symmetry

MDPI

Article

A Secure Three-Factor Anonymous User Authentication Scheme for Internet of Things Environments

Ya-Fen Chang [1], Wei-Liang Tai [2,*], Po-Lin Hou [1] and Kuan-Yu Lai [1]

[1] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung 40401, Taiwan; cyf@nutc.edu.tw (Y.-F.C.); sfj15932@gmail.com (P.-L.H.); derek.kyk@gmail.com (K.-Y.L.)

[2] Department of Information Communications, Chinese Culture University, Taipei 11114, Taiwan

* Correspondence: dwl@ulive.pccu.edu.tw

Abstract: Internet of Things (IoT) is composed of various kinds of devices such as cars, electrical appliances, machines and sensors. With IoT technologies, devices can exchange information through the network, people are allowed to get information collected by devices without interacting with them, and automatic operations for devices are realized. Because of the variety of IoT devices, some of them possess limited computational capability. On the other hand, data transmission in IoT networks is usually through a public channel. To ensure efficiency and security for IoT environments, Lee et al. proposed a three-factor authentication scheme with hash function and XOR operation. They claimed their scheme possessed superior properties and could resist common attacks. After analyzing their scheme, we find that their scheme is vulnerable to five flaws. In this paper, how these found flaws threaten Lee et al.'s scheme is shown in detail. Then, we propose an improvement to overcome the found flaws and preserve the advantages by employing ECC.

Keywords: Internet of Things (IoT); authentication; replay attack; denial-of-service attack; user untraceability; elliptic curve cryptography (ECC)

Citation: Chang, Y.-F.; Tai, W.-L.; Hou, P.-L.; Lai, K.-Y. A Secure Three-Factor Anonymous User Authentication Scheme for Internet of Things Environments. *Symmetry* **2021**, *13*, 1121. https://doi.org/10.3390/sym13071121

Academic Editors: Teen-Hang Meen, Charles Tijus and Jih-Fu Tu

Received: 8 June 2021
Accepted: 21 June 2021
Published: 23 June 2021

1. Introduction

With the rapid development of network technologies, plenty of new applications are proposed and realized. Internet of Things is a new concept that entities can communicate with each other, and entities include various kinds of devices such as cars, electrical appliances, machines and sensors. With IoT technologies, devices can exchange information through the network, people are allowed to get information collected by devices without interacting with them, and automatic operations for devices are realized. There are many IoT applications such as Machine-to-Machine (M2M) communication, Telemedicine Information System (TMIS), Internet of Vehicles (IoV), Smart Home, Industrial of IoT (IIoT), and Smart City. Because data transmission in IoT networks is usually through a public channel, how to protect the security of transmitted data and user privacy for IoT becomes an urgent issue. On the other hand, anonymity becomes an essential issue. Anonymous communication is designed from physical layer to application layer. Various applications and technologies are designed to meet the goal [1–6].

In 2014, Turkanović et al. proposed a new IoT notion-based authentication and key agreement scheme to protect the security of heterogeneous ad hoc wireless sensor networks [7]. Their scheme uses light-weight operations, hash function and XOR operations, and provides functions including mutual authentication, key agreement, password change and dynamic node addition. They also claimed that their scheme could resist various kinds of threats while reducing cost and ensuring performance at the same time. In 2016, Farash et al. [8] found that Turkanović et al.'s scheme suffers from some security flaws such as user traceability, no sensor node anonymity, stolen smart card attack, disclosure of the session key, man-in-the-middle attack. Farash et al. also proposed a new user authentication and key agreement scheme to overcome the flaw that Turkanović et al.'s scheme suffers from and preserve the advantages of it. However, Amin et al. [9] found that Farash et al.'s scheme is still vulnerable to some weaknesses such as compromised user anonymity, known session-specific temporary information attack, offline password guessing attack using stolen smart cards, and insecure secret keys of gateway nodes. Meanwhile, Amin et al. proposed an anonymous-preserving three-factor authenticated key exchange protocol for wireless sensor networks to overcome the security flaws that Farash et al.'s scheme suffers from. In 2017, Jiang et al. [10] found that Amin et al.'s scheme suffers from loss of smart card attacks (SCLA), offline password guessing attack, lack of user untraceability, and known session-specific temporary information attack. Therefore, they proposed a lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. Jiang et al.'s scheme uses the Rabin cryptosystem to withstand tracking and smart card loss attack, uses biometric technologies to realize local password authentication to defend against offline password guessing attack, and uses time stamp mechanism to resist known session-specific temporary information attack. They claimed that their scheme could overcome the weaknesses of Amin et al.'s scheme and ensure key agreement with the higher computational load.

In 2018, Zhang et al. proposed a privacy protection mechanism for E-Health systems by means of dynamic authentication and three-factor key agreement [11]. They claimed that biometric identification on the server could be performed while the server could not get the value of biometrics. In order to protect user anonymity, dynamic authentication instead of traditional password table is adopted. Zhang et al.'s scheme uses hash function and bio-hash function to reduce computational cost and transmission cost and meet the security requirements of electronic medical systems. However, in 2019, Aghily et al. [12] found that there exist serious security vulnerabilities in Zhang et al.'s scheme, including user traceability, desynchronization attack, internal attack and denial-of-service attack. Therefore, they proposed a lightweight three-factor authentication, access control and ownership transfer scheme for E-Health systems in IoT to overcome the weaknesses that Zhang et al.'s scheme suffers from and provide an access control mechanism such that a patient's current doctor can transfer the corresponding authority to a new doctor.

Inspired by the previous mechanisms, Lee et al. proposed a three-factor anonymous user authentication scheme for Internet of Things environments [13]. Lee et al. claimed that their scheme could resist stolen mobile device attack, user impersonation attack, replay attack, stolen-verifier attack, privileged-insider attack, sensor node impersonation attack and session-specific temporary information attack, and it could ensure user anonymity, user untraceability, mutual authentication, session key agreement, local user verification, user-friendly password change, and forward secrecy. They also claimed that their scheme could revoke users' devices to prevent the abuse or disclosure of confidential information when devices are lost or stolen. However, after analyzing their scheme thoroughly, we find that their scheme suffers from some flaws including failure sensor node authentication, failure mobile node authentication, replay attack, denial-of-service attack, and compromised user untraceability.

The rest of this paper is organized as follows. Section 2 reviews Lee et al.'s scheme, and the found flaws are given in Section 3. The proposed scheme is given in Section 4. Security analysis and further discussions are made in Section 5. At last, some conclusions are drawn in Section 6.

2. Review of Lee et al.'s Scheme

Lee et al.'s scheme uses XOR operation, hash function and symmetric cryptography to ensure efficiency. There exist three entities in their scheme, mobile node, IoT node and gateway. At first, mobile nodes and IoT nodes need to register with the gateway. Thereupon, registered users can access services provided by IoT nodes with the smart device via the gateway's help. Lee et al.'s scheme consists of four phases: registration phase, login and authentication phase, password change phase, and revocation phase. Notations used in Lee et al.'s scheme are listed in Table 1. The details are as follows.

Table 1. Notations used in Lee et al.'s scheme.

Notation	Definition
MN_i	Mobile node namely user
N_j	Sensor node
GW	Gateway
ID_i/NID_j	Identity of MN_i/N_j
PW_i	MN_i's password
BIO_i	MN_i's biometrics
T_1, T_2	Timestamps
T_{fresh}	Current timestamp
ΔT	Reasonable transmission delay
n_x, r_x	Random numbers
SK	Session key shared between MN_i and N_j
$E_k(.)/D_k(.)$	Symmetric encryption / decryption
$h(.)$	Hash function
$H(.)$	Bio-hash function
$\vert\vert$	Concatenation operator
$\oplus$	XOR operation
K_G	GW's private secret
K_{GU}	MN_i's private key
K_{GN}	Secret key shared between N_j and GW

2.1. Registration Phase

In Lee et al.'s scheme, registration phase is composed two phases: user registration phase and IoT node registration phase. In the following, these two parts are shown in detail.

2.1.1. User Registration Phase

When a mobile node MN_i wants to access the IoT service, MN_i needs to register with the gateway GW. In this phase, data is transmitted through a secure channel. This phase is depicted in Figure 1, and the details are as follows:

Step 1. First, MN_i chooses its identity ID_i, password PW_i, and biometrics BIO_i. Then MN_i computes $PWB_i = h(PW_i \mid\mid H(BIO_i))$ and $MID_i = h(ID_i \mid\mid H(BIO_i))$.

Step 2. MN_i sends $\{ID_i, PWB_i, MID_i\}$ to GW.

Step 3. Upon receiving $\{ID_i, PWB_i, MID_i\}$, GW selects random numbers r_{GU} and r_D. Then GW computes $RID_i = E_{K_G}(ID_i)$, $PID_i = E_{K_G}(ID_i \mid\mid r_D)$, $x_i = h(ID_i \mid\mid PWB_i)$ and $y_i = h(ID_i \mid\mid PWB_i \mid\mid r_{GU}) \oplus h(K_{GU} \mid\mid ID_i)$. GW stores (RID_i, MID_i) in its database.

Step 4. GW sends $\{PID_i, x_i, y_i, r_{GU}\}$ to MN_i.

Step 5. Upon receiving $\{PID_i, x_i, y_i, r_{GU}\}$, MN_i stores $\{PID_i, x_i, y_i, r_{GU}\}$ in the mobile device.

2.1.2. IoT Node Registration

Before being added to the IoT network, a sensor node N_j needs to register with GW. In this phase, data is transmitted through a public channel. This phase is depicted in Figure 2, and the details are as follows:

Step 1. First, N_j chooses a random number r_j. Then N_j computes $MP_j = h(K_{GN} \mid\mid r_j \mid\mid NID_j)$ and $MI_j = r_j \oplus h(NID_j \mid\mid K_{GN})$

Step 2. N_j sends $\{NID_j, MP_j, MI_j\}$ to GW.

Step 3. After receiving $\{NID_j, MP_j, MI_j\}$, GW uses the secret key K_{GN} to compute $r_j^* = MI_j \oplus h(NID_j \mid\mid K_{GN})$ and $MP_j^* = h(K_{GN} \mid\mid r_j^* \mid\mid NID_j)$. Then GW checks whether $MP_j^* = MP_j$ holds or not. If it holds, GW computes $x_j = h(NID_j \mid\mid K_{GN})$ and $y_j = x_j \oplus MP_j^*$.

Step 4. GW sends $\{y_j\}$ to N_j.

Step 5. Upon receiving $\{y_j\}$, N_j stores $\{y_j\}$ in its memory.

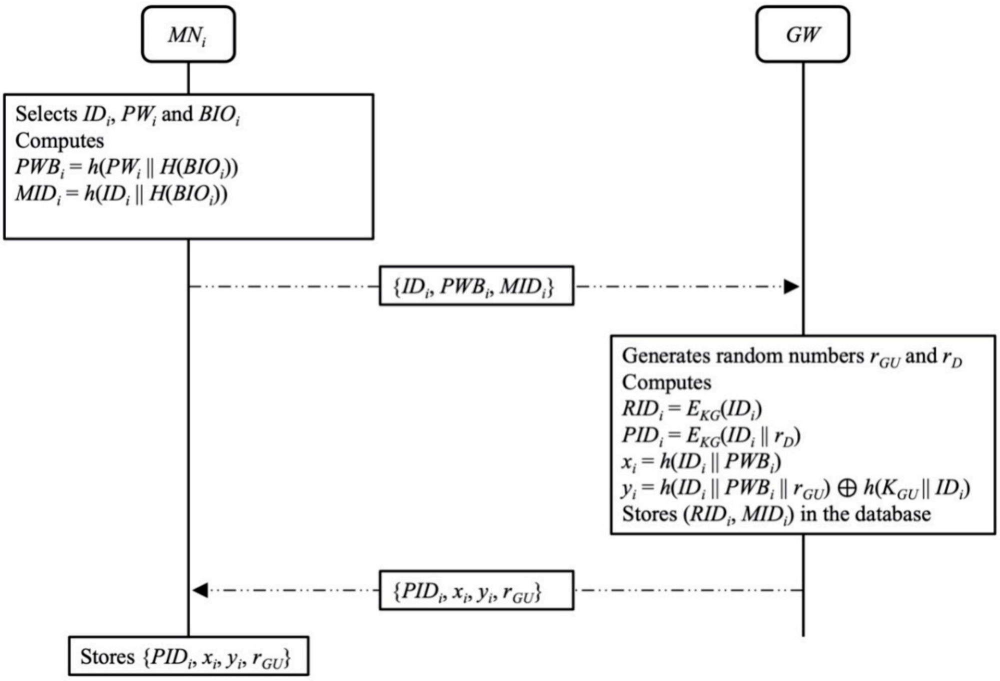

Figure 1. User registration phase of Lee et al.'s scheme.

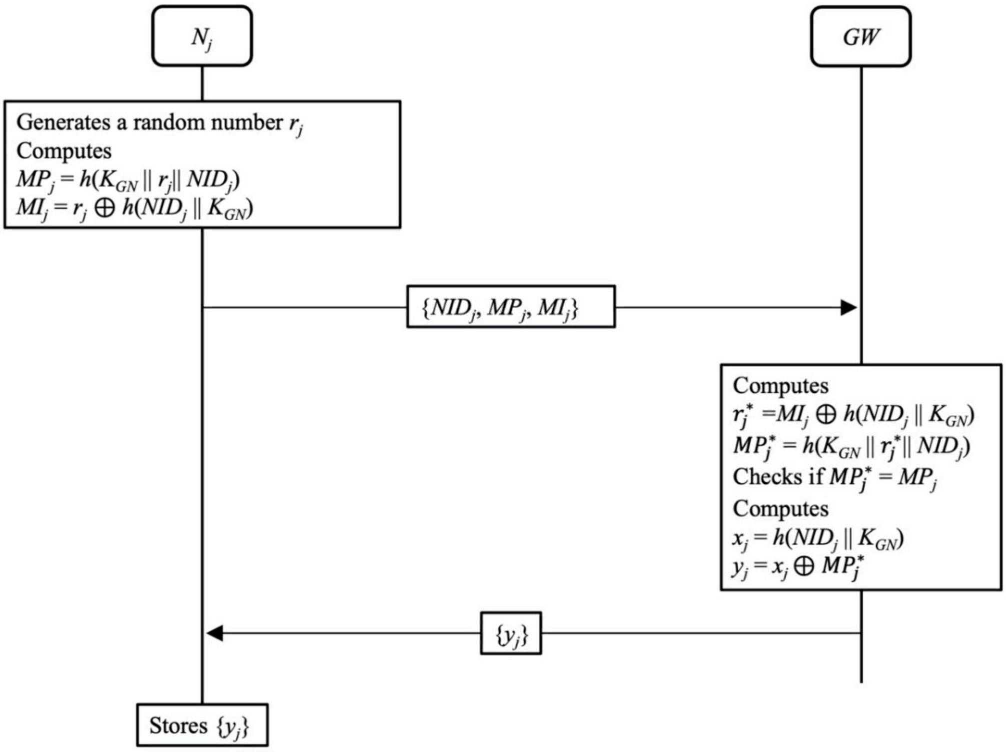

Figure 2. IoT node registration phase of Lee et al.'s scheme.

2.2. Login and Authentication Phase

After registration, registered MN_i can access IoT services provided by registered N_j. In this phase, MN_i and N_j authenticate each other, and a shared session key is negotiated for secure communication. In this phase, data is transmitted through a public channel. This phase is depicted in Figure 3, and the details are as follows:

Step 1. MN_i enters its identity ID_i, password PW_i^{old}, and biometric BIO_i. MN_i computes $PWB_i = h(PW_i \mid\mid H(BIO_i))$ and $x_i^* = h(ID_i \mid\mid PWB_i)$. Then MN_i checks whether $x_i^* = x_i$ holds or not. If it holds, MN_i generates a random number n_i and computes $A_i = y_i \oplus h(ID_i \mid\mid PWB_i \mid\mid r_{GU})$, $UN_i = h(A_i \mid\mid PID_i \mid\mid n_i)$ and $UZ_i = n_i \oplus A_i$; otherwise, MN_i terminates this phase immediately.

Step 2. MN_i gets the current timestamp T_1 and sends $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ to N_j.

Step 3. Upon receiving M_1, N_j checks whether $\mid T_{fresh} - T_1 \mid \leq \Delta T$ holds or not. If it holds, N_j generates a random number n_j and computes $x_j = y_j \oplus h(K_{GN} \mid\mid r_j \mid\mid NID_j)$, $A_j = h(x_j) \oplus n_j$, and $B_j = h(x_j \mid\mid n_j)$.

Step 4. N_j sends $M_2 = \{M_1, NID_j, A_j, B_j\}$ to GW.

Step 5. Upon receiving M_2 from N_j, GW computes $x_j^* = h(NID_j \mid\mid K_{GN})$, $n_j^* = h(x_j^*) \oplus A_j$, and $B_j^* = h(x_j^* \mid\mid n_j^*)$. GW checks whether $B_j^* = B_j$ holds or not. If it does not hold, GW terminates this phase immediately; otherwise, this phase proceeds. GW decrypts PID_i with its private secret K_G to obtain $\{ID_i, r_D\}$ stored in the database for MN_i. Then GW computes $A_i^* = h(ID_i \mid\mid K_{GU})$, $n_i^* = UZ_i \oplus A_i^*$ and $UN_i^* = h(A_i^* \mid\mid PID_i \mid\mid n_i^*)$ and checks whether $UN_i^* = UN_i$ holds or not. If it does not hold, GW terminates this phase immediately; otherwise, this phase proceeds. GW generates a random number r_D^{new} and computes $F_j = h(ID_i \mid\mid n_i^*)$, $G_j = F_j \oplus x_j^*$, $R_{ij} = n_j^* \oplus n_i^*$, $H_j = h(x_j^* \mid\mid n_j^* \mid\mid n_i^* \mid\mid F_j)$, and $PID_i^{new} = E_{K_G}(ID_i, r_D^{new})$.

Step 6. GW sends $M_3 = \{PID_i^{new}, G_j, R_{ij}, H_j\}$ to N_j.

Step 7. Upon receiving M_3 from GW, N_j computes $F_j^* = G_j \oplus x_j$, $n_i^* = R_{ij} \oplus n_j$, and $H_j^* = h(x_j \mid\mid n_j \mid\mid n_i^* \mid\mid F_j^*)$ and checks if whether $H_j^* = H_j$ holds or not. If it does not hold, N_j terminates this phase immediately; otherwise, N_j generates a random number m_j and computes $L_j = h(NID_j \mid\mid n_i^*) \oplus m_j$, $SK_{ji} = h(F_j^* \mid\mid n_i^* \mid\mid m_j)$, and $SV_j = h(SK_{ji} \mid\mid T_1 \mid\mid T_2)$.

Step 8. N_j sends $M_4 = \{PID_i^{new}, L_j, SV_j, T_2\}$ to MN_i.

Step 9. Upon receiving M_4 from N_j, MN_i checks whether $\mid T_{fresh} - T_2 \mid \leq \Delta T$ holds or not. If it holds, N_j computes $m_j^* = L_j \oplus h(NID_j \mid\mid n_i)$, $SK_{ji} = h(h(ID_i \mid\mid n_i) \mid\mid n_i \mid\mid m_j^*)$, and $SV_i = h(SK_{ji} \mid\mid T_1 \mid\mid T_2)$. If SV_i is equal to SV_j, MN_i and N_j have successfully negotiated a session key that can be used to ensure the security of subsequent communication.

2.3. Password Change Phase

MN_i's password can be changed on its smart device, and the details of this phase are shown as follows:

Step 1. MN_i enters its identity ID_i, original password PW_i^{old}, new password PW_i^{new}, and biometric BIO_i. Then MN_i computes $PWB_i^{old} = h(PW_i^{old} \mid\mid H(BIO_i))$ and $x_i^* = h(ID_i \mid\mid PWB_i^{old})$.

Step 2. MN_i checks whether x_i^* is equal to x_i or not. If they are equal, MN_i computes $A_i = y_i \oplus h(ID_i \mid\mid PWB_i^{old} \mid\mid r_{GU})$, $PWB_i^{new} = h(PW_i^{new} \mid\mid H(BIO_i))$, $x_i^{new} = h(ID_i \mid\mid PWB_i^{new})$ and $y_i^{new} = h(ID_i \mid\mid PWB_i^{new} \mid\mid r_{GU}) \oplus A_i \oplus y_i$.

Step 3. At last, MN_i updates x_i^{old} and y_i^{old} with x_i^{new} and y_i^{new}, respectively. Then MN_i's password is updated with PW_i^{new}.

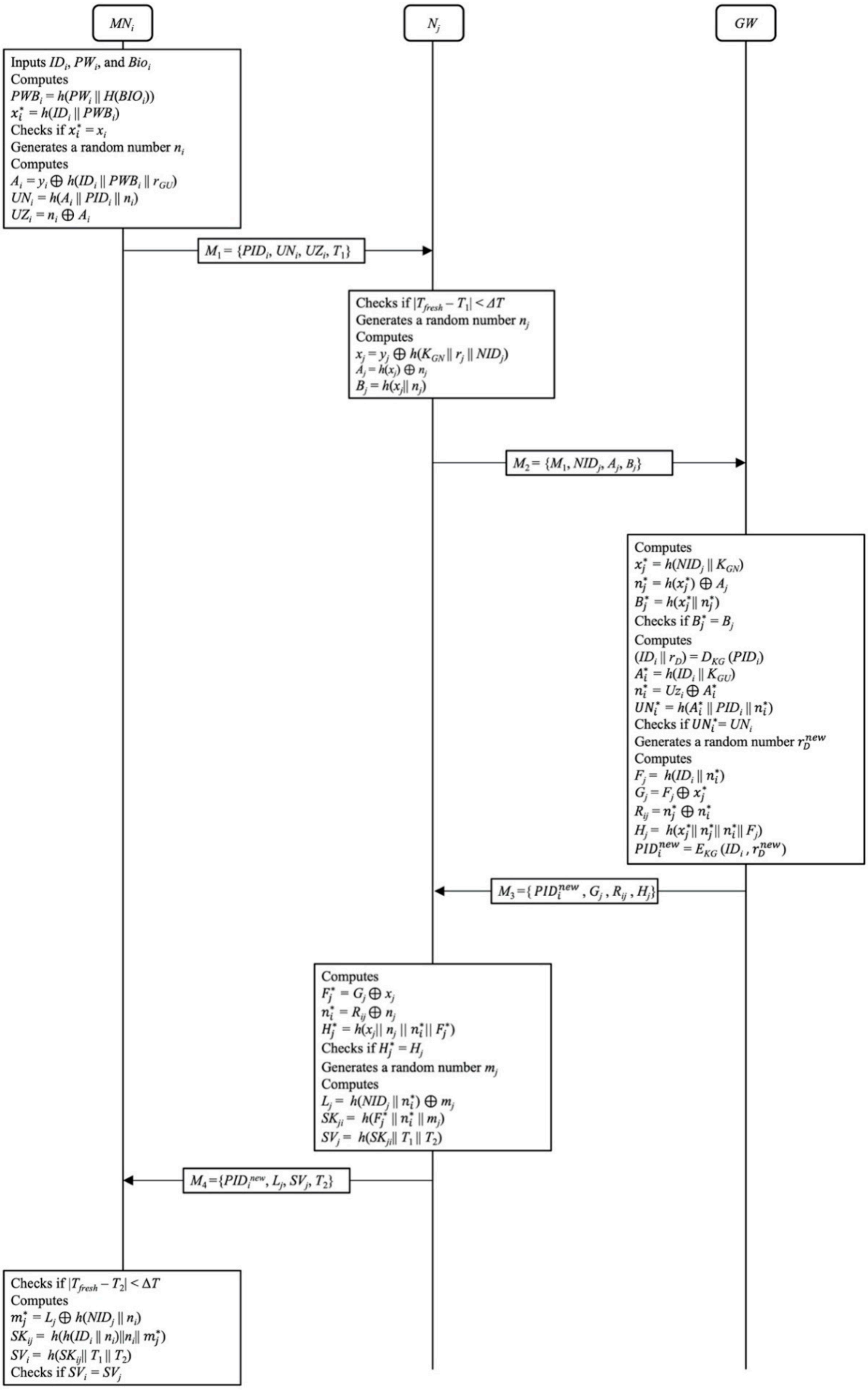

Figure 3. Login and authentication phase of Lee et al.'s scheme.

2.4. Revocation Phase

When MN_i wants to revoke or reissue a secret parameter, revocation phase will be performed. In this phase, data is transmitted through a secure channel, and the details are as follows.

Step 1. MN_i inputs its original identity ID_i^{old}, new identity ID_i^{new}, new password PW_i^{new}, and biometric BIO_i into the mobile device. Then MN_i computes $PWB_i^{new} = h(PW_i^{new} \,||\, H(BIO_i))$, $MID_i^{old} = h(ID_i^{old} \,||\, H(BIO_i))$, and $MID_i^{new} = h(ID_i^{new} \,||\, H(BIO_i))$.

Step 2. MN_i sends the revocation request $\{ID_i^{old}, ID_i^{new}, MID_i^{old}, MID_i^{new}, PWB_i^{new}\}$ to GW.

Step 3. Upon receiving the revocation request from MN_i, GW computes $RID_i^{old} = E_{K_G}(ID_i^{old})$ to verify MN_i's identiy and searches (RID_i, MID_i) in the database to find the specific registered user. If (RID_i, MID_i) is equal to (RID_i^{old}, MID_i^{old}), the identity of MN_i has been verified successfully. Then GW generates new random numbers r_D^{new} and r_{GU}^{new} and computes $PID_i^{new} = E_{K_G}(ID_i \,||\, r_D^{new})$, $RID_i^{new} = E_{K_G}(ID_i^{new})$, $x_i^{new} = h(ID_i \,||\, PWB_i^{new})$ and $y_i^{new} = h(ID_i \,||\, PWB_i^{new} \,||\, r_{GU}^{new}) \oplus h(K_{GU} \,||\, ID_i^{new})$. GW stores ($RID_i^{new}$, MID_i^{new}) in the database.

Step 4. GW sends $\{PID_i^{new}, x_i^{new}, y_i^{new}, r_{GU}^{new}\}$ to MN_i.

Step 5. Upon receiving $\{PID_i^{new}, x_i^{new}, y_i^{new}, r_{GU}^{new}\}$, MN_i stores $\{PID_i^{new}, x_i^{new}, y_i^{new}, r_{GU}^{new}\}$ in the mobile device.

3. Security Analysis

Lee et al. claimed that their scheme could resist stolen mobile device attack, user impersonation attack, replay attack, stolen-verifier attack, privileged-insider attack, sensor node impersonation attack and session-specific temporary information attack, and it could ensure user anonymity, user untraceability, mutual authentication, session key agreement, local user verification, user-friendly password change, and forward secrecy. They also claimed that their scheme could revoke users' devices to prevent the abuse or disclosure of confidential information when devices are lost or stolen. However, after analyzing their scheme thoroughly, we find that their scheme suffers from five flaws. First, when an IoT node N_j registers with the gateway GW, N_j stores y_j sent from GW without checking the integrity of y_j. This approach may make N_j authenticated by GW unsuccessfully. Secondly, in login and authentication phase, $A_i = y_i \oplus h(ID_i \,||\, PWB_i \,||\, r_{GU})$ computed by MN_i is different from $A_i^* = h(ID_i \,||\, K_{GU})$ computed by GW. This may result in mobile node authentication failure. Thirdly, in login and authentication phase, only N_j checks the freshness of T_1, and T_1 is not verified by GW at all. This makes an attacker mount replay attack. Fourth, similar to the third flaw, an attacker can impersonate a mobile node by sending a request to N_j to consume GW and N_j's computational resources. That is, denial of service attack may damage their scheme. Fifthly, user untraceability cannot be ensured as claimed. The details of how these flaws threaten Lee et al.'s scheme and our findings are shown as follows.

3.1. Failure Sensor Node Authentication

In IoT node registration phase, N_j chooses a random number r_j. Then N_j computes $MP_j = h(K_{GN} \,||\, r_j \,||\, NID_j)$ and $MI_j = r_j \oplus h(NID_j \,||\, K_{GN})$ and sends $\{NID_j, MP_j, MI_j\}$ to GW over a public channel. After receiving $\{NID_j, MP_j, MI_j\}$, GW uses the secret key K_{GN} shared between N_j and GW to compute $r_j^* = MI_j \oplus h(NID_j \,||\, K_{GN})$, $MP_j^* = h(K_{GN} \,||\, r_j^* \,||\, NID_j)$. Then GW checks whether $MP_j^* = MP_j$ holds or not. If it holds, the legitimacy of N_j and the integrity of $\{NID_j, MP_j, MI_j\}$ are both ensured, and GW computes $x_j = h(NID_j \,||\, K_{GN})$ and $y_j = x_j \oplus MP_j^*$ and sends $\{y_j\}$ to N_j. After receiving $\{y_j\}$, N_j stores $\{y_j\}$ in its memory. Because messages are transmitted over a public channel, anyone can eavesdrop or interrupt. If an attacker interrupts the transmission of y_j sent by GW and sends the forged y_j' to N_j, N_j will store y_j' immediately with no integrity check, where $y_j' \neq y_j$.

Thereupon, in login and authentication phase, N_j computes $x_j' = y_j' \oplus h(K_{GN} \,||\, r_j \,||\, NID_j) \neq x_j$, $A_j = h(x_j') \oplus n_j$, and $B_j = h(x_j' \,||\, n_j)$, where $y_j = x_j \oplus MP_j$ and $MP_j = h(K_{GN} \,||\, r_j^*$

$|| NID_j$). Then N_j sends $M_2 = \{M_1, NID_j, A_j, B_j\}$ to GW. After receiving M_2, GW computes $x_j^* = h(NID_j || K_{GN})$, $n_j^* = h(x_j^*) \oplus A_j$, and $B_j^* = h(x_j^* || n_j^*)$. GW checks whether $B_j^* = B_j$ holds or not. Unfortunately, it will never hold because $x_j^* \neq x_j'$, and GW will regard that N_j is illegal and terminate login and authentication phase immediately.

3.2. Failure Mobile Node Authentication

In user registration phase, GW computes $RID_i = E_{K_G}(ID_i)$, $PID_i = E_{K_G}(ID_i || r_D)$, $x_i = h(ID_i || PWB_i)$ and $y_i = h(ID_i || PWB_i || r_{GU}) \oplus h(K_{GU} || ID_i)$. GW stores (RID_i, MID_i) in its database. GW sends $\{PID_i, x_i, y_i, r_{GU}\}$ to MN_i through a secure channel. Upon receiving $\{PID_i, x_i, y_i, r_{GU}\}$, MN_i stores $\{PID_i, x_i, y_i, r_{GU}\}$ in the mobile device.

In login and authentication phase, MN_i computes $A_i = y_i \oplus h(ID_i || PWB_i || r_{GU}) = h(ID_i || PWB_i || r_{GU}) \oplus h(K_{GU} || ID_i) \oplus h(ID_i || PWB_i || r_{GU}) = h(K_{GU} || ID_i)$, $UN_i = h(A_i || PID_i || n_i) = h(h(K_{GU} || ID_i) || PID_i || n_i)$ and $UZ_i = n_i \oplus A_i = n_i \oplus h(K_{GU} || ID_i)$. GW computes $A_i^* = h(ID_i || K_{GU})$, $n_i^* = UZ_i \oplus A_i^* = n_i \oplus h(K_{GU} || ID_i) \oplus h(ID_i || K_{GU}) \neq n_i$, and $UN_i^* = h(A_i^* || PID_i || n_i^*) = h(h(ID_i || K_{GU}) || PID_i || n_i^*)$. Then GW checks whether $UN_i^* = UN_i$ holds or not. Unfortunately, it will never hold because $UN_i^* = h(A_i^* || PID_i || n_i^*) = h(h(ID_i || K_{GU}) || PID_i || n_i^*) \neq UN_i$, where $UN_i = h(h(K_{GU} || ID_i) || PID_i || n_i)$. GW will regard that MN_i is illegal and terminate login and authentication phase immediately.

3.3. Vulnerability to Replay Attack

In login and authentication phase, MN_i computes $A_i = y_i \oplus h(ID_i || PWB_i || r_{GU})$, $UN_i = h(A_i || PID_i || n_i)$ and $UZ_i = n_i \oplus A_i$ and sends $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ to N_j, where T_1 is the current timestamp. After receiving M_1, N_j checks the freshness of T_1, computes $x_j = y_j \oplus h(K_{GN} || r_j || NID_j)$, $A_j = h(x_j) \oplus n_j$, and $B_j = h(x_j || n_j)$, and sends $M_2 = \{M_1, NID_j, A_j, B_j\}$ to GW. Upon receiving M_2, GW computes $x_j^* = h(NID_j || K_{GN})$, $n_j^* = h(x_j^*) \oplus A_j$, and $B_j^* = h(x_j^* || n_j^*)$ and checks whether $B_j^* = B_j$ holds or not to verify n_j^* and authenticate N_j. Then GW decrypts PID_i with K_G to retrieve $\{ID_i, r_D\}$, computes $A_i^* = h(ID_i || K_{GU})$, $n_i^* = UZ_i \oplus A_i^*$ and $UN_i^* = h(A_i^* || PID_i || n_i^*)$ and checks whether $UN_i^* = UN_i$ holds or not to verify n_i^* and authenticate MN_i.

From the above, it is found that T_1 is included in none of all parameters computes by GW. And only N_j checks the freshness of T_1. Because messages are transmitted through a public channel, anyone can eavesdrop. That is, an attacker can get $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ easily. Thereupon, the attacker can send $M_1' = \{PID_i, UN_i, UZ_i, T_1'\}$, where T_1' is the current timestamp when the attacker mount replay attack. In the following, login and authentication phase will proceed as usual, GW will regard this request is indeed sent by MN_i, and no entity can detect replay attack.

According to above analysis, it is shown that Lee et al.'s scheme cannot replay attack as claimed.

3.4. Vulnerability to Denial-of-Service Attack

Denial-of-service attack is an attack that an attacker tries to prevent legitimate users from accessing services. In order to launch this attack, an attacker usually consumes as much transmission or computational resources as possible. In login and authentication phase, MN_i computes $A_i = y_i \oplus h(ID_i || PWB_i || r_{GU})$, $UN_i = h(A_i || PID_i || n_i)$ and $UZ_i = n_i \oplus A_i$ and sends $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ to N_j, where T_1 is the current timestamp. After receiving M_1, N_j checks the freshness of T_1, computes $x_j = y_j \oplus h(K_{GN} || r_j || NID_j)$, $A_j = h(x_j) \oplus n_j$, and $B_j = h(x_j || n_j)$, and sends $M_2 = \{M_1, NID_j, A_j, B_j\}$ to GW. Upon receiving M_2, GW computes $x_j^* = h(NID_j || K_{GN})$, $n_j^* = h(x_j^*) \oplus A_j$, and $B_j^* = h(x_j^* || n_j^*)$ and checks whether $B_j^* = B_j$ holds or not to verify n_j^* and authenticate N_j. Then GW decrypts PID_i with K_G to retrieve $\{ID_i, r_D\}$, computes $A_i^* = h(ID_i || K_{GU})$, $n_i^* = UZ_i \oplus A_i^*$ and $UN_i^* = h(A_i^* || PID_i || n_i^*)$ and checks whether $UN_i^* = UN_i$ holds or not to verify n_i^* and authenticate MN_i.

Suppose an attacker impersonates MN_i to send forged M_1 to N_j with fresh T_1. After receiving forged M_1, N_j checks the freshness of T_1. However, T_1 is fresh such that N_j will compute x_j, A_j, and B_j and send M_2 = {M_1, NID_j, A_j, B_j} to GW. Upon receiving M_2, GW computes x_j^*, n_j^*, and B_j^* and checks whether B_j^* = B_j holds or not to verify n_j^* and authenticate N_j. Because B_j is indeed computed by legal N_j, it must hold. Then GW decrypts PID_i with K_G to retrieve {ID_i, r_D}, computes A_i^*, n_i^* and UN_i^* and checks whether UN_i^* = UN_i holds or not to verify n_i^* and authenticate MN_i. Because M_1 is forged, it will not hold. However, this approach has already consumed N_j and GW's computational resources.

That is, if plenty of forged login requests are sent, GW's resources will be exhausted, and legitimate users will be unable to access services. As a result, Lee et al.'s scheme cannot resist denial-of-service attack.

3.5. Compromised User Untraceability

In login and authentication phase, messages are transmitted through a public channel. MN_i sends M_1 = {PID_i, UN_i, UZ_i, T_1} to N_j, and GW sends M_3 = {PID_i^{new}, G_j, R_{ij}, H_j} to N_j, where PID_i = $E_{K_G}(ID_i \,||\, r_D)$ and PID_i^{new} = $E_{K_G}(ID_i, r_D^{new})$. Because whether MN_i replaces PID_i with PID_i^{new} is not explicitly indicated, there are two possible cases for this issue:

Case 1: PID_i is not replaced with PID_i^{new}.

Case 2: PID_i is replaced with PID_i^{new}.

In Case 1, PID_i's transmitted in different sessions are the same. This makes tracing a MN_i with PID_i easy. In Case 2, PID_i's transmitted in different sessions differ from each other. Unfortunately, PID_i and PID_i^{new} are transmitted through a public channel such that it is easy to obtain the correlation between PID_i and PID_i^{new}. As a result, Lee et al.s.'s scheme cannot ensure user untraceability as claimed.

3.6. Our Findings

Lee et al.'s authentication scheme is designed to ensure the security of IoT communications. In their scheme, parameters are generated, computed, or transmitted to achieve the goal with designated processes. However, improper designs result in the found flaws. In this paper, we analyze Lee et al.'s scheme by investigating the processes in detail in Section 3. According to the analyses, we obtain the following. First, the integrity of the transmitted data needs to be ensured. Secondly, because the same input parameters of hash function with different orders obtain different hash values, only rigorous designs can lead to successful verification and authentication. Thirdly, the freshness of a received message needs to be verified explicitly such that the timestamp should be one of the input parameters of hash function to resist replay attack. Fourthly, the gateway is responsible for helping a mobile node and a sensor node to authenticate each other. The gateway should authenticate the mobile node and the sensor node as early as possible to resist denial-of-service attack that may consume the gateway's computational resources. Lastly, user anonymity can be ensured only when the identities PID_i's transmitted in different sessions differ from each other and the correlation cannot be found.

4. The Proposed Authentication Scheme

After analyzing Lee et al.'s three-factor anonymous user authentication scheme for IoT environments, we find that their scheme cannot ensure security as claimed. To overcome the flaws and preserve the advantages, an improvement is proposed. The notations used in the proposed scheme are listed in Table 2.

Table 2. Notations used in the proposed scheme.

Notation	Definition
$U_i/GWN/S_j$	The ith user, the gateway node, the jth sensor node
ID_i/SID_j	U_i's/S_j's identity
PW_i	U_i's password
ID_i/NID_j	Identity of MN_i/N_j
b_i	U_i's biometric
SC	U_i's smart card issued by GWN
K_{GWN}	GWN's master key
K_{GWN-S_j}	The secret key shared between GWN and S_j
$SK_i/SK_j/SK_{GWN}$	The session key computed by $U_i/S_j/GWN$
$h(.)$	A secure one-way hash function
$C \subseteq \{0, 1\}^n$	A set of codewords
$F(.)$	A fuzzy commitment scheme
$f(.)$	A decoding function
$r_i/r_g/r_j$	A random number generated by $U_i/GWN/S_j$
$\vert\vert$	A concatenation operator
$\oplus$	An XOR operator

Different from Lee et al.'s scheme, ECC is employed in our scheme to ensure efficiency. To initialize the scheme, an addition group G over a finite field F_p on the elliptic curve E of prime order n and the generator P of G are selected by GWN. GWN selects its private key $x \in Z_n{}^*$ randomly, computes its public key $X = xP$, and chooses its master key K_{GWN}. GWN publishes $\{E(F_p), G, P, X\}$ while keeping x and K_{GWN} secretly. The proposed scheme is composed of four phases: sensor registration phase, user registration phase, login and authentication phase, and password change phase. The details are as follows.

4.1. Sensor Registration Phase

Before deployment, for each sensor S_j, GWN selects an identity SID_j, computes the secret key $K_{GWN-S_j} = h(SID_j \parallel K_{GWN})$, and stores $\{SID_j, K_{GWN-S_j}\}$ in its memory. After initialization, these initialized sensors are deployed in a particular area to form a wireless sensor network.

4.2. User Registration Phase

When a new user U_i wants to access the services provided by the wireless sensor network such as acquiring sensory data from sensor nodes, U_i has to register with GWN. In this phase, data is transmitted via secure channels. User registration phase is depicted in Figure 4, and the details are as follows:

Step 1. U_i chooses his/her identity ID_i and password PW_i.
Step 2. U_i generates a nonce a_i and computes $RPW_i = h(PW_i \parallel a_i)$.
Step 3. U_i imprints his/her biometric on a special device to get the biometric b_i.
Step 4. U_i sends the registration request $\{ID_i, RPW_i, b_i\}$ to GWN via a secure channel.
Step 5. After receiving U_i's registration request $\{ID_i, RPW_i, b_i\}$, GWN randomly chooses a codeword $c_i \in C$ for U_i.
Step 6. GWN computes $F(c_i, b_i) = (\alpha, \delta)$, $A_i = h(ID_i \parallel RPW_i \parallel c_i)$ and $B_i = h(ID_i \parallel K_{GWN}) \oplus h(RPW_i \parallel c_i)$, where $\alpha = h(c_i)$ and $\delta = c_i \oplus b_i$.
Step 7. GWN stores $\{\delta, A_i, B_i, X, f(.)\}$ into a smart card SC and issues it to U_i via a secure channel.
Step 8. GWN stores ID_i in its database and deletes other information.
Step 9. After obtaining the smart card issued by GWN, U_i stores a_i into SC. Then, SC contains $\{\delta, A_i, B_i, X, f(.), a_i\}$.

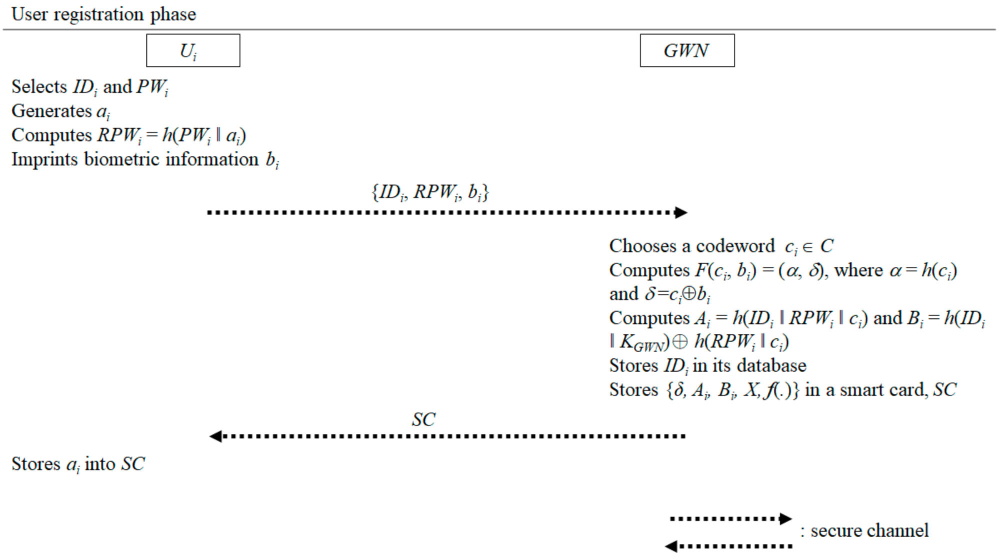

Figure 4. User registration phase of the proposed scheme.

4.3. Login and Authentication Phase

When U_i wants to acquire S_j's sensory data, this phase will be executed. Because only GWN shares secrets with U_i and S_j, only GWN can authenticate U_i and S_j. In this phase, U_i and S_j are authenticated by GWN, and a session key among GWN, U_i and S_j will be generated via GWN's help. This phase is depicted in Figure 5, and the details are as follows:

Step 1. U_i inserts his/her smart card SC into a card reader and imprints his/her biometric $b_i{}'$ on a special device, where SC contains $\{\delta, A_i, B_i, X, f(.), a_i\}$.

Step 2. U_i inputs ID_i and PW_i.

Step 3. SC computes $c_i{}' = f(\delta \oplus b_i{}') = f((c_i \oplus b_i) \oplus b_i{}')$ and $A_i{}' = h(ID_i \| h(PW_i \| a_i) \| c_i{}')$.

Step 4. SC checks whether $A_i{}' = A_i$ holds or not. If it does not hold, this session is terminated by SC; otherwise, U_i's identity ID_i, password PW_i and biometric $b_i{}'$ are verified successfully by SC, and this phase proceeds.

Step 5. SC randomly chooses numbers r_i and $s \in Z_n{}^*$.

Step 6. SC computes $M_1 = B_i \oplus h(h(PW_i \| a_i) \| c_i{}') = h(ID_i \| K_{GWN})$, $M_2 = sP$, $M_3 = sX = sxP$, $M_4 = ID_i \oplus M_3$, $M_5 = h(M_1 \| M_3) \oplus r_i$, $M_6 = h(ID_i \| r_i) \oplus SID_j$ and $M_7 = h(M_1 \| SID_j \| M_3 \| r_i)$.

Step 7. U_i sends the login request $\{M_2, M_4, M_5, M_6, M_7\}$ to GWN.

Step 8. After receiving the login request $\{M_2, M_4, M_5, M_6, M_7\}$ from U_i, GWN computes $M_3{}' = xM_2 = xsP$ and $ID_i{}' = M_4 \oplus M_3{}'$.

Step 9. GWN checks whether $ID_i{}'$ exists in the database or not. If $ID_i{}'$ does not exist, this login request is rejected by GWN; otherwise, this phase proceeds.

Step 10. GWN computes $M_1{}' = h(ID_i{}' \| K_{GWN})$, $r_i{}' = M_5 \oplus h(M_1{}' \| M_3{}')$, $SID_j{}' = M_6 \oplus h(ID_i{}' \| r_i{}')$ and $M_7{}' = h(M_1{}' \| SID_j{}' \| M_3{}' \| r_i{}')$.

Step 11. GWN checks whether $M_7{}' = M_7$ holds or not. If it does not hold, this session is terminated by GWN; otherwise, this phase proceeds.

Step 12. GWN generates a random number r_g.

Step 13. GWN computes $K_{GWN-S_j}{}' = h(SID_j{}' \| K_{GWN})$, $M_8 = ID_i{}' \oplus h(r_g \| K_{GWN-S_j}{}')$, $M_9 = h(M_8 \| SID_j{}' \| K_{GWN-S_j}{}') \oplus r_g$, $M_{10} = h(ID_i{}' \| r_g) \oplus r_i{}'$ and $M_{11} = h(ID_i{}' \| SID_j{}' \| K_{GWN-S_j}{}' \| r_i{}' \| r_g)$.

Step 14. GWN sends $\{M_8, M_9, M_{10}, M_{11}\}$ to S_j.

Step 15. After receiving $\{M_8, M_9, M_{10}, M_{11}\}$, S_j computes $ID_i{}'' = M_8 \oplus h(ID_i{}'' \| K_{GWN-S_j})$, $r_g{}' = M_9 \oplus h(M_8 \| SID_j \| K_{GWN-S_j})$, $r_i{}'' = M_{10} \oplus h(ID_i{}' \| r_g)$, and $M_{11}{}' = h(ID_i{}'' \| SID_j \| K_{GWN-S_j} \| \mathrm{r_i}'' \| \mathrm{r_g}')$.

Step 16. S_j checks whether $M_{11}' = M_{11}$ holds or not. If it does not hold, this session is terminated by S_j; otherwise, this phase proceeds.

Step 17. S_j generates a random number r_j.

Step 18. S_j computes $M_{12} = h(r_g' \parallel r_i'' \parallel K_{GWN-S_j}) \oplus r_j$, $SK_j = h(ID_i'' \parallel SID_j \parallel r_i'' \parallel r_g' \parallel r_j)$ and $M_{13} = h(K_{GWN-S_j} \parallel SK_j \parallel r_j)$.

Step 19. S_j sends the response $\{M_{12}, M_{13}\}$ to GWN.

Step 20. After getting the response $\{M_{12}, M_{13}\}$, GWN computes $r_j' = M_{12} \oplus h(r_g \parallel r_i' \parallel K_{GWN-S_j}')$, $SK_{GWN} = h(ID_i' \parallel SID_j' \parallel r_i' \parallel r_g \parallel r_j')$ and $M_{13}' = h(K_{GWN-S_j}' \parallel SK_{GWN} \parallel r_j')$.

Step 21. GWN checks whether $M_{13}' = M_{13}$ holds or not. If it does not hold, this session is terminated; otherwise, this phase proceeds.

Step 22. GWN computes $M_{14} = h(r_i' \parallel M_1') \oplus r_g$, $M_{15} = h(ID_i' \parallel r_i') \oplus r_j'$ and $M_{16} = h(ID_i' \parallel SK_{GWN} \parallel r_g \parallel r_j')$.

Step 23. GWN sends $\{M_{14}, M_{15}, M_{16}\}$ to U_i.

Step 24. U_i computes $r_g'' = M_{14} \oplus h(r_i' \parallel M_1')$, $r_j'' = M_{15} \oplus h(ID_i' \parallel r_i')$, $SK_i = h(ID_i \parallel SID_j \parallel r_i \parallel r_g'' \parallel r_j'')$ and $M_{16}' = h(ID_i \parallel SK_i \parallel r_g'' \parallel r_j'')$.

Step 25. U_i checks whether $M_{16}' = M_{16}$ holds or not. If it does not hold, this session is terminated; otherwise, the process is completed.

After the above process, U_i can acquire S_j's sensory data via GWN while a session key is shared among U_i, S_j and GWN, where $SK_i = SK_j = SK_{GWN}$.

4.4. Password Change Phase

In the proposed scheme, a user can update his password without GWN involved. This phase is depicted is Figure 6, and the details are as follows:

Step 1. U_i inserts his/her smart card SC into a card reader and imprints his/her biometric b_i' on a special device.

Step 2. U_i inputs ID_i and PW_i.

Step 3. SC computes $c_i' = f(\delta \oplus b_i') = f((c_i \oplus b_i) \oplus b_i')$ and $A_i' = h(ID_i \parallel h(PW_i \parallel a_i) \parallel c_i')$.

Step 4. SC checks whether $A_i' = A_i$ or not. If it does not hold, this request is declined by SC; otherwise, U_i inputs a new password PW_i^*.

Step 5. SC computes $A_i^* = h(ID_i \parallel h(PW_i^* \parallel a_i) \parallel c_i')$ and $B_i^* = B_i \oplus h(h(PW_i \parallel a_i) \parallel c_i') \oplus h(h(PW_i^* \parallel a_i) \parallel c_i')$.

Step 6. SC updates A_i and B_i with A_i^* and B_i^*.

Login and authentication phase

U_i/SC	GWN	S_j

Imprints b_i'
Inputs ID_i and PW_i
Computes $c_i' = f(\delta \oplus b_i') = f(c_i \oplus (b_i \oplus b_i'))$
Computes $A_i' = h(ID_i \parallel h(PW_i \parallel a_i) \parallel c_i')$
Checks if $A_i' = A_i$
Chooses r_i and $s \in Z_n^*$
Computes $M_1 = B_i \oplus h(h(PW_i \parallel a_i) \parallel c_i')$, $M_2 = sP$,
$M_3 = sX = sxP$, $M_4 = ID_i \oplus M_3$, $M_5 = h(M_1 \parallel M_3) \oplus r_i$,
$M_6 = h(ID_i \parallel r_i) \oplus SID_j$ and $M_7 = h(M_1 \parallel SID_j \parallel M_3 \parallel r_i)$

$\{M_2, M_4, M_5, M_6, M_7\}$ →

Computes $M_3' = xM_2 = xsP$ and $ID_i' = M_4 \oplus M_3'$
Checks if ID_i' exists in its database
Computes $M_1' = h(ID_i' \parallel K_{GWN})$,
$r_i' = M_5 \oplus h(M_1' \parallel M_3')$,
$SID_j' = M_6 \oplus h(ID_i' \parallel r_i')$ and
$M_7' = h(M_1' \parallel SID_j' \parallel M_3' \parallel r_i')$
Checks if $M_7' = M_7$
Generates r_g
Computes $K_{GWN\text{-}S_j}' = h(SID_j' \parallel K_{GWN})$,
$M_8 = h(r_g \parallel K_{GWN\text{-}S_j}') \oplus ID_i'$, $M_9 = h(M_8 \parallel SID_j' \parallel K_{GWN\text{-}S_j}') \oplus r_g$,
$M_{10} = h(ID_i \parallel r_g) \oplus r_i'$ and $M_{11} = h(ID_i' \parallel SID_j' \parallel K_{GWN\text{-}S_j}' \parallel r_i' \parallel r_g)$

$\{M_8, M_9, M_{10}, M_{11}\}$ →

Computes $ID_i'' = M_8 \oplus h(r_g' \parallel K_{GWN\text{-}S_j})$,
$r_g' = M_9 \oplus h(M_8 \parallel SID_j \parallel K_{GWN\text{-}S_j})$,
$r_i'' = M_{10} \oplus h(ID_i'' \parallel r_g')$ and
$M_{11}' = h(ID_i'' \parallel SID_j \parallel K_{GWN\text{-}S_j} \parallel r_i'' \parallel r_g')$
Checks if $M_{11}' = M_{11}$
Generates r_j
Computes $M_{12} = h(r_g' \parallel r_i'' \parallel K_{GWN\text{-}S_j}) \oplus r_j$,
$SK_j = h(ID_i'' \parallel SID_j \parallel r_i'' \parallel r_g' \parallel r_j)$ and
$M_{13} = h(K_{GWN\text{-}S_j} \parallel SK_j \parallel r_j)$

← $\{M_{12}, M_{13}\}$

Computes $r_j' = M_{12} \oplus h(r_g \parallel r_i' \parallel K_{GWN\text{-}S_j}')$,
$SK_{GWN} = h(ID_i' \parallel SID_j' \parallel r_i' \parallel r_g \parallel r_j')$ and
$M_{13}' = h(K_{GWN\text{-}S_j}' \parallel SK_{GWN} \parallel r_j')$
Checks if $M_{13}' = M_{13}$
Computes $M_{14} = h(r_i' \parallel M_1') \oplus r_g$,
$M_{15} = h(r_i' \parallel ID_i) \oplus r_j'$ and
$M_{16} = h(ID_i' \parallel SK_{GWN} \parallel r_g \parallel r_j')$

← $\{M_{14}, M_{15}, M_{16}\}$

Computes $r_g'' = M_{14} \oplus h(r_i \parallel M_1)$, $r_j'' = M_{15} \oplus h(r_i' \parallel ID_i)$,
$SK_i = h(ID_i \parallel SID_j \parallel r_i \parallel r_g'' \parallel r_j'')$ and
$M_{16}' = h(ID_i \parallel SK_i \parallel r_g'' \parallel r_j'')$
Checks if $M_{16}' = M_{16}$.

Figure 5. Login and authentication phase of the proposed scheme.

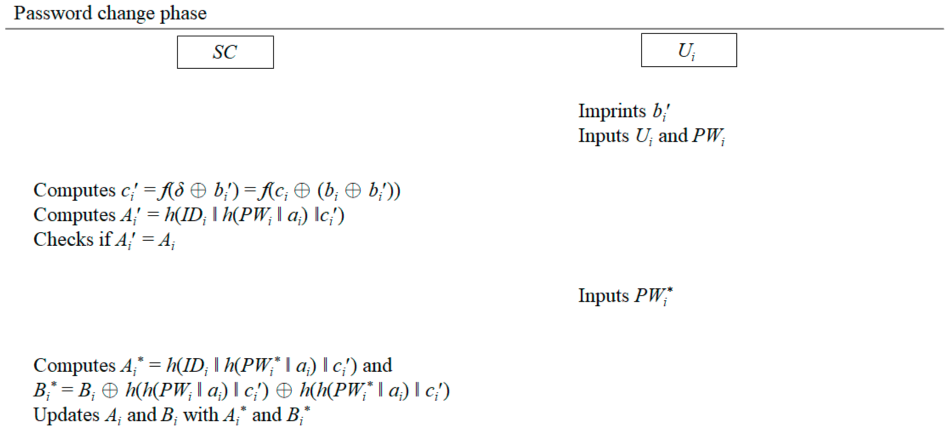

Figure 6. Password change phase of the proposed protocol.

5. Security Analysis and Further Discussions

In this section, security analysis is first made to show that the proposed scheme can not only overcome the drawbacks that Lee et al.'s scheme suffers from but also resist common attack. The further discussions are made to demonstrate the properties that the proposed scheme possesses. The details are as follows.

5.1. Resistance against Leakage of the Secret Key Shared between GWN and S_j

In login and authentication phase, *GWN* sends $\{M_8, M_9, M_{10}, M_{11}\}$ to where $K_{GWN-S_j}{}'$ = $h(SID_j' \| K_{GWN})$, $M_8 = ID_i' \oplus h(r_g \| K_{GWN-S_j}{}')$, $M_9 = h(M_8 \| SID_j' \| K_{GWN-S_j}{}') \oplus r_g$, $M_{10} = h(ID_i' \| r_g) \oplus r_i'$ and $M_{11} = h(ID_i' \| SID_j' \| K_{GWN-S_j}{}' \| r_i' \| r_g)$. After getting $\{M_8, M_9, M_{10}, M_{11}\}$, S_j computes $ID_i'' = M_8 \oplus h(ID_i'' \| K_{GWN-S_j})$, $r_g' = M_9 \oplus h(M_8 \| SID_j \| K_{GWN-S_j})$, $r_i'' = M_{10} \oplus h(ID_i' \| r_g)$, and $M_{11}' = h(ID_i'' \| SID_j \| K_{GWN-S_j} \| r_i'' \| r_g')$. Then S_j checks if $M_{11}' = M_{11}$ to determine whether the communication party is *GWN* and to ensure the correctness of the obtained r_i'', r_g', and ID_i''. Thereupon, S_j generates a random number r_j, computes $M_{12} = h(r_g' \| r_i'' \| K_{GWN-S_j}) \oplus r_j$, $SK_j = h(ID_i'' \| SID_j \| r_i'' \| r_g' \| r_j)$ and $M_{13} = h(K_{GWN-S_j} \| SK_j \| r_j)$, and sends the response $\{M_{12}, M_{13}\}$ to *GWN*.

The secret key $K_{GWN-S_j} = h(SID_j \| K_{GWN})$ shared between *GWN* and S_j is contained in $M_8 = ID_i' \oplus h(r_g \| K_{GWN-S_j}{}')$, $M_9 = h(M_8 \| SID_j' \| K_{GWN-S_j}{}') \oplus r_g$, $M_{11} = h(ID_i' \| SID_j' \| K_{GWN-S_j}{}' \| r_i' \| r_g)$, $M_{12} = h(r_g' \| r_i'' \| K_{GWN-S_j}) \oplus r_j$, and $M_{13} = h(K_{GWN-S_j} \| SK_j \| r_j)$. A legal user U_i knows ID_i, SID_j, r_i, r_g, r_j, and SK_i. From M_8, M_9, M_{11}, M_{12}, and M_{13}, U_i can retrieve $h(r_g \| K_{GWN-S_j}{}')$, $h(M_8 \| SID_j' \| K_{GWN-S_j}{}')$, and $h(r_g' \| r_i'' \| K_{GWN-S_j})$. However, K_{GWN-S_j} is concealed by the secure one-way hash function. According to the properties of one-way hash function, it is infeasible to retrieve the input from a hash value. It denotes that K_{GWN-S_j} cannot be retrieved from $h(r_g \| K_{GWN-S_j}{}')$, $h(M_8 \| SID_j' \| K_{GWN-S_j}{}')$, and $h(r_g' \| r_i'' \| K_{GWN-S_j})$. According to the above, it is ensured that no one even a legal user can retrieve the secret key $K_{GWN-S_j} = h(SID_j \| K_{GWN})$ shared between *GWN* and S_j.

5.2. Resistance against Sensor Node Impersonation Attack

We have explained why the proposed scheme can protect the secret key K_{GWN-S_j} = $h(SID_j \| K_{GWN})$ shared between *GWN* and S_j from being revealed. If an adversary wants to impersonate S_j, he needs to send $M_{12} = h(r_g' \| r_i'' \| K_{GWN-S_j}) \oplus r_j$, and $M_{13} = h(K_{GWN-S_j} \| SK_j \| r_j)$ to *GWN*. In each session, random numbers r_i, r_g, and r_j will be generated. It denotes that r_i, r_g, r_j and SK_j differ from those in other sessions, where $SK_j = h(ID_i'' \| SID_j \| r_i'' \| r_g' \| r_j)$. Because K_{GWN-S_j} is unknown, the adversary cannot retrieve correct ID_i' and r_g' from M_8 and M_9, where $M_8 = ID_i' \oplus h(r_g \| K_{GWN-S_j}{}')$ and $M_9 = h(M_8 \| SID_j' \| K_{GWN-S_j}{}') \oplus r_g$. As a result, r_i' cannot be retrieved as well. Because the adversary is not aware of ID_i', r_g', r_i'' and K_{GWN-S_j}, he cannot compute correct M_{12} and M_{13} to cheat GWN. If the adversary retransmits M_{12} and M_{13} in a previous session, GWN will detect that M_{12} and M_{13} are not correct. It is because r_i, r_g, r_j and SK_j in one session differ from those in other sessions. According to the above, it is ensured that no one can impersonate S_j.

5.3. Resistance against Gateway Node Impersonation Attack

After sensor registration phase, GWN and S_j, GWN share the secret key $K_{GWN-S_j} = h(SID_j \parallel K_{GWN})$. After user registration phase, U_i gets a smart card SC containing $\{\delta, A_i, B_i, X, f(.), a_i\}$, where $F(c_i, b_i) = (\alpha, \delta)$, $\alpha = h(c_i)$, $\delta = c_i \oplus b_i$, $A_i = h(ID_i \parallel RPW_i \parallel c_i)$ and $B_i = h(ID_i \parallel K_{GWN}) \oplus h(RPW_i \parallel c_i)$. In login and authentication phase, GWN sends $\{M_8, M_9, M_{10}, M_{11}\}$ to S_j. After receiving $\{M_8, M_9, M_{10}, M_{11}\}$, S_j computes $ID_i'' = M_8 \oplus h(ID_i'' \parallel K_{GWN-S_j})$, $r_g' = M_9 \oplus h(M_8 \parallel SID_j \parallel K_{GWN-S_j})$, $r_i'' = M_{10} \oplus h(ID_i' \parallel r_g)$, and $M_{11}' = h(ID_i'' \parallel SID_j \parallel K_{GWN-S_j} \parallel r_i'' \parallel r_g')$ and checks if $M_{11}' = M_{11}$. If it holds, it denotes that the computed ID_i'', r_g' and r_i'' and the shared secret key K_{GWN-S_j} are correct. If an adversary retransmits $\{M_8, M_9, M_{10}, M_{11}\}$ of a previous session, $M_{11}' = M_{11}$ must hold. However, because K_{GWN-S_j} and $M_1 = h(ID_i \parallel K_{GWN})$ are unknown, the adversary cannot obtain r_g, r_i' and r_j'. That is, the adversary cannot obtain $SK_{GWN} = h(ID_i' \parallel SID_j' \parallel r_i' \parallel r_g \parallel r_j')$ such that no sensory data collected by S_j will be revealed.

On the other hand, in login and authentication phase, U_i sends the login request $\{M_2, M_4, M_5, M_6, M_7\}$ to GWN. After receiving the login request $\{M_2, M_4, M_5, M_6, M_7\}$ from U_i, GWN computes $M_3' = xM_2 = xsP$ and $ID_i' = M_4 \oplus M_3'$ and checks if ID_i' exists in the database. Then, GWN computes $M_1' = h(ID_i' \parallel K_{GWN})$, $r_i' = M_5 \oplus h(M_1' \parallel M_3')$, $SID_j' = M_6 \oplus h(ID_i' \parallel r_i')$ and $M_7' = h(M_1' \parallel SID_j' \parallel M_3' \parallel r_i')$ and checks if $M_7' = M_7$. Then the phase proceeds. After getting $\{M_{14}, M_{15}, M_{16}\}$ from GWN, U_i computes $r_g'' = M_{14} \oplus h(r_i' \parallel M_1')$, $r_j'' = M_{15} \oplus h(ID_i' \parallel r_i')$, $SK_i = h(ID_i \parallel SID_j \parallel r_i \parallel r_g'' \parallel r_j'')$ and $M_{16}' = h(ID_i \parallel SK_i \parallel r_g'' \parallel r_j'')$ and checks if $M_{16}' = M_{16}$. If it holds, it denotes that GWN is legal and the session key SK_i is negotiated successfully. Because only GWN knows K_{GWN}, only GWN can compute $M_1' = h(ID_i' \parallel K_{GWN})$ to retrieve r_i' and ID_i'. That is, only GWN can compute M_{14}, M_{15}, and M_{16} to have itself authenticated by U_i.

According to the above, the proposed scheme can resist gateway node impersonation attack.

5.4. Resistance against User Impersonation Attack

After user registration phase, U_i gets a smart card SC containing $\{\delta, A_i, B_i, X, f(.), a_i\}$, where $F(c_i, b_i) = (\alpha, \delta)$, $\alpha = h(c_i)$, $\delta = c_i \oplus b_i$, $A_i = h(ID_i \parallel RPW_i \parallel c_i)$ and $B_i = h(ID_i \parallel K_{GWN}) \oplus h(RPW_i \parallel c_i)$. In login and authentication phase, U_i sends the login request $\{M_2, M_4, M_5, M_6, M_7\}$ to GWN. After receiving the login request $\{M_2, M_4, M_5, M_6, M_7\}$ from U_i, GWN computes $M_3' = xM_2 = xsP$ and $ID_i' = M_4 \oplus M_3'$ and checks if ID_i' exists in the database. Then, GWN computes $M_1' = h(ID_i' \parallel K_{GWN})$, $r_i' = M_5 \oplus h(M_1' \parallel M_3')$, $SID_j' = M_6 \oplus h(ID_i' \parallel r_i')$ and $M_7' = h(M_1' \parallel SID_j' \parallel M_3' \parallel r_i')$ and checks if $M_7' = M_7$. If it holds, it denotes that the computed M_1', r_i' and SID_j' are correct. If an adversary retransmits the login request $\{M_2, M_4, M_5, M_6, M_7\}$ of a previous session to GWN, $M_7' = M_7$ must hold. The phase will proceed. Then GWN sends $\{M_{14}, M_{15}, M_{16}\}$ to U_i.

Although $\{M_2, M_4, M_5, M_6, M_7\}$ and $\{M_{14}, M_{15}, M_{16}\}$ are transmitted via public channels, the adversary cannot retrieve $M_1 = h(ID_i \parallel K_{GWN})$ because of the properties of one-way hash function, where $M_1 = h(ID_i \parallel K_{GWN})$, $M_2 = sP$, $M_3 = sX = sxP$, $M_4 = ID_i \oplus M_3$, $M_5 = h(M_1 \parallel M_3) \oplus r_i$, $M_6 = h(ID_i \parallel r_i) \oplus SID_j$, $M_7 = h(M_1 \parallel SID_j \parallel M_3 \parallel r_i)$, $M_{14} = h(r_i' \parallel M_1') \oplus r_g$, $M_{15} = h(ID_i' \parallel r_i') \oplus r_j'$ and $M_{16} = h(ID_i' \parallel SK_{GWN} \parallel r_g \parallel r_j')$. After getting $\{M_{14}, M_{15}, M_{16}\}$ from GWN, the adversary cannot obtain r_g'', r_j'', and SK_i. As a result, the adversary cannot obtain the sensory data from S_j. From the above, the proposed scheme can defend against user impersonation attack.

5.5. Resistance against Replay Attack

Sensory data collected by S_j will be concealed by the session key. U_i, S_j and GWN obtain SK_i, SK_j and SK_{GWN}, respectively. $SK_i = SK_j = SK_{GWN} = h(ID_i \| SID_j \| r_i \| r_g \| r_j)$. r_i, r_g, and r_j are random numbers generated by U_i, GWN and S_j, respectively. In each session, r_i, r_g, and r_j are fresh. If an adversary wants to mount replay attack, he cannot obtain the fresh session key to obtain S_j's sensory data.

5.6. Resistance against Stolen Smart Card Attack

$\{\delta, A_i, B_i, X, f(.), a_i\}$ are stored in SC, where $RPW_i = h(PW_i \| a_i)$, $F(c_i, b_i) = (\alpha, \delta)$, $A_i = h(ID_i \| RPW_i \| c_i)$, $B_i = h(ID_i \| K_{GWN}) \oplus h(RPW_i \| c_i)$, $\alpha = h(c_i)$ and $\delta = c_i \oplus b_i$. If an attacker gets U_i's smart card SC and has the ability to reveal the stored parameters, the attacker gets δ, A_i, B_i, $X, f(.)$ and a_i. However, only the one who has b_i can obtain c_i, and only the one who has c_i and knows PW_i can get $h(ID_i \| K_{GWN})$. Moreover, in login and authentication phase, only the one who knows $h(ID_i \| K_{GWN})$ and ID_i can be authenticated by GWN. As a result, even if an attacker steals a smart card and reveals the stored data, he still cannot get essential authentication information.

5.7. User Anonymity and Untraceability

In login and authentication phase, data is transmitted via public channels such that a malicious user can eavesdrop. It denotes that the malicious user can get $M_4 = ID_i \oplus M_3$ and $M_8 = ID_i' \oplus h(r_g \| K_{GWN-S_j}')$. SC randomly chooses numbers r_i and $s \in Z_n^*$ and computes $M_3 = sX = sxP$. Only GWN can obtain M_3 because only GWN knows its private key x. As a result, only GWN can retrieve ID_i from M_4. Moreover, only GWN and S_j know K_{GWN-S_j} such that only S_j can retrieve ID_i' from M_8.

On the other hand, all transmitted parameters are computed with fresh random numbers such that no constant parameter is transmitted. This makes tracing a specific user is impossible. According to the above, the proposed scheme can ensure user anonymity and untraceability.

5.8. Further Assessment

In our scheme, parameters are generated, computed, or transmitted to achieve the goal with designated processes. Only proper designs result in secure mechanisms. We analyze our scheme thoroughly by investigating the processes with various attack scenarios and assessing user anonymity and untraceability in Section 5. According to the analyses, we obtain the following. Firstly, our scheme can resist leakage of the secret key shared between GWN and S_j because the secret key $K_{GWN-S_j} = h(SID_j \| K_{GWN})$ shared between GWN and S_j is concealed by hash function. No one can retrieve it from the transmitted parameters because of the properties of one-way hash function. Secondly, our scheme can resist various impersonation attacks. It is because one party can authenticate another by checking whether it knows the essential secret or not. And the integrity and the freshness of the transmitted data are verified at the same time. Thirdly, our scheme can resist replay attack. Different from Lee et al.'s scheme, random numbers instead of timestamps are used to verify the freshness. This approach also eliminates the burden of synchronization. Lastly, our scheme ensures user anonymity and untraceability. It is because the real identity is concealed with M_3, and M_3's in different sessions differ from each other.

6. Conclusions

Lee et al. proposed a three-factor authentication scheme by using hash and bio-hash functions to ensure the security of IoT communications. With through analyses, we find that their scheme suffers from failure sensor node authentication, failure mobile node authentication, replay attack, denial-of-service attack, and compromised user untraceability. Only with proper improvements, Lee et al. scheme can ensure security, efficiency, and important properties as claimed. We propose an improvement with ECC to overcome the drawbacks that Lee et al.'s scheme suffers from and preserve the advantages. According to the corresponding analysis, it is ensured that the proposed scheme achieves the goal to be realized and utilized in the real world.

Author Contributions: Conceptualization, Y.-F.C. and W.-L.T.; methodology, P.-L.H.; validation, Y.-F.C.; formal analysis, Y.-F.C.; investigation, K.-Y.L.; writing—original draft preparation, P.-L.H. and K.-Y.L.; writing—review and editing, Y.-F.C. and W.-L.T.; supervision, Y.-F.C. and W.-L.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Ministry of Science and Technology under the Grants MOST 109-2410-H-025-013- and MOST 108-2221-E-034-006-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, H.; Ben-Othman, J.; Mokdad, L.; Son, J.; Li, C. Research challenges and security threats to AI-driven 5G virtual emotion applications using autonomous vehicles, drones, and smart devices. *IEEE Netw.* **2020**, *34*, 288–294. [CrossRef]
2. Yang, M.; Luo, J.; Ling, Z.; Fu, X.; Yu, W. De-anonymizing and countermeasures in anonymous communication networks. *IEEE Commun. Mag.* **2015**, *53*, 60–66. [CrossRef]
3. Tai, W.L.; Chang, Y.F.; Li, W.H. An IOT notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks. *Inf. Secur. Appl.* **2017**, *34*, 133–141. [CrossRef]
4. Wei, Z.; Liu, F.; Masouros, C.; Vincent Poor, H. Fundamentals of physical layer anonymous communications: Sender detection and anonymous precoding. *arXiv* **2020**, arXiv:2010.09122.
5. Chang, Y.F.; Tai, W.L.; Hsu, M.H. A secure mobility network authentication scheme ensuring user anonymity. *Symmetry* **2017**, *9*, 307. [CrossRef]
6. Lin, C.C.; Chang, Y.F.; Chang, C.C.; Zheng, Y.Z. A fair and secure reverse auction for government procurement. *Sustainability* **2020**, *12*, 8567. [CrossRef]
7. Turkanović, M.; Brumen, B.; Hölbl, M. 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.* **2014**, *20*, 96–112. [CrossRef]
8. Farash, M.; Turkanović, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176. [CrossRef]
9. Amin, R.; Islam, S.; Biswas, G.; Khan, M.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62. [CrossRef]
10. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392. [CrossRef]
11. Zhang, L.; Zhang, Y.; Tang, S.; Luo, H. Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement. *IEEE Trans. Ind. Electron.* **2018**, *65*, 2795–2805. [CrossRef]
12. Aghili, S.; Mala, H.; Shojafar, M.; Peris-Lopez, P. LACO: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT. *Future Gener. Comput. Syst.* **2019**, *96*, 410–424. [CrossRef]
13. Lee, H.; Kang, D.; Ryu, J.; Won, D.; Kim, H.; Lee, Y. A three-factor anonymous user authentication scheme for Internet of Things environments. *J. Inf. Secur. Appl.* **2020**, *52*, 102494. [CrossRef]