sensors

MDPI

Article

Efficient and Security Enhanced Anonymous Authentication with Key Agreement Scheme in Wireless Sensor Networks

Jaewook Jung, Jongho Moon, Donghoon Lee and Dongho Won *

Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 440-746, Korea; jwjung@security.re.kr (J.J.); jhmoon@security.re.kr (J.M.); dhlee@security.re.kr (D.L.)

* Correspondence: dhwon@security.re.kr; Tel.: +82-31-290-7213

Academic Editor: Kemal Akkaya
Received: 23 December 2016; Accepted: 17 March 2017; Published: 21 March 2017

Abstract: At present, users can utilize an authenticated key agreement protocol in a Wireless Sensor Network (WSN) to securely obtain desired information, and numerous studies have investigated authentication techniques to construct efficient, robust WSNs. Chang et al. recently presented an authenticated key agreement mechanism for WSNs and claimed that their authentication mechanism can both prevent various types of attacks, as well as preserve security properties. However, we have discovered that Chang et al's method possesses some security weaknesses. First, their mechanism cannot guarantee protection against a password guessing attack, user impersonation attack or session key compromise. Second, the mechanism results in a high load on the gateway node because the gateway node should always maintain the verifier tables. Third, there is no session key verification process in the authentication phase. To this end, we describe how the previously-stated weaknesses occur and propose a security-enhanced version for WSNs. We present a detailed analysis of the security and performance of our authenticated key agreement mechanism, which not only enhances security compared to that of related schemes, but also takes efficiency into consideration.

Keywords: wireless sensor networks; session key agreement; off-line password guessing attack; lightweight computation; formal proof

1. Introduction

Wireless Sensor Networks (WSNs) are distributed networks composed of tiny autonomous sensors capable of collecting information related to the environment or physical conditions of a target region [1]. WSNs can be implemented in various use cases—including military battlefields, healthcare services and smart grid networks—to provide convenience to users [2]. Figure 1 illustrates the WSN system architecture. As shown in Figure 1, WSN systems are comprised of three parties, including the user, the gateway nodes and the sensor nodes [1,2]. WSN is made of sensor nodes that are wirelessly connected to a gateway that is then connected to a user. On the other hand, in some WSNs, the sensor nodes can also be connected to each other in order to facilitate multi-hop wireless mesh networks.

Although users enjoy the simplicity and efficiency in WSNs, security has emerged as a major issue in both academia and industry [3]. Specifically, confidential information including the user's identity and password should not be exposed even if an unauthorized user eavesdrops on data packets transmitted in the WSN [4]. To guarantee reliability among the communicating parties, an authentication mechanism can afford confidentiality and integrity when users access WSNs [3,4]. At this point, in order to design a secure authentication mechanism for WSNs, the following security requirements should be commonly considered [5–13].

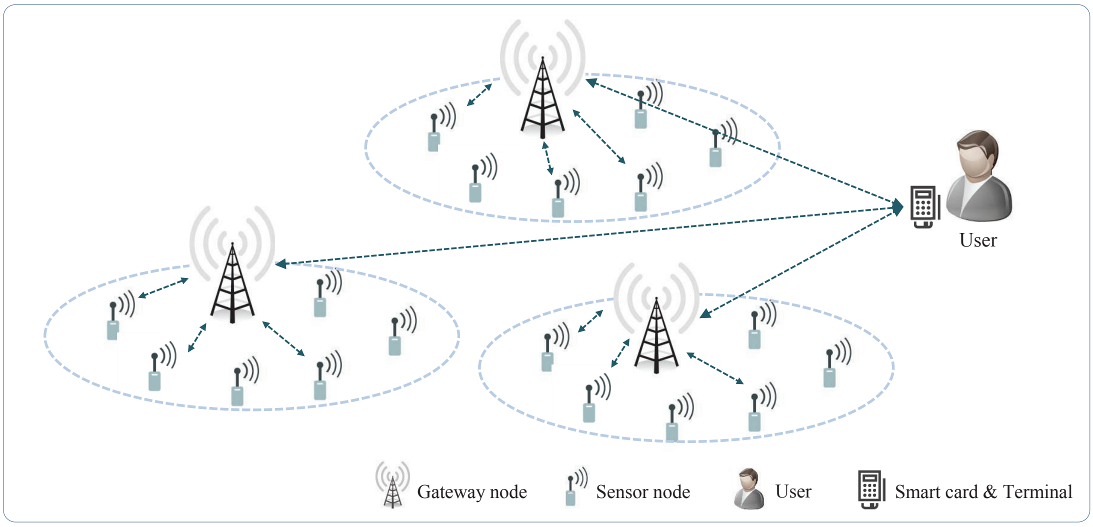

Figure 1. WSN system architecture.

- User anonymity: Even if an attacker extracts some information stored in the user's smart card or if it eavesdrops on the messages transmitted in the communication group, the user's identity should be protected.
- Mutual authentication: An authentication mechanism should execute several steps to achieve mutual authentication, which is to test all transmitted messages to judging the legitimacies.
- Session key agreement: After the mutual authentication process has completed, the session key should be securely assigned to communication parties on the network.
- Password verification process: If a user mistakenly enters an incorrect password in the login phase, the password should be promptly detected before performing the authentication phase.
- User friendliness: An authentication mechanism provides a password change procedure with which a user can freely update their password without communicating with the gateway node.
- Robustness: User authenticated key agreement schemes should withstand different types of attacks, such as off-line password guessing attacks, replay attacks, insider attacks and impersonation attacks.

Furthermore, the efficiency aspect should be considered when applying the authentication mechanism to the WSN environment because the sensor nodes are limited in terms of their computing resources and power [5]. In other words, when constructing an authentication mechanism for WSNs, a hash function-based method is recommended for use since it requires less computation overhead than public-key cryptosystems, such as RSA, elliptic curves cryptography (ECC) and El-gamal, all of which have a high computational overhead [6,7]. Therefore, the authentication protocol implemented for WSNs should be simple and efficient while also conforming to the required security.

1.1. Related Studies

In 2006, Wong et al. [8] first presented a lightweight user authentication protocol for WSNs. Their protocol improved the efficiency by only employing a one-way hash function and exclusive-OR operation. However, Das [9] pointed out that Wong et al.'s scheme [8] could not withstand many logged-in users with the same login identity attacks and stolen-verifier attacks. Das [9] then suggested an improved version that solved the flaws present in Wong et al.'s method. Unfortunately, Khan and Alghathbar [10] demonstrated in 2010 that Das's scheme [9] could not withstand a privileged-insider attack and gateway node bypass attack and proposed an enhanced new strategy. In the same year, Chen and Shih [11] also demonstrated that Das's scheme [9] overlooks parallel session attacks and cannot support a mutual authentication property. Chen and Shih [11] then proposed an enhanced version. In 2012, Vaidya et al. [12] pointed out that Das's scheme [9], Khan and Alghathbar's scheme [10] and Chen and Shih's scheme [11] contained the same vulnerabilities against a lost smart card attack and sensor node impersonation attack. To compensate for these defects, Vaidya et al. [12] suggested their own authentication scheme, arguing that it can withstand various attack types. However, Kim et al. [13] proved in 2014 that Vaidya et al.'s scheme [12] has some weaknesses, such as to user impersonation attacks and gateway node bypass attacks, and thus proposed an upgraded scheme. In 2015, Chang et al. [14] demonstrated that Kim et al.'s scheme [13] could not prevent an impersonation attack, lost smart card attack or man-in-the-middle attack, and it did not provide session key security. Chang et al. [14] then proposed an improved scheme. However, Park and Park [15] pointed out recently that Chang et al's scheme [14] still had some weaknesses, such as off-line password guessing attack, perfect forward secrecy problem and incorrectness of password change, and proposed an enhanced new version.

In particular, various cryptography techniques were employed in their protocols in order to improve the security for WSNs. Lee [16] and Kumari et al. [17] apply a chaotic map technique in their authentication mechanism. In 2015, Cheng et al. [18] presented an RSA-based authentication method for WSNs. In addition, Yeh et al. [19] proposed an authentication protocol based on elliptic curves cryptography (ECC) for WSNs. However, Han [20] pointed out that Yeh et al.'s scheme [19] could not achieve perfect forward secrecy and fails to provide mutual authentication. To address these weaknesses, Shi and Gong [21] presented a new authentication mechanism for WSNs using an ECC technique. However, Choi et al. [22] demonstrated that Shi and Gong's mechanism [21] could not satisfy security requirements because their scheme is unsafe against lost smart card attacks and does not provide session key security.

1.2. Motivations and Contributions

In 2015, Chang et al. [14] presented a two-factor user authenticated key agreement scheme for WSNs. They claimed that their scheme could resist an off-line password guessing attack and an impersonation attack, as well as provide session key security. However, we have discovered that Chang et al.'s scheme [14] comprises critical security weaknesses. Their scheme (i) still cannot guarantee protection against an off-line password guessing attack or user impersonation attack, (ii) fails to provide session key security, (iii) is faced with a scalability problem because the gateway nodes in their scheme always maintain verifier tables (iv) and cannot provide session key verification processes.

Our main contribution in this study is as follows. First, we concretely explain the weaknesses in Chang et al.'s scheme. Second, we propose a more developed authentication protocol for WSNs. Third, we show that the proposed mechanism satisfies various security requirements. Finally, we demonstrate that the proposed protocol has better performance than other related studies in terms of the computation cost and time consumption.

1.3. Preliminaries

In this subsection, we first introduce the biohash function [23], which is used in our proposed scheme. Then, we list the notations of Chang et al.'s scheme [14] and our proposed scheme in Table 1.

Table 1. Notations.

Value	Description
U_i	Remote user
S_j	Sensor node
GWN	Gateway node
ID_i, PW_i	Identity and password of U_i
Bio_i	Biometric information of U_i
PW_i^{new}	New password of U_i

Table 1. *Cont.*

Value	Description
u	Random number of U_i
ID_s	Identity of smart card
TID_i	Temporary identity for U_i's next login
SID_j	Identity of S_j
K	Secret key generated by the GWN
RN_r, RN_G, R	Random numbers
$h(\cdot)$	One-way hash function
$H(\cdot)$	Biohash function
$f(x, k)$	Pseudo-random function of variable s with key k
$X\|\|Y$	Concatenate operation
$\oplus$	XOR operation
T_1, T_2, T_3, T_4	Current time stamp values
K_S	Session key
ΔT	The maximum of the transmission delay time

1.3.1. Biohash Function

The user's biometric information is very sensitive data. Thus, when user identification is carried out using biometric data, a secure and sophisticated matching technique is required. In order to handle this concern, in 2004, Jin et al. [23] presented a fingerprint-based function to identify the user's legitimacy. The biohash technique employs the particular tokenized pseudo-random numbers to each of the users measuring biometric feature arbitrarily onto two-fold strands. Figure 2 describes the user recognition mechanism employing the user's biometric information and biohashing technique. When a device recognizes user's biometric template T, it transforms T into the form of feature vector and then transmits to transform function $H(\cdot)$. Transform function $H(\cdot)$ creates transformed template $H(T, K)$ by inputting the transmitted template T and random key K. Furthermore, the device creates biohash code, $H(Q, K)$ from the random key K and the stored value, which is a biometric query, in order to judge whether the user is registered or not, comparing to the new value, $H(T, K)$. The biohashing technique is also applied in our scheme, illustrated in Section 5. We use an input value Bio as a combination of the user's biometric information and a random key for convenience, like other authentication schemes [24–27] using the biohashing technique.

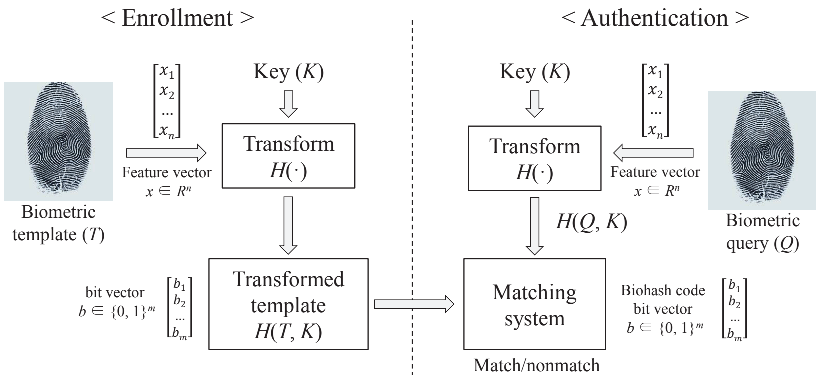

Figure 2. Authentication mechanism using the biohashing approach.

The biohash function $H(\cdot)$ is a one-way function with a feature that can reduce the probability of the denial of service. That is to say, the identical biometric information creates the identical value of $H(Bio)$, and it is impossible to calculate an input value Bio from the result value of $H(Bio)$. Until now, many authentication studies have been conducted [24–27] based on the biohashing technique. Our proposed scheme also adopts the user's biometric information applying a biohashing, and the details are given below in Section 5.

1.3.2. Scalability and Practicability in Terms of Authentication Using Biometric Information

The three-factor authentication protocol has been frequently employed in recent days, which complements the two-factor authentication protocol using the identity and password by adding biometric information. Basically, an authentication mechanism using biometric information requires a smart card terminal capable of recognizing a smart card and a device capable of recognizing the user's biometric (fingerprint) information. To reduce this inconvenience, Baratelli [28] and Kozlay [29] devised a new smart card-based fingerprint identification technology by adding a fingerprint recognition device in the smart card, and Clancy [30] proposed a self-fingerprint authentication technique using a smart card. In other words, a new device that combines a smart card terminal and a fingerprint reader has already been developed. In fact, authentication research does not really mean the inconvenience of fingerprint terminal devices and assumes that devices that can recognize both smart cards and fingerprints are used. In addition, a number of research works with respect to three-factor authentication protocol already [24–27] have applied user's biometric information.

First of all, the most important reason for using biometric information in the authentication mechanism is to increase the security of the protocol by preventing identity/password guessing attack. For this reason, our proposed scheme also uses the biometric information of the user, and it is confirmed that the proposed scheme is very safe. A detailed description of the protocol can be found in Section 4, and a security analysis can be found in Section 5.

1.3.3. Notations

The notations used in this paper are listed in Table 1.

1.4. Organization of the Paper

The remainder of this paper is structured as follows. In Section 2, we briefly explain Chang et al.'s authentication scheme. Section 3 demonstrates the vulnerabilities in Chang et al.'s scheme. A detailed explanation of our proposed scheme is provided in Section 4. In Section 5, we evaluate whether our proposed scheme can withstand various attacks. Further, we conduct a formal security proof using the random oracle model in Section 6. In Section 7, we analyze the performance of the proposed scheme, and in Section 8, we provide the conclusion to the paper.

2. Review of Chang et al.'s Scheme

In this section, we briefly review Chang et al.'s authenticated key agreement scheme [14] to then cryptanalyze their scheme. It is composed of four phases: registration, login, authentication and password change. In Chang et al.'s scheme [14], there are three communication parties, including a user U_i, a gateway node GWN and a sensor node S_j. We describe each phase in detail, and Table 1 shows the notations used in Chang et al.'s scheme.

2.1. Registration Phase

(1) U_i selects ID_i and PW_i, and U_i then generates a random number RN_r. U_i computes $HPW_i = h(PW_i||RN_r)$ and sends a registration request $\langle ID_i, HPW_i\rangle$ to GWN through a secure channel.

(2) GWN computes $HID_i = h(ID_i||K)$, $X_{S_i} = h(HID_i||K)$, $A_i = h(HPW_i||X_{S_i}) \oplus HID_i$, $B_i = h(HPW_i \oplus X_{S_i})$ and $C_i = X_{S_i} \oplus h(ID_s||HPW_i)$ and maintains $(TID_i, TID_i^{\circ}, HID_i)$ in its database for U_i, where $TID_i = RN_G$ and $TID_i^{\circ} = ''''$. GWN chooses a smart card and writes $\{ID_s, A_i, B_i, C_i, TID_i, h(\cdot)\}$ into the smart card's memory. Then, GWN sends the smart card to U_i through a secure channel.

(3) U_i computes $XPW_i = h(PW_i) \oplus RN_r$ and stores XPW_i in the smart card's memory. Finally, the smart card contains the information $\{ID_s, A_i, B_i, C_i, TID_i, h(\cdot), XPW_i\}$.

2.2. Login Phase

(1) U_i inserts U_i's smart card into a terminal and inputs the ID_i and PW_i. The smart card computes $RN_r^* = h(PW_i) \oplus XPW_i$, $HPW_i^* = h(PW_i||RN_r^*)$, $X_{S_i}^* = C_i \oplus h(ID_s||HPW_i^*)$, $B_i^* = h(HPW_i^* \oplus X_{S_i}^*)$ and compares B_i^* with the stored value B_i. If this condition is satisfied, the smart card acknowledges the legitimacy of U_i and proceeds with the next step. Otherwise, it terminates this phase.

(2) The smart card computes $k_i = h(X_{S_i}^*||T_1)$, $DID_i = h(HPW_i^*||X_{S_i}^*) \oplus k_i$ and $M_{U_i,G} = h(A_i||X_{S_i}^*||T_1)$.

(3) Finally, U_i sends a login request $\langle DID_i, M_{U_i,G}, T_1, TID_i \rangle$ to GWN through a public channel.

2.3. Authentication Phase

(1) GWN first checks the validity of the time stamp $|T_1' - T_1| < \Delta T$ and retrieves HID_i from TID_i corresponding to TID_i in its database. If GWN cannot search the TID_i, GWN retrieves HID_i from TID_i°. GWN, then computes $X_{S_i} = h(HID_i||K)$, $k_i = h(X_{S_i}||T_1)$, $X^* = DID_i \oplus k_i$, $M_{U_i,G}^* = h(X^* \oplus HID_i||X_{S_i}||T_i)$ and compares $M_{U_i,G}^*$ with the received value $M_{U_i,G}$. If this condition is satisfied, GWN acknowledges the legitimacy of the U_i and proceeds with the next step. Otherwise, it terminates this phase.

(2) GWN computes $X_{S_j} = h(SID_j||K)$, $M_{G,S_j} = h(DID_i||SID_j||X_{S_j}||T_2)$, then sends the message $\langle DID_i, M_{G,S_j}, T_2 \rangle$ to S_j through a public channel.

(3) S_j checks whether $|T_2' - T_2| < \Delta T$. S_j then computes $M_{G,S_j}^* = h(DID_i||SID_j||X_{S_j}^*||T_2)$ and compares M_{G,S_j}^* with the received value M_{G,S_j}. If this condition is satisfied, S_j believes that the GWN is authentic. Otherwise, it terminates this phase.

(4) S_j computes $k_j = h(X_{S_j}^*||T_3)$, $z_i = M_{G,S_j}^* \oplus k_j$, $K_S = f(DID_i, k_j)$, $M_{S_j,G} = h(z_i||X_{S_j}^*||T_3)$ and then sends the message $\langle M_{S_j,G}, T_3 \rangle$ to GWN through a public channel.

(5) GWN checks whether $|T_3' - T_3| < \Delta T$. GWN then computes $k_j = h(X_{S_j}||T_3)$, $z_i^* = M_{G,S_j} \oplus k_j$, $M_{S_j,G}^* = h(z_i^*||X_{S_j}||T_3)$ and compares $M_{S_j,G}^*$ with the received value $M_{S_j,G}$. If true, GWN believes that the S_j is authentic. Otherwise, GWN terminates this phase.

(6) GWN computes $M_{G,U_i} = h(DID_i||M_{U_i,G}^*||k_j||X_{S_i}||T_4)$, $y_i = k_j \oplus h(k_i)$, $TID_{i\_new} = h(HID_i||T_1)$ and updates (TID_i, TID_i°) as (TID_{i_new}, TID_i) in its database. GWN then sends the message $\langle y_i, M_{G,U_i}, T_4 \rangle$ to U_i through a public channel.

(7) U_i checks whether $|T_4' - T_4| \leq \Delta T$. U_i then computes $k_j^* = y_i \oplus h(k_i)$, $M_{G,U_i}^* = h(DID_i||M_{U_i,G}||k_j^*||X_{S_i}||T_4)$ and compares M_{G,U_i}^* with the received value M_{G,U_i}. If the verification does not hold, this phase is terminated. Otherwise, U_i believes that the GWN is authentic and computes the shared session key $K_S = f(DID_i, k_j^*)$.

(8) U_i computes $HID_i = A_i \oplus h(HPW_i^*||X_{S_i}^*)$ and $h(HID_i||T_1)$. Lastly, U_i updates TID_i as $h(HID_i||T_1)$ and successfully ends the authentication phase.

2.4. Password Change Phase

(1) U_i inserts U_i's smart card into a card reader and inputs ID_i, the old password PW_i and new password PW_i^{new}. The smart card computes $RN_r^* = h(PW_i) \oplus XPW_i$, $HPW_i^* = h(PW_i||RN_r^*)$, $X_{S_i}^* = C_i \oplus h(ID_s||HPW_i^*)$, $B_i^* = h(HPW_i^* \oplus X_{S_i}^*)$ and compares B_i^* with the stored value B_i. If this condition is not satisfied, it terminates this phase. Otherwise, the smart card proceeds with the next step.

(2) The smart card computes $HPW_i^{new} = h(PW_i^{new}||RN_r^*)$, $A_i^{new} = h(HPW_i^{new}||X_{S_i}) \oplus HID_i$, $B_i^{new} = h(HPW_i^{new} \oplus X_{S_i})$ $C_i^{new} = X_{S_i} \oplus h(ID_s||HPW_i^{new})$ and $XPW_i^{new} = h(PW_i^{new}) \oplus RN_r$.

(3) The smart card replaces the existing value (A_i, B_i, C_i, XPW_i) with the new values $(A_i^{new}, B_i^{new}, C_i^{new}, XPW_i^{new})$.

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

In this section, we show that Chang et al.'s scheme [14] possesses a number of security vulnerabilities. The following vulnerabilities are based on the two assumptions that

- An attacker can extract all parameters stored in the smart card by physically monitoring its power consumption [31].
- An attacker can eavesdrop or reform any messages in the public channel [32,33].

Under these two assumptions, the following problems have been found, and their detailed descriptions are given below.

3.1. Off-Line Password Guessing Attack

This attack attempts to input a password until the correct password is discovered because many users have a tendency to employ simple, brief passwords for the sake of convenience. For this reason, the authentication mechanism for all passwords should be invented to guarantee protection against a guessing attack. However, Chang et al.'s scheme [14] has a weakness in this situation, and we therefore propose a scenario for an off-line password-guessing attack. The following is a detailed description:

Step 1. An attacker extracts $\{ID_s, A_i, B_i, C_i, TID_i, h(\cdot), XPW_i\}$ from U_i's stolen smart card by physically monitoring its power consumption [31].
Step 2. The attacker collects a valid login request $\langle DID_i, M_{U_i,G}, T_1, TID_i\rangle$ from the previous session [32,33].
Step 3. The attacker selects a password candidate PW_i^*.
Step 4. The attacker computes $HPW_i^* = h(PW_i^*||h(PW_i^*) \oplus XPW_i)$ using the password candidate PW_i^*.
Step 5. The attacker then computes:

$$\begin{aligned} X_{S_i}^* &= C_i \oplus h(ID_s||HPW_i^*) \\ &= C_i \oplus h(ID_s||h(PW_i^*||h(PW_i^*) \oplus XPW_i)) \\ B_i^* &= h(HPW_i^* \oplus X_{S_i}^*) \\ &= h(h(PW_i^*||h(PW_i^*) \oplus XPW_i) \oplus C_i \oplus h(ID_s||h(PW_i^*||h(PW_i^*) \oplus XPW_i))) \end{aligned}$$

Step 6. The attacker repeats the steps above from 3–5 until the computed result B_i^* equals the breached secret B_i.
Step 7. If they correspond with each other, PW_i^* would be an accurate password. If not, the attacker repeats the above steps until the correct password is found.

Therefore, we can realize that Chang et al.'s scheme [14] is vulnerable to the off-line password guessing attack.

3.2. User Impersonation Attack

The security of the password-based authentication mechanism relies on the complexity of the password. Thus, if an attacker obtains a password, the attacker can pretend to be a legal user. Unfortunately, Chang et al.'s scheme [14] allows an attacker to impersonate a legal user if the attacker obtains the user's password PW_i through a guessing attack. The following is a detailed description of this scenario:

Step 1. An attacker extracts $\{ID_s, A_i, B_i, C_i, TID_i, h(\cdot), XPW_i\}$ from U_i's stolen smart card [31].
Step 2. The attacker collects a valid login request $\langle DID_i, M_{U_i,G}, T_1, TID_i\rangle$ from the previous session.
Step 3. The attacker obtains the user's PW_i through an off-line password guessing attack.

Step 4. The smart card computes:

$$\begin{aligned} DID_i^* &= h(HPW_i||X_{S_i}) \oplus k_i \\ &= h(HPW_i||X_{S_i}) \oplus h(X_{S_i}||T_1), \text{where } X_{S_i} = C_i \oplus h(ID_s||h(PW_i||h(PW_i) \oplus XPW_i)) \\ M_{U_i,G}^* &= h(A_i||X_{S_i}||T_1) \\ &= h(A_i||C_i \oplus h(ID_s||h(PW_i||h(PW_i) \oplus XPW_i))||T_1) \end{aligned}$$

Step 5. The attacker then sends a counterfeited login request $\langle DID_i^*, M_{U_i,G}^*, T_1, TID_i\rangle$ to GWN through a public channel.

Step 6. After receiving the $\langle DID_i^*, M_{U_i,G}^*, T_1, TID_i\rangle$, GWN computes $X_{S_i} = h(HID_i||K)$, $k_i = h(X_{S_i}||T_i)$, $X = DID_i^* \oplus k_i$ and $M_{U_i,G} = h(X \oplus HID_i||X_{S_i}||T_i)$.

Step 7. GWN compares the computed value $M_{U_i,G}$ with the received value $M_{U_i,G}^*$. Finally, GWN successfully finishes the verification process because $M_{U_i,G}^*$, which is computed by the attacker, is correctly equal to $M_{U_i,G}$, which is computed by the GWN.

Through the aforementioned descriptions, the attacker can successfully pass the checking process and be disguised as a legal user under Chang et al.'s scheme [14].

3.3. Session Key Compromise

In Chang et al.'s scheme [14], if an attacker knows U_i's password PW_i, the attacker can establish the session key $K_S = f(DID_i, k_j)$ shared between U_i and S_j. First, the attacker can extract $\{ID_s, A_i, B_i, C_i, TID_i, h(\cdot), XPW_i\}$ from U_i's stolen smart card. Second, the attacker can obtain DID_i and y_i after eavesdropping on the messages $\langle DID_i, M_{U_i,G}, T_1, TID_i\rangle$ and $\langle y_i, M_{G,U_i}, T_4\rangle$. Then, the attacker can try to compute $k_j = y_i \oplus h(k_i) = y_i \oplus h(h(C_i \oplus h(ID_s||h(PW_i||h(PW_i) \oplus XPW_i))||T_1))$ using the acquired PW_i, which has been previously compromised as in Section 3.1. With the combined $\{y_i, C_i, ID_s, PW_i, XPW_i, T_1\}$ values, the attacker can successfully construct the $K_S = f(DID_i, k_j)$.

3.4. Scalability Problem

In order to provide convenience, Chang et al. [14] suggested that the GWN maintains a verifier table in the database to save the information, such as the user's temporary identities (TID_i, TID_i°) and $HID_i = h(ID_i||K)$ value. Accordingly, the GWN should always need to retain each user's verifier table. However, the increased amount of user information that needs to be retained places greater burden on the GWN since the number of verifier tables will increase as the number of users' increases. Moreover, the use of the verifier table is inefficient in terms of the computation time since the changed values at each phase need to be updated in the verifier table.

3.5. Absence of a Session Key Verification Process

According to [34,35], the authenticated key agreement mechanism recommends a verification procedure to verify the coherence of the generated session keys between the communicating parties. In the authentication phase in Chang et al.'s scheme [14], U_i generates his/her own session key K_S after verifying the message $\langle y_i, M_{G,U_i}, T_4\rangle$ through $M_{G,U_i}^* \stackrel{?}{=} M_{G,U_i}$. However, in this case, because of the $M_{S_j,G} = h(z_i||X_{S_j}^*||T_3)$ has no information about the session key generated by S_j, and the U_i can hardly be sure whether a new generated session key K_S is precisely the same as the S_j's session key or not. Therefore, the following procedures [34] are required to ensure an accurate session key distribution between a U_i and a S_j: (1) after generating a session key, S_j sends a message, including information regarding the generated session key; (2) the U_i should guarantee the accuracy of the session key from the S_j, verifying the received message.

4. The Proposed Scheme

In this section, we suggest an improved version of the authenticated key agreement mechanism for the WSN in order to provide improved security by resolving Chang et al.'s [14] weaknesses. In the proposed scheme, to guarantee protection from the off-line password guessing attack, we employ biometrics information with the biohashing technique $H(\cdot)$ [23], as mentioned in Section 1.3. By preventing an off-line password guessing attack, our scheme can guarantee protection against an impersonation attack and against session key compromise. In addition, we remove the verifier table stored in *GWN* to increase efficiency. Our proposed scheme also consists of four phases: registration, login, authentication and password change. We describe each phase in detail, and Figures 3–5 describe our scheme. The notation used in the proposed scheme is displayed in Table 1.

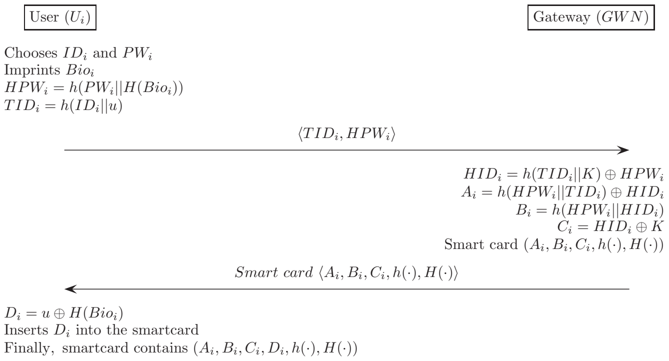
User (U_i)
Gateway (GWN)
Chooses ID_i and PW_i
Imprints Bio_i
$HPW_i = h(PW_i||H(Bio_i))$
$TID_i = h(ID_i||u)$
$\langle TID_i, HPW_i \rangle$
$HID_i = h(TID_i||K) \oplus HPW_i$
$A_i = h(HPW_i||TID_i) \oplus HID_i$
$B_i = h(HPW_i||HID_i)$
$C_i = HID_i \oplus K$
Smart card $(A_i, B_i, C_i, h(\cdot), H(\cdot))$
$Smart\ card\ \langle A_i, B_i, C_i, h(\cdot), H(\cdot) \rangle$
$D_i = u \oplus H(Bio_i)$
Inserts D_i into the smartcard
Finally, smartcard contains $(A_i, B_i, C_i, D_i, h(\cdot), H(\cdot))$

Figure 3. Registration phase for the proposed scheme.

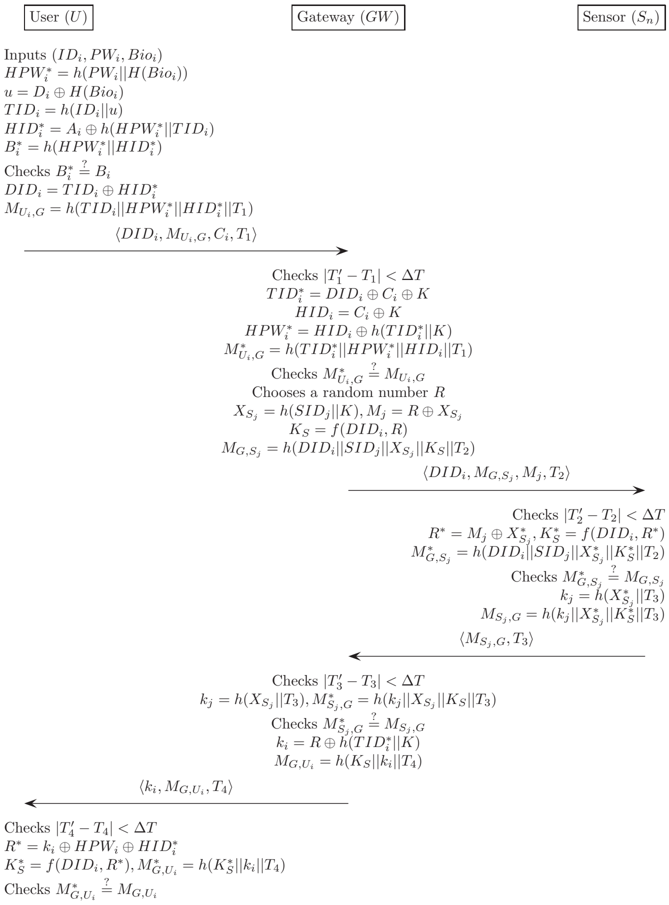
User (U)
Gateway (GW)
Sensor (S_n)
Inputs (ID_i, PW_i, Bio_i)
$HPW_i^* = h(PW_i||H(Bio_i))$
$u = D_i \oplus H(Bio_i)$
$TID_i = h(ID_i||u)$
$HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$
$B_i^* = h(HPW_i^*||HID_i^*)$
Checks $B_i^* \stackrel{?}{=} B_i$
$DID_i = TID_i \oplus HID_i^*$
$M_{U_i,G} = h(TID_i||HPW_i^*||HID_i^*||T_1)$
$\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$
Checks $|T_1' - T_1| < \Delta T$
$TID_i^* = DID_i \oplus C_i \oplus K$
$HID_i = C_i \oplus K$
$HPW_i^* = HID_i \oplus h(TID_i^*||K)$
$M_{U_i,G}^* = h(TID_i^*||HPW_i^*||HID_i||T_1)$
Checks $M_{U_i,G}^* \stackrel{?}{=} M_{U_i,G}$
Chooses a random number R
$X_{S_j} = h(SID_j||K), M_j = R \oplus X_{S_j}$
$K_S = f(DID_i, R)$
$M_{G,S_j} = h(DID_i||SID_j||X_{S_j}||K_S||T_2)$
$\langle DID_i, M_{G,S_j}, M_j, T_2 \rangle$
Checks $|T_2' - T_2| < \Delta T$
$R^* = M_j \oplus X_{S_j}^*, K_S^* = f(DID_i, R^*)$
$M_{G,S_j}^* = h(DID_i||SID_j||X_{S_j}^*||K_S^*||T_2)$
Checks $M_{G,S_j}^* \stackrel{?}{=} M_{G,S_j}$
$k_j = h(X_{S_j}^*||T_3)$
$M_{S_j,G} = h(k_j||X_{S_j}^*||K_S^*||T_3)$
$\langle M_{S_j,G}, T_3 \rangle$
Checks $|T_3' - T_3| < \Delta T$
$k_j = h(X_{S_j}||T_3), M_{S_j,G}^* = h(k_j||X_{S_j}||K_S||T_3)$
Checks $M_{S_j,G}^* \stackrel{?}{=} M_{S_j,G}$
$k_i = R \oplus h(TID_i^*||K)$
$M_{G,U_i} = h(K_S||k_i||T_4)$
$\langle k_i, M_{G,U_i}, T_4 \rangle$
Checks $|T_4' - T_4| < \Delta T$
$R^* = k_i \oplus HPW_i \oplus HID_i^*$
$K_S^* = f(DID_i, R^*), M_{G,U_i}^* = h(K_S^*||k_i||T_4)$
Checks $M_{G,U_i}^* \stackrel{?}{=} M_{G,U_i}$

Figure 4. Login and authentication phase for the proposed scheme.

$\boxed{\text{User } (U)}$

Inputs $(ID_i, PW_i, PW_i^{new}, Bio_i)$
$HPW_i^* = h(PW_i||H(Bio_i)), u = D_i \oplus H(Bio_i)$
$TID_i = h(ID_i||u)$
$HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$
$B_i^* = h(HPW_i^*||HID_i^*)$
Checks $B_i^* \overset{?}{=} B_i$
$HPW_i^{new} = h(PW_i^{new}||H(Bio_i))$
$A_i^{new} = h(HPW_i^{new}||TID_i) \oplus HID_i$
$B_i^{new} = h(HPW_i^{new}||HID_i)$
Replaces the existing values (A_i, B_i) with the new values (A_i^{new}, B_i^{new}), respectively.
Finally, the smart card contains the information $\{A_i^{new}, B_i^{new}, C_i, D_i, h(\cdot), H(\cdot)\}$

Figure 5. Password change phase for the proposed scheme.

4.1. Registration Phase

The registration phase begins when the U_i sends a request message for registration to GWN through a secure channel. The GWN then issues a smart card, including some information, and sends it to U_i. Meanwhile, S_j stores pre-defined values SID_j and $X_{S_j}^*$ in its memory, where $X_{S_j}^* = h(SID_j||K)$. The following describes this process in detail, and Figure 2 illustrates the registration phase for our proposed scheme.

(1) U_i selects ID_i and PW_i, and U_i then imprints his/her biometrics Bio_i. U_i computes $HPW_i = h(PW_i||H(Bio_i))$, generates a random number u and computes $TID_i = h(ID_i||u)$. U_i sends a registration request $\langle TID_i, HPW_i \rangle$ to GWN through a secure channel.
(2) 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 chooses a smart card and writes $\{A_i, B_i, C_i, h(\cdot), H(\cdot)\}$ into the smart card's memory. Then, GWN sends the smart card to U_i through a secure channel.
(3) Upon receiving the smart card, U_i computes $D_i = u \oplus H(Bio_i)$ and stores it in the smart card. Finally, the smart card contains the information $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$.

4.2. Login Phase

The login phase is executed whenever the U_i wants to gain access to WSN using his/her ID_i, PW_i and smart card. In this phase, U_i sends the login request to GWN. Figure 3 illustrates the login and authentication phase for our proposed scheme. The following describes this process in detail.

(1) U_i inserts U_i's smart card into a terminal and inputs the ID_i, PW_i and imprints biometric Bio_i. The smart card computes $HPW_i^* = h(PW_i||H(Bio_i))$, $u = D_i \oplus H(Bio_i)$, $TID_i = h(ID_i||u)$, $HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$, $B_i^* = h(HPW_i^*||HID_i^*)$ and compares B_i^* with the stored value B_i. If this condition is satisfied, the smart card acknowledges the legitimacy of the U_i and proceeds to the next step. Otherwise, it terminates this phase.
(2) The smart card computes $DID_i = TID_i \oplus HID_i^*$ and $M_{U_i,G} = h(TID_i||HPW_i^*||HID_i^*||T_1)$.
(3) Finally, U_i sends a login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$ to GWN through a public channel.

4.3. Authentication Phase

The authentication phase begins when GWN receives the login request from the U_i. This phase performs several steps to achieve mutual authentication, as well as a session key agreement between U_i, GWN and S_j involved within the WSN. The following describes this process in detail.

(1) GWN first checks the validity of the time stamp $|T_1' - T_1| < \Delta T$ and computes $TID_i^* = DID_i \oplus C_i \oplus K$, $HID_i = C_i \oplus K$ and $HPW_i^* = HID_i \oplus h(TID_i^*||K)$. GWN further computes $M_{U_i,G}^* = h(TID_i^*||HPW_i^*||HID_i||T_1)$ and compares it with the received value $M_{U_i,G}$. If this

condition is satisfied, GWN acknowledges the legitimacy of the U_i and proceeds with the next step. Otherwise, it terminates this phase.

(2) GWN generates 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) S_j checks whether $|T_2' - T_2| < \Delta T$ and computes $R^* = M_j \oplus X_{S_j}^*$ and $K_S^* = f(DID_i, R^*)$. S_j further computes $M_{G,S_j}^* = h(DID_i||SID_j||X_{S_j}^*||K_S^*||T_2)$ and compares it with the received value M_{G,S_j}. If this condition is satisfied, S_j believes that the GWN is authentic. Otherwise, it terminates this phase.

(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 then sends the message $\langle M_{S_j,G}, T_3 \rangle$ to GWN through a public channel.

(5) GWN checks whether $|T_3' - T_3| < \Delta T$. GWN computes $k_j = h(X_{S_j}||T_3)$, $M_{S_j,G}^* = h(k_j||X_{S_j}||K_S||T_3)$ and compares $M_{S_j,G}^*$ with the received value $M_{S_j,G}$. If true, GWN believes that the S_j is authentic. Otherwise, GWN terminates this phase.

(6) GWN computes $k_i = R \oplus h(TID_i^*||K)$ and $M_{G,U_i} = h(K_S||k_i||T_4)$. GWN then sends the message $\langle k_i, M_{G,U_i}, T_4 \rangle$ to U_i through a public channel.

(7) U_i checks whether $|T_4' - T_4| \leq \Delta T$ and computes $R^* = k_i \oplus HPW_i \oplus HID_i^*$ and $K_S^* = f(DID_i, R^*)$. U_i further computes $M_{G,U_i}^* = h(K_S^*||k_i||T_4)$ and compares it with the received value M_{G,U_i}. If this condition is not satisfied, this phase is terminated. Otherwise, U_i believes that the GWN is authentic and successfully ends the authentication phase

4.4. Password Change Phase

The password change phase begins when the U_i intends to change the original password PW_i to a new password PW_i^{new}. Figure 4 illustrates the password change phase for our proposed scheme. The following describes this process in detail.

(1) U_i inserts U_i's smart card into a terminal, inputs ID_i, PW_i, PW_i^{new} and then imprints biometric Bio_i. The smart card computes $HPW_i^* = h(PW_i||H(Bio_i))$, $u = D_i \oplus H(Bio_i)$, $TID_i = h(ID_i||u)$, $HID_i^* = A_i \oplus h(HPW_i^*||TID_i)$, $B_i^* = h(HPW_i^*||HID_i^*)$ and compares B_i^* with the stored value B_i. If this condition is not satisfied, it terminates this phase. Otherwise, the smart card proceeds with the next step.

(2) The smart card computes $HPW_i^{new} = h(PW_i^{new}||H(Bio_i))$, $A_i^{new} = h(HPW_i^{new}||TID_i) \oplus HID_i$ and $B_i^{new} = h(HPW_i^{new}||HID_i)$.

(3) The smart card replaces the existing values A_i and B_i with the new values A_i^{new} and B_i^{new}, respectively. Finally, the smart card contains the information $\{A_i^{new}, B_i^{new}, C_i, D_i, h(\cdot), H(\cdot)\}$.

5. Security Analysis and Proof of the Proposed Scheme

In this section, we first describe whether the proposed scheme can withstand various attacks and also satisfy the basic requirements. Moreover, we adopt Burrows–Abadi–Needham (BAN) logic [36] to prove that a session key can be correctly generated between U_i and S_j. The results are described as follows.

5.1. Informal Security Analysis of the Proposed Scheme

In this subsection, our proposed scheme is examined against various attacks and is evaluated according to the suitability of the basic requirements [5–13]. We also conduct a comparative analysis [10,12–15], which is illustrated in Table 2.

Table 2. Security comparison of our proposed scheme and other related schemes.

Features	Khan et al. [10]	Vaidya et al. [12]	Kim et al. [13]	Chang et al. [14]	Park et al. [15]	Our Scheme
User anonymity	×	×	√	√	√	√
Mutual authentication	×	√	√	√	√	√
Stolen smart card attack	×	×	×	×	√	√
Replay attack	√	√	√	√	√	√
Off-line PW guessing attack	×	√	√	×	√	√
U_i impersonation attack	×	×	√	×	√	√
S_j impersonation attack	×	√	×	√	√	√
Password verification	√	√	√	√	√	√
Session key verification	×	×	×	×	×	√
Privileged-insider attack	√	√	√	√	√	√
Session key security	×	×	×	×	√	√
Efficient password change	√	√	√	√	√	√
GWN bypass attack	×	×	√	√	√	√
Off-line ID guessing attack	×	×	√	√	√	√
No verifier table	√	√	√	×	×	√
Formal proof	×	√	×	√	√	√

- The proposed scheme preserves user anonymity:

User anonymity is a valuable property for the user authentication protocol because the exposure of a user's identity can allow an unauthorized party to track the user's login pattern. Suppose that the attacker has intercepted U_i's login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$ and extracted information $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$ in a stolen smart card [31]. The attacker may then try to compute ID_i through $h(ID_i||u) = DID_i \oplus HID_i$. However, it is impossible to know HID_i since HID_i consists of $(C_i \oplus K)$ and the secret key K is only known to GWN. In addition, u includes $H(Bio_i)$ information that is only known to U_i. Therefore, the attacker cannot acquire the user's ID_i.

- The proposed scheme achieves mutual authentication:

In the authentication phase of our scheme, U_i, GWN and S_j authenticate each other through some checking processes. In detail, GWN first verifies the login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$ by checking whether $M^*_{U_i,G} = M_{U_i,G}$. S_j also verifies the message $\langle DID_i, M_{G,S_j}, M_j, T_2 \rangle$ by checking whether $M^*_{G,S_j} = M_{G,S_j}$. In addition, GWN and U_i verify the messages $\langle M_{S_j,G}, T_3 \rangle$ and $\langle k_i, M_{G,U_i}, T_4 \rangle$ by checking $M^*_{S_j,G} \overset{?}{=} M_{S_j,G}$ and $M^*_{G,U_i} \overset{?}{=} M_{G,U_i}$, respectively. Thus, all transmitted messages in our scheme are successfully verified, and our scheme can achieve mutual authentication.

- The proposed scheme withstands stolen smart card attacks:

In our scheme, even if an attacker extracts secret values $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$ stored in a stolen smart card through the power consumption technique [31], the attack cannot lead to other malicious attacks. In order to obtain the ID_i, the attack has to know the secret key K and $H(Bio_i)$. However, it is impossible to know the K and $H(Bio_i)$. Therefore, if the attacker does not know the user's ID_i, the attacker cannot impersonate a legitimate user. Thus, our proposed scheme can withstand a stolen smart card attack.

- The proposed scheme withstands replay attacks:

In our scheme, all transmitted messages include current time stamp values, such as T_1, T_2, T_3 or T_4. Therefore, even if an attacker intercepts the login request message and tries to login GWN, the attacker cannot pass the time stamp checking process during the authentication phase. Thus, our proposed scheme can withstand a replay attack.

- The proposed scheme withstands off-line password guessing attacks:

An off-line password guessing attack occurs when an attacker attempts to guess a password and eventually finds the exact user's password in an off-line environment. This comes from the tendency that many users create simple and brief passwords for their personal convenience, which makes the attacker easily acquire the users' password by guessing the off-line password without a time limit [37]. For these reasons, the authentication schemes for all password-based users should be designed to prevent a guessing attack.

In our scheme, the attacker can obtain $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$ from the stolen smart card [31] and can intercept the login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$. Using these values, the attacker may try to guess the correct identity ID_i' and password PW_i' through $B_i' = h(h(PW_i'||H(Bio_i))||A_i \oplus h(h(PW_i'||H(Bio_i)||TID_i'))$ or $DID_i \oplus A_i = TID_i' \oplus h(h(PW_i'||H(Bio_i)||TID_i')$. However, without knowing Bio_i, the attacker cannot guess PW_i'. In addition, $H(Bio_i)$ is hashed biometric information, which is only known by U_i. Therefore, our proposed scheme is secure against off-line password guessing attacks.

- The proposed scheme withstands user impersonation attacks:

In order to impersonate a legitimate U_i, the attacker should modify the login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$ after obtaining the value of ID_i. However, as we mentioned above, it is impossible for an attacker to obtain the value of ID_i. Thus, the attacker fails to compute $DID_i = TID_i \oplus HID_i$ and cannot generate a sufficient login request to cheat GWN. Therefore, our proposed scheme can withstand a user impersonation attack.

- The proposed scheme withstands sensor node impersonation attacks with node capture:

Suppose that the attacker captures the sensor node S_j and extracts information $(SID_j, X_{S_j}^*)$ [13]. The attacker then tries to modify the message $\langle M_{S_j,G}, T_3 \rangle$ to impersonate a legitimate S_j. However, the attacker cannot generate a valid message because $X_{S_j}^*$ consists of $h(SID_j||K)$, and it is not feasible to obtain the K. Therefore, the attacker cannot impersonate a valid sensor node.

- The proposed scheme provides password verification process:

There is a possibility that a user inputs an incorrect password by mistake. However, for the password verification procedure, the incorrect password will be detected after performing the authentication phase. Our scheme considers this kind of inefficiency situation, verifying the correctness of password PW_i by checking the value B_i at the beginning of the login phase.

- The proposed scheme provides the session key verification process:

In our scheme, after generating a session key $K_S^* = f(DID_i, R^*)$, S_j computes $M_{S_j,G} = h(k_j||X_{S_j}^*||K_S^*||T_3)$ and sends the message $\langle M_{S_j,G}, T_3 \rangle$ to GWN. GWN then 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. After receiving the message, U_i computes $R^* = k_i \oplus HPW_i \oplus HID_i^*$, $K_S^* = f(DID_i, R^*)$ and $M_{G,U_i}^* = h(K_S^*||k_i||T_4)$ and then compares M_{G,U_i}^* with the received value M_{G,U_i}. Since M_{G,U_i} includes the information of the session key K_S, U_i may be sure that the K_S generated by S_j and GWN is accurate if the comparison result $M_{G,U_i}^* = M_{G,U_i}$ is correct. Therefore, our scheme provides a session key verification process.

- The proposed scheme withstands privileged-insider attacks:

An insider attack means that an insider can directly obtain the user's password from the server and can then access the user's account in another server by using the same password. During the registration phase of our scheme, PW_i is transmitted not as a revealed condition, but as a form of $HPW_i = h(PW_i||H(Bio_i))$ when U_i sends a registration request $\langle TID_i, HPW_i \rangle$ to GWN. Accordingly, the insider attacker in GWN cannot identify the U_i's PW_i. Thus, our scheme can withstand an insider attack.

- The proposed scheme provides session key security:

  In our scheme, in order to compromise the session key $K_S = f(DID_i, R)$, the attacker should know the random number R. Therefore, the attacker may try to obtain R through $R = M_j \oplus h(SID_j||K)$. However, it is impossible for an attacker to compute R because the attacker cannot obtain K, which is only known to GWN. Thus, our authentication scheme ensures session key security.

- The proposed scheme provides an efficient password change phase:

  In general, when a password change occurs, it is encouraged for the verification process to be carried out without any assistance from the GWN to ensure user friendliness and efficiency [24]. Our proposed scheme performs existing password checks in the self-verification process within the smart card. After checking the process through $B_i^* = B_i$, the computed values (A_i^{new}, B_i^{new}) from the new password PW_i^{new} will be switched with the existed values (A_i, B_i) in a convenient and efficient way.

- The proposed scheme withstands gateway node bypass attacks:

  During the authentication phase of our scheme, the attacker may try to construct the message $\langle DID_i, M_{G,S_j}, M_j, T_2 \rangle$ using the parameters $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$ stored in the stolen smart card [31] in order to impersonate a legitimate GWN. However, the attacker cannot compute $X_{S_j} = h(SID_j||K)$ because K is not public information. Thus, the attacker cannot construct a sufficient message to cheat S_j. Eventually, the attacker cannot impersonate a valid GWN.

- The proposed scheme withstands off-line identity guessing attacks:

  Suppose that the attacker extracts all of the secret information $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$ from the smart card and intercepts U_i's login request $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$. Using these values, the attacker may try to guess the correct identity ID_i' through $TID_i' = h(ID_i'||u)$, $HID_i' = DID_i \oplus TID_i'$, $K' = C_i \oplus HID'$, $HPW_i' = HID_i' \oplus h(TID_i'||K')$ and $B_i' = h(DID_i \oplus TID_i' \oplus h(TID_i'||K')||DID_i \oplus TID_i')$. However, in order to successfully guess the ID_i', the attacker should know the random number u. Even though the attacker knows the D_i, the attacker fails to compute $u = D_i \oplus H(Bio_i)$ because $H(Bio_i)$ is not public information. Therefore, our proposed scheme can withstand an off-line identity guessing attack.

5.2. Authentication Proof Using BAN Logic

In this subsection, we use BAN logic to verify the legitimacy of the session keys distributed to participants who communicate in the proposed scheme. BAN logic [36] is applied as a well-known formal logic to analyze the security of cryptographic protocols. The basic notation for BAN logic is as follows.

- $U \triangleleft C$: U sees condition C.
- $U \mid\equiv C$: Condition C is believed by U
- $\sharp(C)$: It makes a fresh C.
- $U \mid\sim C$: U expresses the condition C.
- $U \overset{K}{\longleftrightarrow} S$: U and S share a secret key K.
- $U \Rightarrow C$: Condition C is handled by U.
- $(C)_K$: Perform the hash operation on C using K.

BAN logic also offers five logic rules as follows.

- Rule 1. Message-meaning rule: $\frac{U|\equiv U \overset{K}{\leftrightarrow} S, U \triangleleft <C>_K}{U|\equiv S|\sim C}$: if U trusts that the key K is shared with S, U sees the C combined with K, then U trusts S once said C.
- Rule 2. Nonce-verification rule: $\frac{U|\equiv\#(C), U|\equiv S|\sim C}{U|\equiv S|\equiv C}$: if U trusts that C's freshness and U trusts S once said C, then U trusts that S trusts C.

- Rule 3. Believe rule: $\frac{U|\equiv C, U|\equiv M}{A|\equiv (C,M)}$: if U trusts C and M, (C, M) are also trusted by U.
- Rule 4. Freshness-conjuncatenation rule: $\frac{U|\equiv \#(C)}{A|\equiv \#(C,M)}$: if freshness of C is trusted by U, then U can trust the freshness of full condition.
- Rule 5. Jurisdiction rule: $\frac{U|\equiv S|\Rightarrow C, U|\equiv S|\equiv C}{U|\equiv C}$: if U trusts that S has jurisdiction over C, and U trusts that S trusts a condition C, then U also trusts C.

Through our analysis, we will intend to satisfy the following four goals.

- Goal 1: $U_i \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$
- Goal 2: $S_j \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$
- Goal 3: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$
- Goal 4: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$

Next, all transmitted messages can be transmuted into an idealized form as follows.

- Message 1: $U_i \rightarrow GWN : (ID_i, HPW_i, K, T_1)_{HID_i}$
- Message 2: $GWN \rightarrow S_j : (ID_i, SID_j, R, T_2)_{X_{S_j}}$
- Message 3: $S_j \rightarrow GWN : (ID_i, R, T_3)_{X_{S_j}}$
- Message 4: $GWN \rightarrow U_i : (ID_i, K, R, T_4)_{HID_i}$

In order to analyze our authentication mechanism, we define some assumptions as follows.

- A1: $GWN \mid\equiv \sharp(T_1)$
- A2: $S_j \mid\equiv \sharp(T_2)$
- A3: $GWN \mid\equiv \sharp(T_3)$
- A4: $U_i \mid\equiv \sharp(T_4)$
- A5: $GWN \mid\equiv (GWN \overset{X_{S_j}}{\longleftrightarrow} S_j)$
- A6: $S_j \mid\equiv (GWN \overset{X_{S_j}}{\longleftrightarrow} S_j)$
- A7: $U_i \mid\equiv (U_i \overset{HID_i}{\longleftrightarrow} GWN)$
- A8: $GWN \mid\equiv (U_i \overset{HID_i}{\longleftrightarrow} GWN)$
- A9: $U_i \mid\equiv S_j \Rightarrow (U_i \overset{K_S}{\longleftrightarrow} S_j)$
- A10: $S_j \mid\equiv U_i \Rightarrow (U_i \overset{K_S}{\longleftrightarrow} S_j)$

Now, we describe our main proof as follows. In order to describe our proof, we use predefined information, including five logic rules, four messages and ten assumptions.

- According to the Message 1, we could derive the following:
  V1: $GWN \triangleleft (ID_i, HPW_i, K, T_1)_{HID_i}$
- Based on Assumption A8 and Rule 1, we derive:
  V2: $GWN \mid\equiv U_i \mid\sim (ID_i, HPW_i, K, T_1)_{HID_i}$
- Based on Assumption A1 and Rule 4, we derive:
  V3: $GWN \mid\equiv \sharp(ID_i, HPW_i, K, T_1)_{HID_i}$
- Based on V2, V3 and Rule 2, we derive:
  V4: $GWN \mid\equiv U_i \mid\equiv (ID_i, HPW_i, K, T_1)_{HID_i}$
- According to Message 2, we derive:
  V5: $S_j \triangleleft (ID_i, SID_j, R, T_2)_{X_{S_j}}$

- Based on Assumption A6 and Rule 1, we derive:
  V6: $S_j \mid\equiv GWN \mid\sim (ID_i, SID_j, R, T_2)_{X_{S_j}}$
- Based on Assumption A2 and Rule 4, we derive:
  V7: $S_j \mid\equiv \sharp(ID_i, SID_j, R, T_2)_{X_{S_j}}$
- Based on V6, V7 and Rule 2, we derive:
  V8: $S_j \mid\equiv GWN \mid\equiv (ID_i, SID_j, R, T_2)_{X_{S_j}}$
- According to Message 3, we derive:
  V9: $GWN \triangleleft (ID_i, R, T_3)_{X_{S_j}}$
- Based on Assumption A5 and Rule 1, we derive:
  V10: $GWN \mid\equiv S_j \mid\sim (ID_i, R, T_3)_{X_{S_j}}$
- Based on Assumption A3 and Rule 4, we derive:
  V11: $GWN \mid\equiv \sharp(ID_i, R, T_3)_{X_{S_j}}$
- Based on V10, S11 and Rule 2, we derive:
  V12: $GWN \mid\equiv S_j \mid\equiv (ID_i, R, T_3)_{X_{S_j}}$
- According to Message 4, we derive:
  V13: $U_i \triangleleft (ID_i, K, R, T_4)_{HID_i}$
- Based on Assumption A7 and Rule 1, we derive:
  V14: $U_i \mid\equiv GWN \mid\sim (ID_i, K, R, T_4)_{HID_i}$
- Based on Assumption A4 and Rule 4, we derive:
  V15: $U_i \mid\equiv \sharp(ID_i, K, R, T_4)_{HID_i}$
- Based on V14, V15 and Rule 2, we derive:
  V16: $U_i \mid\equiv GWN \mid\equiv (ID_i, K, R, T_4)_{HID_i}$
- Based on V12, V16 and the session key $K_S = f(DID_i, R)$, we derive:
  V17: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$ (Goal 3)
- Based on V4, V8 and the session key $K_S = f(DID_i, k_i \oplus HPW_i \oplus HID_i)$, we derive:
  V18: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$ (Goal 4)
- Based on Assumption A9, V17 and Rule 5, we derive:
  V19: $U_i \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$ (Goal 1)
- Based on assumption A10, V18 and Rule 5, we derive:
  V20: $S_j \mid\equiv (U_i \overset{K_S}{\longleftrightarrow} S_j)$ (Goal 2)

The above description clearly shows that U_i, GWN and S_j achieve the mutual authentication property. In addition, based on Goal 1, Goal 2, Goal 3 and Goal 4, we can assure that the session key K_S is securely shared between them.

6. Formal Security Proof of the Proposed Scheme

In this section, we have demonstrated that the proposed scheme is secure through a formal proof using the random oracle model. First, we specify a cryptographic one-way hash function as follows.

Definition 1. *A hash function $f : \{0,1\}^* \to \{0,1\}^n$ is a one-direction function [38,39] that takes the input $x \in \{0,1\}^*$ of arbitrary length and outputs a bit string with a fixed-length $f(x) \in \{0,1\}^n$, which is referred to as the "message digest" or "hash value". When using cryptographic hash functions, the following three common levels of security must be considered:*

- It is impossible to acquire the input x under the conditions of the hash value $y = h(x)$ and the given hash function $h(\cdot)$.
- It is impossible to acquire another input x', when given the input x and $f(x') = f(x)$.
- It is impossible to acquire the inputs (x, x'), where $x \neq x'$, when given $f(x) = f(x')$.

Reveal: Given the hash result $y = h(x)$, this random oracle will unconditionally output the input x.

Theorem 1. *A one-way hash function $h(\cdot)$ is assumed to operate like an oracle. Under this assumption, our proposed mechanism is provably secure against an attacker $\mathcal{A}$ to protect U_i's personal information, such as identity ID_i, password PW_i, biometrics Bio_i and the GWN's secret key K.*

Proof. A similar method as that used in [26] is applied in our authentication mechanism to formally verify the security. For the proof, we assume that an attacker $\mathcal{A}$ is able to derive U_i's identity ID_i, password PW_i, biometrics Bio_i and the GWN's secret key K. For this, $\mathcal{A}$ runs the experimental algorithm that is shown in Algorithm 1, $EXP1^{AUAKAS}_{HASH,A}$ for our anonymous user authentication with key agreement scheme (AUAKAS). We define the success probability for $EXP1^{AUAKAS}_{HASH,A}$ as $Success1^{ABUAKAS}_{HASH,A} = |Pr[EXP1^{ABUAKAS}_{HASH,A} = 1] - 1|$, where $Pr(\cdot)$ means the probability of $EXP1^{AUAKAS}_{HASH,A}$. The advantage function for this experiment becomes $Adv^{ABUAKAS}_{HASH,A}(t, q_R) = max_A\{Success1^{ABUAKAS}_{HASH,A}\}$ in which the maximum is determined by three factors: all of $\mathcal{A}$, the execution time t and the number of queries q_R derived from the Reveal oracle. If attacker $\mathcal{A}$ is assumed to be able to resolve the hash function problem, $\mathcal{A}$ could directly obtain U_i's identity ID_i, password PW_i, biometrics Bio_i and the GWN's secret key K. Refer to the attack experiment described in Algorithm 1. In this case, $\mathcal{A}$ will discover the complete connections between U_i and GWN. However, it is computationally infeasible to invert a one-way hash function $h(\cdot)$, i.e., $Adv^{AUAKAS}_{HASH,A}(t) \leq \epsilon$, $\forall \epsilon > 0$. Then, we have $Adv^{AUAKAS}_{HASH,A}(t, q_R) \leq \epsilon$, since $Adv^{AUAKAS}_{HASH,A}(t, q_R)$ depends on $Adv^{AUAKAS}_{HASH,A}(t)$. Therefore, our proposed scheme is provably secure against the attacker $\mathcal{A}$ for deriving ID_i, PW_i, Bio_i and K. □

Algorithm 1: Algorithm $EXP1^{AUAKAS}_{HASH,A}$.

```
Eavesdrop login request message $\langle DID_i, M_{U_i,G}, C_i, T_1 \rangle$ during the login phase.
Call the Reveal oracle. Let $(TID'_i, HPW'_i, HID'_i, T'_1) \leftarrow Reveal(M_{U_i,G})$
Call the Reveal oracle. Let $(ID'_i, u') \leftarrow Reveal(TID'_i)$
Computes $DID'_i = h(ID'_i||u') \oplus HID'_i$
if$(DID'_i = DID_i)$ then
    Accepts $ID'_i$ as the correct $ID_i$ of user $U_i$
    Call the Reveal oracle. Let $(PW'_i, Bio'_i) \leftarrow Reveal(HPW'_i)$
    Computes $M'_{U_i,G} = h(TID'_i||h(PW'_i||Bio'_i)||HID'_i||T'_1)$
    if$(M'_{U_i,G} = M_{U_i,G})$ then
        Accepts $Bio'_i$ and $PW'_i$ as the correct $Bio_i$ and $PW_i$ of user $U_i$
        Call the Reveal oracle. Let $(HID''_i, K') \leftarrow Reveal(C_i)$
        if$(HID''_i = HID'_i)$ then
            Accept $K'$ as the correct secret key $K$ of gateway node $GWN$
            return 1 (Success)
        else
            return 0
        end if
    else
        return 0
    end if
else
    return 0
end if
```

Theorem 2. *The one-way hash function $h(\cdot)$ is assumed to perform as an oracle, and the smart card for U_i is stolen by an adversary $\mathcal{A}$. Under these assumptions, our proposed mechanism is secure against an adversary $\mathcal{A}$ to derive the password PW_i of a user U_i.*

Proof. We assume that an attacker $\mathcal{A}$ is able to derive the U_i's password PW_i after extracting the parameters $\{A_i, B_i, C_i, h(\cdot), H(\cdot)\}$ stored in the smart card by physically monitoring its power consumption [31]. $\mathcal{A}$ then runs the experimental algorithm $EXP2^{AUAKAS}_{HASH,A}$ that is shown in Algorithm 2. We define the success probability for $EXP2^{AUAKAS}_{HASH,A}$ as $Success2^{ABUAKAS}_{HASH,A} = |Pr[EXP2^{ABUAKAS}_{HASH,A} = 1] - 1|$, where $Pr(\cdot)$ means the probability of $EXP2^{AUAKAS}_{HASH,A}$. The advantage function for this experiment becomes $Adv2^{ABUAKAS}_{HASH,A}(t_2, q_R) = max_A\{Success2^{ABUAKAS}_{HASH,A}\}$ in which the maximum is determined by three factors: all of $\mathcal{A}$, the execution time t_2 and the number of queries q_R derived from the Reveal oracle. If $Adv2^{AUAKAS}_{HASH,A}(t_2) \leq \epsilon, \forall \epsilon > 0$, our scheme is provably secure against the attacker $\mathcal{A}$ to derive PW_i. According to the attack experiment described in Algorithm 2, $\mathcal{A}$ could obtain U_i's password PW_i if $\mathcal{A}$ is able to resolve the hash function problem. However, as shown in Definition 1, it is computationally infeasible to invert a one-way hash function $h(\cdot)$. Then, we have $Adv2^{AUAKAS}_{HASH,A}(t_2, q_R) \leq \epsilon$, since $Adv2^{AUAKAS}_{HASH,A}(t_2, q_R)$ depends on $Adv2^{AUAKAS}_{HASH,A}(t_2)$. As a result, the proposed scheme is provably secure against attacker $\mathcal{A}$ to derive PW_i even if the smart card is stolen by $\mathcal{A}$. □

Algorithm 2: Algorithm $EXP2^{AUAKAS}_{HASH,A}$.

```
Extract the information $\{A_i, B_i, C_i, D_i, h(\cdot), H(\cdot)\}$ stored in the smart card
  through physically monitoring its power consumption [31].
Call the Reveal oracle. Let $(HPW_i', TID_i', HID_i') \leftarrow Reveal(A_i)$
Call the Reveal oracle. Let $(PW_i', Bio_i') \leftarrow Reveal(HPW_i')$
Computes $HID_i' = A_i \oplus h(h(PW_i'||Bio_i')||TID_i')$
Computes $B_i' = h(HPW_i'||HID_i') = h(h(PW_i'||Bio_i')||A_i \oplus h(h(PW_i'||Bio_i')||TID_i'))$
if$(B_i' = B_i)$ then
    Accepts $PW_i'$ as the correct $PW_i$ of user $U_i$
    return 1 (Success)
else
    return 0
end if
```

7. Performance Analysis of the Proposed Scheme

In this section, we performed a comparison of the computational costs and execution time for the proposed scheme relative to other, related schemes [10,12–15]. In general, the computational cost is examined based on the respective operations in the authentication protocol. Accordingly, this analysis of the computational cost concentrates on the operations that are conducted by the participant, such as U_i, GWN and S_j. To evaluate the computational costs, we define the following computational parameters.

- T_H: the time to execute a one-way hash/pseudo-random function/biohash function.
- T_X: the time to execute a XOR operation.
- T_E: the time to execute a ECC multiplication.
- T_F: the time to execute a fuzzy extractor.

Table 3 provides a summary of the comparison of the computational overhead of the related schemes [10,12–15]. The results show that Khan and Alghathbar's scheme [10], Vaidya et al.'s scheme [12], Kim et al.'s scheme [13], Change et al.'s scheme [14], Park and Park's scheme [15] and the proposed scheme require total computational overheads of $16T_H + 6T_X$, $30T_H + 24T_X$, $37T_H + 30T_X$, $37T_H + 21T_X$, $39T_H + 19T_X + 3T_F + 4T_E$ and $34T_H + 15T_X$, respectively.

Table 3. Comparison of the computational cost between our scheme and other hash-based schemes.

Phases		Khan et al. [10]	Vaidya et al. [12]	Kim et al. [13]	Chang et al. [14]	Park et al. [15]	Proposed Scheme
Registration	U_i	$1T_H$	$1T_H$	$2T_H + 1T_X$	$2T_H + 1T_X$	$1T_H + 1T_F$	$3T_H$
	GWN	$2T_H + 1T_X$	$4T_H + 2T_X$	$6T_H + 3T_X$	$5T_H + 3T_X$	$5T_H + 3T_X$	$3T_H + 3T_X$
	S_j	–	–	–	–	–	–
Login	U_i	$3T_H + 1T_X$	$6T_H + 4T_X$	$7T_H + 5T_X$	$7T_H + 4T_X$	$6T_H + 3T_X$ $+1T_F + 1T_E$	$6T_H + 2T_X$
	GWN	–	–	–	–	–	–
	S_j	–	–	–	–	–	–
Authen tication	U_i	–	$2T_H + 3T_X$	$2T_H + 4T_X$	$4T_H + 2T_X$	$4T_H + 2T_X + 1T_E$	$2T_H + 2T_X$
	GWN	$5T_H + 2T_X$	$6T_H + 6T_X$	$8T_H + 8T_X$	$6T_H + 4T_X$	$11T_H + 4T_X$	$8T_H + 5T_X$
	S_j	$2T_H$	$3T_H + 2T_X$	$3T_H + 2T_X$	$4T_H + 1T_X$	$4T_H + 1T_X + 2T_E$	$4T_H + 1T_X$
Password change	U_i	$3T_H + 2T_X$	$8T_H + 6T_X$	$9T_H + 7T_X$	$9T_H + 6T_X$	$8T_H + 6T_X + 1T_F$	$8T_H + 2T_X$
	GWN	–	–	–	–	–	–
	S_j	–	–	–	–	–	–
Total cost		$16T_H + 6T_X$	$30T_H + 24T_X$	$37T_H + 30T_X$	$37T_H + 21T_X$	$39T_H + 19T_X$ $+3T_F + 4T_E$	$34T_H + 15T_X$
Execution time		≈0.008 s	≈0.015 s	≈0.0185 s	≈0.0185 s	≈0.4605 s	≈0.017 s

Based on the total cost results in Table 3, we have performed an experiment on the execution time to obtain an objective comparison between our scheme and other related schemes [10,12–15]. The following methods are generally used to measure the execution time for the authentication protocol: (i) determine computational overhead; (ii) measure the execution time of the cryptographic operations used in the protocol; and (iii) substitute the measured time obtained by (ii) into (i). We have measured the execution times using these measurement methods, and the results are shown in the execution time field of Table 3.

The results of the simulation in Li et al.'s and Wazid et al.'s research [40,41] show that the actual execution time for the cryptographic one-way hash function T_H and ECC multiplication T_E is 0.0005 s and 0.063 s, respectively. In addition, according to [41], the execution time of the fuzzy extractor operation T_F is almost the same as the ECC multiplication operation T_E. Thus, we assumed that the time consumption of these two operations is the same. On the other hand, XOR operation T_X is not considered in our measurement because the execution time of the XOR operation T_X is extremely short. Based on the $T_H \approx 0.0005$, $T_E \approx 0.063$, $T_F \approx 0.063$ and the total computation cost, we finally analyze the execution time. As shown in Table 3, we observed that the execution time of our proposed scheme is of only 0.017 s ($34T_H \approx 34 \times 0.0005$ s), so it can be considered as a negligible significance. In contrast, the execution times of Kim et al.'s scheme [13], Chang et al.'s scheme [14] and Park and Park's scheme [15] are 0.0185 s ($37T_H \approx 37 \times 0.0005$ s), 0.0185 s ($37T_H \approx 37 \times 0.0005$ s) and 0.4605 s ($39T_H + 3T_F + 4T_E \approx 39 \times 0.0005$ s + 7×0.063 s), respectively. Therefore, our scheme turned out to have a slightly better efficiency than these schemes [13–15]. Even if our scheme requires slightly more computation time than Khan and Alghathbar's scheme [10] and Vaidya et al.'s scheme [12], this is acceptable because our scheme has more effective security features and a higher security level, as shown in Table 2.

8. Conclusions

In this paper, we have demonstrated that Chang et al.'s scheme has a number of critical weaknesses, and we propose an authentication mechanism with enhanced security to overcome these weaknesses. Our proposed scheme has been thoroughly verified in terms of its variety of security features, and the proof result demonstrates that our scheme can guarantee protection against various types of attacks, even if the smart card is stolen by an attacker. In addition, a performance comparison for the proposed scheme in relation to the schemes proposed in other studies was carried out, and we consider that our proposed scheme has sufficient efficiency for WSNs.

Acknowledgments: This work was supported by Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R0126-15-1111, The Development of Risk-based Authentication·Access Control Platform and Compliance Technique for Cloud Security).

Author Contributions: J.J., J.M. and D.L. conceived of and designed the experiments. J.J. performed the experiments. J.J. and Y.M. analyzed the data. J.J. and D.W. wrote the paper.

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

References

1. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330.
2. Chong, C.Y.; Kumar, S.P. Sensor networks: Evolution, opportunities, and challenges. *Proc. IEEE* **2003**, *91*, 1247–1256.
3. Choi, Y.; Nam, J.; Lee, D.; Kim, J.; Jung, J.; Won, D. Security Enhanced Anonymous Multi-Server Authenticated Key Agreement Scheme Using Smart Cards and Biometrics. *Sci. World. J.* **2014**, *2014*, 281305.
4. Nam, J.; Kim, M.; Paik, J.; Lee, Y.; Won, D. A provably-secure ECC-based authentication scheme for wireless sensor networks. *Sensors* **2014**, *14*, 21023–21044.
5. Claycomb, W.R.; Shin, D. A novel node level security policy framework for wireless sensor networks. *J. Netw. Comput. Appl.* **2011**, *34*, 418–428.
6. Watro, R.; Kong, D.; Cuti, S.F.; Gardiner, C.; Lynn, C.; Kruus, P. TinyPK: Securing sensor networks with public key technology. In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington, DC, USA, 25 October 2004; pp. 59–64.
7. Hwang, M.S.; Li, L.H. A new remote user authentication scheme using smart cards. *IEEE Trans. Consum. Electron.* **2000**, *46*, 28–30.
8. 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, pp. 1–9.
9. Das, M.L. Two-factor user authentication scheme in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
10. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. *Sensors* **2010**, *10*, 2450–2459.
11. Chen, T.H.; Shih, W.K. A Robust Mutual Authentication Protocol for Wireless Sensor Networks. *ETRI J.* **2010**, *32*, 704–712.
12. Vaidya, B.; Makrakis, D.; Mouftah, H. Two-factor mutual authentication with key agreement in wireless sensor networks. *Secur. Commun. Netw.* **2012**, *9*, 171–183 .
13. 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.
14. 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.
15. Park, Y.; Park, Y. Three-Factor User Authentication and Key Agreement Using Elliptic Curve Cryptosystem in Wireless Sensor Networks. *Sensors* **2016**, *16*, 2123.
16. Lee, T.F. Efficient and Secure Temporal Credential-Based Authenticated Key Agreement Using Extended Chaotic Maps for Wireless Sensor Networks. *Sensors* **2015**, *15*, 14960–14980.
17. Kumari, S.; Li, X.; Wu, F.; Das, A.K.; Arshad, H.; Khan, M.K. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. *Future Gen. Comput. Syst.* **2016**, *63*, 56–75.
18. Cheng, C.Y.; Lin, I.C.; Huang, S.Y. An RSA-like scheme for multiuser broadcast authentication in wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *2015*, 200.
19. 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.
20. Han, W. Weakness of a Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. Available online: http://eprint.iacr.org/2011/293 (accessed on 27 June 2011).

21. Shi, W.; Gong, P. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*, 730831.
22. Choi, Y.; Lee, D.; Kim, J.; Jung, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2014**, *14*, 10081–10106.
23. Jin, A.T.B.; Ling, D.N.C.; Goh, A. Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recogn.* **2004**, *37*, 2245–2255.
24. Moon, J.; Choi, Y.; Kim, J.; Won, D. An Improvement of Robust and Efficient Biometrics Based Password Authentication Scheme for Telecare Medicine Information Systems Using Extended Chaotic Maps. *J. Med. Syst.* **2016**, *40*, 1–11.
25. Mishra, D.; Das, A.K.; Mukhopadhyay, S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Syst. Appl.* **2014**, *41*, 8129–8143.
26. Moon, J.; Choi, Y.; Jung, J.; Won, D. An Improvement of Robust Biometrics-Based Authentication and Key Agreement Scheme for Multi-Server Environments Using Smart Cards. *PLoS ONE* **2015**, *10*, e0145263.
27. Das, A.K.; Goswami, A. A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care. *J. Med. Syst.* **2013**, *37*, 1–16.
28. Baratelli, P.J. Smart Card with Integrated Fingerprint Reader. U.S. Patent No. 6325285 B1, 4 December 2001.
29. Kozlay, D. Design & Method for Manufacturing Low-Cost Smartcards with Embedded Fingerprint Authentication System Modules. U.S. Patent No. US20050139685 A1, 6 July 2004.
30. Clancy, T.C.; Kiyavash, N.; Lin, D.J. Secure smartcard-based fingerprint authentication. In Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, Berkley, CA, USA, 2–8 November 2003; pp. 45–52.
31. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Advances in Cryptology-CRYPTO'99, LNCS, Santa Barbara, CA, USA, 15–19 December 1999; Volume 1666, pp. 388–397.
32. Chung, Y.; Choi, S.; Lee, Y.; Park, N.; Won, D. An Enhanced Lightweight Anonymous Authentication Scheme for a Scalable Localization Roaming Service in Wireless Sensor Networks. *Sensors* **2016**, *16*, 1653.
33. Kang, D.; Jung, J.; Mun, J.; Lee, D.; Choi, Y.; Won, D. Efficient and robust user authentication scheme that achieve user anonymity with a Markov chain. *Secur. Commun. Netw.* **2016**, *9*, 1462–1476.
34. Blake-Wilson, S.; Johnson, D.; Menezes, A. Key agreement protocols and their security analysis. In Proceedings of the IMA International Conference on Cryptography and Coding, Cirencester, UK, 17–19 December 1997; pp. 30–45.
35. Islam, S.H.; Khan, M.K.; Li, X. Security analysis and improvement of 'a more secure anonymous user authentication scheme for the integrated EPR information system'. *PLoS ONE* **2015**, *10*, e0131368.
36. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. Lond. A. Math. Phys. Sci.* **1989**, *426*, 233–271.
37. Ma, C.G.; Wang, D.; Zhao, S.D. Security flaws in two improved remote user authentication schemes using smart cards. *Int. J. Commun. Syst.* **2014**, *27*, 2215–2227.
38. Stallings W. *Cryptography and Network Security: Principles and Practices*; Pearson Education: Upper Saddle River, NJ, USA, 2006.
39. FIPS P. 180-1. *Secure Hash Standard*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 1995.
40. Li, C.T.; Hwang, M.S.; Chu, Y.P. A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks. *Comput. Commun.* **2008**, *31*, 2803–2814.
41. Wazid, M.; Das, A.K.; Kumari, 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.