Aghili, S.F., Mala, H., Peris-Lopez, P. (2018). Securing Heterogeneous Wireless Sensor Networks: Breaking and Fixing a Three-Factor Authentication Protocol. *Sensors*, 18 (11), 3663.

DOI: https://doi.org/10.3390/s18113663

sensors

MDPI

Article

Securing Heterogeneous Wireless Sensor Networks: Breaking and Fixing a Three-Factor Authentication Protocol

Seyed Farhad Aghili [1], Hamid Mala [1,*] and Pedro Peris-Lopez [2]

[1] Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib St., Isfahan 81746-73441, Iran; sf.aghili@eng.ui.ac.ir

[2] Department of Computer Science, University Carlos III of Madrid, Avda. de la Universidad 30, 28911 Leganés, Spain; pperis@inf.uc3m.es

* Correspondence: h.mala@eng.ui.ac.ir; Tel.: +98-31-379-35608

Received: 3 August 2018; Accepted: 21 September 2018; Published: 29 October 2018

Abstract: Heterogeneous wireless sensor networks (HWSNs) are employed in many real-time applications, such as Internet of sensors (IoS), Internet of vehicles (IoV), healthcare monitoring, and so on. As wireless sensor nodes have constrained computing, storage and communication capabilities, designing energy-efficient authentication protocols is a very important issue in wireless sensor network security. Recently, Amin et al. presented an untraceable and anonymous three-factor authentication (3FA) scheme for HWSNs and argued that their protocol is efficient and can withstand the common security threats in this sort of networks. In this article, we show how their protocol is not immune to user impersonation, de-synchronization and traceability attacks. In addition, an adversary can disclose session key under the typical assumption that sensors are not tamper-resistant. To overcome these drawbacks, we improve the Amin et al.'s protocol. First, we informally show that our improved scheme is secure against the most common attacks in HWSNs in which the attacks against Amin et al.'s protocol are part of them. Moreover, we verify formally our proposed protocol using the BAN logic. Compared with the Amin et al.'s scheme, the proposed protocol is both more efficient and more secure to be employed which renders the proposal suitable for HWSN networks.

Keywords: heterogeneous wireless sensor networks; authentication; traceability attack; de-synchronization attack

1. Introduction

In wireless sensor networks (WSNs) there are many sensor nodes scattered in a defined area [1]. These networks can be categorized into two important classes: homogeneous and heterogeneous sensor networks. On the one hand, in homogeneous sensor networks, all the sensor nodes are equal in terms of energy and hardware complexity. On the other hand, heterogeneous sensor networks (HWSNs) include various types of wireless sensor nodes with different capabilities and functions. In HWSNs, the sensors share their functions and increase the reliability of the network without increasing the cost of implementation [2–5]. Some of these sensors are low-cost, low-power and consequently have constrained computational power, transmission range, storage capacity and battery life [6]. It is clear that there are great needs to design energy-efficient protocols for such networks. In HWSN, users communicate to the sensor nodes to acquire data of their own interest. Therefore, the user and sensor node authentication is an important line of research in HWSN security which has recently awakened interest from the network security research community. In HWSN, the gateway node (GWN) plays an essential part in the authorization procedure since this element is the connection (input/output) with the all the elements outside the network. As shown in Figure 1, there are

five models in authenticating users and sensor nodes in HWSN [7]. In these five schemes, a user, a gateway node and a sensor node implement the authentication protocol by exchanging four messages (e.g., Figure 1(a.1–a.4)). In each scheme, there are four steps: (1) the gateway node authenticates the user (e.g., Figure 1(a.1)); (2) the sensor node authenticates the legitimate user and the gateway node (e.g., Figure 1(a.2)); (3) the sensor node verifies the legitimacy of the gateway node (e.g., Figure 1(a.3)); and finally, (4) in the last step, the user authenticates the legitimate sensor node (e.g., Figure 1(a.4)). Since HWSN nodes face with to many limitations in power consumption and communication range, models, in which a user and a sensor are a long way apart, are not practical, Figure 1e,b,d [8,9].

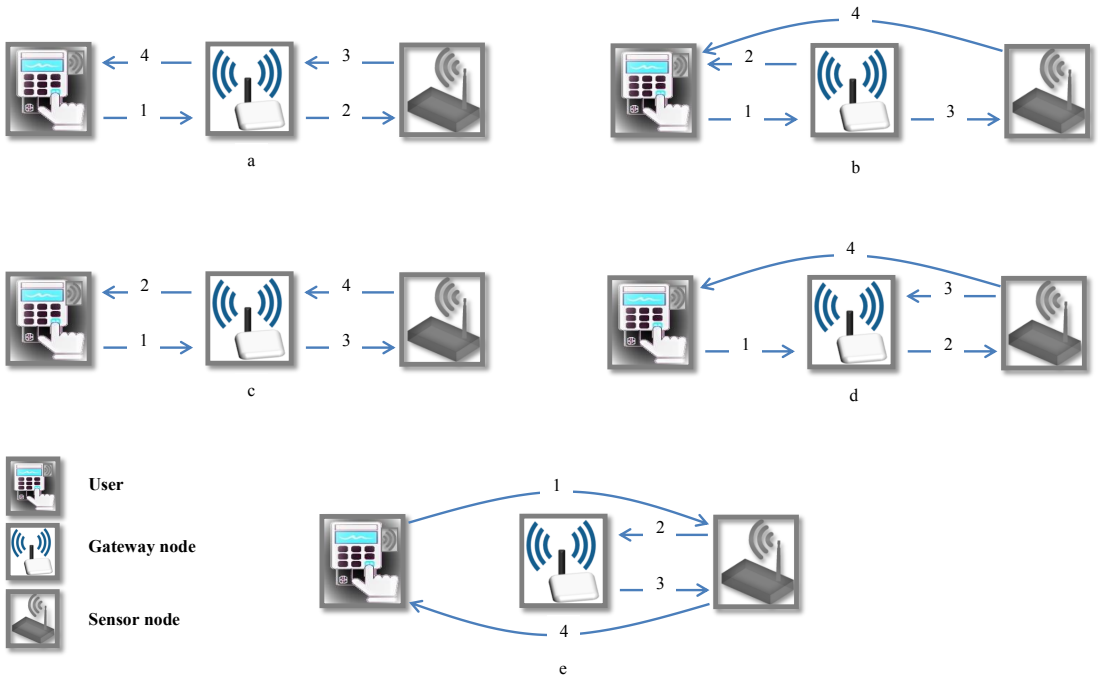

Figure 1. Five user authentication models in HWSN [7].

To tackle with security challenges of HWSN networks, we need lightweight enough and secure schemes. In the literature, authentication protocols are the most common adopted solution [7,10–14]. Unfortunately, most of them do not provide the required security and present important security pitfalls or are not energy-efficient. In this vein, recently, Amin et al. presented an untraceable and anonymous 3FA scheme for HWSNs. They used the model depicted in Figure 1a to design their protocol and asserted that their protocol can resist all common attacks known in the context of HWSN [15]. Nevertheless, in this article, we cryptanalyzed this protocol to show that this scheme is vulnerable against user impersonation, de-synchronization and session key disclosure attacks and also the adversary can trace the user. In order to hinder these attacks, we improve the Amin et al.'s protocol.

1.1. Our Contribution

The contributions of this article are summarized as below:

- At first, we present several serious security attacks against the Amin et al.'s scheme [15]. Our proposed attacks include de-synchronization, user impersonation, user traceability and session disclosure attacks.
- In order to increase the security level offered by Amin et al.'s protocol, we remedy the security faults found in their scheme.

- The security of the proposed scheme has be scrutinized from a formal and informal point of view. The attacks mentioned in Amin et al.'s protocol and other common security attacks have been considered in the design of the new protocol.
- The efficiency of our proposal is higher than the offered by Amin et al.'s scheme. Therefore, our scheme can be used for resource constrained sensors as the ones employed in HWSNs.

1.2. Paper Organization

The organization of the article is as follows. In Section 2, some related work are presented. Section 3 introduces the required preliminaries and notations. We review Amin et al.'s protocol in Section 4. Section 5 shows the security pitfalls of this scheme. We propose the improved scheme in Section 6. Then, we discuss the security of the proposed protocol in an informally way in Section 7, while, in Section 8, a formal analysis is presented. Finally, we extract some conclusions in Section 9.

2. Related Work

In a wireless sensor network, to allow a legitimate user to obtain information from a target sensor, the system needs to verify the validity of user by running an authentication protocol. In this section, we briefly discuss some existing schemes that aim to increase the security level of these networks.

Two-factor Authentication Schemes: Several two-factor authentication (2FA) schemes have been proposed for WSN, where the login phase of these protocols is based on passwords and smartcards.

In 2006, Wong et al. [16] presented a 2FA protocol based on the use of a hash function for wireless sensor networks, but the authors in [11] found that the protocol suffers from serious security pitfalls (i.e., replay, stolen-verifier and forgery attacks). To overcome these important weaknesses, authors in [11] proposed a new 2FA protocol based on passwords and smartcards. However, this protocol also is not immune against denial of service attacks and the nodes can be compromised [17].

In 2010, to improve the [11] protocol, Chen et al. [10] presented a bilateral authentication protocol in which three entities are involved (i.e., users, sensor nodes and the gateway node). In the same year, Khan et al. [12] showed that [11] fails in the authentication and in the key updating mechanism and presented a new protocol that they claimed it hinders the mentioned attacks. Later, Vaidya et al. [18] introduced several security vulnerabilities in [10–12] based on the stolen smartcard assumption. Xue et al. in 2013 presented a mutual authentication protocol based on temporal credentials, which is mainly based on the use of hash functions [7]. Nevertheless, He et al. [19] showed how the above protocol [7] is not resistant against user node and sensor node impersonation attacks and proposed a new temporal-credential-based protocol to overcome these weaknesses. In addition, Mir et al. [20] compromised the security of the healthcare system designed by He et al. [21], uncovering impersonation and password disclosure attacks. In addition, Turkanovic et al. [22] presented another bilateral authentication scheme in the context of HWSNs. However, Amin and Biswas [23] examined the Turkanovic et al. scheme and identified certain security problems (e.g., offline identity and password guessing attacks) and finally claimed to remove these security pitfalls in an efficient protocol. In the same year, Farash et al. [6] showed also some security shortcomings in [22] and proposed a new lightweight protocol. In the context of lightweight cryptography, Gope et al. [24] presented a 2FA protocol with especial security features including user anonymity and forward/backward secrecy. Soon, in [25], the authors analyzed the Gope's protocol by presenting a session key disclosure attack.

Three-Factor Authentication Schemes: In 2016, Amin et al. [26] pointed out how the Farash et al. protocol is susceptible to a number of attacks and proposed a new mechanism which was claimed to be resistant against these attacks. To enhance the security flaws of 2FA protocols, Amin et al. proposed a three-factor authentication (3FA) scheme based on password, smartcard and biometric trait linked to the legitimate user. However, Arasteh et al. [27] proposed replay and Denial-of-Service (DoS) attacks against Amin et al.'s scheme. In 2017, the authors in [28] presented an smartcard loss attack

against Amin et al.'s 3FA protocol [26]. They also showed that the attacker can reveal the session keys in other sessions of the protocol. To overcome the security flaws of this protocol, they proposed the enhanced scheme based on the Rabin's cryptosystem. In the same year, Jiang et al. [29] presented a solution to enhance the security of another 3FA protocol [30] that suffers from important security faults including traceability, identity guessing, offline password guessing, user impersonation and server impersonation attacks.

Chang et al. in [31] found several vulnerabilities in the Turkanovic et al. 2FA protocol [22] and presented an enhancement solution, but the scheme was shown to be vulnerable to a wide set of attacks such as traceability, information disclosure or session key attacks [15]. Eventually, Amin et al. [15] presented a new untraceable and anonymous 3FA scheme for HWSNs which was argued to be the improved version of Chang et al. scheme. Nevertheless, in this article, we scrutinize the security of this 3FA protocol and show how it is vulnerable to user impersonation, de-synchronization and session key disclosure attacks and also the adversary can trace the user. To prevent these attacks, we upgrade the Amin et al.'s protocol and analyze its security from a formal and informal perspective.

Privacy Schemes: In some of the protocols mentioned, the authors have stated that their schemes can preserve the user's privacy. To do this, the user's identifier is encoded using a dynamic identity. This anonymous identifier is used when the user communicates with the gateway node, and this information is useless for the attacker to reveal the user's identity [24]. In detail, in schemes [7,32,33], the authors claim that their proposals preserve users' privacy. Unfortunately, all of them fail in this purpose [24].

Threat Model: Our threat model mainly follows the Dolev–Yao model [34]. Therefore, the adversary can intercept, modify, delete and change any of messages transmitted over the insecure communication channel. The adversary can also execute side channel attacks and then obtain the secrets stored on the smartcard. In addition, the adversary can capture the sensors and reveal their private information stored in their memory as these devices do not have tamper protection mechanisms [24].

3. Preliminaries and Notations

This section first shows the notations used in this paper and then revises the proposed fuzzy extractor function for extracting the biometric parameters required for the third factor of the authentication procedure.

3.1. Notations

The notation used through this article is summarized in the Table 1.

3.2. Fuzzy Extractor

The facts that biometric tokens cannot be easily guessed, are difficult to be copied, shared and forged, and are not lost or forgotten makes biometric based authentication more preferable than traditional password based ones [35,36].

A fuzzy extractor can generate cryptography keys over noisy data. In other words, they are error tolerant. In detail, this is composed of two processes, a probabilistic algorithm GEN and a deterministic algorithm REP as described below:

1. The generation procedure (GEN): given a biometric input B_i, this probabilistic algorithm generates a secret key ψ_i and a non-secret string θ_i, i.e., $GEN(B_i) = (\psi_i, \theta_i)$.
2. The reproduction procedure (REP): given the noisy input B_i^* and the corresponding auxiliary string θ_i, this algorithm is able to recover the same key ψ_i as in the generation process, i.e., $\psi_i = REP(B_i^*, \theta_i)$.

Table 1. Notations.

Notation	Description
U_i	The *i-th* user
GWN	The gateway node
SC_i	The smartcard of U_i
S_j	The *j-th* sensor node
Z_q^*	Multiplicative group, where q is a large prime, $Z_q^* = \{x : 0 < x < q, gcd(x, q) = 1\}$
ID_i	Identity of U_i
SID_j	Identity of S_j
X_{GWN}	Secret key of GWN
f_i	Secret key linked to U_i
f_j	Secret key linked to S_j
PW_i	Password linked to U_i
B_i	Biometric trait linked to U_i
K_i	Nonce generated by U_i
K_j	Nonce generated by S_j
SK_i, SK_j, SK_G	Session key
$REP(\cdot), GEN(\cdot)$	Fuzzy extractor operations
ψ_i, θ_i	Outputs of $GEN(\cdot)$ algorithm
T_i	Timestamp
ΔT	Allowable transmission delay
$h(\cdot)$	One-way hash function
$\oplus$	Bitwise XOR operation
$\|$	Concatenation operation

4. Review of Amin et al.'s Scheme

In this section, we scrutinize the security of the authentication protocol proposed by Amin et al., which is composed of nine phases: (1) pre-deployment; (2) user registration; (3) login; (4) authentication and key agreement; (5) updating; (6) post-deployment; (7) password recovery; (8) password change; and (9) smartcard revocation.

4.1. Pre-Deployment Phase

Firstly, the gateway node GWN chooses X_{GWN} as a long-term secret key and assigns identities SID_j to the sensor nodes S_j ($1 \leq j \leq m$ for a population of m sensor nodes in the network). Then, the GWN calculates $f_j = h(SID_j \| X_{GWN})$ and stores $\langle SID_j, f_j \rangle$ into the memory of S_j.

4.2. User Registration Phase

Using a secure channel, the user U_i executes the following steps in conjunction with the GWN.

Step 1. U_i chooses an identity ID_i, attaches to it a personal credentials (e.g., social security number), and submits both values to the GWN.

Step 2. If the GWN does not find ID_i in the database, it generates $r_i \in_R Z_q^*$ and calculates $MI_i = h(ID_i \| r_i)$ and $f_i = h(MI_i \| X_{GWN})$. Both values ($\langle MI_i, f_i \rangle$) are stored in a new smartcard SC_i and the device is handed over to U_i.

Step 3. Once receiving the smartcard, U_i chooses a password PW_i and then uses a sensor device to obtain his biometric information B_i and finally writes $\langle PW_i, ID_i, B_i \rangle$ to the SC_i.

Step 4. SC_i uses the fuzzy extractor technique to calculate $(\psi_i, \theta_i) = GEN(B_i)$, it then computes $A_i = h(ID_i \| PW_i \| \psi_i)$, $E_i = \theta_i \oplus h(ID_i \| PW_i)$, $C_i = f_i \oplus h(PW_i \| \psi_i)$, $REC = PW_i \oplus h(ID_i \| \psi_i)$, $REG_i = h(ID_i \oplus \psi_i)$ and deletes f_i.

Finally, the smartcard contains the tuple $\langle MI_i, C_i, E_i, A_i, REC, REG_i, GEN(), REP(), h() \rangle$.

4.3. Login Phase

The user U_i follows these steps to access the data collected by sensor S_j.

Step 1. U_i inserts SC_i into the terminal and then enters ID_i' and PW_i' and also uses the sensor device to imprint his biometric information B_i'.

Step 2. SC_i retrieves $\theta_i' = E_i \oplus h(ID_i' \| PW_i')$ and computes $\psi_i' = REP(B_i', \theta_i')$, $f_i' = C_i \oplus h(PW_i' \| \psi_i')$ and $A_i' = h(ID_i' \| PW_i' \| \psi_i')$. SC_i verifies the correctness of A_i'. If so, SC_i concludes $ID_i' = ID_i$, $PW_i' = PW_i$ and $B_i' = B_i$; otherwise, SC_i denies U_i.

Step 3. SC_i generates $K_i \in_R Z_q^*$ and computes $N_i = h(MI_i \| K_i \| f_i \| T_1 \| SID_j)$, $L_i = K_i \oplus h(MI_i \| f_i \| T_1)$, $P_i = SID_j \oplus h(f_i \| T_1)$ and $Q_i = h(ID_i) \oplus h(K_i \| T_1)$ —T_1 represents the current timestamp.

Finally, SC_i sends the tuple $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ to GWN through an insecure channel.

4.4. Authentication and Session Key Agreement Phase

Two goals are achieved in this phase (see Figure 2): (1) U_i and S_j are authenticated through GWN; and (2) U_i and S_j set a session key. In particular, the following five steps are executed.

Step 1. After receiving the message $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ in login phase, the GWN checks whether the timestamp condition $\mid T_1 - T_2 \mid \leq \Delta T$ holds, T_2 being the current time of GWN. If the condition is fulfilled, the GWN aborts the connection. Otherwise, it calculates $f_i' = h(MI_i \| X_{GWN})$ and then decodes $K_i' = L_i \oplus h(MI_i \| f_i' \| T_1)$, $h(ID_i) = Q_i \oplus h(K_i' \| T_1)$ and $SID_j' = P_i \oplus h(f_i' \| T_1)$. It then computes $N_i' = h(MI_i \| K_i' \| f_i' \| T_1 \| SID_j')$ and checks the validity of the received N_i. If so, the GWN identifies to U_i as an authorized user. If not, it aborts the connection.

Step 2. Then, GWN calculates $f_j' = h(SID_j \| X_{GWN})$, $N_j = h(h(ID_i) \| f_j' \| T_2 \| K_i)$, $SS_j = h(ID_i) \oplus h(f_j' \| T_2)$ and $V_j = K_i \oplus h(ID_i)$. GWN then sends the tuple $\langle N_j, SS_j, V_j, T_2 \rangle$ to S_j.

Step 3. Upon receiving the message $\langle N_j, SS_j, V_j, T_2 \rangle$, S_j checks the validity of timestamp T_2. If $\mid T_2 - T_3 \mid > \Delta T$, it terminates the connection. Otherwise, S_j computes $h(ID_i) = SS_j \oplus h(f_j \| T_2)$, $K_i' = V_j \oplus h(ID_i)$ and the $N_j' = h(h(ID_i) \| f_j \| T_2 \| K_i')$ and verifies the validity of received N_j. If it is invalid, then S_j aborts the session. Otherwise, it generates $K_j \in_R Z_q^*$ and computes $SK_j = h(h(ID_i) \| SID_j \| K_i' \| K_j)$ as a session key and then computes $W_j = h(SK_j \| T_3)$ and $K_{ij} = K_i \oplus K_j$. Then, S_j sends the tuple $\langle W_j, K_{ij}, T_3 \rangle$ to GWN.

Step 4. Once the message $\langle W_j, K_{ij}, T_3 \rangle$ is received, the GWN verifies the freshness of T_3. If $\mid T_3 - T_4 \mid > \Delta T$, GWN aborts the connection. Otherwise, it decodes $K_j' = K_{ij} \oplus K_i$ and calculates the session key $SK_G = h(h(ID_i) \| SID_j' \| K_i' \| K_j')$. It then computes $W_j' = h(SK_G \| T_3)$ to verify the correctness of the received W_j. If the above verification fails, then GWN discontinues the session. Otherwise, it calculates $M_1 = h(SK_G \| K_j' \| T_4)$ and forwards the message $\langle M_1, K_{ij}, T_4 \rangle$ to U_i.

Step 5. Once the message $\langle M_1, K_{ij}, T_4 \rangle$ is received, U_i checks whether the condition $\mid T_4 - T_5 \mid \leq \Delta T$ is satisfied. If it is not fulfilled, U_i aborts the session. Otherwise, it calculates $K_j' = K_{ij} \oplus K_i$, $SK_i = h(h(ID_i) \| SID_j \| K_i \| K_j')$ and $M_1' = h(SK_i \| K_j' \| T_4)$ to verify the correctness of the received M_1. Now the entities are mutually authenticated and a session key $SK_i = SK_G = SK_j$ has been negotiated.

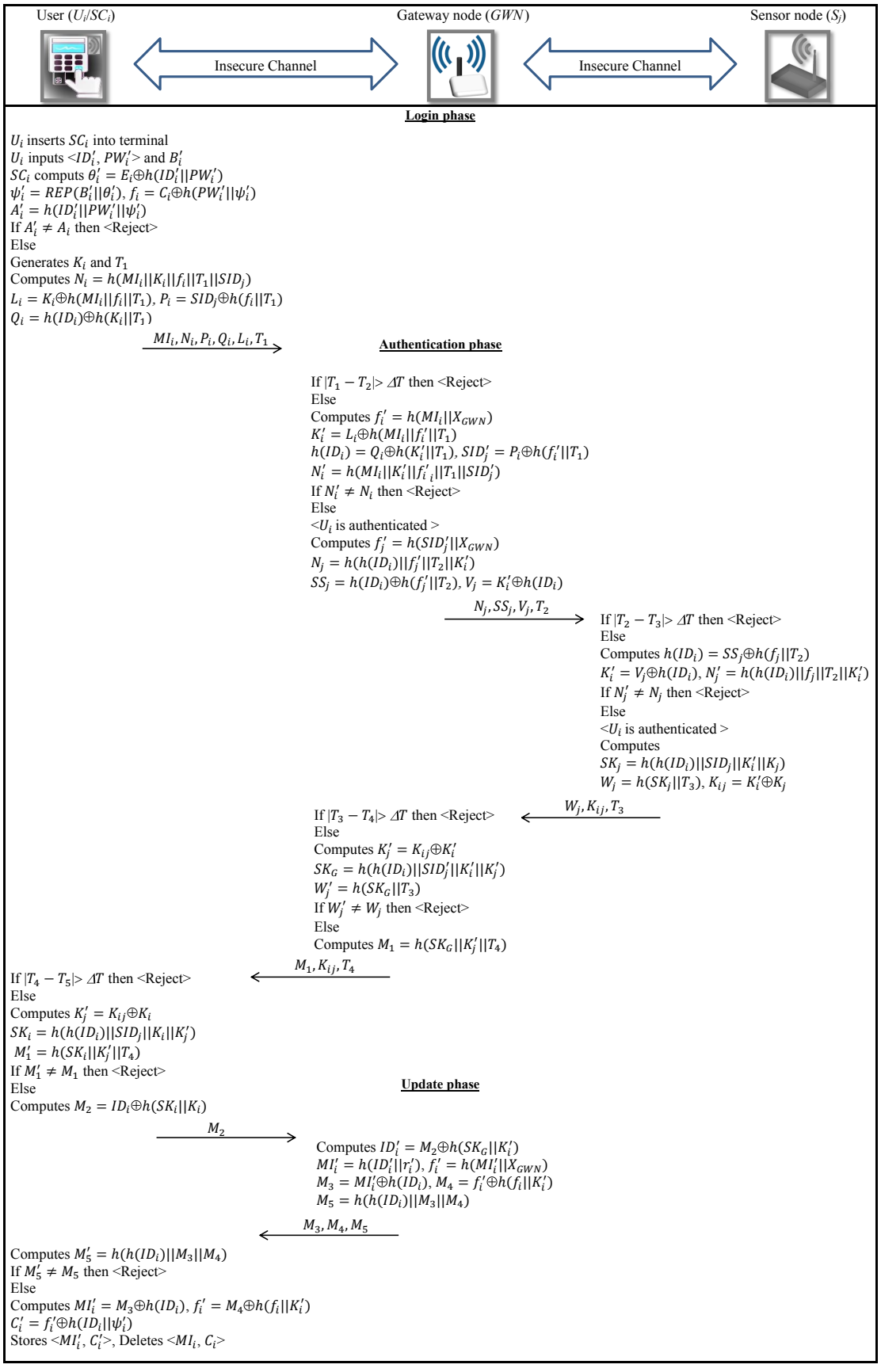

Figure 2. Authentication and key agreement phases in Amin et al.'s protocol [15].

4.5. Update Phase

In this phase, in order to achieve user untraceability, U_i updates $\langle MI_i, C_i \rangle$ as follows:

Step 1. U_i computes $M_2 = ID_i \oplus h(SK_i \| K_i)$ and sends it to GWN as a confirmation message. After receiving the message, GWN decodes $ID_i = M_2 \oplus h(SK_G \| K_i')$ and updates $MI_i' = h(ID_i \| r_i')$ and $f_i' = h(MI_i' \| X_{GWN})$, where $r_i' \in_R Z_q^*$. It then computes $M_3 = MI_i' \oplus h(ID_i)$, $M_4 = f_i' \oplus h(f_i \| K_i')$ and $M_5 = h(h(ID_i) \| M_3 \| M_4)$ and sends the tuple $\langle M_3, M_4, M_5 \rangle$ to U_i.

Step 2. After receiving the message $\langle M_3, M_4, M_5 \rangle$, U_i calculates $M_5' = h(h(ID_i) \| M_3 \| M_4)$ to check the validity of the received M_5. If so, it extracts $MI_i' = M_3 \oplus h(ID_i)$ and $f_i' = M_4 \oplus h(f_i \| K_i')$ and computes $C_i' = f_i' \oplus h(ID_i \| \psi_i)$. Then, U_i rewrites $\langle MI_i', C_i' \rangle$ to SC_i instead of previous $\langle MI_i, C_i \rangle$.

4.6. Post-Deployment Phase

A new sensor node S_k is used in this phase to replace a damaged sensor node S_j. The GWN generates a new identity SID_k and then calculates $f_k = h(SID_k \| X_{GWN})$ and stores $\langle SID_k, f_k \rangle$ in S_k's memory.

4.7. Password Recovery Phase

U_i executes this phase when he forgets his password. U_i needs to insert SC_i in the card reader and enter his identity ID_i along with B_i. Now, the SC_i computes $\psi_i' = REP(B_i' \| \theta_i)$ and $REG_i' = h(ID_i \| \psi_i')$. Then, SC_i checks whether $REG_i' = REG_i$. If so, then it computes $PW_i = REC \oplus h(ID_i \| \psi_i)$ and sends the recovered password to the user.

4.8. Password Change Phase

The password of the user U_i can be updated by executing the updating procedure with SC_i and without the intervention of GWN. In detail, the following steps show how the user can update the old password PW_i for a new one PW_i^{new}.

Step 1. U_i inserts SC_i in to the terminal and enters $\langle ID_i', PW_i' \rangle$ along with biometric information B_i'.

Step 2. SC_i uses the fuzzy extractor technique to calculate $(\psi_i', \theta_i') = GEN(B_i')$, it then computes $A_i^\star = h(ID_i' \| PW_i' \| \psi_i')$ and $f_i' = C_i \oplus h(PW_i' \| \psi_i')$. If $(A_i^\star = A_i)$, then SC_i requests U_i to enter a new password PW_i^{new} at SC_i; otherwise, SC_i aborts this procedure.

Step 3. Now, SC_i calculates $A_i^{new} = h(ID_i \| PW_i^{new} \| \psi_i')$, $E_i^{new} = \theta_i' \oplus h(ID_i \| PW_i^{new})$, $C_i^{new} = C_i \oplus h(PW_i \| \psi_i' \oplus h(PW_i^{new} \| \psi_i'))$, $REC^{new} = PW_i^{new} \oplus h(ID_i \| \psi_i')$ and replaces $\langle A_i, E_i, C_i, REC \rangle$ with $\langle A_i^{new}, E_i^{new}, C_i^{new}, REC^{new} \rangle$.

4.9. Smartcard Revocation Phase

Generally, smartcards can be lost, stolen or damaged. Thus, the smartcard revocation phase is very important. This phase is executed as described below:

Step 1. U_i submits ID_i and a personal credential (e.g., social security number) to the smartcard issuer.

Step 2. If the smartcard issuer can find ID_i in the database, it generates $r_i \in_R Z_q^*$ and calculates $MI_i^{new} = h(ID_i \| r_i)$ and $f_i^{new} = h(MI_i^{new} \| X_{GWN})$. It then writes $\langle MI_i^{new}, f_i^{new} \rangle$ into a new smartcard SC_i^{new} and delivers it to the user U_i.

Step 3. Once SC_i^{new} is received, U_i chooses a password PW_i^{new}, receives new biometric information B_i^{new} from the sensor and writes $\langle PW_i, ID_i, B_i \rangle$ to the SC_i.

Step 4. SC_i uses the fuzzy extractor technique to calculate $(\psi_i, \theta_i) = GEN(B_i^{new})$. It then computes $A_i^{new} = h(ID_i \| PW_i^{new} \| \psi_i)$, $E_i^{new} = \theta_i \oplus h(ID_i \| PW_i^{new})$, $C_i^{new} = f_i^{new} \oplus h(PW_i^{new} \| \psi_i)$, $REC^{new} = PW_i^{new} \oplus h(ID_i \| \psi_i)$ and $REG_i^{new} = h(ID_i \oplus \psi_i)$, and implants $\langle C_i^{new}, E_i^{new}, A_i^{new}, REC^{new}, REG_i^{new}, GEN(), REP(), h() \rangle$ into SC_i and deletes f_i^{new}.

5. Security Analysis of Amin et al.'s Protocol

In [15], the authors claimed that the adversary/attacker A cannot trace or identify the user U_i using the transmitted messages. Moreover, they claimed that the attacker cannot impersonate the user by accessing to the old login eavesdropped messages.

Unfortunately, for Amin et al.'s protocol, we show how the proposed protocol is not immune against user impersonation and de-synchronization attacks. The user can be also tracked by an attacker who eavesdrops on only one protocol session. In addition, we provide evidence of how an adversary can easily obtain the session key under the assumption that sensors are not tamper-resistant.

5.1. User Impersonation Attack

In this attack, we point out how an adversary A is authenticated by both the gateway node GWN and the sensor node S_j. The attack is described below:

- A eavesdrops on the message $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ sent by U_i to the GWN, then he changes the Q_i value to Q_i'.
- After receiving the message $\langle MI_i, N_i, P_i, Q_i', L_i, T_1 \rangle$ in the login phase, the GWN checks two issues: (1) timestamp condition $\mid T_1 - T_2 \mid \leq \Delta T$ and (2) validity of the received $N_i = h(MI_i \| K_i \| f_i \| T_1 \| SID_j)$, which does not depend on Q_i. Thus, the GWN accepts these two conditions and computes $h(ID_i)^\star = Q_i' \oplus h(K_i' \| T_1)$ and SID_j'. It then calculates N_i'. Now, the GWN believes that A is an authorized user.
- Then, GWN calculates f_j' and then computes $N_j = h(h(ID_i)^\star \| f_j' \| T_2 \| K_i)$, $SS_j = h(ID_i)^\star \oplus h(f_j' \| T_2)$ and $V_j = K_i \oplus h(ID_i)^\star$ and sends the tuple $\langle N_j, SS_j, V_j, T_2 \rangle$ to S_j.
- S_j check the correctness of timestamp and computes $h(ID_i)^\star = SS_j \oplus h(f_j \| T_2)$, $K_i' = V_j \oplus h(ID_i)^\star$ and $N_j' = h(h(ID_i)^\star \| f_j \| T_2 \| K_i')$ and checks validity of the received N_j. It generates $K_j \in_R Z_q^*$ and computes $SK_j = h(h(ID_i)^\star \| SID_j \| K_i' \| K_j)$ as a session key and then computes W_j and K_{ij}. Now, the S_j also believes that A is an authorized user and sends the tuple $\langle W_j, K_{ij}, T_3 \rangle$ to GWN.
- The GWN checks the validity of T_3. It decodes K_j' and computes the session key $SK_G = h(h(ID_i)^\star \| SID_j' \| K_i' \| K_j')$. It then computes $W_j' = h(SK_G \| T_3)$ and checks validity of the received W_j and computes $M_1 = h(SK_G \| K_j' \| T_4)$ and sends the message $\langle M_1, K_{ij}, T_4 \rangle$ to U_i which is the adversary. At this point, the adversary sends the random number M_2 to GWN as a confirmation message. After receiving the message, GWN uses the message to obtain ID_i which is the random number. Due to the absence of any checking process, it employs this value to compute M_3, M_4 and M_5 and then sends the tuple $\langle M_3, M_4, M_5 \rangle$ to the adversary.

Following this attack, the adversary cheats GWN and S_j to pass the protocol with the success probability of "1". Moreover, GWN and S_j establish the wrong session key along with $h(ID_i)^\star$.

5.2. De-Synchronization Attack

In Amin et al.'s authentication phase, an adversary A by eavesdropping only one session can reveal the $h(ID_i)$ of the user U_i and uses it to render the user to a de-synchronization state as follows. Note that, in the proposed attack, the superscript j indicates the parameters of the j-th run of protocol, $j = 1, 2$. In addition, in the Amin et al. scheme, the values of $h(ID_i)$ of the user U_i is a constant value. In detail, the attack can be executed following the steps described below:

- A eavesdrops on the message $M_3^1 = MI_i^2 \oplus h(ID_i)$ from session 1;
- A eavesdrops on the message MI_i^2 from session 2;
- A obtains $h(ID_i)$ from equation $h(ID_i) = M_3^1 \oplus MI_i^2$;
- In Step 6 of the authentication phase, A intercepts $\langle M_3^2, M_4^2, M_5^2 \rangle$ and modifies them to $M_3^\star$, $M_4^\star$ and $M_5^\star = h(h(ID_i) \| M_3^\star \| M_4^\star)$;
- A sends the tuple $\langle M_3^\star, M_4^\star, M_5^\star \rangle$ to U_i;

- U_i calculates $M_5^\star = h(h(ID_i)\|M_3^\star\|M_4^\star)$ and then checks validity of the received $M_5^\star$. Then, it extracts $MI_i' = M_3^\star \oplus h(ID_i)$ and $f_i' = M_4^\star \oplus h(f_i\|K_i')$ and computes $C_i' = f_i' \oplus h(ID_i\|\psi_i)$. Then, U_i rewrites $\langle MI_i', C_i'\rangle$ to SC_i instead of previous $\langle MI_i, C_i\rangle$.

Following this attack, the adversary compels the U_i to insert the wrong $\langle MI_i, C_i\rangle$ into SC_i's memory. Now, U_i cannot use SC_i to do the login.

5.3. User Traceability Attack

Following the privacy model proposed by Ouafi and Phan [37], the attacker can perform following phases to mount a traceability attack.

Step 1. In round n, A sends an $Execute\ query(GWN, U_0, n)$ and eavesdrops on messages $MI_{0,n}^{U_0}$, $Q_{0,n}^{U_0} = h(ID_0)_n^{U_0} \oplus h(K_{0,n}^{U_0}\|T_{1,n}^{U_0})$, $T_{1,n}^{U_0}$, $V_{j,n}^{S_j} = K_{0,n}^{U_0} \oplus h(ID_0)_n^{U_0}$ and $M_{3,n}^{GWN}$;

Step 2. The adversary A selects two users U_0 and U_1 and sends a $Test\ query(U_1, U_0, n+1)$ and depending on the random bit $b \in \{0,1\}$ the adversary A receives a $h(ID_b)^{U_b} \in \{h(ID_0)^{U_0}, h(ID_1)^{U_1}\}$ corresponding to users $\{U_0, U_1\}$;

Step 3. A sends an $Execute\ query(GWN, U_b, n+1)$ and eavesdrops on messages $MI_{b,n+1}^{U_b}$, $Q_{b,n+1}^{U_b} = h(ID_b)_{n+1}^{U_b} \oplus h(K_{b,n+1}^{U_b}\|T_{1,n+1}^{U_b})$, $T_{1,n+1}^{U_b}$, $V_{j,n+1}^{S_j} = K_{b,n+1}^{U_b} \oplus h(ID_b)_{n+1}^{U_b}$ and $M_{3,n+1}^{GWN}$;

Step 4. A guesses the random bit $b = 0$ if $h(ID_0)_n^{U_0} = h(ID_b)_{n+1}^{U_b}$ with a probability higher than a random coin flip following the procedure described below.

Step 5. We have,

$$h(ID_b)_{n+1}^{U_b} = Q_{b,n+1}^{U_b} \oplus h((V_{j,n+1}^{S_j} \oplus (MI_{b,n+1}^{U_b} \oplus M_{3,n}^{GWN})\|T_{1,n+1}^{U_b}),$$

$$h(ID_0)_n^{U_0} = Q_{0,n}^{U_0} \oplus h((V_{j,0}^{S_j} \oplus (MI_{b,n+1}^{U_b} \oplus M_{3,n}^{GWN})\|T_{1,n}^{U_0}),$$

- As $h(ID_i)^{U_i}$ is constant and the user does not update it,
- If $h(ID_b)_{n+1}^{U_b} = h(ID_0)_n^{U_0}$, then $U_b = U_0$.

Step 6. As a result, we can express $Adv_A^{UNT}(k) = |Pr[A\ guesses\ b\ correctly] - \frac{1}{2}| = |1 - \frac{1}{2}| = \frac{1}{2} \gg \epsilon(k)$;

Following the described attack, the attacker can trace any target user U_i. In other words, Amin et al.'s scheme is not resistant against user traceability attack.

5.4. Session Key Disclosure Attack

As described in Section 5.2, A can extract $h(ID_i)$ belonged to U_i. Thus, if we assume that the sensor S_j is not equipped with tamper-resistant, A obtains $\langle SID_j, f_j\rangle$ from sensor's memory—note that the adversary does not require f_j to execute the proposed attack. Then, it executes the session key disclosure attack as follows:

- A eavesdrops on messages T_1 and $V_j = K_i' \oplus h(ID_i)'$;
- A obtains K_i from equation $K_i = V_j \oplus h(ID_i)$;
- A obtains K_j from equation $K_j = K_{ij} \oplus K_i$;
- A computes the session key SK_j using the $SK_j = h(h(ID_i)\|SID_j\|K_i'\|K_j)$.

Therefore, an adversary can disclose the session key in Amin et al.'s protocol.

Finally, we would like to highlight that all our proposed attacks exploit the fact that the bitwise XOR operation is a source of vulnerability against passive and active attacks [38–40].

6. Our Proposed Protocol

We present an enhanced version of Amin et al.'s protocol to remedy its security pitfalls. The scheme, as the original proposal, is split into night phases: (1) pre-deployment; (2) user registration;

(3) login; (4) authentication and key agreement; (5) update; (6) post-deployment; (7) password recovery; (8) password change; and (9) smart revocation. As we only enhanced the (3), (4), and (5) phases, these are the ones that we describe.

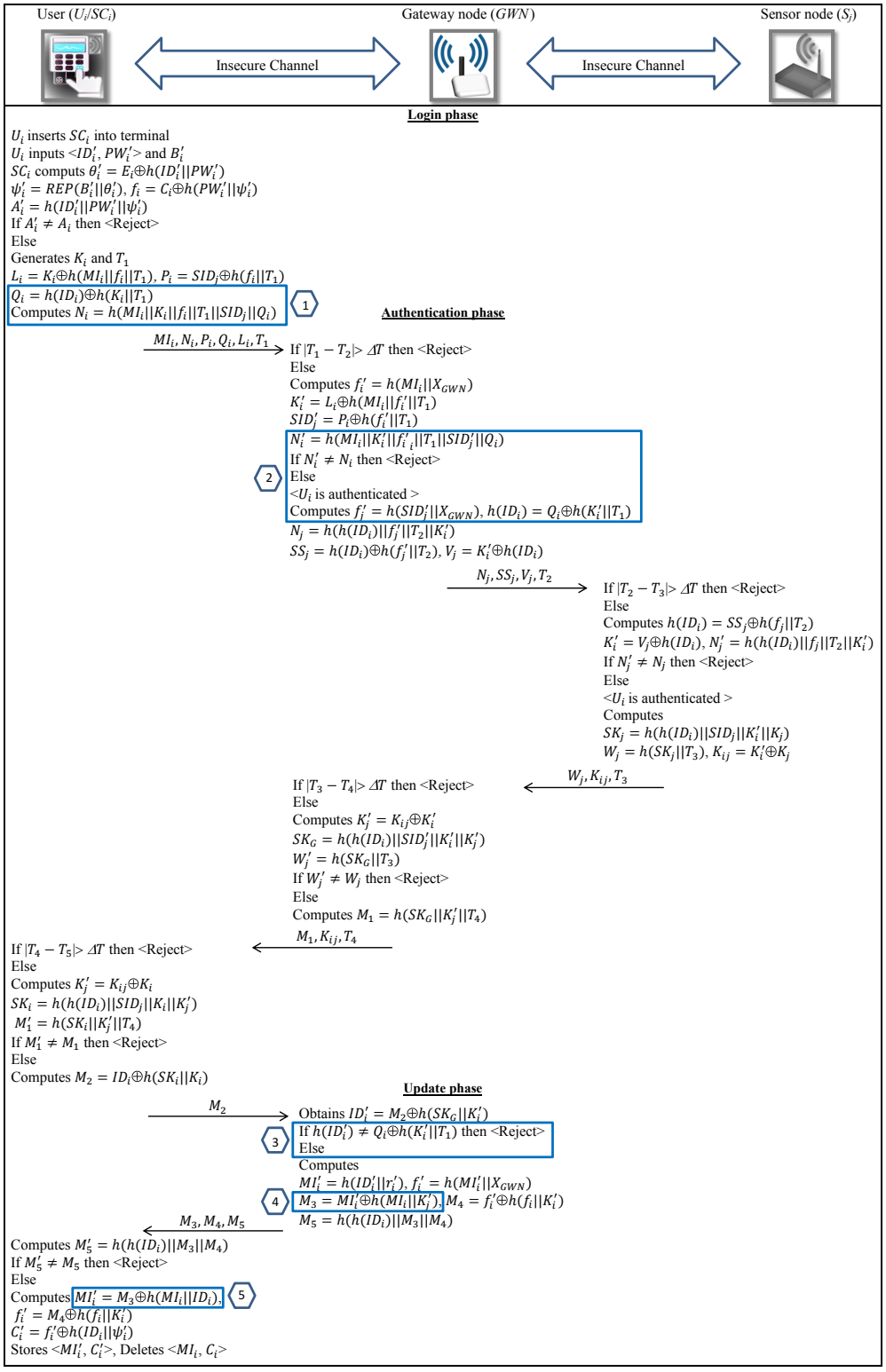

Figure 3. Modified Amin et al.'s authentication and key agreement phase. Changes are highlighted by boxes in the proposed scheme.

In summary, the enhanced authentication and key agreement phase, and update phase of the proposed scheme, as shown in the blue boxes in Figure 3, have five important changes. To prevent the user impersonation attack, the user makes uses of Q_i in the message N_i. Subsequently, the gateway node GWN verifies this value to authenticate the legitimate user (boxes number 1 and 2). To overcome the de-synchronization attack, we change the format of message M_3 as well as the equation the user employs to update MI_i. Therefore, the attacker cannot obtain $h(ID_i)$ by XORing these two values (boxes number 4 and 5). To avoid the replay attack, the gateway node GWN checks the validity of M_2 by verifying the value of $h(ID_i)$ (box number 3).

6.1. Login Phase

In this phase, we employ the Q_i in N_i to guarantee the integrity of Q_i. U_i performs the following steps to login when it wishes to access data collected by S_j:

Step 1. U_i inserts SC_i into the terminal and then enters ID_i' and PW_i' and also uses the sensor device to imprint his biometric information B_i'.

Step 2. SC_i retrieves $\theta_i' = E_i \oplus h(ID_i' \| PW_i')$ and computes $\psi_i' = REP(B_i', \theta_i')$, $f_i' = C_i \oplus h(PW_i' \| \psi_i')$ and $A_i' = h(ID_i' \| PW_i' \| \psi_i')$. SC_i checks validity of A_i'. If so, SC_i implies $ID_i' = ID_i$, $PW_i' = PW_i$ and $B_i' = B_i$; otherwise, SC_i denies U_i.

Step 3. SC_i generates $K_i \in_R Z_q^*$ and calculates $L_i = K_i \oplus h(MI_i \| f_i \| T_1)$, $P_i = SID_j \oplus h(f_i \| T_1)$, $Q_i = h(ID_i) \oplus h(K_i \| T_1)$ and $N_i = h(MI_i \| K_i \| f_i \| T_1 \| SID_j \| Q_i)$, T_1 being the current timestamp.

After this, SC_i forwards the tuple $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ to GWN using a public communication channel.

6.2. Authentication and Session Key Agreement Phase

At this point, U_i and S_j are authenticated through GWN and a session key is set between both entities. In addition, we modify the message M_3 to tackle the attacker when she tries to obtain $h(ID_i)$ in the next session. In Figure 3, we summarize the details of this phase:

Step 1. Once the message $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ is received in the Login phase, the GWN checks whether the timestamp condition $| T_1 - T_2 | \leq \Delta T$ holds. If the condition is fulfilled, the GWN terminates the connection. Otherwise, it calculates $f_i' = h(MI_i \| X_{GWN})$ and then decodes $K_i' = L_i \oplus h(MI_i \| f_i' \| T_1)$ and $SID_j' = P_i \oplus h(f_i' \| T_1)$. It then calculates $N_i' = h(MI_i \| K_i' \| f_i' \| T_1 \| SID_j' \| Q_i)$ and checks validity of the received N_i. If so, the GWN identifies to U_i as an authorized user. If not, it terminates the connection.

Step 2. Then, GWN obtains $h(ID_i) = Q_i \oplus h(K_i' \| T_1)$ and calculates $f_j' = h(SID_j \| X_{GWN})$ and then computes $N_j = h(h(ID_i) \| f_j' \| T_2 \| K_i)$, $SS_j = h(ID_i) \oplus h(f_j' \| T_2)$ and $V_j = K_i \oplus h(ID_i)$, T_2 being the current timestamp. GWN then forwards the tuple $\langle N_j, SS_j, V_j, T_2 \rangle$ to S_j.

Step 3. Once the message $\langle N_j, SS_j, V_j, T_2 \rangle$ is received, S_j checks validity of the timestamp T_2. If $| T_2 - T_3 | > \Delta T$, it terminates the connection. Otherwise, S_j calculates $h(ID_i) = SS_j \oplus h(f_j \| T_2)$, $K_i' = V_j \oplus h(ID_i)$ and $N_j' = h(h(ID_i) \| f_j \| T_2 \| K_i')$ and checks validity of the received N_j. If the verification fails, then S_j aborts the session. Otherwise, it generates $K_j \in_R Z_q^*$ and computes $SK_j = h(h(ID_i) \| SID_j \| K_i' \| K_j)$ as the session key and then computes $W_j = h(SK_j \| T_3)$ and $K_{ij} = K_i \oplus K_j$. Finally, S_j sends the tuple $\langle W_j, K_{ij}, T_3 \rangle$ to GWN.

Step 4. Once the message $\langle W_j, K_{ij}, T_3 \rangle$ is received, the GWN verifies the correctness of T_3. If $| T_3 - T_4 | > \Delta T$, GWN aborts the connection. Otherwise, it decodes $K_j' = K_{ij} \oplus K_i$ and computes the session key $SK_G = h(h(ID_i) \| SID_j' \| K_i' \| K_j')$. It then computes $W_j' = h(SK_G \| T_3)$ and checks the validity of the received W_j. If the above verification fails, then GWN discontinues the session. Otherwise, it calculates $M_1 = h(SK_G \| K_j' \| T_4)$ and forwards the message $\langle M_1, K_{ij}, T_4 \rangle$ to U_i.

Step 5. Once the message $\langle M_1, K_{ij}, T_4 \rangle$ is received, U_i checks whether the condition $\mid T_4 - T_5 \mid \leq \Delta T$ is satisfied. If it does not fulfilled, U_i ends the session. Otherwise, it calculates $K'_j = K_{ij} \oplus K_i$, $SK_i = h(h(ID_i) \| SID_j \| K_i \| K'_j)$ and $M'_1 = h(SK_i \| K'_j \| T_4)$ and checks the validity of M_1. At this point, the entities are mutually authenticated and a session key $SK_i = SK_G = SK_j$ has been negotiated.

6.3. Update Phase

In this phase, U_i updates $\langle MI_i, C_i \rangle$ in order to achieve user untraceability, as described in the next steps and depicted in Figure 3:

Step 1. U_i computes $M_2 = ID_i \oplus h(SK_i \| K_i)$ and sends it to GWN as a confirmation message. After receiving the message, GWN decodes $ID_i = M_2 \oplus h(SK_G \| K'_i)$ and checks if the condition $h(ID_i) = Q_i \oplus h(K'_i \| T_1)$ holds. If the verification fails, then GWN aborts the session. Otherwise, it updates $MI'_i = h(ID_i \| r'_i)$ and $f'_i = h(MI'_i \| X_{GWN})$, where $r'_i \in_R Z^*_q$. It then computes $M_3 = MI'_i \oplus h(MI_i \| K'_j)$, $M_4 = f'_i \oplus h(f_i \| K'_i)$ and $M_5 = h(h(ID_i) \| M_3 \| M_4)$ and sends the tuple $\langle M_3, M_4, M_5 \rangle$ to U_i.

Step 2. After receiving the message $\langle M_3, M_4, M_5 \rangle$, U_i calculates $M'_5 = h(h(ID_i) \| M_3 \| M_4)$ and then checks validity of M_5. If so, it extracts $MI'_i = M_3 \oplus h(MI_i \| K'_j)$ and $f'_i = M_4 \oplus h(f_i \| K'_i)$ and computes $C'_i = f'_i \oplus h(ID_i \| \psi_i)$. Then, U_i rewrites $\langle MI'_i, C'_i \rangle$ to SC_i instead of previous $\langle MI_i, C_i \rangle$.

7. Security Analysis of the Proposed Protocol

The proposed protocol is analyzed from an informal and formal point of view. This analysis shows how the proposed scheme withstands relevant and common security attacks.

The informal security analysis of a security scheme discusses its robustness against the common attacks known in its context. However, the formal security analysis methods employ mathematics or logic tools such as BAN-logic [41], AVISPA [42] or Proverif [43] to formally scrutinize the security of a cryptographic protocol. In this article, we employ the BAN-logic tool to formally verify our proposed protocol.

7.1. Informal Security Analysis

In this section, we point out how our proposed protocol withstands against relevant and well-known attacks.

7.1.1. Stolen Smartcard Attack

In our proposal, if the smartcard SC_i is stolen or lost, the adversary can access its memory and obtain all the information MI_i, A_i, E_i, C_i, REC and REG_i stored in the smartcard. Note that, in our protocol, the smartcard is not tamper-resistant. Since some values (ID_i, PW_i and B_i) are unknown for the adversary, s/he cannot compute $\theta'_i = E_i \oplus h(ID'_i \| PW'_i)$, $\psi'_i = REP(B'_i, \theta'_i)$ and $f'_i = C_i \oplus h(PW'_i \| \psi'_i)$ without having any information about these parameters. Furthermore, it is also computationally unfeasible for the attacker to disclose the ID_i, PW_i and the secret biometric information B_i of the user U_i thanks to the collision-resistance property of the one-way hash function. Thus, the proposed protocol is secure against the stolen smartcard attack.

7.1.2. Offline Password Guessing Attack

In our scheme, the password PW_i of the user U_i is involved in A_i, E_i, C_i and REC values, which are stored in the smartcard. As discussed above, the adversary A cannot use any of these stored items to obtain the password. In addition, using the messages transferred from the user U_i, the attacker cannot relate these messages to the items stored on the smartcard to find useful information to verify her/his guess about PW_i. Therefore, our proposed scheme is robust against offline password guessing attack.

7.1.3. Privileged Insider Attack

In this kind of attack, the insider attacker tries to impersonate the legitimate user by using this user's password. However, in the user registration phase of our scheme, U_i only submits ID_i as a registration request. In addition, all the messages transmitted via a public channel are independent of ID_i. Thus, by no means can the insider of GWN get U_i's password. That is, our proposed protocol is resistant against the privileged insider attack.

7.1.4. Offline Identity Guessing Attack

On this occasion, the adversary tries to obtain knowledge about the real identity ID_i of a user U_i —the user and GWN are the unique entities who know this information. In our proposal, the adversary cannot derive ID_i from information obtained from the smartcard. In addition, ID_i is never passed over the public communication channel. As a consequence of using the one-way hash function $h(\cdot)$, the adversary cannot find any useful information related to ID_i to verify her/his guess. Therefore, our proposed scheme is robust against identity guessing attack.

7.1.5. User Impersonation Attack

In this attack, the adversary aims to cheat GWN by attempting to take the place of a legitimate user in the logging phase. S/he may use the eavesdropped login message $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ of the previous sessions to conduct her/his attack. We show how our scheme is resistant against this attack. Once the eavesdropped message is received, the GWN checks the legitimacy of the user U_i by validating $N_i = h(MI_i \| K_i \| f_i \| T_1 \| SID_j \| Q_i)$. A has to possess f_i and $h(ID_i)$ to forge N_i. However, without having any knowledge about the password ID_i, the biometric key and the SID_j of the smartcard, the adversary A cannot calculate a valid N_i. Therefore, our proposed scheme is secure against user impersonation attack.

7.1.6. Gateway Node Impersonation Attack

To impersonate the gateway node, the adversary has to forge the message $\langle N_j, SS_j, V_j, T_2 \rangle$. Thus, the adversary A needs to know f_j, K_i and $h(ID_i)$ to compute $N_j = h(h(ID_i) \| f_j' \| T_2 \| K_i)$, which is impossible. Thus, A cannot forge the aforementioned message. In addition, A cannot compute $M_1 = h(SK_G \| K_j' \| T_4)$ and $K_{ij} = K_i \oplus K_j$, which are created by GWN. Therefore, our proposed scheme resists GWN impersonation attack.

7.1.7. Sensor Node Impersonation Attack

In the authentication phase, the typical sensor node S_j computes $W_j = h(SK_j \| T_3)$ and $K_{ij} = K_i \oplus K_j$ and sends these values along with T_3 to the gateway node GWN. To forge the messages W_j and K_{ij}, the adversary A must compute $SK_j = h(h(ID_i) \| SID_j \| K_i \| K_j)$ and must know K_i and K_j. Moreover, A cannot compute SK_j without the knowing $h(ID_i)$ and SID_j. Therefore, A cannot compute S_j's messages to execute a sensor node impersonation attack.

7.1.8. Session Key Security

In the authentication and session key agreement, the attacker can eavesdrop the messages $W_j = h(SK_j \| T_3)$ and $M_1 = h(SK_G \| K_j \| T_4)$. Nevertheless, the session key $SK_j = SK_G = h(h(ID_i) \| SID_j \| K_i \| K_j)$ is protected by the usage of the one-way hash function $h(\cdot)$. For this, it is computationally impossible for the adversary to derive the used key. Thus, our proposed scheme provides session key security.

7.1.9. User Anonymity

In our proposed protocol, the identity ID_i of user U_i is never passed in plain-text over an insecure communication channel. In this sense, $h(ID_i)$ is the value transmitted in the public

messages. Due to the collision-resistant property of the one-way hash function $h(\cdot)$, deriving ID_i from $h(ID_i)$ is computationally impossible for the attacker. Therefore, our proposed scheme preserves user anonymity.

7.1.10. Preserving User Untraceability

In this attack, an adversary A aims to determine whether two messages are generated by the same (unknown) user. Luckily, in our proposal, the attacker cannot be able to find any relationship between Q_i, M_2 and user's identity ID_i. Furthermore, it must be noted that, in our proposed protocol, all the parameters used in the messages $\langle MI_i, N_i, P_i, Q_i, L_i, T_1 \rangle$ are random. Moreover, when the update phase of the protocol is executed, U_i updates $\langle MI_i, C_i \rangle$ for the next session. Therefore, A cannot determine whether two protocol sessions are linked to the same user. Therefore, in our proposed protocol, users cannot be tracked.

7.1.11. Replay Attack

In the replay attack, the adversary forwards eavesdropped messages of the protocol (previous sessions) to try to deceive legitimate entities. The timestamp values and random numbers used in all messages of the protocol prevents any replay efforts from attacker. Therefore, replay attacks can be identified by verifying the freshness of the timestamp values and random numbers. Therefore, the replay attack does not work in our scheme.

7.2. Formal Security Analysis

We use BAN-logic [41] to conduct the security analysis of the authentication and key agreement phase of our proposal. Table 2 summarizes the used notation. Thereupon, we introduce the two main rules used in our analysis.

R1 (Shared key rule). $\frac{P|\equiv P \overset{k}{\longleftrightarrow} Q, P \triangleleft [X]_k}{P|\equiv Q|\sim X}$, if P believes that s/he shared the key K with Q, and P receives the message $[X]_k$; then, P believes that Q sent X.

R2 (Belief rule). $\frac{P|\equiv Q|\sim (X,Y)}{P|\equiv Q|\sim X}$, if P believes Q sends the message set (X, Y); then, P believes Q sends the message X.

Our formal security analysis is split into the following steps:

Step 1. Protocol messages.

PM1: $MI_i, N_i, P_i, Q_i, L_i, T_1$,

PM2: N_j, SS_j, V_j, T_2,

PM3: W_j, K_ij, T_3,

PM4: M_1, K_ij, T_4,

Step 2. Idealizing the protocol messages. At this point, the protocol messages are converted into the idealized format based on the BAN-logic notations. The results are denoted by IM1, ..., IM9 as below:

IM1 ($U_i \rightarrow GWN$): $GWN \triangleleft \{K_i\}_{h(MI_i \| X_{GWN})}$,

IM2 ($U_i \rightarrow GWN$): $GWN \triangleleft \{SID_j\}_{h(MI_i \| X_{GWN})}$,

IM3 ($U_i \rightarrow GWN$): $GWN \triangleleft (MI_i, K_i, T_1, SID_j, Q_i)_{h(MI_i \| X_{GWN})}$,

IM4 ($U_i \rightarrow GWN$): $GWN \triangleleft \{h(ID_i)\}_{K_i}$,

IM5 ($GWN \rightarrow S_j$): $S_j \triangleleft (h(ID_i), T_2, K_i)_{h(SID_j \| X_{GWN})}$,

IM6 ($S_j \rightarrow GWN$): $GWN \triangleleft \{K_j\}_{K_i}$,

IM7 ($S_j \rightarrow GWN$): $GWN \triangleleft (SK_j)_{T_3}$,

IM8 ($GWN \rightarrow U_i$): $U_i \triangleleft \{K_j\}_{K_i}$,

IM9 ($GWN \rightarrow U_i$): $U_i \triangleleft (SK_G)_{K_j}$.

Step 3. Explicit assumptions. The seven assumptions on the proposed scheme are described by A1, ..., A7 as below:

A1: $U_i \mid\equiv \sharp(K_i, T_1, T_4)$,

A2: $GWN \mid\equiv \sharp(T_1, T_2, T_3, T_4)$,
A3: $S_j \mid\equiv \sharp(K_j, T_2, T_3)$,
A4: $U_i \mid\equiv U_i \overset{h(MI_j \| X_{GWN})}{\longleftrightarrow} GWN$,
A5: $GWN \mid\equiv GWN \overset{h(MI_j \| X_{GWN})}{\longleftrightarrow} U_i$,
A6: $GWN \mid\equiv GWN \overset{h(SID_j \| X_{GWN})}{\longleftrightarrow} S_j$,
A7: $S_j \mid\equiv S_j \overset{h(SID_j \| X_{GWN})}{\longleftrightarrow} GWN$.

Step 4. Security goals. The nine security goals which are expected to be verified after analyzing the protocol by BAN-logic are listed by G1, ..., G9 as below. For instance, the goal G1 states that the gateway node must believe that the user U_i has sent the key K_i:

G1: $GWN \mid\equiv U_i \mid\sim K_i$,
G2: $GWN \mid\equiv U_i \mid\sim SID_j$,
G3: $GWN \mid\equiv U_i \mid\sim h(ID_i)$,
G4: $S_j \mid\equiv GWN \mid\sim K_i$,
G5: $S_j \mid\equiv GWN \mid\sim h(ID_i)$,
G6: $GWN \mid\equiv S_j \mid\sim K_j$,
G7: $GWN \mid\equiv S_j \mid\sim SK_j$,
G8: $U_i \mid\equiv GWN \mid\sim K_j$,
G9: $U_i \mid\equiv GWN \mid\sim SK_G$.

Step 5. Deriving the security goals. Finally, to show the achievement of the above-mentioned goals, we apply logical rules of the BAN-logic to the idealized messages and initial premises as described below.

In accordance with IM1, A5 and R1:
Result1: $GWN \mid\equiv U_i \mid\sim K_i$ (satisfy **G**1);
Given the IM2, A5 and R1:
Result2: $GWN \mid\equiv U_i \mid\sim SID_j$ (satisfy **G**2);
In accordance with IM4, Result1 and R1:
Result3: $GWN \mid\equiv U_i \mid\sim h(ID)_i$ (satisfy **G**3);
Given the IM5, A7 and R1:
Result4: $S_j \mid\equiv GWN \mid\sim (h(ID_i), T_2, K_i)$;
Taking into account Result4 and R2:
Result5: $S_j \mid\equiv GWN \mid\sim K_i$ (satisfy **G**4);
Result6: $S_j \mid\equiv GWN \mid\sim h(ID_i)$ (satisfy **G**5);
In accordance with IM6, A6 and R1:
Result7: $GWN \mid\equiv S_j \mid\sim K_j$ (satisfy **G**6);
In accordance with IM7, A2 and R1:
Result8: $GWN \mid\equiv S_j \mid\sim SK_j$ (satisfy **G**7);
In accordance with IM8, A1 and R1:
Result9: $U_i \mid\equiv GWN \mid\sim K_j$ (satisfy **G**8);
In accordance with IM9, Result9 and R1:
Result10: $U_i \mid\equiv GWN \mid\sim SK_G$ (satisfy **G**9).

Table 2. BAN-logic notations.

Notation	Description
$P \mid\equiv X$	P believes a proposition X
$P \triangleleft X$	P receives a message X
$P \mid\sim X$	P sent a message X
$P \overset{k}{\rightleftharpoons} X$	P and X share the secret key k and only these two entities can use k to prove its identity to each other.
$\sharp(X)$	It means that X is fresh
$\{X\}_k$	Encryption of X using the secret k
$(X)_k$	Hash computation of X using the secret k
$P \overset{k}{\leftrightarrow} Q$	P and Q share a secret k
$\frac{P}{Q}$	If P then Q

Given the above steps, it can easily be concluded that the protocol can meet all preset goals. Therefore, we can state that our proposed scheme is secure.

8. Performance Comparison

In this work, we propose a new 3FA protocol to overcome the security weaknesses of the Amin et al. [15] scheme. We show how our enhanced protocol is not only secure but also efficient enough to be used in HWSNs. The discussion about the security features, computational overhead and computational cost offered by our proposed scheme and other related schemes, such as Amin et al. [15], Yeh et al. [32], Xue et al. [7], Das et al. [44], Jiang et al. [33], Das et al. [45] and Gope et al. [24] is presented in this section.

8.1. Security Features' Comparison

In Table 3, we sum up the security features offered by our proposed protocol and other similar ones. The symbol "Yes" indicates that the scheme is secure against the related attack and the symbol "No" indicates the contrary. From this, we can conclude that our proposal satisfies all the security features required and offers a higher security level than its predecessors. In addition, protocols [7,24,32,33] do not provide three-factor authentication while our scheme does.

Table 3. Security features' comparison.

Security Features	Amin et al. [15]	Yeh et al. [32]	Xue et al. [7]	Das [44]	Jiang et al. [33]	Das[45]	Gope et al. [24]	Ours
Protection of user untraceability	No	No	No	Yes	Yes	Yes	No	**Yes**
Resistance against replay attack	Yes	No	Yes	Yes	Yes	Yes	Yes	**Yes**
Resistance against user impersonation attack	No	No	No	Yes	No	Yes	Yes	**Yes**
Resistance against gateway node impersonation attack	Yes	No	No	No	No	No	Yes	**Yes**
Resistance against sensor node impersonation attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
Resistance to de-synchronization attack	No	No	No	No	No	No	Yes	**Yes**
Support of dynamic node addition	Yes	No	No	Yes	No	Yes	Yes	**Yes**
Robustness against insider attack	Yes	Yes	No	Yes	No	Yes	Yes	**Yes**
Robustness against stolen smartcard attack	Yes	No	No	Yes	No	Yes	Yes	**Yes**
User anonymity	Yes	No	No	No	Yes	Yes	Yes	**Yes**
Resistance against identity guessing attack	Yes	No	No	Yes	Yes	Yes	Yes	**Yes**
Support of three-factor security	Yes	No	No	Yes	No	Yes	No	**Yes**
Supports correct password update	Yes	No	No	Yes	No	Yes	No	**Yes**
Resistance against session key disclosure attack	No	Yes	Yes	Yes	Yes	Yes	No	**Yes**

8.2. Overall Computational Overhead Comparison

In HWSNs, sensors have limited energy so any authentication protocol designed for these networks should be lightweight and energy efficient. Moreover, we use the model represented in Figure 1a to design our scheme. In our scheme, we use the hash, and the fuzzy extractor functions, which are both efficient. In fact, using the low-power cryptographic functions, rather than a very demanding one, can reduce energy consumption [46]. According to the results of the experiments presented in [24], each modular exponential operation in ECC-160 algorithm consumes 1.2 Ws energy and takes t_{Exp} = 11.69 ms execution time. Moreover, for symmetric key encryption/decryption (128-bit AES-CBC), the running time and energy consumption are approximately t_{sym} = 4.62 ms and 0.72 Ws and for hash function (SHA-256) these two values are approximately t_{Hash} = 1.06 ms and 0.27 Ws, respectively. These results were obtained using the MSB-430 sensor boards with the TI MSP430 micro controller [24]. Moreover, the time that the fuzzy extractor takes t_f is about 17.1 ms [47]. In Table 4, previous works [7,15,24,32,33,44,45] and our proposed scheme are compared in terms of computational cost. As shown in this table, in our proposal, the total computational cost is only $25 \times t_{Hash} + t_f$. Although our proposed scheme consumes slightly more time than some proposals [7,24,33], these extra time is because of the additional operations needed for securing the scheme (improving security pitfalls of its predecessors) and the three-factor capability, which is critical for secure HWSN networks. Finally, it is worth noticing that our results are similar to [15,45], but we offer a higher security level.

Table 4. Overall computational overhead of the authentication phase.

Scheme	User	GW	Sensor Node	Total Cost	Rough Estimation
Amin et al. [15]	$10t_{Hash} + t_f$	$11t_{Hash}$	$4t_{Hash}$	$25t_{Hash} + t_f$	43 ms
Yeh et al. [32]	$2t_{Exp} + t_{Hash}$	$4t_{Exp} + 4t_{Hash}$	$2t_{Exp} + 3t_{Hash}$	$8t_{Hash} + 8t_{Exp}$	100 ms
Xue et al. [7]	$7t_{Hash}$	$10t_{Hash}$	$5t_{Hash}$	$22t_{Hash}$	23 ms
Das [44]	$7t_{Hash} + t_f$	$t_{Sym} + 2t_{Hash}$	$t_{Sym} + 2t_{Hash}$	$11t_{Hash} + 2t_{Sym} + t_f$	38 ms
Jiang et al. [33]	$7t_{Hash}$	$10t_{Hash}$	$5t_{Hash}$	$22t_{Hash}$	23 ms
Das [45]	$9t_{Hash} + t_f$	$11t_{Hash}$	$5t_{Hash}$	$25t_{Hash} + t_f$	43 ms
Gope et al. [24]	$7t_{Hash}$	$9t_{Hash}$	$3t_{Hash}$	$19t_{Hash}$	20 ms
Ours	$10t_{Hash} + t_f$	$11t_{Hash}$	$4t_{Hash}$	$25t_{Hash} + t_f$	43 *ms*

8.3. Computational Cost and Execution Time

To achieve better efficiency and taking into account the energy restrictions of sensor nodes, the computation costs of sensors should be kept as low as possible. In Table 5, we summarize both the computational cost and execution time of our proposal and its predecessors [7,15,32,33,44,45]. From this, it is clear that our proposal is one of the most efficient in terms of energy and execution time. That is, our proposal can be fitted in resource-limited sensor nodes.

Table 5. Computational cost and execution time comparison.

Scheme	Computational Cost	Execution Time
Amin et al. [15]	1.08 Ws	4.24 ms
Yeh et al. [32]	3.21 Ws	26.56 ms
Xue et al. [7]	1.35 Ws	5.3 ms
Das [44]	1.53 Ws	7.8 ms
Jiang et al. [33]	1.35 Ws	5.3 ms
Das [45]	1.35 Ws	5.3 ms
Gope et al. [24]	0.81 Ws	3.18 ms
Ours	1.08 Ws	4.24 ms

9. Conclusions

In heterogeneous wireless sensor networks (HWSNs), we find sensors with different capabilities and functionalities and dispersed within a defined area. Generally, their capabilities, such as computation and energy, are very limited. The security of these devices is pivotal and challenging due to its constrained resources. In this vein, we propose a secure and efficient three-factor authentication (3FA) scheme that is suitable for HWSNs and enhances the security of a recent proposed protocol [15]. Meanwhile, we showed how [15] is not resistant to user impersonation and de-synchronization attacks and also the attacker can track the user by eavesdropping only one session. In addition, an adversary can disclose the session key under the common assumption that the hardware of sensors is not tamper-resistant. To scrutinize the security of our proposal, we informally and formally analyze its security and show how our protocol guarantees all the security features and provides the highest security level in comparison with their predecessors. Moreover, in relation to performance, our scheme consumes only few milliseconds and is very efficient in terms of energy consumption. All of this renders our scheme adequate for HWSNs in which sensors generally have very limited resources. Therefore, as a future work, we aim to propose a new scheme to support user access control that guarantees authorized users to access the information allowed in HWSNs.

Author Contributions: All authors contributed equally to this work in all tasks.

Funding: This work was partially supported by the MINECO grant TIN2016-79095-C2-2-R (SMOG-DEV—Security mechanisms for fog computing: advanced security for devices); and by the CAM grant S2013/ICE-3095 (CIBERDINE: Cybersecurity, Data, and Risks).

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

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

References

1. Jiang, Q.; Ma, J.; Yang, C.; Ma, X.; Shen, J.; Chaudhry, S.A. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Comput. Electr. Eng.* **2017**, *63*, 182–195. [CrossRef]
2. Karl, H.; Willig, A. *Protocols and Architectures for Wireless Sensor Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
3. Yarvis, M.; Kushalnagar, N.; Singh, H.; Rangarajan, A.; Liu, Y.; Singh, S. Exploiting heterogeneity in sensor networks. In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005, Miami, FL, USA, 13–17 March 2005; Volume 2, pp. 878–890.
4. Castiglione, A.; D'Arco, P.; De Santis, A.; Russo, R. Secure group communication schemes for dynamic heterogeneous distributed computing. *Future Gener. Comput. Syst.* **2017**, *74*, 313–324. [CrossRef]
5. Zhong, H.; Shao, L.; Cui, J.; Xu, Y. An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks. *J. Parallel Distrib. Comput.* **2018**, *111*, 1–12. [CrossRef]
6. Farash, M.S.; 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]
7. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323. [CrossRef]
8. Pal, V.; Singh, G.; Yadav, R. Effect of Heterogeneous nodes location on the performance of clustering algorithms for wireless sensor networks. *Procedia Comput. Sci.* **2015**, *57*, 1042–1048. [CrossRef]
9. Castiglione, A.; Palmieri, F.; Fiore, U.; Castiglione, A.; De Santis, A. Modeling energy-efficient secure communications in multi-mode wireless mobile devices. *J. Comput. Syst. Sci.* **2015**, *81*, 1464–1478. [CrossRef]
10. Chen, T.H.; Shih, W.K. A robust mutual authentication protocol for wireless sensor networks. *ETRI J.* **2010**, *32*, 704–712. [CrossRef]
11. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090. [CrossRef]

12. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors* **2010**, *10*, 2450–2459. [CrossRef] [PubMed]
13. Nyang, D.; Lee, M.K. Improvement of Das's Two-Factor Authentication Protocol in Wireless Sensor Networks. *IACR Cryptol. ePrint Arch.* **2009**, *2009*, 631.
14. Sun, D.Z.; Li, J.X.; Feng, Z.Y.; Cao, Z.F.; Xu, G.Q. On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. *Pers. Ubiquitous Comput.* **2013**, *17*, 895–905. [CrossRef]
15. Amin, R.; Islam, S.H.; Kumar, N.; Choo, K.K.R. An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks. *J. Netw. Comput. Appl.* **2017**, *104*, 133–144. [CrossRef]
16. Wong, K.H.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, Taiwan, 5–7 June 2006; Volume 1.
17. Das, A.K.; Sharma, P.; Chatterjee, S.; Sing, J.K. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1646–1656. [CrossRef]
18. Vaidya, B.; Makrakis, D.; Mouftah, H. Two-factor mutual authentication with key agreement in wireless sensor networks. *Secur. Commun. Netw.* **2016**, *9*, 171–183. [CrossRef]
19. He, D.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inf. Sci.* **2015**, *321*, 263–277. [CrossRef]
20. Mir, O.; Munilla, J.; Kumari, S. Efficient anonymous authentication with key agreement protocol for wireless medical sensor networks. *Peer-to-peer Netw. Appl.* **2017**, *10*, 79–91. [CrossRef]
21. He, D.; Kumar, N.; Chen, J.; Lee, C.C.; Chilamkurti, N.; Yeo, S.S. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimed. Syst.* **2015**, *21*, 49–60. [CrossRef]
22. 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]
23. Amin, R.; Biswas, G. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80. [CrossRef]
24. Gope, P.; Hwang, T. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Trans. Ind. Electron.* **2016**, *63*, 7124–7132. [CrossRef]
25. Adavoudi-Jolfaei, A.; Ashouri-Talouki, M.; Aghili, S.F. Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks. *Peer-to-peer Netw. Appl.* **2017**, 1–17. [CrossRef]
26. Amin, R.; Islam, S.H.; Biswas, G.; Khan, M.K.; 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]
27. Arasteh, S.; Aghili, S.F.; Mala, H. A new lightweight authentication and key agreement protocol for Internet of Things. Information Security and Cryptology (ISCISC). In Proceedings of the 2016 13th International Iranian Society of Cryptology Conference, Tehran, Iran, 7–8 September 2016; pp. 52–59.
28. 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]
29. Jiang, Q.; Chen, Z.; Li, B.; Shen, J.; Yang, L.; Ma, J. Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. *J. Ambient Intell. Humaniz. Comput.* **2018**, *9*, 1061–1073. [CrossRef]
30. Lu, Y.; Li, L.; Peng, H.; Yang, Y. An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. *J. Med. Syst.* **2015**, *39*, 32. [CrossRef] [PubMed]
31. Chang, C.C.; Le, H.D. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 357–366. [CrossRef]
32. Yeh, H.L.; Chen, T.H.; Liu, P.C.; Kim, T.H.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779. [CrossRef] [PubMed]
33. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-peer Netw. Appl.* **2015**, *8*, 1070–1081. [CrossRef]
34. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]

35. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
36. Odelu, V.; Das, A.K.; Goswami, A. A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1953–1966. [CrossRef]
37. Ouafi, K.; Phan, R.C.W. Privacy of recent rfid authentication protocols. In *International Conference on Information Security Practice and Experience*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 263–277.
38. Shin, S.; Kwon, T. Two-Factor Authenticated Key Agreement Supporting Unlinkability in 5G-Integrated Wireless Sensor Networks. *IEEE Access* **2018**, *6*, 11229–11241. [CrossRef]
39. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* **2017**, *89*, 72–85. [CrossRef]
40. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Sangaiah, A.K.; Choo, K.K.R. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *J. Netw. Comput. Appl.* **2018**, *103*, 194–204. [CrossRef]
41. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
42. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuéllar, J.; Drielsma, P.H.; Héam, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *International Conference on Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285.
43. Blanchet, B. Automatic verification of security protocols in the symbolic model: The verifier proverif. In *Foundations of Security Analysis and Design VII*; Springer: New York, NY, USA, 2014; pp. 54–87.
44. Das, A.K. A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wirel. Pers. Commun.* **2015**, *82*, 1377–1404. [CrossRef]
45. Das, A.K. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-to-peer Netw. Appl.* **2016**, *9*, 223–244. [CrossRef]
46. Hellaoui, H.; Koudil, M.; Bouabdallah, A. Energy-efficient mechanisms in security of the internet of things: A survey. *Comput. Netw.* **2017**, *127*, 173–189. [CrossRef]
47. He, D.; Kumar, N.; Lee, J.H.; Sherratt, R. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electron.* **2014**, *60*, 30–37.