sensors

MDPI

Article

A Lightweight Three-Factor Authentication and Key Agreement Scheme in Wireless Sensor Networks for Smart Homes

Sooyeon Shin and Taekyoung Kwon *

Graduate School of Information, Yonsei University, Seoul 03722, Korea; shinsy80@yonsei.ac.kr

* Correspondence: taekyoung@yonsei.ac.kr; Tel.: +82-2-2123-4523

Received: 29 March 2019; Accepted: 24 April 2019; Published: 29 April 2019

Abstract: A wireless sensor network (WSN) is used for a smart home system's backbone that monitors home environment and controls smart home devices to manage lighting, heating, security and surveillance. However, despite its convenience and potential benefits, there are concerns about various security threats that may infringe on privacy and threaten our home life. For protecting WSNs for smart homes from those threats, authentication and key agreement are basic security requirements. There have been a large number of proposed authentication and key agreement scheme for WSNs. In 2017, Jung et al. proposed an efficient and security enhanced anonymous authentication with key agreement scheme by employing biometrics information as the third authentication factor. They claimed that their scheme resists on various security attacks and satisfies basic security requirements. However, we have discovered that Jung et al.'s scheme possesses some security weaknesses. Their scheme cannot guarantee security of the secret key of gateway node and security of session key and protection against user tracking attack, information leakage attack, and user impersonation attack. In this paper, we describe how those security weaknesses occur and propose a lightweight three-factor authentication and key agreement scheme in WSNs for smart homes, as an improved version of Jung et al.'s scheme. We then present a detailed analysis of the security and performance of the proposed scheme and compare the analysis results with other related schemes.

Keywords: three-factor authentication; key agreement; password; smart card; biometrics; anonymity; untraceability; wireless sensor networks; Internet of Things; smart home

1. Introduction

Wireless sensor networks (WSNs), composed of many low-cost and low-power sensor nodes, have become a popular technology for various applications including Internet of Things (IoT) applications such as health-care, smart homes, smart factoring, and smart city [1]. For example, a smart home is defined as a networking technology to integrate devices and appliances so that many smart home devices with sensors monitor home environments, capture users' offline activities, and control lighting, windows, doors, heating, security and surveillance, and so on. In the smart home application, internal and external users need to directly access the WSN for real-time control and data acquisition will increase through direct access. According to Zion Market Research [2], the global smart home market was valued at USD 39.68 billion in 2017 and is expected to reach a value of USD 159.68 billion by 2023. The growing market for smart home provides a more comfortable and easier way of life to users while presenting new challenges for preserving privacy. Moreover, due to the inherent characteristics of WSNs, such as resource constraints and the use of wireless medium, they are likely to be exposed to various attacks. In such situations, cryptographic techniques such as encryption and message authentication should be applied to protect user privacy and WSN against various attacks. To apply cryptographic techniques, user authentication and key agreement are basically required.

1.1. Related Works

To improve security of WSNs, many user authentication and key agreement schemes have been proposed in the last decade [3–14]. In 2006, based on lightweight operations, such as XOR operations and one-way hash function, Wong et al. [3] proposed a lightweight strong-password authentication scheme for WSNs. However, Das [4] pointed out that Wong et al.'s scheme is vulnerable to same login identity attack, replay attack and stolen-verifier attack. Das then presented a two-factor authenticated key establishment scheme for WSNs as an improved version of Wong et al.'s scheme. Unfortunately, many papers [5,15–18] have revealed that Das's scheme is vulnerable to various attacks such as privileged insider, gateway node bypass, smart card loss, and parallel session attacks.

Although many improved versions of Das's scheme have been proposed to solve the above-mentioned security flaws, they still have some security problems. As one of the improved versions, Vaidya et al. [5] proposed a novel two-factor user authentication scheme with key agreement for WSNs, but in 2014, Kim et al. [6] pointed out that Vaidya et al.'s scheme could not withstand both user impersonation attack and gateway node bypass attack. Kim et al. then proposed a user authentication and key agreement scheme that resisted those attacks. In 2015, Chang et al. [8] found that Kim et al.'s scheme is vulnerable to impersonation, lost smart card, man-in-the-middle attacks and does not provide session key security and user privacy. Chang et al. then presented an enhanced two-factor authentication and key agreement using dynamic identities. However, recently, Park et al. [9] and Jung et al. [13] pointed out that Chang et al.'s scheme has security flaws such as off-line password guessing attack, user impersonation attack, perfect forward secrecy problem, and incorrectness of password change. Park et al. and Jung et al. then proposed improved schemes in 2016 and 2017, respectively. Park et al. proposed a three-factor user authentication and key agreement scheme using ECC (Elliptic Curve Cryptosystem) and Jung et al. proposed an efficient anonymous authentication with key agreement scheme using only lightweight operations. However, we found that Jung et al.'s scheme still has some security weaknesses [19].

On the other hand, user authentication and key agreement schemes based on the concept of IoT have been proposed. In 2014, Turkanović et al. [7] proposed an energy-efficient user authentication scheme with high security and low computational cost using the concept of the IoT. However, Farash et al. [10] found that Turkanović et al.'s scheme has security weaknesses and then proposed an improved scheme. In 2016, Amin et al. [11] claimed that Farash et al.'s scheme has some security problems such as known session-specific temporary information attack, off-line password guessing attack using a stolen-smart card, a new-smart card-issue attack, user impersonation attack, insecurity of the secret key of the gateway node, and insecurity of user anonymity. Amin et al. then proposed an anonymity-preserving three-factor authenticated key exchange scheme for IoT-based WSNs. Unfortunately, recently, Jiang et al. [12] found several security flaws in Amin et al.'s scheme, such as smart card loss attack, known session-specific temporary information attack, and tracking attack. Jiang et al. then proposed a lightweight three-factor authentication and key agreement scheme for Internet-integrated WSNs based on Rabin cryptosystem. Jiang et al.'s scheme provides various security features but this scheme is difficult to implement and deploy in practical applications because of heavyweight decryption of Rabin cryptosystem.

1.2. Research Contributions

As shown in the section on related works, most of the proposed user authentication and key agreement schemes for WSNs fail to provide adequate security protection or still suffer from various security attacks. To overcome these weaknesses, we design a lightweight authentication and key agreement scheme. Our research contributions are as follows.

- We analyze the most recent three-factor authentication and key agreement scheme of Jung et al.'s scheme and present its security weaknesses. We show that Jung et al.'s scheme [13] does not provide strong anonymity and the secrecy of the secret key of the gateway node. We also show

that Jung et al.'s scheme is vulnerable to a tracing attack, information leakage attack, session key recovery attack, and user impersonation attack.
- We introduce a system model suitable for smart homes based on WSNs. Under this model, we propose a lightweight three-factor authentication and key agreement scheme as an improved version of Jung et al.'s scheme. The proposed scheme not only satisfies various security requirements but also uses lightweight operations, such as XOR and hash functions, which are very suitable for the resource constrained WSNs.
- We formally prove the security of the proposed scheme using both random oracle model and BAN (Burrows-Abadi-Needham) logic. We then verify the proposed scheme on popular and robust security verification tool, AVISPA (Automated Validation of Internet Security Protocols and Applications).
- Through informal security analysis, we show that the proposed scheme can satisfy the required security properties and withstand various attacks. We then compare it with other related schemes in terms of security features.
- Through a performance evaluation, we compare the performance of the proposed scheme with other related schemes in terms of their computational cost and communication cost.

The remaining parts of this paper are as follows: Section 2 briefly reviews Jung et al.'s scheme; Section 3 demonstrates the security weaknesses of Jung et al.'s scheme; the details of the proposed scheme is illustrated in Section 4; Sections 5 and 6 give the formal and informal security analysis of the proposed scheme, respectively; Section 7 shows the performance evaluation of the proposed scheme; Section 8 concludes the paper.

1.3. Preliminary

A conventional hash function may return different outputs even if there is little variation in the inputs because its output is sensitive [20]. On the other hand, since biometric information is prone to various noises during data acquisition, it is difficult to re-product actual biometric in common practice. A fuzzy extractor method has been used to solve these problems [20–23]. The fuzzy extractor can extract a uniformly-random string and a public information from the biometric template with a given error tolerance t. In other words, even if the input changes slightly, the fuzzy extractor could output the same random string with the help of the public information. The fuzzy extractor consists of the following two algorithms.

- $GEN(Bio_i) = (b_i, par_i)$: Given a biometric template Bio_i as an input, this probabilistic algorithm outputs a secret biometric key b_i and a helper string par_i.
- $REP(Bio_i', par_i) = (b_i)$: Given a noisy biometric Bio_i' and a helper string par_i as inputs, this deterministic algorithm reproduces the biometric key b_i.

2. Review of Jung et al.'s Scheme

In this section, we briefly review Jung et al.'s anonymous authentication with key agreement scheme in WSNs [13]. Jung et al.'s scheme consists of four phases: user registration, login, authentication, and password change. We describe the first three phases related to the security weaknesses in detail. Table 1 shows the notations used in Jung et al.'s scheme.

Before a sensor node S_j is deployed, it keeps SID_j and $X_{S_j}^*$ in its storage, where $X_{S_j}^* = h(SID_j||K)$.

Table 1. Notations for Jung et al.'s scheme.

Notation	Description	Notation	Description
U_i	Remote user	u	Random number of U_i
S_j	Sensor node	R	Random number
GWN	Gateway node	K	Secret key generated by the GWN
ID_i, PW_i	Identity and password of U_i	K_S	Session key
Bio_i	Biometric information of U_i	$f(v,k)$	Pseudo-random function of variable v with key k
TID_i	Temporary identity of U_i's next login	$h(\cdot), H(\cdot)$	One-way hash function and biohash function
SID_j	Identity of S_j	$T, \Delta T$	Timestamp and the transmission delay time

2.1. Registration Phase

In the registration phase, U_i sends a request message for registration to GWN then GWN issues a smart card for U_i. All messages in this phase are transmitted through a secure channel.

(1) U_i chooses ID_i, PW_i, and a random number u and imprints his/her biometrics Bio_i. U_i computes $HPW_i = h(PW_i||H(Bio_i))$ and $TID_i = h(ID_i||u)$ and sends a registration request $\langle TID_i, HPW_i \rangle$ to GWN.
(2) Upon receiving the registration request, GWN computes $HID_i = h(TID_i||K) \oplus HPW_i, A_i = h(HPW_i||TID_i) \oplus HID_i, B_i = h(HPW_i||HID_i)$, and $C_i = HID_i \oplus K$. GWN then issues a smart card by storing $(A_i, B_i, C_i, h(\cdot), H(\cdot))$ in its memory and sends the smart card to U_i.
(3) Upon receiving the smart card, U_i computes $D_i = u \oplus H(Bio_i)$ and additionally stores it into the smart card.

2.2. Login Phase

In the login phase, U_i sends the service request to GWN using his/her smart card, identity, password, and biometric information.

(1) U_i inserts own smart card into a terminal, enters ID_i and PW_i, and imprints Bio_i.
(2) The smart card computes $HPW_i^* = h(PW_i||H(Bio_i))$, $u = D_i \oplus H(Bio_i), TID_i = h(ID_i||u), HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$, and $B_i^* = h(HPW_i^*||HID_i^*)$. The smart card then checks whether B_i^* matches with the received B_i. If it does not hold, the smart card terminates this phase. Otherwise, the smart card confirms the legitimacy of U_i and computes $DID_i = TID_i \oplus HID_i^*$ and $M_{U_i,G} = h(TID_i||HPW_i^*||HID_i^*||T_1)$.
(3) The smart card sends the login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$ to GWN through a public channel.

2.3. Authentication Phase

The authentication phase begins when GWN receives the login request from U_i. In this phase, U_i, GWN, and S_j authenticate each other and establish a session key K_S.

(1) GWN checks the validity of T_1 and computes $TID_i^* = DID_i \oplus C_i \oplus K, HID_i = C_i \oplus K, HPW_i^* = HID_i \oplus h(TID_i||K)$ and $M_{U_i,G}^* = h(TID_i^*||HPW_i^*||HID_i||T_1)$. GWN then checks whether $M_{U_i,G}^*$ matches with the received $M_{U_i,G}$. If it does not hold, it terminates this phase. Otherwise, GWN believes that U_i is authentic and proceeds with the next step.
(2) GWN chooses a random number R and computes $X_{S_j} = h(SID_j||K), M_j = R \oplus X_{S_j}, K_S = f(DID_i, R)$ and $M_{G,S_j} = h(DID_i||SID_j||X_{S_j}||K_S||\ T_2)$. GWN then sends the message $\langle DID_i, M_{G,S_j}, M_j, T_2 \rangle$ to S_j through a public channel.
(3) Upon receiving the message, S_j checks the validity of T_2 and computes $R^* = M_j \oplus X_{S_j}^*, K_S^* = f(DID_i, R^*)$, and $M_{G,S_j}^* = h(DID_i||SID_j||X_{S_j}^*||K_S^*||T_2)$. S_j checks whether M_{G,S_j}^* matches with the received M_{G,S_j}. If it does not hold, S_j terminates this phase. Otherwise, S_j believes the GWN is authentic.

(4) S_j computes $k_j = h(X^*_{S_j}||T_3)$ and $M_{S_j,G} = h(k_j||X^*_{S_j}||K^*_S||T_3)$. S_j finally sends the message $\langle M_{S_j,G}, T_3\rangle$ to GWN through a public channel.
(5) Upon receiving the message, GWN checks the validity of T_3 and computes $k_j = h(X_{S_j}||T_3)$ and $M^*_{S_j,G} = h(k_j||X_{S_j}||K_S||T_3)$. GWN then checks whether $M^*_{S_j,G}$ matches with the received $M_{S_j,G}$. If it does not hold, GWN terminates this phase. Otherwise, GWN believes that S_j is authentic and proceeds with the next step.
(6) GWN computes $k_i = R \oplus h(TID^*_i||K)$ and $M_{G,U_i} = h(K_S||k_i||T_4)$ and sends the message $\langle k_i, M_{G,U_i}, T_4\rangle$ to U_i through a public channel.
(7) Upon receiving the message, U_i checks the validity of T_4 and computes $R^* = k_i \oplus HPW_i \oplus HID^*_i, K^*_S = f(DID_i, R^*)$, and $M^*_{G,U_i} = h(K^*_S, k_i, T_4)$. U_i then checks whether M^*_{G,U_i} matches with the received M_{G,U_i}. If it does not hold, this phase is terminated. Otherwise, U_i believes that GWN is authentic and successfully ends the authentication phase.

3. Security Weaknesses of Jung et al.'s Scheme

In this section, we show that Jung et al.'s scheme [13] has security weaknesses.

3.1. Tracing Attack

As the concern for privacy increases in our lives, user anonymity has become a vital security requirement in various applications including WSN applications. For example, the personalized services in smart home applications (e.g., home energy management system) provide users with better convenience, but breach of privacy has been a serious concern [24]. In general, the preservation of identity privacy in the context of an authentication protocol requires not only anonymity but also untraceability [25]. Although untraceability is not a necessary condition of anonymity, strong anonymity with untraceability is required for fully protecting user privacy. In Jung et al.'s scheme, every time U_i uses the fixed values DID_i and C_i to login the WSN thus anyone can trace U_i according to these strings constantly. Therefore, Jung et al.'s scheme is prone to user tracing attack and fails to provide untraceability.

3.2. Insecurity of the Secret Key of the Gateway Node

In Jung et al.'s scheme, the secret key K of GWN is used to compute critical parameters of users' smart cards and secret keys of all sensor nodes. The security of Jung et al.'s scheme thus depends on the security of the secret key K. Unfortunately, any authorized user can easily extract K using his/her identity, password, biometrics and values stored in the smart card. Assume that an authorized user U_i retrieves the information $\langle A_i, B_i, C_i, D_i\rangle$ from his/her smart card, where $A_i = h(HPW_i||TID_i) \oplus HID_i$, $B_i = h(HPW_i||HID_i)$, $C_i = HID_i \oplus K$, and $D_i = u \oplus H(Bio_i)$. As the smart card calculates at the login phase, U_i then computes $u = D_i \oplus H(Bio_i)$, $TID'_i = h(ID_i||u)$, and $HID'_i = A_i \oplus h(HPW_i||TID'_i)$. Based on HID'_i and C_i, U_i computes K', where $K' = C_i \oplus HID'_i$. Since he or she now knows the secret key K', U_i can impersonate GWN and launch the following attacks.

3.3. Information Leakage Attack

We described how an authorized user U_j can know K' in Section 3.2. After getting K', U_j who acts as an adversary $\mathcal{A}$ can achieve secret information required for authentication and key agreement as follows:

(1) $\mathcal{A}$ intercepts the user U_i's login message $\langle DID_i, M_{U_i,G}, C_i, T_1\rangle$, where $DID_i = TID_i \oplus HID_i$, $M_{U_i,G} = h(TID_i||HPW_i||HID_i||T_1)$, and $C_i = HID_i \oplus K$.
(2) $\mathcal{A}$ computes $HID'_i = C_i \oplus K'$ and $TID'_i = DID_i \oplus HID'_i$.
(3) $\mathcal{A}$ then computes $HPW'_i = HID'_i \oplus h(TID'_i||K')$.

Thus, $\mathcal{A}$ can obtain all secret values TID_i', HID_i', and HPW_i' need to login the WSN and launch session key recovery attack and user impersonation attack.

3.4. Session Key Compromise

We assume that $\mathcal{A}$ can obtain the secret information by intercepting the U_i's login message and also can intercept the last message of the authentication phase. After getting the secret information in Section 3.3 and k_i, $\mathcal{A}$ can successfully launch a session key recovery attack as follows:

(1) $\mathcal{A}$ intercepts the last message $\langle k_i, M_{G,U_i}, T_4 \rangle$ sent from GWN, where $k_i = R \oplus h(TID_i||K')$ and $M_{G,U_i} = h(K_S||k_i||T_4)$.
(2) $\mathcal{A}$ computes $R' = k_i \oplus h(TID_i'||K)$.
(3) $\mathcal{A}$ discovers the session key K_S' between the user U_i, GWN, and the sensor node S_j by computing $K_S' = f(DID_i||R')$.

Thus, according to the above procedure, an adversary can successfully construct the session key K_S' between U_i, GWN, and S_j.

3.5. User Impersonation Attack

Once an adversary $\mathcal{A}$ achieves the GWN's secret key K and secret information TID_i', HID_i', and HPW_i' as described in Sections 3.2 and 3.3, respectively, $\mathcal{A}$ can also impersonate a user U_i in Jung et al.'s scheme without the target user's identity ID_i, password PW_i, and biometric information Bio_i as follows:

(1) $\mathcal{A}$ computes $DID_i' = TID_i' \oplus HID_i', C_i' = HID_i' \oplus K'$ and $M_{U_i,G}' = h(TID_i'||HPW_i'||HID_i'||T_1')$, where T_1' is the current time stamp used by $\mathcal{A}$. Of course, since DID_i and C_i are the fixed values, it is possible to use the previously intercepted one.
(2) $\mathcal{A}$ sends the login message $\langle DID_i', M_{U_i,G}', C_i', T_1' \rangle$.
(3) At GWN, user authentication is successfully performed and $\mathcal{A}$ calculates the session key K_S' after receiving the last message as described in Section 3.4.

It is clear from the above discussion that $\mathcal{A}$ can masquerade as a valid user U_i to login to the WSN without ID_i, PW_i and Bio_i. Thus, Jung et al.'s scheme is vulnerable to the user impersonation attack.

4. Proposed Scheme for Smart Homes

In this section, we propose a three-factor authentication and key agreement scheme in WSNs for smart homes in which we find the aforementioned security weaknesses found in Jung et al.'s scheme. Figure 1 illustrates a system model of WSNs for a smart home monitoring and control system. The system model includes three types of entities: a user (U_i), a home gateway node (HG), and sensor nodes (S_j). After registration and mutual authentication with the help of HG, U_i can access the WSN to monitor and control smart home.

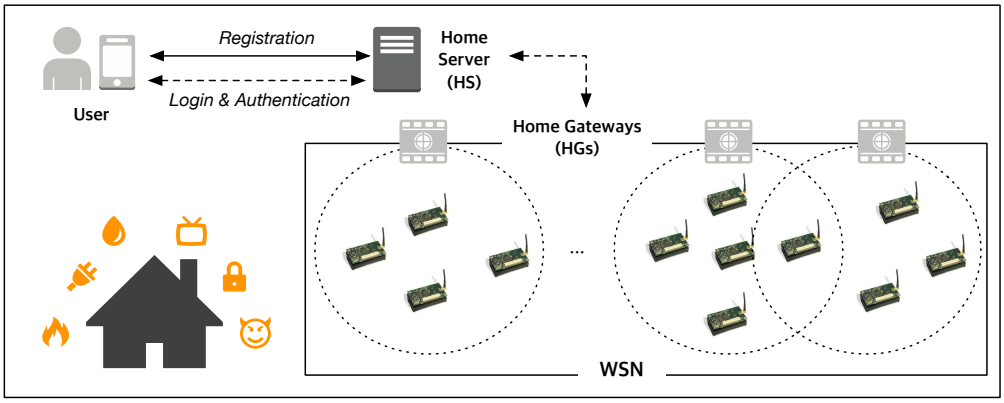

Figure 1. An example of smart home monitoring and control system based on WSNs.

The proposed scheme consists of five phases: system setup, user registration, login, authentication, and password change. We use the additional notation for the proposed scheme listed in Table 2.

Table 2. Notations used for the proposed scheme.

Notation	Description	Notation	Description
HG	Home Gateway	SC_i	Smart card for U_i
TID_i	Temporary identity of U_i	SK	Session key
PID_i^l	One-time pseudonym of U_i for the l-th login	$GEN(\cdot)$	Fuzzy generator function
K_U	Secret key generated by the HG for users	$REP(\cdot)$	Fuzzy reproduction function
K_S	Secret key generated by the HG for sensor nodes		

4.1. System Setup Phase

This phase is executed by home gateway (HG) in an off-line mode before deployment of sensor nodes in a target field.

(1) HG generates randomly two master secrets K_U and K_S for all users and all sensor nodes, respectively, which are only known to HG.
(2) HG selects a unique identity SID_j and computes $X_{S_j} = h(SID_j||K_S)$ for each sensor node S_j.
(3) Finally, each sensor node is deployed in the target field after storing SID_j and X_{S_j} into its memory in a secure manner.

4.2. User Registration Phase

The user registration phase begins when a user U_i sends a request message for registration to HG over a secure channel. Figure 2 illustrates the user registration phase for the proposed scheme. This phase is described below.

(1) U_i selects the desired identity ID_i and password PW_i and imprints his/her biometrics Bio_i. U_i generates a random secret number u_i and computes $(b_i, par_i) = GEN(Bio_i)$, $HPW_i = h(PW_i||b_i)$ and $TID_i = h(ID_i||u_i)$. U_i then sends a registration request $\langle TID_i, HPW_i \rangle$ to HG over a secure channel.
(2) Upon receiving the user's registration request, HG randomly selects a unique one-time pseudonym PID_i^1 for U_i. HG computes $HID_i = h(TID_i||K_U)$, $A_i = h(HPW_i||TID_i) \oplus HID_i$, $B_i = h(HPW_i||HID_i)$, and $C_i^1 = h(TID_i||HID_i) \oplus PID_i^1$. HG issues a smart card SC_i for U_i after saving $\{A_i, B_i, C_i^1, h(\cdot)\}$ in it. HG then sends SC_i to U_i over a secure channel and stores $\{PID_i^1, TID_i\}$ into its memory.
(3) After receiving the smart card SC_i, U_i computes $D_i = u \oplus h(ID_i||b_i)$ and saves D_i, par_i, $GEN(\cdot)$, and $REP(\cdot)$ in SC_i. Finally, SC_i contains $\{A_i, B_i, C_i^1, D_i, par_i, h(\cdot), GEN(\cdot), REP(\cdot)\}$.

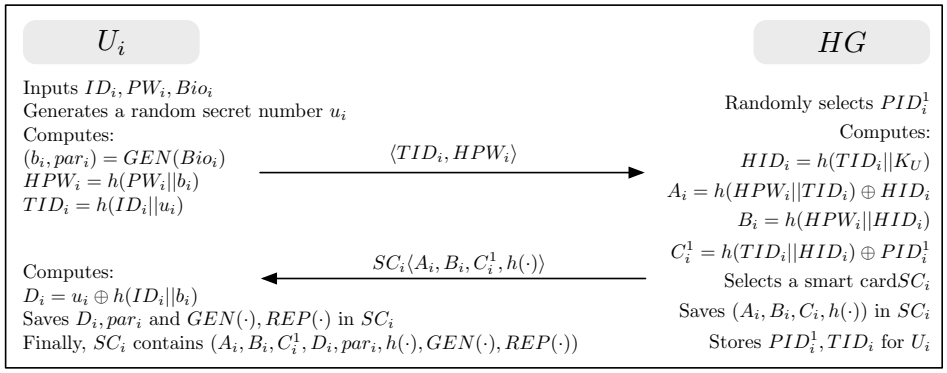

Figure 2. User registration phase for the proposed scheme.

4.3. Login Phase

The login phase is executed when U_i wants to gain access to the WSN using his/her SC_i, ID_i, PW_i, and Bio_i. Figure 3 illustrates the login and authentication phases for the proposed scheme. This phase contains the following steps.

(1) U_i inserts own SC_i, inputs his/her ID_i and PW_i, and imprints his/her biometrics Bio_i into a terminal (i.e., a smart card reader or a smartphone embedded with SC_i).

(2) SC_i computes $b_i = REP(Bio_i, \beta_i)$, $u_i = D_i \oplus h(ID_i||b_i)$, $TID_i = h(ID_i||u_i)$, $HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$, and $B_i^* = h(HPW_i^*||HID_i^*)$. SC_i checks whether B_i^* matches with the stored B_i. If it matches SC_i ensures that U_i has provided correct ID_i, PW_i, and Bio_i. SC_i then selects a random number r_i and computes $PID_i^1 = C_i^1 \oplus h(TID_i||HID_i^*)$, $R_i = h(TID_i||PID_i^1||r_i)$, $M_i = r_i \oplus h(TID_i||HID_i^*||T_1)$ and $M_{U_i,G} = h(TID_i||HID_i^*||\ PID_i^1||R_i||T_1)$.

(3) Finally, U_i sends a login request $\langle PID_i^1, M_i, M_{U_i,G}, T_1 \rangle$ to HG over a public channel.

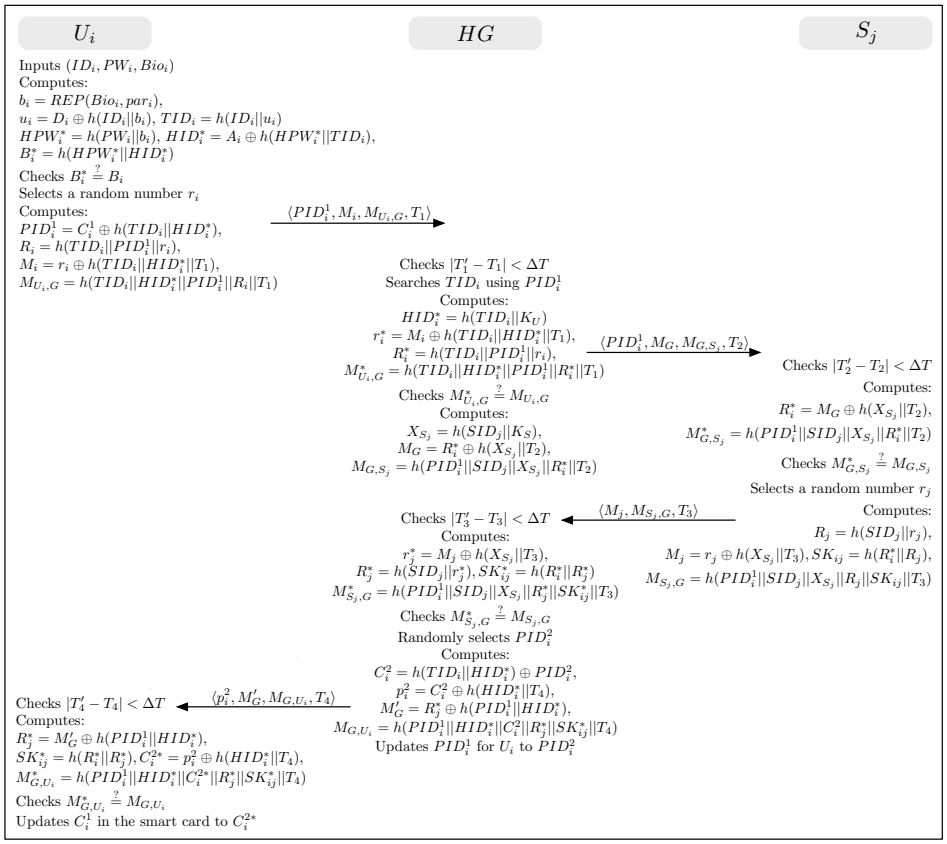

Figure 3. Login and authentication phases for the proposed scheme.

4.4. Authentication Phase

The authentication phase begins when HG receives the login request from U_i. For achieving mutual authentication and session key agreement, this phase executes in several steps as followings.

(1) HG checks the validity of the timestamp $|T_1' - T_1| < \Delta T$ and searches TID_i using PID_i^1. HG computes $HID_i^* = h(TID_i||K_U)$, $r_i^* = M_i \oplus h(TID_i||HID_i^*||T_1)$, $R_i^* = h(TID_i||PID_i^1||r_i^*)$, and $M_{U_i,G}^* = h(TID_i||HID_i^*||PID_i^1||R_i^*||T_1)$. Then, HG compares $M_{U_i,G}^*$ with the received value $M_{U_i,G}$. If this condition is not satisfied, HG terminates this phase. Otherwise, HG believes that U_i is a legitimate user. HG then chooses an appropriate sensor node S_j for the user's needs and computes $X_{S_j} = h(SID_j||K_S)$, $M_G = R_i^* \oplus h(X_{S_j}||T_2)$, and $M_{G,S_j} = h(PID_i^1||SID_j||X_{S_j}||R_i^*||T_2)$. HG sends the message $\langle PID_i^1, M_G, M_{G,S_j}, T_2\rangle$ to S_j over a public channel.

(2) Upon receiving the message from HG, S_j checks the validity of the timestamp $|T_2' - T_2| < \Delta T$ and compute $R_i^* = M_G \oplus h(X_{S_j}||T_2)$ and $M_{G,S_j} = h(PID_i^1||SID_j||X_{S_j}||R_i^*||T_2)$. S_j then compares M_{G,S_j}^* with the received value M_{G,S_j}. If this condition is not satisfied, S_j terminates this phase since HG fails to prove to be a legitimate home gateway. Otherwise, S_j believes that HG is authentic. S_j then selects a random number r_j and computes $R_j = h(SID_j||r_j)$, $M_j = r_j \oplus h(X_{S_j}||T_3)$, $SK_{ij} = h(R_i^*||R_j)$, and $M_{S_j,M} = h(PID_i^1||SID_j||X_{S_j}||R_j||SK_{ij}||T_3)$. S_j sends the message $\langle M_j, M_{S_j,G}, T_3\rangle$ to HG over a public channel.

(3) Upon receiving the message from S_j, HG checks the validity of the timestamp $|T_3' - T_3| < \Delta T$ and computes $r_j^* = M_j \oplus h(X_{S_j}||T_3)$, $R_j^* = h(SID_j||r_j^*)$, $SK_{ij}^* = h(R_i^*||R_j^*)$, and $M_{S_j,G}^* = h(PID_i^1||SID_j||X_{S_j}||R_j^*||SK_{ij}^*||T_3)$. HG compares $M_{S_j,G}^*$ with the received value $M_{S_j,G}$. If this condition is not satisfied, HG terminates this phase. Otherwise, HG believes that S_j is a legitimate sensor node. HG then randomly selects another unique one-time pseudonym PID_i^2 for U_i's next login session and computes $C_i^2 = h(TID_i||HID_i^*) \oplus PID_i^2$, $p_i^2 = C_i^2 \oplus h(HID_i^*||T_4)$, $M_G' = R_j^* \oplus h(PID_i^1||HID_i^*)$, and $M_{G,U_i} = h(PID_i^1||HID_i^*||C_i^2||R_j^*||SK_{ij}^*||T_4)$. Finally, HG sends the message $\langle p_i^2, M_G', M_{G,U_i}, T_4\rangle$ to U_i over a public channel and updates PID_i^1 stored in its memory to PID_i^2 for U_i.

(4) Upon receiving the message from HG, U_i checks the validity of the timestamp $|T_4' - T_4| < \Delta T$ and computes $R_j^* = M_G' \oplus h(PID_i^1||HID_i^*)$, $SK_{ij}^* = h(R_i||R_j^*)$, $C_i^2 = p_i^2 \oplus h(HID_i^*||T_4)$, and $M_{G,U_i}^* = h(PID_i^1||HID_i^*||C_i^2||R_j^*||SK_{ij}^*||T_4)$. U_i then compares M_{G,U_i}^* with the received value M_{G,U_i}. If this condition is not verified, U_i terminates this phase since HG fails to prove to be a legitimate home gateway. Otherwise, U_i believes that HG is authentic and updates C_i^1 in SC_i to C_i^2 for the next session.

4.5. Password Change Phase

The password change phase begins when U_i wants to change the original password PW_i to a new password PW_i^{new}. Figure 4 illustrates this phase for the proposed scheme. This phase contains the following steps.

(1) U_i inserts own SC_i, inputs his/her ID_i, PW_i, and a new password PW_i^{new} and imprints his/her biometrics Bio_i into a terminal.

(2) SC_i computes $b_i = REP(Bio_i, par_i)$, $u_i = D_i \oplus H(Bio_i)$, $TID_i = h(ID_i||u_i)$, $HPW_i^* = h(PW_i||b_i)$, $HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$, and $B_i^* = h(HPW_i^*||HID_i^*)$. SC_i then compares B_i^* with the stored B_i. If this condition is not satisfied, SC_i terminates this phase. Otherwise, SC_i performs the next step.

(3) SC_i computes $HPW_i^{new} = h(PW_i^{new}||H(Bio_i))$, $A_i^{new} = h(HPW_i^{new}||TID_i) \oplus HID_i^*$, and $B_i^{new} = h(HPW_i^{new}||HID_i^*)$. SC_i replaces the stored values A_i and B_i with the newly computed values A_i^{new} and B_i^{new}, respectively. Finally, SC_i contains $\{A_i^{new}, B_i^{new}, C_i^{\ell}, D_i, h(\cdot), H(\cdot)\}$, where ℓ is the index of the next login.

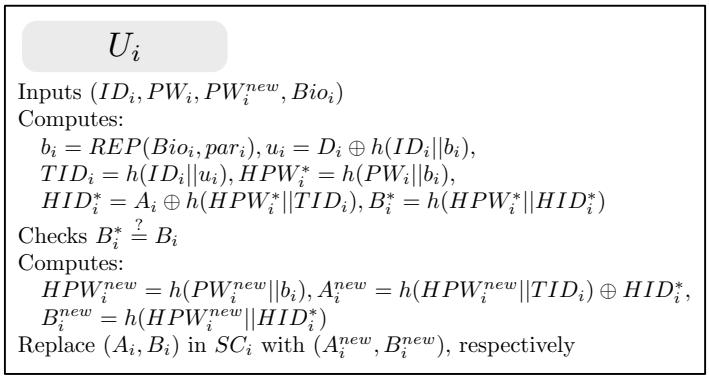

Figure 4. Password change phase for the proposed scheme.

5. Formal Security Analysis of the Proposed Scheme

In this section, we formally analyze the security of the proposed scheme in three ways. First of all, in Section 5.1, we conduct a formal security proof in the random oracle model since the proposed scheme heavily depends on the security of a one-way hash function. Through the rigorous formal poof using the random oracle, we show that the proposed scheme is probabilistically secure against an adversary both to protect the long-term secret information of the user and home gateway and to protect the session key shared between the user and sensor node. In Section 5.2, we then perform the logical verification using BAN logic [26] to confirm the correctness that the authenticated participants share the session key securely in the proposed scheme. In Section 5.3, we automatically validate the proposed scheme using AVISPA tool [27,28] to ensure that the proposed scheme is secure against active and passive attacks (i.e., replay and man-in-the-middle attacks) defined in the simulation tool.

5.1. Security Proof Using Random Oracle Model

Through a formal proof using the random oracle model, we show that the proposed scheme is secure against an adversary. We follow the formal security proof of the proposed scheme similar to that in [13,22] and consider the method of contradiction proof. Based on the random oracle model, the following Theorems 1 and 2 show that the proposed scheme can resist various security attacks. For this purpose, we assume that there exists the following random oracle as illustrated in Definition 1.

Definition 1. *Reveal: Given a hash value y=h(x), this random oracle unconditionally outputs the input x.*

Theorem 1. *Under the assumption that a one-way hash function h(·) behaves like an oracle, the proposed scheme is probably secure against an adversary $\mathcal{A}$ for deriving the identity ID_i, the password PW_i, the biometric key b_i of a legal user U_i and the secret key K_U of the HG, even if the user U_i's smart card SC_i is lost/stolen.*

Proof of Theorem 1. For the proof, we assume that an adversary $\mathcal{A}$ is able to derive the identity ID_i, the password PW_i, and the biometric key b_i of a legal user U_i, and the secret key K_U of the HG. We also assume that the adversary $\mathcal{A}$ has the lost/stolen smart card SC_i of the user U_i and $\mathcal{A}$ can extract all the sensitive information stored in the SC_i using the power analysis attack [29–31]. For this, $\mathcal{A}$ uses the *Reveal* oracle to run an experimental algorithm $EXP1^{3FAKA}_{HASH,\mathcal{A}}$ shown in Algorithm 1 for the proposed three-factor authentication and key agreement ($3FAKA$). We define the success probability for $EXP1^{3FAKA}_{HASH,\mathcal{A}}$ as $Succ1^{3FAKA}_{HASH,\mathcal{A}} = |Pr[EXP1^{3FAKA}_{HASH,\mathcal{A}} = 1] - 1|$, where $Pr[E]$ is the probability of an event E. The advantage function for this experiment becomes $Adv1^{3FAKA}_{HASH,\mathcal{A}}(t_1, q_R) = max_{\mathcal{A}}\{Succ1^{3FAKA}_{HASH,\mathcal{A}}\}$ in which the maximum is taken over all $\mathcal{A}$ with execution time t_1 and the number of queries q_R made to the *Reveal* oracle. According to the attack experiment described in Algorithm 1, if the adversary $\mathcal{A}$ has the ability to invert the one-way hash function $h(\cdot)$, then $\mathcal{A}$ can directly obtain U_i's ID_i, PW_i, and b_i and HG's K_U, and win the game. However, it is computationally infeasible problem to invert $h(\cdot)$, i.e., $Adv1^{3FAKA}_{HASH,\mathcal{A}}(t_1) < \epsilon$, for any sufficiently small $\epsilon > 0$. Then, we have

$Adv1^{3FAKA}_{HASH,\mathcal{A}}(t_1, q_R) \leq \epsilon$, since $Adv1^{3FAKA}_{HASH,\mathcal{A}}(t_1, q_R) \leq \epsilon$ depends on $Adv1^{3FAKA}_{HASH,\mathcal{A}}(t_1)$. Therefore, the proposed scheme is provably secure against the adversary $\mathcal{A}$ for deriving ID_i, PW_i, b_i, and K_U, even if the smart card SC_i is lost/stolen by $\mathcal{A}$. □

Algorithm 1 $EXP1^{3FAKA}_{HASH,\mathcal{A}}$

Extract the information $\{A_i, B_i, C_i^1, D_i\}$ from SC_i using the power analysis attack [29–31].
Call the *Reveal* oracle. Let $(HPW_i', HID_i') \leftarrow Reveal(B_i)$
Compute $a = A_i \oplus HID_i'$
Call the *Reveal* oracle. Let $(HPW_i'', TID_i') \leftarrow Reveal(a)$
if $(HPW_i'' = HPW_i')$ **then**
  Compute $PID_i^{1'} = C_i^1 \oplus h(TID_i' || HID_i')$
  Intercept the login request message $\langle PID_i^1, M_i, M_{U_i,G}, T_1 \rangle$
  Call the *Reveal* oracle. Let $(TID_i^*, HID_i^*, PID_i^{1*}, R_i^*, T_1^*) \leftarrow Reveal(M_{U_i,G})$
  if $(PID_i^{1*} = PID_i^{1'})$ and $(TID_i^* = TID_i')$ and $(HID_i^* = HID_i')$ and $(T_1^* = T_1)$ **then**
    Call the *Reveal* oracle. Let $(TID_i^{**}, K_U^{**}) \leftarrow Reveal(HID_i')$
    Call the *Reveal* oracle. Let $(PW_i^{**}, b_i^{**}) \leftarrow Reveal(HPW_i')$
    Call the *Reveal* oracle. Let $(ID_i^{**}, u_i^{**}) \leftarrow Reveal(TID_i')$
    Compute $D_i' = u_i^{**} \oplus h(ID_i^{**} || b_i^{**})$
    if $(D_i' = D_i)$ **then**
      Accept ID_i^{**}, PW_i^{**}, and b_i^{**} as the correct identity ID_i, password PW_i, biometric key b_i of the user U_i, and K_U^{**} as the correct secret key K_U of HG.
      return 1
    else
      return 0
    end if
  else
    return 0
  end if
else
  return 0
end if

Theorem 2. *Under the assumption that a one-way hash function $h(\cdot)$ behaves like an oracle, the proposed scheme is probably secure against an adversary $\mathcal{A}$ for deriving the session key SK_{ij} shared between a legal user U_i and a sensor node S_j.*

Proof of Theorem 2. The proof of this theorem is similar to that in Theorem 1. We assume that an adversary $\mathcal{A}$ is able to derive the session key SK_{ij} shared between a legal user U_i and a sensor node S_j. For this, $\mathcal{A}$ uses the *Reveal* oracle to run an experimental algorithm $EXP2^{3FAKA}_{HASH,\mathcal{A}}$ shown in Algorithm 2 for the proposed three-factor authentication and key agreement ($3FAKA$). We define the success probability for $EXP2^{3FAKA}_{HASH,\mathcal{A}}$ as $Succ2^{3FAKA}_{HASH,\mathcal{A}} = |Pr[EXP2^{3FAKA}_{HASH,\mathcal{A}} = 1] - 1|$. The advantage function for this experiment becomes $Adv2^{3FAKA}_{HASH,\mathcal{A}}(t_2, q_R) = max_{\mathcal{A}}\{Succ2^{3FAKA}_{HASH,\mathcal{A}}\}$ in which the maximum is taken over all $\mathcal{A}$ with execution time t_2 and the number of queries q_R made to the *Reveal* oracle. According to the attack experiment described in Algorithm 2, if the adversary $\mathcal{A}$ has the ability to invert the one-way hash function $h(\cdot)$, then $\mathcal{A}$ can easily derive SK_{ij} and win the game. However, it is computationally infeasible problem to invert $h(\cdot)$, i.e., $Adv2^{3FAKA}_{HASH,\mathcal{A}}(t_2) < \epsilon$, for any sufficiently small $\epsilon > 0$. Then, we have $Adv2^{3FAKA}_{HASH,\mathcal{A}}(t_2, q_R) \leq \epsilon$, since $Adv2^{3FAKA}_{HASH,\mathcal{A}}(t_2, q_R) \leq \epsilon$ is also dependent

on $Adv2^{3FAKA}_{HASH,\mathcal{A}}(t_2)$. Therefore, the proposed scheme is provably secure against the adversary $\mathcal{A}$ for deriving SK_{ij}. □

Algorithm 2 $EXP2^{3FAKA}_{HASH,\mathcal{A}}$

Intercept the login request message $\langle PID_i^1, M_i, M_{U_i,G}, T_1 \rangle$ during the login phase.
Call the *Reveal* oracle. Let $(TID_i', HID_i', PID_i^{1'}, R_i', T_1') \leftarrow Reveal(M_{U_i,G})$
if $(PID_i^{1'} = PID_i^1)$ and $(T_1' = T_1)$ **then**
  Compute $r_i' = M_i \oplus h(TID_i'||HID_i'||T_1)$
  Compute $R_i'' = h(TID_i'||PID_i^1||r_i')$
  if $(R_i'' = R_i')$ **then**
    Intercept the message $\langle M_j, M_{S_j,G}, T_3 \rangle$ during the authentication phase.
    Call the *Reveal* oracle. Let $(PID_i^{1*}, SID_j^*, X_{S_j}^*, R_j^*, SK_{ij}^*, T_3^*) \leftarrow Reveal(M_{S_j,G})$
    if $(PID_i^{1*} = PID_i^1)$ and $(T_3^* = T_3)$ **then**
      Compute $SK_{ij}' = h(R_i'||R_j^*)$
      if $(SK_{ij}' = SK_{ij}^*)$ **then**
        Accept SK_{ij}^* as the correct session key shared between U_i and S_j.
        return 1
      else
        return 0
      end if
    else
      return 0
    end if
  else
    return 0
  end if
else
  return 0
end if

5.2. Security Verification using BAN Logic

In this section, we use BAN logic to verify the legitimacy of the session key shared between participants who communicate in the proposed scheme. Tables 3 and 4 illustrate notations and rules used in BAN logic, respectively.

Table 3. Notations in BAN logic.

Notation	Description	Notation	Description
$P\| \equiv X$	P believes X	$\#(X)$	X is fresh
$P \triangleleft X$	P sees X	$P \overset{K}{\leftrightarrow} Q$	K is the shared key between P and Q
$P\| \sim X$	P said X	$\langle X \rangle_Y$	X combined with the formula Y
$P \Rightarrow X$	P has jurisdiction over X	$(X)_K$	X hashed under the key K

Table 4. Rules in BAN logic.

Rule	Description
$\frac{P\|\equiv P\overset{K}{\leftrightarrow}Q, P\triangleleft\langle X\rangle_K}{P\|\equiv Q\|\sim X}$	**[Rule 1: Message-meaning rule]** if P believes that the K is shared with Q and P sees X combined with K, then P believes Q said X
$\frac{P\|\equiv \#(X), P\|\equiv Q\|\sim X}{P\|\equiv Q\|\equiv X}$	**[Rule 2: Nonce-verification rule]** if P believes that X is fresh and P believes Q said X, then P believes that Q believes X
$\frac{P\|\equiv \#(X)}{P\|\equiv \#(X,Y)}$	**[Rule 3: Freshness-conjuncation rule]** if P believes that X is fresh, then P believes that (X, Y) is fresh
$\frac{P\|\equiv Q\|\Rightarrow X, P\|\equiv Q\|\equiv X}{P\|\equiv X}$	**[Rule 4: Jurisdiction rule]** if P believes that X has jurisdiction over X and P believes that Q believes X, then P also believes X

To ensure the security of the proposed scheme under BAN logic, the proposed scheme needs to satisfy the following goals.

- Goal 1: $U_i\| \equiv S_j\| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$
- Goal 2: $U_i\| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$
- Goal 3: $S_j\| \equiv U_i\| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$
- Goal 4: $S_j\| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$

We first transfer all transmitted messages into idealized form as follows.

- M_1: $U_i \rightarrow HG$: $(PID_i^{\ell}, R_i, K_U, T_1)_{HID_i}$
- M_2: $HG \rightarrow S_j$: $(PID_i^{\ell}, SID_j, R_i, K_S, T_2)_{X_{S_j}}$
- M_3: $S_j \rightarrow HG$: $(PID_i^{\ell}, SID_j, R_j, K_S, T_3)_{X_{S_j}}$
- M_4: $HG \rightarrow U_i$: $(PID_i^{\ell}, PID_i^{\ell+1}, R_j, K_U, T_4)_{HID_i}$

We secondly define some assumptions as initiative premises as follows.

- P_1: $HG\| \equiv \#(T_1)$
- P_2: $S_j\| \equiv \#(T_2)$
- P_3: $HG\| \equiv \#(T_3)$
- P_4: $U_i\| \equiv \#(T_4)$
- P_5: $U_i\| \equiv (U_i \overset{HID_i}{\longleftrightarrow} HG)$
- P_6: $HG\| \equiv (U_i \overset{HID_i}{\longleftrightarrow} HG)$
- P_7: $S_j\| \equiv (S_j \overset{X_{S_j}}{\longleftrightarrow} HG)$
- P_8: $HG\| \equiv (S_j \overset{X_{S_j}}{\longleftrightarrow} HG)$
- P_9: $U_i\| \equiv S_j\| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$
- P_{10}: $S_j\| \equiv U_i\| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$

We then prove the proposed scheme achieves the security goals based on the idealized form of the messages, assumptions, and BAN logic rules.

- According to M_1, we get
  V_1: $HG \triangleleft (PID_i^{\ell}, R_i, K_U, T_1)_{HID_i}$
- According to P_6 and Rule 1, we get
  V_2: $HG\| \equiv U_i\| \sim (PID_i^{\ell}, R_i, K_U, T_1)_{HID_i}$

- According to P_1 and Rule 3, we get
  V_3: $HG| \equiv \#(PID_i^{\ell}, R_i, K_U, T_1)_{HID_i}$
- According to V_2, V_3, and Rule 2, we get
  V_4: $HG| \equiv U_i| \equiv (PID_i^{\ell}, R_i, K_U, T_1)_{HID_i}$
- According to M_2, we get
  V_5: $S_j \triangleleft (PID_i^{\ell}, SID_j, R_i, K_S, T_2)_{X_{S_j}}$
- According to P_7 and Rule 1, we get
  V_6: $S_j| \equiv HG| \sim (PID_i^{\ell}, SID_j, R_i, K_S, T_2)_{X_{S_j}}$
- According to P_2 and Rule 3, we get
  V_7: $S_j| \equiv \#(PID_i^{\ell}, SID_j, R_i, K_S, T_2)_{X_{S_j}}$
- According to V_6, V_7, and Rule 2, we get
  V_8: $S_j| \equiv HG| \equiv (PID_i^{\ell}, SID_j, R_i, K_S, T_2)_{X_{S_j}}$
- According to M_3, we get
  V_9: $HG \triangleleft (PID_i^{\ell}, SID_j, R_j, K_S, T_3)_{X_{S_j}}$
- According to P_8 and Rule 1, we get
  V_{10}: $HG| \equiv | \sim (PID_i^{\ell}, SID_j, R_j, K_S, T_3)_{X_{S_j}}$
- According to P_3 and Rule 3, we get
  V_{11}: $HG| \equiv \#(PID_i^{\ell}, SID_j, R_j, K_S, T_3)_{X_{S_j}}$
- According to V_{10}, V_{11}, and Rule 2, we get
  V_{12}: $HG| \equiv S_j| \equiv (PID_i^{\ell}, SID_j, R_j, K_S, T_3)_{X_{S_j}}$
- According to M_4, we get
  V_{13}: $U_i \triangleleft (PID_i^{\ell}, PID_i^{\ell+1}, R_j, K_U, T_4)_{HID_i}$
- According to P_5 and Rule 1, we get
  V_{14}: $U_i| \equiv HG| \sim (PID_i^{\ell}, PID_i^{\ell+1}, R_j, K_U, T_4)_{HID_i}$
- According to P_4 and Rule 3, we get
  V_{15}: $U_i| \equiv \#(PID_i^{\ell}, PID_i^{\ell+1}, R_j, K_U, T_4)_{HID_i}$
- According to V_{14}, V_{15}, and Rule 2, we get
  V_{16}: $U_i| \equiv HG| \equiv (PID_i^{\ell}, PID_i^{\ell+1}, R_j, K_U, T_4)_{HID_i}$
- As $SK_{ij} = h(R_i||R_j)$ and combining V_{12}, V_{16}, we get
  V_{17}: $U_i| \equiv S_j| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$ (Goal 1)
- $SK_{ij} = h(R_i||R_j)$ and combining V_4, V_8, we get
  V_{18}: $S_j| \equiv U_i| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$ (Goal 3)
- According to P_9, V_{17} and Rule 4, we get
  V_{19}: $U_i| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$ (Goal 2)
- According to P_{10}, V_{18} and Rule 4, we get
  V_{20}: $S_j| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j)$

Therefore, the above logic proves that the proposed scheme achieves Goals 1–4 successfully. In other words, the proposed scheme achieves mutual authentication and the session key SK_{ij} is securely shared between parties.

5.3. Security Verification Using AVISPA

We simulate the proposed scheme using the AVISPA software, a widely accepted tool for automatically validating the security features of the protocols. We describe the implementation of the proposed scheme using HLPSL (High-Level Protocols Specification Language) and then present the simulation results.

5.3.1. HLPSL Specification of the Proposed Scheme

We now briefly discuss the simulation process of the proposed scheme for the roles of the participants, U_i, HG, and S_j, the session, the goal, and the environment. Tables 5–7 present the roles of U_i, HG, and S_j in HLPSL language, respectively. Table 8 presents the session, environment, and goal roles in the HLPSL language. In the implementation, the following seven secrecy goals and two authentication properties were verified.

- Goal 1: The secrecy_of subs1 represents that $\langle ID_i, PW_i \rangle$ are kept secret to (U_i) only.
- Goal 2: The secrecy_of subs2 represents that $\langle TID_i, HID_i \rangle$ are kept secret to (U_i, HG) only.
- Goal 3: The secrecy_of subs3 represents that $\langle R_i, R_j \rangle$ are kept secret to (U_i, HG, S_j) only.
- Goal 4: The secrecy_of subs4 represents the negotiated session key SK_{ij} is only known to (U_i, HG, S_j).
- Goal 5: The secrecy_of subs5 represents that the secret key K_U of HG is permanently kept secret, known to only (HG).
- Goal 6: The secrecy_of subs6 represents that the secret key K_S of HG is permanently kept secret, known to only (HG).
- Goal 7: The secrecy_of subs7 represents that the shared secret X_{S_j} is only known to (HG, S_j).
- Authentication Property 1: The authentication_on user_gateway_rri represents that U_i generates R_i. If HG securely receives R_i through a message, it authenticates U_i.
- Authentication Property 2: The authentication_on gateway_sensor_rrj represents that S_j generates R_j. If HG securely receives R_j through a message, it authenticates S_j.

Table 5. Role specification of U_i in HLPSL.

```
1: role user(Ui, HG, Sj: agent, SKey1: symmetric_key, SKey2: symmetric_key,
             H, GEN, REP: hash_func, Snd, Rcv: channel(dy))
2: played_by Ui
3: def=
4:    local State: nat, IDi, PWi, Bioi, BBi, Pari, TIDi, HPWi, HIDi, PID1i, Si, Ai, Bi, Ci, C1i, C2i, Di, RRi, RRj, Ri,
            T1, T4: text, Mi, Muig, SKij, P2i, Mg, Mgui: message,
5:    Inc: hash_func
6:    const user_gateway, gateway_user, sensor_user, subs1, subs2, subs3, subs4, subs5, subs6, subs7: protocol_id
7:    init State :=0
8:    transition
9:       1. State = 0 ∧ Rcv(start) = |>
10:      State' := 1 ∧ IDi' := new() ∧ PWi' := new() ∧ Si' := new() ∧ BBi' := GEN(Bioi) ∧ Pari' := GEN(Bioi)
        ∧ HPWi' := H(PWi'.BBi') ∧ TIDi' := H(IDi'.Si') ∧ Snd(TIDi'.HPWi'_SKey1) ∧ secret(IDi,PWi, subs1, Ui)
11:      2. State = 1 ∧ Rcv({Ai'.Bi'.C1i'}_SKey1) = |>
12:      State' := 2 ∧ Ri' := new() ∧ T1' := new() ∧ BBi' := GEN(Bioi) ∧ Di' := xor(ui,H(IDi.BBi')) ∧ TIDi' := H(IDi.ui)
        ∧ HPWi' := h(PWi.BBi') ∧ Ai' := xor(HIDi, H(HPWi'.TIDi')) ∧ Bi' := H(HPWi'.HIDi)
        ∧ PID1i' := xor(C1i', H(TIDi'.HIDi)) ∧ RRi' := H(TIDi'.PID1i'.Ri') ∧ Mi' := xor(Ri', H(TIDi'.HIDi.T1'))
        ∧ Muig' := H(TIDi'.HIDi.PID1i'.RRi'.T1')
        ∧ Snd(PID1i'.Mi'.Muig'.T1') ∧ secret({TIDi,HIDi}, subs2, {Ui, HG}) ∧ witness(Ui, HG, user_gateway, RRi')
13:      3. State = 2 ∧ Rcv(P2i'.Mg'.Mgui'.T4') = |>
14:      State' := 3 ∧ RRj' := xor(Mg', H(PID1i.HIDi)) ∧ SKij' := H(RRi.RRj')
        ∧ secret({RRi,RRj}, subs3, {Ui, HG, Sj}) ∧ secret({SKij}, subs4, {Ui, HG, Sj})
15: end role
```

Table 6. Role specification of HG in HLPSL.

```
1: role gateway(Ui, HG, Sj: agent, SKey1: symmetric_key, SKey2: symmetric_key,
                H, GEN, REP: hash_func, Snd, Rcv: channel(dy))
2: played_by HG
3: def=
4:    local State: nat, Ku, Ks, TIDi, HPWi, PID1i, PID2i, HIDi, Ai, Bi, C1i, C2i, SIDj, Xsj, RRi, RRj, Ri, Rj,
            T1, T2, T3, T4: text, Mi, Muig, Mg, Mg2, Mgsj, Mgui, Mj, Msjg, SKij, P2i: message,
5:    Inc: hash_fun
6:    const user_gateway, gateway_user, sensor_user, subs1, subs2, subs3, subs4, subs5, subs6, subs7: protocol_id
7:    init State :=0
8:   transition
9:       1. State = 0 ∧ Rcv({TIDi'.HPWi'}_SKey1) =|>
10:       State' := 1 ∧ PID1i' := new() ∧ HIDi' := H(TIDi'.Ku) ∧ Ai' := xor(HIDi', H(HPWi'.TIDi'))
        ∧ Bi' := H(HPWi'.HIDi') ∧ C1i' := xor(PID1i', H(TIDi'.HIDi')) ∧ Snd({Ai'.Bi'.C1i'}_SKey1)
        ∧ SIDj' := new() ∧ Xsj' := H(SIDj'.Ks)
        ∧ Snd({SIDj'.Xsj'}_SKey2) ∧ secret(Ku, subs5, HG) ∧ secret(Ks, subs6, HG) ∧ secret(Xsj, subs7, HG, Sj)
11:       2. State = 1 ∧ Rcv(PID1i'.Mi'.Muig'.T1') =|>
12:       State' := 2 ∧ HIDi' := H(TIDi.Ku) ∧ Ri' := xor(Mi', H(TIDi.HIDi'.T1')) ∧ RRi' := H(TIDi.PID1i.Ri')
        ∧ T2' := new() ∧ Xsj' := H(SIDj.Ks) ∧ Mg2' := xor(RRi', H(Xsj'.T2')) ∧ Mgsj' := H(PID1i'.SIDj.Xsj'.RRi'.T2')
        ∧ Snd(PID1i'.Mg2'.Mgsj'.T2')
13:       3. State = 2 ∧ Rcv(Mj'.Msjg'.T3') =|>
14:       State' := 3 ∧ Rj' := xor(Mj', H(Xsj.T3')) ∧ RRj' := H(SIDj.Rj') ∧ SKij' := H(RRi.RRj') ∧ PID2i' := new()
        ∧ T4' := new() ∧ C2i' := xor(PID2i', H(TIDi.HIDi)) ∧ P2i' := xor(C2i', H(HIDi.T4'))
        ∧ Mg' := xor(RRj', H(PID1i.HIDi)) ∧ Mgui' := H(PID1i.HIDi.C2i'.RRj'.SKij'.T4')
        ∧ Snd(P2i'.Mg'.Mgui'.T4')
15: end role
```

Table 7. Role specification of S_j in HLPSL.

```
1: role sensor(Ui, HG, Sj: agent, SKey1: symmetric_key, SKey2: symmetric_key,
               H, GEN, REP: hash_func, Snd, Rcv: channel(dy))
2: played_by Sj
3: def=
4:    local State: nat, PID1i, SIDj, Xsj, RRi, RRj, Rj, T2, T3: text, Mg2, Mgsj, Mgui, Mj, Msjg, SKij: message,
5:    Inc: hash_func
6:    const user_gateway, gateway_sensor, sensor_user, subs1, subs2, subs3, subs4, subs5, subs6, subs7: protocol_id
7:    init State :=0
8:    transition
9:       1. State = 0 ∧ Rcv({SIDj'.Xsj'}_SKey2) =|>
10:       State' := 1 ∧ T3' := new()
11:       2. State = 1 ∧ Rcv(PID1i'.Mg2'.Mgsj'.T2') =|>
12:       State' := 2 ∧ RRi' := xor(Mg2', H(Xsj.T2')) ∧ Rj' := new() ∧ T3' := new() ∧ RRj' := H(SIDj.Rj')
        ∧ Mj' := xor(Rj', H(Xsj.T3')) ∧ SKij' := H(RRi'.RRj') ∧ Msjg' := H(PID1i'.SIDj.Xsj.Rj'.SKij'.T3')
        ∧ Snd(Mj'.Msjg'.T3') ∧ witness(Sj, HG, gateway_sensor, RRj')
13: end role
```

Table 8. Specification of the session, environment, and goal in HLPSL.

```
1: role session(Ui, HG, Sj:agent, SKey1: symmetric_key, SKey2: symmetric_key,
                H, GEN, REP: hash_func)
2: def=
3:    local SI, SJ, RI, RJ, PI, PJ: channel(dy)
4:    composition
5:       user(Ui, HG, Sj, SKey1, SKey2, H, GEN, REP, SI, RI)
6:       ∧ gateway(Ui, HG, Sj, SKey1, SKey2, H, GEN, REP, SJ, RJ)
7:       ∧ sensor(Ui, HG, Sj, SKey1, SKey2, H, GEN, REP, PI, PJ)
8: end role

1: role environment()
2: def=
3:    const ui, hg, sj: agent, skey1 : symmetric_key, skey2 : symmetric_key, h, gen, rep: hash_func,
4:          idi, bioi, sidj, pwi, ai, bi, ci, t1, t2, t3, t4, rri, rrj, skij, mi, mj, mg, mg2, muig, mgui, mgsj, msjg: text,
5:          user_gateway_rri, gateway_sensor_rrj, sensor_user,
6:          subs1, subs2, subs3, subs4, subs5, subs6, subs7: protocol_id
7:    intruder_knowledge = ui, hg, sj, h, gen, rep, mi, muig, mg2, mgsj, mj, msjg, mg, mgui
8:    composition
9:       session(hg, ui, sj, skey1, skey2, h, gen, rep)
10:      ∧ session(ui, hg, sj, skey1, skey2, h, gen, rep)
11:      ∧ session(sj, ui, hg, skey1, skey2, h, gen, rep)
12: end role

1: goal
2:    secrecy_of subs1 secrecy_of subs2 secrecy_of subs3 secrecy_of subs4
3:    secrecy_of subs5 secrecy_of subs6 secrecy_of subs7
4:    authentication_on user_gateway_rri authentication_on gateway_sensor_rrj 5: end goal
environment()
```

5.3.2. Simulation Results

We execute the HLPSL specifications using SPAN (Security Protocol ANimator for AVISPA) [32]. Figure 5a,b show the simulation results based on OFMC (On-the-Fly-Model-Checker) and CL-AtSe (Constraint-Logic-based Attack Searcher) models, respectively. From these results, we find that the proposed scheme is SAFE under OFMC and CL-AtSe against active and passive attacks. Therefore, we demonstrate that the proposed scheme is secure.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/3factor_protocol.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.13s
  visitedNodes: 8 nodes
  depth: 3 plies
```

(**a**)

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/3factor_protocol.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.09 seconds
  Computation: 0.00 seconds
```

(**b**)

Figure 5. Simulation results of the proposed scheme using AVISPA tool: (**a**) OFMC model, (**b**) CL-AtSe model.

6. Implication of Security Analysis

We further describe the implication of our security analysis with regard to security properties of the proposed scheme. Saying, we show how the proposed scheme satisfies the security requirements

for user authentication and session key agreement and resists various kinds of known attacks. We then compare the security of the proposed scheme with other related schemes.

6.1. Security Properties

6.1.1. Mutual Authentication

In steps (1) and (4) of Section 4.4, U_i and HG authenticate each other by verifying the correctness of $M_{U_i,G}$ and M_{G,U_i}. An adversary cannot generate legal $M_{U_i,G} = h(TID_i||HID_i^*||PID_i^1|R_i||T_1)$ and $M_{G,U_i} = h(PID_i^1||HID_i^*||C_i^2||R_j^*||SK_{ij}^*||T_4)$ without knowing HID_i. Even if the adversary obtains SC_i of U_i and stored values, the adversary cannot derive the correct HID_i without having the corresponding U_i's ID_i, PW_i, and Bio_i. As a result, the proposed scheme can achieve mutual authentication between U_i and HG.

In steps (2) and (3) of Section 4.4, HG and S_j authenticate each other by verifying the correctness of M_{G,S_j} and $M_{S_j,G}$. An adversary cannot generate legal $M_{G,S_j} = h(PID_i^1||SID_j||X_{S_j}||R_i^*||T_2)$ and $M_{S_j,G} = h(PID_i^1||SID_j||X_{S_j}||R_j||SK_{ij}||T_3)$ without knowing their shared secret information X_{S_j}. As a result, the proposed scheme can achieve mutual authentication between HG and S_j.

6.1.2. Session Key Agreement

In the login and authentication phases, the session key $SK_{ij} = h(R_i||R_j) = h(h(TID_i||PID_i^\ell||r_i)||h(SID_j||r_j))$ is established between U_i and S_j for protecting future communication. In the proposed scheme, the secrecy of SK_{ij} is dependent on the secrecy of the random values r_i and r_j. These values are carefully protected by the secret keys shared between U_i and HG and between HG and S_j, respectively. Even if an adversary obtains SK_{ij} for the ℓ-th session, he/she cannot compute any of the past and future session keys by using this disclosed SK_{ij} because SK_{ij} is protected by $h(\cdot)$ and the random values r_i and r_j including one-time psuedonym PID_i^ℓ are different in each session. As a result, the proposed scheme achieves both session key agreement and known key security.

6.1.3. User Anonymity with Untraceability

As we mentioned in Section 3.1, for fully protecting user privacy, strong anonymity with untraceability is required. In the proposed scheme, the U_i's actual identity ID_i is not transmitted during all phases, including the registration phase. Therefore, even if an adversary eavesdrops on all communication messages, it is not possible to obtain ID_i directly from the messages. In addition, even if the adversary gets TID_i, it cannot retrieve ID_i from TID_i because ID_i is masked with u_i and u_i is protected by Bio_i only known to U_i. Similarly, even if the adversary gets HID_i, it cannot retrieve ID_i from HID_i without knowing a secret key, K_U, which is only known to HG.

Furthermore, M_i and $M_{U_i,G}$ in the login request message are computed with random values r_i and T_1 and U_i uses an one-time pseudonym PID_i^ℓ every session. In other words, all values in the login request message are different in sessions. Therefore, any adversary cannot trace the different sessions of the same user from exchanged messages via public channels and the proposed scheme achieves the feature of strong anonymity with untraceability.

6.1.4. Resisting Stolen Smart Card Attack

In the proposed scheme, U_i's smart card SC_i contains $\{A_i, B_i, C_i^\ell, D_i, par_i, h(\cdot), GEN(\cdot), REP(\cdot)\}$ where $A_i = h(HPW_i||TID_i) \oplus HID_i$, $B_i = h(HPW_i||HID_i)$, $C_i^\ell = h(TID_i||HID_i) \oplus PID_i^\ell$ and $D_i = u_i \oplus h(ID_i||b_i)$. Even if SC_i is stolen by an adversary and all contained values in it are retrieved by the adversary through side-channel attacks such as power analysis attack [29–31], the adversary cannot guess HPW_i, TID_i, and HID_i including ID_i, PW_i, and Bio_i by using A_i, B_i, C_i, and D_i and also cannot guess PID_i^ℓ from C^ℓ without knowing b_i, u_i, and K_U because it is impossible to know these key values.

Without knowing U_i's real identity ID_i, password PW_i, and biometric Bio_i, the adversary cannot impersonate as the user. As a result, the proposed scheme can resist the stolen smart card attack.

6.1.5. Resisting Offline Guessing Attack

An adversary may attempt to guess U_i's identity ID_i, password PW_i and biometric key b_i by extracting the values stored in the smart card SC_i. However, the adversary cannot derive b_i using only par_i without knowing the U_i's biometric Bio_i. The adversary also cannot derive ID_i and b_i from TID_i and D_i, respectively, without knowing the random value u_i . Therefore, the adversary cannot guess the correct ID_i, PW_i, and b_i without knowing Bio_i and u_i due to the collision-resistant property of the one-way hash function $h(\cdot)$. As a result, the proposed scheme can resist the offline guessing attack.

6.1.6. Resisting Privileged Insider Attack

In practice, users tend to use same password to register across different systems. If a privileged insider obtain the user's password, he/she can use it to access other systems by impersonating as this user. In the proposed scheme, U_i submits the hashed password HPW_i instead of the plaintext of real password PW_i during the registration phase. HPW_i is also masked by U_i's secret biometric key b_i. Therefore, an insider cannot obtain U_i's real password and the proposed scheme can resist the privileged insider attack.

6.1.7. Resisting Stolen-Verifier Attack

To succeed in the stolen-verifier attack, an adversary should obtain the verification information (e.g., the plaintexts of passwords, hashed passwords, biometric key data, or hashed biometric key data) stored in the server. However, in the proposed scheme, the server maintains only $\{PID_i^1, TID_i\}$ which is both password-independent and biometric-key-independent information. Therefore, the proposed scheme can resist the stolen-verifier attack.

6.1.8. Resisting Known Session-Specific Temporary Information Attack

In the proposed scheme, both randomly selected values r_i and r_j, from U_i and S_j, respectively, are always masked by the secret values HID_i and X_{S_j}. Even if an adversary knows r_i and r_j, he/she cannot compute $SK_{ij} = h(R_i||R_j) = h(h(TID_i||PID_i^\ell||r_i)||h(SID_j||r_j))$ without knowing U_i's temporary identity TID_i and one-time pseudonym PID_i^ℓ and S_j's identity SID_j. Moreover, as we described, the adversary has no way to compute TID_i and SID_j. As a result, in the proposed scheme, a leakage of the session-specific temporary information r_i and r_j does not affect the security of the established session key.

6.1.9. Resisting User Impersonation Attack

To impersonate a user U_i, an adversary should obtain the values in SC_i and intercepts the messages exchanged in the previous sessions. In the proposed scheme, even if the adversary succeeded the above things, the adversary cannot produce a legal login request $\langle PID_i^1, M_i, M_{U_i,G}, T_1 \rangle$ without knowing all the authentication factors, i.e., SC_i, PW_i, and Bio_i including ID_i and u_i. As we mentioned above, it is impossible for an adversary to obtain ID_i, PW_i, u_i, and b_i. Therefore, the proposed scheme can resist the user impersonation attack.

6.1.10. Resisting Sensor Node Impersonation and Node Capture Attacks

To impersonate a sensor node S_j, an adversary should intercept the messages exchanged in the previous sessions. However, in the proposed scheme, the adversary cannot produce a legal message $\langle M_j, M_{S_j,G}, T_3 \rangle$ without knowing $X_{S_j} = h(SID_j||K_S)$ because the adversary does not know the HG's secret key K_S even if he/she obtains SID_j.

Even if the adversary captures a sensor node S_j and obtains X_{S_j} stored in S_j, the adversary's further attacks using the compromised sensor node only affect communications related to that node. Since each sensor node has a different key $X_{S_m} = h(SID_m||K_S)$, the adversary cannot derive other non-compromised sensor nodes' keys without knowing K_S and thus the further attacks will not affect other communications. As a result, the proposed scheme can resist both sensor node impersonation attack and node capture attack.

6.2. Comparison of Security Features

We compare the security features of the proposed scheme with other related three-factor authentication and key agreement schemes [9,11–13]. Table 9 shows the comparison results. From Table 9, we can see that first three related schemes do not guarantee all security features, in especial, untraceability required for strong anonymity. The proposed scheme and Jiang et al.'s scheme achieves more ideal security features and resist most of attacks. However, Jiang et al.'s scheme is expensive to implement and deploy in practical applications due to the low performance of Rabin cryptosystem. As shown in Section 7, Jiang et al.'s scheme is five times slower than the proposed scheme in total running time.

Table 9. Security feature comparison of the proposed scheme with other related three-factor authentication and key agreement schemes.

Security Feature	Amin et al. [11]	Park et al. [9]	Jung et al. [13]	Jiang et al. [12]	Proposed Scheme
Mutual authentication	O	O	O	O	O
Session key security	O	O	X	O	O
User anonymity	O	O	O	O	O
Untraceability	X	X	X	O	O
Resistance to Stolen smart card attack	X	O	X	O	O
Offline guessing attack	O	O	O	O	O
Privileged insider attack	O	O	O	O	O
Stolen-verifier attack	O	X	O	O	O
Known session-specific temporary information attack	X	O	O	O	O
User impersonation attack	O	O	X	O	O
Sensor node impersonation attack	O	O	O	O	O

O: The scheme can provide the security feature or resist the attack; X: The scheme cannot provide the security feature or resist the attack.

7. Performance Analysis of the Proposed Scheme

We analyze the performance of the proposed scheme and compare it with other related schemes in terms of computational cost and communication cost.

7.1. Computational Cost Analysis

For computational cost analysis, we compare the computation cost of the proposed scheme with the four related schemes [9,11–13]. We only focus on comparing the login and authentication phases because the registration and password change phases are not performed frequently. Since the time for executing of a bitwise XOR operation is negligible, we do not consider XOR operations for computational cost analysis. To facilitate analysis, we use the following notations.

- T_H: time for executing a one-way hash function
- T_B: time for executing a biohash function
- T_F: time for executing a fuzzy extractor
- T_P: time for executing an ECC point multiplication
- T_M: time for a modular exponentiation

Wang et al. [33] implemented several operations on three kinds of common PCs and measured their execution time by using C/C++ library MIRACL. According to the experimental results in

Wang et al.'s research [33], we assume that the executing time for the cryptographic one-way hash function T_H (SHA-1), ECC point multiplication T_P (ECC sect163r1 [34]), and modular exponentiation T_M ($|n| = 512$) on common PCs (Intel T5870 2.00 GHz, Intel, Santa Clara, CA, US) are 2.58 µs, 1.226 ms, and 2.573 ms, respectively. Moreover, the execution time for the fuzzy extractor operation T_F is almost the same as the ECC point multiplication T_P [35] and it is also assumed that $T_B = T_F \approx T_P$ according to [36]. We consider possible real sensor devices with 8-bit ATmega128L micorocontroller (i.e., MICAz of Crossbow Technology). According to the experimental results on those sensor nodes [37,38], we assume that the executing time for the cryptographic one-way hash function T'_H (SHA-1) and ECC point multiplication T'_P (ECC sect163r1 [34]) are 3.6 ms and 114 ms, respectively.

In Table 10, we summarize the computational cost and running time of the proposed scheme and of the related schemes for user, gateway node, and sensor node. The total running time of the proposed scheme for the login and authentication phases is $T_F + 28T_H + 6T'_H \approx 22.9$ ms. It shows that the proposed scheme is almost 10 times more efficient than and Park et al. scheme [9]. The proposed scheme also has a higher security level than both Amin et al.'s scheme [11] and Jung et al.'s scheme [13] as shown in Table 9 and it is as efficient as them. Although Jiang et al.'s scheme [12] has similar security level with the proposed scheme, the proposed scheme is slightly efficient and easily implemented than Jiang's et al.'s scheme since the proposed scheme uses only lightweight operations such as XOR and hash functions not complex public-key cryptographic operations. Therefore, the proposed scheme can achieve all security features in Table 9 without deteriorating efficiency in terms of the computational cost.

Table 10. Comparison of computation costs for the login and authentication phases of the proposed scheme and other related schemes.

Entity	Amin et al. [11]	Park et al. [9]	Jung et al. [13]	Jiang et al. [12]	Proposed Scheme
User	$T_B + 12T_H$	$T_F + 2T_P + 10T_H$	$T_B + 8T_H$	$T_B + T_M + 8T_H$	$T_F + 13T_H$
Gateway node	$15T_H$	$11T_H$	$9T_H$	$T_M + 12T_H$	$15T_H$
Sensor node	$5T'_H$	$2T'_P + 4T'_H$	$4T'_H$	$5T'_H$	$6T'_H$
Total cost	$T_B + 27T_H + 5T'_H$	$T_F + 2T_P + 2T'_P$ $+21T_H + 4T'_H$	$T_B + 17T_H + 4T'_H$	$T_B + 2T_M$ $+20T_H + 5T'_H$	$T_F + 28T_H + 6T'_H$
Total running time	≈ 19.3 ms	≈ 246.1 ms	≈ 15.7 ms	≈ 24.4 ms	≈ 22.9 ms

7.2. Communication Cost Analysis

We also analyze the communication cost of the proposed scheme for login and authentication phases and compare it with that of the related schemes [9,11–13]. For communication cost analysis, we evaluate the communication cost in terms of the size of message in bits and the number of values in a message. We assume that the lengths of the identity, password, random number, and output of the hash function are each 128 bits. We also assume that the lengths of modulo n for rabin cryptosystem used in [12] and prime p for ECC used in [9] are each 1024 bits.

The communication cost of user, gateway node, and sensor node of the proposed scheme and related schemes are summarized in Table 11. The total communication cost of the proposed scheme is 1920 bits. From comparison in Table 11, the proposed scheme require lower communication cost than the above related schemes expect Jung et al.'s scheme. Although the proposed scheme is slightly less efficient than Jung et al.'s scheme in terms of communication cost, the difference (512 bits) is not significant since the proposed scheme has a higher security level as shown in Table 9.

Table 11. Comparison of communication costs for the login and authentication phases of the proposed scheme and other related schemes: the size of message in bits (the number of values in a message).

Communication	Amin et al. [11]	Park et al. [9]	Jung et al. [13]	Jiang et al. [12]	Proposed Scheme
User→Gateway node	768 bits (6)	1536 bits (5)	512 bits (4)	1408 bits (4)	512 bits (4)
Gateway node→Sensor node	640 bits (5)	1408 bits (4)	512 bits (4)	640 bits (5)	512 bits (4)
Sensor node→Gateway node	384 bits (3)	1280 bits (3)	256 bits (2)	384 bits (3)	384 bits (3)
Gateway node→User	384 bits (3)	1408 bits (4)	384 bits (3)	256 bits (2)	512 bits (4)
Total	2176 bits	5632 bits	1664 bits	2688 bits	1920 bits

8. Conclusions

In this paper, we have identified the security weaknesses in the recent three-factor authentication and key agreement scheme. Then, we have introduced the system model for smart homes based on WSNs. Based on this model, we have proposed a secure and lightweight three-factor authentication and key agreement scheme using the smart card, password, and biometrics. We have presented security proof using random oracle model and BAN logic. Afterwards, we have performed the security verification using AVISPA. Through formal and informal security analysis, we have demonstrated the proposed scheme fulfills the desirable security requirements and resists against various attacks. We have also evaluated the performance of the proposed scheme with regard to the computational and communication overheads. Finally, we have presented the comparative analysis of the proposed scheme with other related schemes, which justify that the proposed scheme has advantages in terms of efficiency and security.

In the future work, we expect to evaluate the performance of the proposed scheme by implementing and conducting experiments on actual devices (e.g., smart phones and sensor motes) for smart homes based on WSNs. Based on the experimental results, it will be possible to further examine the effectiveness of the proposed scheme.

Author Contributions: S.S. discovered the proclaimed weaknesses and proposed the improved scheme. T.K. directed this research and worked on the overall improvement.

Funding: This work was supported as part of Military Crypto Research Center(UD170109ED) funded by Defense Acquisition Program Administration(DAPA) and Agency for Defense Development(ADD).

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

References

1. Shih, C.S.; Chou, J.J.; Lin, K.J. WuKong: Secure Run-Time Environment and Data-Driven IoT Applications for Smart Cities and Smart Buildings. *J. Internet Serv. Inf. Secur.* **2018**, *8*, 1–17.
2. Zion Market Research. Available online: https://www.zionmarketresearch.com/news/smart-home-market (accessed on 2 February 2019).
3. Wong, K.H.M.; 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 (SUTC'06), Taichung, Taiwan, 5–7 June 2006; Volume 1.
4. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090. [CrossRef]
5. Vaidya, B.; Makrakis, D.; Mouftah, H. Two-factor mutual authentication with key agreement in wireless sensor networks. *Secur. Commun. Netw.* **2012**, *9*, 171–183. [CrossRef]
6. Kim, J.; Lee, D.; Jeon, W.; Lee, Y.; Won, D. Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks. *Sensors* **2014**, *14*, 6443–6462. [CrossRef] [PubMed]
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. Chang, I.P.; Lee, T.F.; Lin, T.H.; Liu, C.M. Enhanced Two-Factor Authentication and Key Agreement Using Dynamic Identities in Wireless Sensor Networks. *Sensors* **2015**, *15*, 29841–29854. [CrossRef] [PubMed]
9. Park, Y.; Park, Y. Three-Factor User Authentication and Key Agreement Using Elliptic Curve Cryptosystem in Wireless Sensor Networks. *Sensors* **2016**, *16*, 2123. [CrossRef]
10. 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]
11. 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]

12. 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]
13. Jung, J.; Moon, J.; Lee, D.; Won, D. Efficient and Security Enhanced Anonymous Authentication with Key Agreement Scheme in Wireless Sensor Networks. *Sensors* **2017**, *17*, 644. [CrossRef] [PubMed]
14. Shin, S.; Kwon, T. Two-Factor Authenticated Key Agreement Supporting Unlinkability in 5G-Integrated Wireless Sensor Networks. *IEEE Access* **2018**, *6*, 11229–11241. [CrossRef]
15. 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]
16. Chen, T.H.; Shih, W.K. A Robust Mutual Authentication Protocol for Wireless Sensor Networks. *ETRI J.* **2010**, *32*, 704–712. [CrossRef]
17. Huang, H.F.; Chang, Y.F.; Liu, C.H. Enhancement of Two-Factor User Authentication in Wireless Sensor Networks. In Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'10), Darmstadt, Germany, 15–17 October 2010; pp. 27–30.
18. He, D.; Gao, Y.; Chan, S.; Chen, C.; Bu, J. An Enhanced Two-factor User Authentication Scheme in Wireless Sensor Networks. *Ad Hoc Sens. Wirel. Netw.* **2010**, *10*, 361–371.
19. Shin, S.; Kwon, T. Cryptanalysis of the Anonymous Authentication with Key Agreement Scheme in Wireless Sensor Networks. In *Research Briefs on Information & Communication Technology Evolution (ReBICTE)*; ISYOU: Seoul, Korea, 2018; Volume 4.
20. Das, A.K.; Goswami, A. A robust anonymous biometric-based remote user authentication scheme using smart cards. *J.King Saud Univ. Comput. Inf. Sci.* **2015**, *27*, 193–210. [CrossRef]
21. Burnett, A.; Byrne, F.; Dowling, T.; Duffy, A. A Biometric Identity Based Signature Scheme. *Int. J. Netw. Secur.* **2007**, *5*, 317–326.
22. 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]
23. 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**, *12*, 43–59. [CrossRef]
24. Rahman, M.S.; Nakamura, T.; Base, A.; Takasaki, H.; Kiyomoto, S. PPM: Privacy Policy Manager for Home Energy Management System. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2018**, *9*, 42–56.
25. Jiang, Q.; Ma, J.; Wei, F.; Tian, Y.; Shen, J.; Yang, Y. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. *J. Netw. Comput. Appl.* **2016**, *76*, 37–48. [CrossRef]
26. Burrows, M.; Abadi, M.; Needham, R. A Logic of Authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
27. AVISPA (Automated Validation of Internet Security Protocols and Applications). Available online: http://www.avispa-project.org/ (accessed on 15 April 2019).
28. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuellar, J.; Drielsma, P.H.; Heám, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *Computer Aided Verification*; Etessami, K., Rajamani, S.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285.
29. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In *Advances in Cryptology, Proceedings of the CRYPTO' 99*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
30. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2002**, *51*, 541–552. [CrossRef]
31. Mahanta, H.J.; Azad, A.K.; Khan, A.K. Power analysis attack: A vulnerability to smart card security. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; pp. 506–510.
32. SPAN a Security Protocol ANimator for AVISPA. Available online: http://people.irisa.fr/Thomas.Genet/span/ (accessed on 2 February 2019).
33. Wang, D.; He, D.; Wang, P.; Chu, C. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals are Beyond Attainment. *IEEE Trans. Dependable Secur. Comput.* **2015**, *12*, 428–442. [CrossRef]

34. Certicom Research. Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters. Available online: http://www.secg.org/download/aid-784/sec2-v2.pdf (accessed on 5 March 2019).
35. Wazid, M.; Das, A.K.; Kummari, S.; Li, X.; Wu, F. Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS. *Secur. Commun. Netw.* **2016**, *9*, 1983–2001. [CrossRef]
36. He, D.; Kumar, N.; Lee, J.; Sherratt, R.S. Enhanced Three-factor Security Protocol for Consumer USB Mass Storage Devices. *IEEE Trans. Consum. Electron.* **2014**, *60*, 30–37.
37. Sankar, R.; Le, X.; Lee, S.; Wang, D. Protection of data confidentiality and patient privacy in medical sensor networks. In *Implantable Sensor Systems for Medical Applications*; Inmann, A., Hodgins, D., Eds.; Woodhead Publishing Series in Biomaterials; Woodhead Publishing: Sawston/Cambridge, UK, 2013; pp. 279–298.
38. Seo, S.C.; Han, D.G.; Kim, H.C.; Hong, S. TinyECCK: Efficient Elliptic Curve Cryptography Implementation over GF(2M) on 8-Bit Micaz Mote. *IEICE Trans. Inf. Syst.* **2008**, *E91-D*, 1338–1347. [CrossRef]