(*--The identify of user--*)
free PWi: bitstring[private].(*--The password of user--*)
free fng: fingerprint[private].(*--The fingerprint of user--*)
free Kgwn: bitstring[private]. (*--The secret parameter of GWN--*)
free SKu: bitstring[private]. (*--The session key of user--*)
free SKj: bitstring[private]. (*--The session key of sensor--*)
free SIDj: bitstring. (*-- The identity of Sensor Node--*)
free P: bitstring. (*--The base point--*)
free user: User.
free server: Server.
free sensor: Sensor.
(*--Hash operation--*)
fun Hash(bitstring): bitstring.
(*--Fuzzy Extractor algorithm operation--*)
fun BH(bitstring,fingerprint): bitstring.
fun Gen(fingerprint): bitstring.
fun Rep(fingerprint,bitstring): bitstring.
(*--Bit operation--*)
fun bit_timestamp(timestamp): bitstring.
fun bit_key(key): bitstring.
fun bit_nonce(nonce): bitstring.
fun key_bit(bitstring): key.
(*--ECC operation--*)
fun EccMul(bitstring, bitstring): bitstring.
fun EccAdd(bitstring, bitstring): bitstring.
reduc forall p: bitstring, m1: bitstring, d: key, m2: bitstring;
EccSub(EccAdd(p,m1),d,m2)=p.
(*--XOR operation--*)
fun XOR(bitstring, bitstring): bitstring.
equation forall x: bitstring, y: bitstring;
XOR(XOR(x, y), y) = x.
(*--Concat operation--*)
fun Con(bitstring, bitstring): bitstring.
reduc forall x:bitstring, y:bitstring;
Split(Con(x, y))=(x,y).
(*--Check timestamp Fresh operation--*)
fun checktimestampfresh(bitstring, bool): bool
reduc forall T: bitstring;
checktimestampfresh(T, true) = true
otherwise forall T: bitstring;
checktimestampfresh(T, false) = false.
```

Figure 2: Definitions.

5.8. Anonymity and Unlinkability. In the authentication phase of the proposed scheme, the user's identity is contained in the message $\mathrm{MES}_1 = \{M_1, M_2, M_3, T_1\}$, where $M_1 = m_i \cdot P$, $M_2 = (\mathrm{ID}_i^* \| \mathrm{SID}_j) \oplus h(m_i \cdot \mathrm{PK}_G \| T_1)$, and $M_3 = h(a_i^* \| M_1 \| M_2 \| T_1)$. The user's identity ID_i^* is protected by $h(m_i \cdot \mathrm{PK}_G \| T_1)$; only the gateway can obtain the user's real identity. So, our scheme meets the requirement of anonymity. At the same time, because the random number

```
(*--events--*)
event ULoginPhase(User).
event UAuthenticationPhase(User).
event UserSessionKey(User).
event SNSessionKey(Sensor).
event GWNAuthentication(Server).
(*--queries--*)
query attacker(SKj).
query attacker(SKu).
query attacker(PWi).
query attacker(Kgwn).
query inj-event(UAuthenticationPhase(user)) ==> inj-event(ULoginPhase(user)).
query inj-event(GWNAuthentication(server)) ==> inj-event(UAuthenticationPhase(user)).
query inj-event(SNSessionKey(sensor)) ==> inj-event(GWNAuthentication(server)).
query inj-event(UserSessionKey(user)) ==> inj-event(SNSessionKey(sensor)).
```

Figure 3: Events and queries.

```
(*--process of user--*)
let UserProcess(IDi:bitstring, PWi:bitstring, fng:fingerprint, MPWi:bitstring, taoi:bitstring, Fi:bitstring, PKg:bitstring,
P:bitstring, SIDj:bitstring)=
        let sigma=Rep(fng,taoi) in
        let nMPWi=Hash(Con(IDi,Con(PWi,sigma))) in
        if nMPWi=MPWi then
                event ULoginPhase(user);
        new rmi: nonce;
        new Time1: timestamp;
        let mi=bit_nonce(rmi) in
    let T1=bit_timestamp(Time1) in
        let ai=XOR(Fi,Hash(Con(IDi,Con(sigma,PWi)))) in
        let M1=EccMul(mi,P) in
        let M2=XOR(Con(IDi,SIDj),Hash(Con(EccMul(mi,PKg),T1))) in
        let M3=Hash(Con(ai,Con(M1,Con(M2,T1)))) in
        out (chn1,(M1,M2,M3,T1));
        event UAuthenticationPhase(user);
        in (chn1,(N3:bitstring,N4:bitstring,T2:bitstring,T3:bitstring));
        if checktimestampfresh(T3, true) then
                let nSi=Hash(Con(ai,Con(SIDj,T2))) in
                let SKu=Hash(Con(EccMul(mi,N3),Con(SIDj,Con(IDi,nSi)))) in
                let nN4=Hash(Con(nSi,Con(SKu,Con(N3,Con(IDi,T3))))) in
                if nN4=N4 then
                event UserSessionKey(user).
```

Figure 4: Process of the user.

m_i and the timestamp T_1 are contained in M_2, which is changed in each session, therefore, our scheme is also unlinkability.

5.9. Perfect Forward Secrecy. In the proposed scheme, the session key $SK = h(c_j \cdot m_i \cdot P \| SID_j \| ID_i \| S_i')$. Even if an adversary can know the user's all secret information and the secret key of GWN, c_jP, and m_iP, but he/she still cannot compute $c_j \cdot m_i \cdot P$ because of the intractability of the computational Diffie–Hellman (CDH) problem. So, the proposed scheme can achieve perfect forward secrecy.

6. Formal Security Analysis Using ProVerif

ProVerif is a formalized cryptographic protocol verification tool based on the Dolev–Yao model, which can describe various cryptographic primitives. When using the ProVerif tool to validate a cryptographic protocol, the tool will present a corresponding sequence of attacks if the protocol is vulnerable.

As shown in Figure 2, we defined channels, basic types, and functions. The proposed scheme involves 5 events, namely, ULoginPhase(), UAuthenticationPhase(), UserSessionKey(), SNSessionKey(), and GWNAuthentication().

```
(*--process of GWN--*)
let GWNProcess(PKg:bitstring, P:bitstring, Kgwn:bitstring)=
        in (chn1,(M1:bitstring,M2:bitstring,M3:bitstring,T1:bitstring));
        if checktimestampfresh(T1, true) then
                let (IDi:bitstring,SIDj:bitstring)=Split(XOR(M2,Hash(Con(EccMul(M1,Kgwn),T1)))) in
                let nai=Hash(Con(IDi,Kgwn)) in
                let nM3=Hash(Con(nai,Con(M1,Con(M2,T1)))) in
                if nM3=M3 then
                        event GWNAuthentication(server);
        new Time2: timestamp;
    let T2=bit_timestamp(Time2) in
    let Si=Hash(Con(nai,Con(SIDj,T2))) in
        let bj=Hash(Con(SIDj,Kgwn)) in
        let N1=XOR(Con(Si,IDi),Hash(Con(bj,Con(SIDj,T2)))) in
        let N2=Hash(Con(M1,Con(IDi,Con(Si,Con(SIDj,T2))))) in
        out (chn2,(M1,N1,N2,T2)).
```

Figure 5: Process of GWN.

```
(*--process of Sensor Node--*)
let SensorProcess(bj:bitstring, SIDj:bitstring, P:bitstring)=
        in (chn2,(M1:bitstring,N1:bitstring,N2:bitstring,T2:bitstring));
        if checktimestampfresh(T2, true) then
        let (Si:bitstring,IDi:bitstring)=Split(XOR(N1,Hash(Con(bj,Con(SIDj,T2))))) in
        let nN2=Hash(Con(M1,Con(IDi,Con(Si,Con(SIDj,T2))))) in
        if nN2=N2 then
                new cmj:nonce;
                new Time3:timestamp;
                let cj=bit_nonce(cmj) in
    let T3=bit_timestamp(Time3) in
                let N3=EccMul(cj,P) in
                let SKj=Hash(Con(EccMul(cj,M1),Con(SIDj,Con(IDi,Si)))) in
                let N4=Hash(Con(Si,Con(SKj,Con(N3,Con(IDi,T3))))) in
                event SNSessionKey(sensor);
                out (chn1,(N3,N4,T2,T3)).
```

Figure 6: Process of sensor node.

```
(*--Main process--*)
process
        let ai=Hash(Con(IDi,Kgwn)) in
        let PKg=EccMul(Kgwn,P) in
        let bj=Hash(Con(SIDj,Kgwn)) in
        let (sigma:bitstring,taoi:bitstring)=Gen(fng) in
        let MPWi=Hash(Con(IDi,Con(PWi,sigma))) in
        let Fi=XOR(ai,Hash(Con(IDi,Con(sigma,PWi)))) in
        (
        (!UserProcess(IDi,PWi,fng,MPWi,taoi,Fi,PKg,P,SIDj)) |
        (!GWNProcess(PKg,P,Kgwn)) |
        (!SensorProcess(bj,SIDj,P))
    )
```

Figure 7: Main process.

```
------------------------------------------------------------
Verification summary:
Query not attacker(SKj[]) is true.
Query not attacker(SKu[]) is true.
Query not attacker(PWi[]) is true.
Query not attacker(Kgwn[]) is true.
Query inj-event(UAuthenticationPhase(user[])) ==> inj-event(ULoginPhase(user[])) is true.
Query inj-event(GWNAuthentication(server[])) ==> inj-event(UAuthenticationPhase(user[])) is true.
Query inj-event(SNSessionKey(sensor[])) ==> inj-event(GWNAuthentication(server[])) is true.
Query inj-event(UserSessionKey(user[])) ==> inj-event(SNSessionKey(sensor[])) is true.
------------------------------------------------------------
```

Figure 8: Results.

Table 2: Comparison of our scheme and related schemes in attack/properties.

Attacks/properties	[19]	[20]	[21]	[23]	[24]	Ours
Privileged-insider attack	✓	✗	✓	✓	✓	✓
Offline password guessing attack	✓	✗	✓	✓	✓	✓
Denial-of-service attack	✓	✓	✓	✓	✗	✓
Forger and impersonation attack	✗	✗	✓	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓
Man-in-middle attack	✓	✓	✓	✓	✓	✓
Smart card loss attack	✗	✗	✓	✓	✓	✓
Sensor node capture attack	✗	✓	✓	✓	✓	✓
Stolen-verifier attack	✓	✓	✓	✓	✗	✓
Desynchronization attack	✓	✓	✓	✓	✗	✓
Perfect forward secrecy	✓	✓	✓	✓	✗	✓
Identity anonymity	✗	✓	✓	✓	✓	✓
Mutual authentication	✗	✓	✓	✓	✓	✓
Untraceability	✗	✗	✓	✓	✓	✓

✓, resist (attacks)/possess (properties); ✗, suffer (attacks)/no (properties).

Table 3: The computational cost comparison.

Schemes	U_i(user)	SN_j(sensor)	GWN	Total
[19]	$T_H + 2T_{ECC}$	$3T_H + 2T_{ECC}$	$4T_H + 4T_{ECC}$	$8T_H + 8T_{ECC}$ (20.552ms)
[20]	$7T_H + 2T_{ECC}$	$5T_H + 2T_{ECC}$	$9T_H$	$21T_H + 4T_{ECC}$ (11.432ms)
[21]	$5T_H + 5T_{ECC}$	$3T_H + 4T_{ECC}$	$5T_H + 4T_{ECC}$	$13T_H + 13T_{ECC}$ (33.397ms)
[23]	$8T_H + 3T_{ECC} + T_{SE}$	$5T_H + 2T_{ECC} + T_{SE}$	$7T_H + T_{ECC} + 2T_{SE}$	$20T_H + 6T_{ECC} + 4T_{SE}$ (18.606ms)
[24]	$7T_H + 2T_{SE}$	$10T_H + 2T_{SE}$	$4T_H$	$21T_H + 4T_{SE}$ (3.668ms)
Ours	$7T_H + 3T_{ECC}$	$7T_H + T_{ECC}$	$4T_H + 2T_{ECC}$	$18T_H + 6T_{ECC}$ (16.230ms)

ULoginPhase() indicates the login phase of the user, UAuthenticationPhase() indicates the user sends authentication request, GWNAuthentication() indicates the gateway pass the authentication of the user, SNSessionKey() indicates sensor node agrees on the session key, and UserSessionKey() indicates the user agrees the session key. Figure 3 shows the above events and queries.

The operations of the user, GWN, and sensor node are shown in Figure 4, Figure 5, and Figure 6, respectively. Figure 7 exhibits the main process. According to the result in Figure 8, the proposed scheme can provide security of the session key, the password of the user, and the secret parameter of GWN. Meanwhile, the process of mutual authentication is executed in sequence.

7. Performance Comparison

In this section, we analyze the security and performance comparison between our schemes with some related schemes. Table 2 shows the comparison of attacks/properties of the schemes. Compared with Shuai et al.'s scheme, our scheme is more secure to various known attacks and has some good properties. As shown in Table 3, we can see the comparison of computational cost between the proposed scheme and the related schemes [19–21, 23, 24], where T_H represents hash operation time, T_{SE} is the time of the symmetric encryption/decryption operation, and T_{ECC} denotes the time cost of ECC operation. In the environment [18] of Windows 10 64 bit laptop, Intel (R) Core (TM)

i5-6300HQ CPU @ 2.30 GHz, 12 GB RAM, we get T_H = 0.068ms (millisecond), T_{ECC} = 2.501ms, and T_{SE} = 0.56ms. It can be seen from Table 3 that our scheme takes less time than related schemes. Compared with Shuai et al.'s scheme, our scheme overcomes the problem of Shuai et al.'s scheme, the computation cost is a little more than Shuai et al.'s scheme to achieve the perfect forward secrecy.

8. Conclusion

In this paper, we first pointed out Shuai et al.'s scheme is vulnerable to desynchronization attack, stolen-verifier attack, and no perfect forward security. In addition, we propose a new three-factor authentication using ECC and fuzzy extractor algorithm, which not only defends against the above attacks but also defends other attacks as shown in informal security analysis. We also simulate the proposed scheme for its formal security verification using the ProVerif tool to prove the security. Its performance analysis shows that it has less communication cost than the related schemes, and it can be applied to WSN in IoT. In the future, we will design block chain-based anonymous authentication scheme for WSN in IoT.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key R&D Program of China (Grant no. 2017YFB0802000, URL: http://www.most.gov.cn/) and the National Natural Science Foundation of China (Grant nos. 61702152 and 61702153, URL: http://www.nsfc.gov.cn/).

References

[1] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on emerging and selected topics in circuits and systems*, vol. 3, no. 1, pp. 45–54, 2013.

[2] Q. Xie, W. Liu, S. Wang, L. Han, B. Hu, and T. Wu, "Improvement of a uniqueness-and-anonymity-preserving user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 38, no. 9, 2014.

[3] F. Li and P. Xiong, "Practical secure communication for integrating wireless sensor networks into the internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3677–3684, 2013.

[4] W. K. Seah, Z. A. Eu, and H. P. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges," in *Proceedings of the 2009 1st International Conference on Wireless Communication, Vehicular Technology*, pp. 1–5, IEEE, Aalborg, Denmark, May 2009.

[5] M. Turkanović, B. Brumen, and 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 Networks*, vol. 20, pp. 96–112, 2014.

[6] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.

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

[8] Q. Xie, J. Zhao, and X. Yu, "Chaotic maps-based three-party password-authenticated key agreement scheme," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1021–1027, 2013.

[9] S. A. Chaudhry, "Designing an efficient and secure message exchange protocol for internet of vehicles," *Security and Communication Networks*, vol. 2021, Article ID 5554318, 9 pages, 2021.

[10] M. Rana, A. Shafiq, I. Altaf et al., "A secure and lightweight authentication scheme for next generation IoT infrastructure," *Computer Communications*, vol. 165, pp. 85–96, 2021.

[11] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, 2009.

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

[13] P. Kumar, S. G. Lee, and H. J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[14] D. He, N. Kumar, J. Chen, C. C. Lee, N. Chilamkurti, and S. S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[15] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655, 2016.

[16] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and anonymous two-factor authentication protocol for healthcare applications with wireless medical sensor networks," *Multimedia Systems*, vol. 23, no. 2, pp. 195–205, 2017.

[17] O. Mir, J. Munilla, and S. Kumari, "Efficient anonymous authentication with key agreement protocol for wireless medical sensor networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 79–91, 2017.

[18] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.

[19] H. L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim, and H. W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[20] C. C. Chang and L. D. Hai, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 357–366, 2015.

[21] S. Challa, M. Wazid, A. K. Das et al., "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[22] M. Tanveer, G. Abbas, Z. H. Abbas, M. Bilal, A. Mukherjee, and K. S. Kwak, "LAKE-6SH: lightweight user authenticated key exchange for 6LoWPAN-based smart homes," *IEEE Internet of Things Journal*, vol. 14, no. 8, pp. 1–14, 2021.

[23] Q. Xie, K. Li, X. Tan, L. Han, W. Tang, and B. Hu, "A secure and privacy-preserving authentication protocol for wireless sensor networks in smart city," *EURASIP Journal on Wireless Communications and Networking*, vol. 119, no. 1, pp. 1–17, 2021.

[24] M. Shuai, N. Yu, H. Wang, L. Xiong, and Y. Li, "A lightweight three-factor Anonymous authentication scheme with privacy protection for personalized healthcare applications," *Journal of Organizational and End User Computing*, vol. 33, no. 3, pp. 1–18, 2021.

[25] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW-14)*, pp. 82–96, Cape Breton, NS, Canada, June 2014.

[26] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.