sensors

MDPI

Article

Three-Factor User Authentication and Key Agreement Using Elliptic Curve Cryptosystem in Wireless Sensor Networks

YoHan Park and YoungHo Park *

School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; hanny12@ee.knu.ac.kr

* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Academic Editor: Kemal Akkaya

Received: 19 October 2016; Accepted: 12 December 2016; Published: 14 December 2016

Abstract: Secure communication is a significant issue in wireless sensor networks. User authentication and key agreement are essential for providing a secure system, especially in user-oriented mobile services. It is also necessary to protect the identity of each individual in wireless environments to avoid personal privacy concerns. Many authentication and key agreement schemes utilize a smart card in addition to a password to support security functionalities. However, these schemes often fail to provide security along with privacy. In 2015, Chang et al. analyzed the security vulnerabilities of previous schemes and presented the two-factor authentication scheme that provided user privacy by using dynamic identities. However, when we cryptanalyzed Chang et al.'s scheme, we found that it does not provide sufficient security for wireless sensor networks and fails to provide accurate password updates. This paper proposes a security-enhanced authentication and key agreement scheme to overcome these security weaknesses using biometric information and an elliptic curve cryptosystem. We analyze the security of the proposed scheme against various attacks and check its viability in the mobile environment.

Keywords: user authentication; key agreement; biometric information; elliptic curve cryptosystem; wireless sensor networks

1. Introduction

Wireless sensor networks (WSNs) are ad hoc networks composed of a number of sensor nodes with limited power, computation, storage and communication capabilities [1]. They provide effective solutions to a wide array of monitoring problems in various environments, such as battlefields, healthcare services and the smart grid [2]. Recently, sensor-attached things that communicate with neighboring things are enabling the development of the Internet of Things (IoT) environment [3]. For these reasons, WSNs have gained wide attention, in both the academic and industrial fields. However, the issue of securing and authenticating communication is problematic, because the nodes are vulnerable to attacks and do not have enough capacity for the secure storage of keys [4–6]. To solve these security issues, authentication and key agreement schemes using two-factor security, passwords and smart cards have attracted attention and have been studied widely in an effort to guarantee secure communication [7–14]. Unfortunately, many of them still suffer from various attacks and do not provide secure communication.

Several authentication and key agreement schemes for WSNs have been proposed. In 2010, Das [8] proposed a two-factor user authentication protocol for WSNs. He insisted the scheme withstood various attacks from users with the same identity, as well as from stolen-verifier attacks. However, He et al. [9], Khan and Alghathbar [10] and Chen and Shih [11] pointed out that Das's scheme was vulnerable to insider and impersonation attacks, gateway node bypassing attacks and privileged-insider attacks and did not provide mutual authentication. Subsequently, each proposed their own authentication scheme to provide secure user authentication in WSNs. In 2012, Vaidya et al. [12] demonstrated that Das's scheme [8], Khan and Alghathbar's scheme [10] and Chen and Shih's scheme [11] had security problems and that none of them provided key agreement. Vaidya et al. proposed a two-factor mutual user authentication scheme with key agreement for WSNs. In 2014, Kim et al. [13] presented that both gateway node bypassing attacks and user impersonation attacks were possible in Vaidya et al.'s scheme [12]. They proposed an authentication and key agreement scheme that resisted user impersonation and gateway node bypassing attacks. However, in 2015, Chang et al. [14] analyzed Kim et al.'s scheme [13] and found it had security vulnerabilities in the following areas: impersonation attacks, lost smart card attacks, man-in-the-middle attacks, violation of session key security and failure to protect user privacy. To solve these problems, Chang et al. [14] proposed a scheme that provided user privacy by using dynamic identities and provided better security functionality than Kim et al.'s scheme. However, we point out that Chang et al.'s scheme does not withstand several types of attacks and fails to provide a password update.

Recently, to improve the security of two-factor authentication schemes that are vulnerable to guessing attacks and subject to inefficient password change policies in WSNs, biometric-based user authentication schemes, combined with smart cards and passwords, have drawn considerable attention in research [15–19]. Biometric-based user authentication in the WSN becomes inherently more reliable and secure than traditional two-factor user authentication schemes [20]. Several advantages can be derived from the use of biometric keys over traditional passwords because they cannot be lost; they are unforgettable, difficult to copy, hard to forge and difficult to break. Therefore, biometric-based user authentication is considered to be more secure and reliable than conventional authentication schemes [20].

In this paper, we cryptanalyze Chang et al.'s scheme [14] and demonstrate the security weaknesses, such as password guessing attacks, lack of forward secrecy and inaccurate password updates. Further, we propose a biometric-based user authentication and key agreement scheme for WSNs using fuzzy extraction and an elliptic curve cryptosystem (ECC). The proposed scheme withstands security threats from malicious adversaries and insider users by using an ECC-based session key. Our scheme is also suitable for WSNs when compared to traditional authentication and key agreement schemes because it performs simple ECC operations, hash functions and exlusive OR (XOR) operations. We prove that our scheme provides mutual authentication using Burrows-Abadi-Needham (BAN) logic [21].

The remainder of this paper is organized as follows: In Section 2, we present our preliminary details, and Chang et al.'s scheme is reviewed in Section 3. In Section 4, we cryptanalyze Chang et al.'s scheme, and our proposed scheme is presented in Section 5. Finally, we analyze our proposed scheme in Section 6 and conclude with the findings of this work in Section 7.

2. Preliminaries

In this section, we introduce the notations used in this paper and then define the cryptographic system and primitives used as building blocks in our security system. Finally, we define security requirements for user authentication in WSNs.

2.1. Notations

The notations used throughout this paper are described in Table 1.

2.2. Elliptic Curves Cryptosystem

Let p, q be two large primes, and $E/\mathbb{F}_p$ indicates an elliptic curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. We denote by $\mathbb{G}_1$ a q-order subgroup of the additive group of points of $E/\mathbb{F}_p$. The discrete logarithm problem (DLP) is required to be hard in $\mathbb{G}_1$. Mathematical problems in ECC are given as follows [22]:

Definition 1 (Elliptic curve discrete logarithm (ECDL) problem). *Given a point element $Q \in \mathbb{G}_1$, find an integer $a \in \mathbb{Z}_p^*$, such that $Q = a \times P$, where $a \times P$ indicates that the point P is added to itself for a times by the elliptic curve operation.*

Definition 2 (Elliptic curve computational Diffie–Hellman (ECDH) problem). *For $a, b \in \mathbb{Z}_p^*$, given two point elements $a \times P, b \times P \in \mathbb{G}_1$, compute $a \times b \times P \in \mathbb{G}_1$.*

Definition 3 (Elliptic curve decisional Diffie–Hellman (ECDDH) problem). *For $a, b, c \in \mathbb{Z}_p^*$, given three point elements $a \times P, b \times P, c \times P \in \mathbb{G}_1$, decide whether $c \times P = a \times b \times P$ or not.*

We assume that the ECDDH problem is intractable, which may guarantee that there is no probabilistic polynomial time (PPT) algorithm to solve ECDDHP, ECCDHP and ECDDLP with non-negligible probability.

Table 1. Notations.

Notation	**Meaning**
p, q	two large primes
U_i	user i
S_j	sensor node j
GWN	gateway node
SC_i	smart card of the user U_i
ID_i / pw_i	identity/password of U_i
BIO_i	biometric template of U_i
TID_i	temporal identity of U_i
SID_j	identity of S_j
ID_S	identity of SC_i
$\mathcal{A}$	adversary
K	a master secret of GWN
$\mathbb{G}_1$	cyclic group of order q
P	generator of $\mathbb{G}_1$
T_i, T_j, T_G	timestamps
$\bigoplus$	XOR operation
$\vert\vert$	concatenate operation
$h(\cdot)$	a secure one-way hash function

2.3. Fuzzy Extraction

We briefly describe the extraction process of key data from the given biometrics of a user using a fuzzy extractor. The output of a conventional hash function is sensitive, and it may also return completely different outputs even if there is little variation in the inputs. Note that the biometric information is prone to various noises during data acquisition, and the reproduction of the actual biometrics is hard in common practice. To avoid such a problem, a fuzzy extractor method [23] is preferred, which can extract a uniformly-random string and public information from the biometric template with a given error tolerance. In the reproduction process, the fuzzy extractor recovers the original biometric key data for noisy biometrics using a helper string. The fuzzy extractor consists of Gen (generate) and Rep (reproduce).

- $\text{Gen}(BIO_i) = (R_i, P_i)$. This probabilistic algorithm takes a biometric template BIO_i as an input and then outputs a biometric key R_i, which is a uniform and random string, and a helper string P_i. R_i can be the same under the assistance of P_i even if the biometric information changes slightly.

- Rep$(BIO_i', P_i) = (R_i)$. This deterministic algorithm takes noisy biometric information BIO_i' and a helper string P_i as inputs, then reproduces the biometric key R_i. To reproduce the same R_i, the metric space distances between BIO_i and BIO_i' have to meet the given verification threshold.

2.4. Network Model

- U_i: A user who receives a smart card from GWN and uses it to access multiple servers. After a successful authentication process with S_j, the user is given access to mobile services. Furthermore, the user's smart card is not tamper-resistant and can be lost or stolen by an adversary.
- S_j: A sensor node that collects information and provides services to users who successfully complete the authentication process. Sensors are not equipped with tamper-resistant hardware due to cost constraints, thus an adversary will know all of the keying materials stored in that sensor's memory.
- GWN: A trusted third-party that generates system parameters. It provides smart cards to users and pre-shared keys to sensors. GWN is assumed to be trustworthy and never compromised by an adversary.

2.5. Security Requirements

According to recent studies [24,25], the user authentication scheme for WSNs should satisfy the following security requirements: (1) mutual authentication: the user U_i and the sensor node S_j should authenticate each other with the help of the gateway node GWN; (2) anonymity: any adversary $\mathcal{A}$ should not be able to obtain the real identity of the user U_i; (3) session key generation: after executing the authentication and key agreement phase, the user U_i and the sensor node S_j should generate a session key; (4) unconstrained by GWN: the GWN should not have or be able to compute the registered user's information, such as the password and biometric template; (5) attack resistance: the scheme should withstand various attacks, such as off-line identity/password guessing, impersonation, smart card loss, man-in-the-middle and reply attacks; (6) efficient password update: it is required to change or update the users' password without the participation of GWN.

3. Review of Chang et al.'s Authentication and Key Agreement Scheme

In this section, we review Chang et al.'s authenticated key agreement scheme. It comprises four phases: registration, login, authentication and key agreement, as well as password change.

3.1. Registration Phase

Step 1: U_i chooses ID_i, pw_i and a random number RN_r, then computes $HPW_i = h(pw_i||RN_r)$ and sends $\{ID_i, HPW_i\}$ to GWN via a secure channel.

Step 2: GWN computes $HID_i = h(ID_i||K)$, $X_{S_i} = h(HID_i||K)$, $A_i = h(HPW_i||X_{S_i}) \oplus HID_i$, $B_i = h(HPW_i \oplus X_{S_i})$, $C_i = X_{S_i} \oplus h(ID_S||HPW_i)$. Then, GWN sends the smart card $SC_i = (ID_S, h(\cdot), A_i, B_i, C_i, TID_i)$ to U_i via a secure channel. GWN stores $(TID_i, TID_i^\circ, HID_i)$ in its storage, where $TID_i = RN_G$, RN_G is a nonce, and $TID_i^\circ = ''''$, where $TID_i^\circ = ''''$ means TID_i° contains nothing.

Step 3: U_i computes $XPW_i = h(pw_i) \oplus RN_r$ and inserts it into SC_i.

3.2. Login Phase

Step 1: U_i inputs ID_i^* and pw_i^* into SC_i.

Step 2: SC_i computes $RN_r^* = h(pw_i^*) \oplus XPW_i$, $HPW_i^* = h(pw_i^*||RN_r^*)$, $X_{S_i}^* = C_i \oplus h(ID_S||HPW_i^*)$, $B_i^* = h(HPW_i^* \oplus X_{S_i}^*$. Then, SC_i verifies $B_i^* \overset{?}{=} B_i$. If it is valid, SC_i computes $k_i = h(X_{S_i}^*||T_i)$, $DID_i = h(HPW_i^*||X_{S_i}^*) \oplus k_i$, $M_{U_i,G} = h(A_i||X_{S_i}^*||T_i)$, where T_i is the timestamp.

Step 3: U_i sends $\{DID_i, M_{U_i,G}, T_i, TID_i\}$ to GWN.

3.3. Authentication and Key Agreement Phase

Step 1: GWN checks the validity of T_i and retrieves HID_i from TID_i. Then, GWN computes $X_{S_i} = h(HID_i||K)$, $k_i = h(X_{S_i}||T_i)$, $X^* = DID_i \oplus k_i$, $M^*_{U_i,G} = h((X^* \oplus HID_i)||X_{S_i}||T_i)$, then checks $M^*_{U_i,G} \overset{?}{=} M_{U_i,G}$. If it is correct, GWN computes $X_{S_j} = h(SID_j||K)$, $M_{G,S_j} = h(DID_i||SID_j||X_{S_j}||T_G)$, then sends $\{DID_i, M_{G,S_j}, T_G\}$ to S_j, where T_G is the timestamp.

Step 2: S_j checks the validity of T_G and computes $M^*_{G,S_j} = h(DID_i||SID_j||X^*_{S_j}||T_G)$, then checks $M^*_{G,S_j} \overset{?}{=} M_{G,S_j}$. If it is successful, S_j computes $k_j = h(X_{S_j}||T_j)$, $Z_i = M^*_{G,S_j} \oplus k_j$, $K_S = f(DID_i, k_j)$, $M_{S_j,G} = h(Z_i||X^*_{S_j}||T_j)$, then sends $\{M_{S_j,G}, T_j\}$ to GWN, where T_j is the timestamp.

Step 3: GWN checks the validity of T_j and computes $k_j = h(X_{S_j}||T_j)$, $Z^*_i = M^*_{G,S_j} \oplus k_j$, $M^*_{S_j,G} = h(Z_i||X^*_{S_j}||T_j)$, then checks $M^*_{S_j,G} \overset{?}{=} M_{S_j,G}$. If it is correct, GWN computes $M_{G,U_i} = h(DID_i||M^*_{U_i,G}||k_j||X_{X_i}||T'_G)$, $y_i = k_j \oplus h(k_i)$, $TID_{i_{new}} = h(HID_i||T_i)$, then sends $\{y_i, M_{G,U_i}, T'_G\}$, where T'_G is the timestamp. Additionally, GWN updates (TID_i, TID°) as $(TID_{i_{new}}, TID_i)$.

Step 4: U_i checks the validity of T'_G and computes $k_j = y_i \oplus h(k_i)$, $M^*_{G,U_i} = h(DID_i||M_{U_i,G}||k_j||X_{S_i}||T'_G)$, then checks $M^*_{G,U_i} \overset{?}{=} M_{G,U_i}$ If it is correct, U_i computes $K_S = f(DID_i, k_j)$ and updates TID_i as $h(HID_i||T_i)$.

3.4. Password Change Phase

Step 1: U_i inputs $\{ID^*_i, pw^*_i, pw_{ni}\}$ into SC_i, where pw_{ni} is a new password.

Step 2: The smart card computes $RN^*_r = h(pw^*_i) \oplus XPW_i$, $HPW^*_i = h(pw^*_i||RN^*_r)$, $X^*_{S_i} = C_i \oplus h(ID_s||HPW^*_i)$, $B^*_i = h(HPW^*_i \oplus X^*_{S_i})$, then checks $B^*_i \overset{?}{=} B_i$. If it is correct, SC_i computes updated values $HPW_{ni} = h(pw_{ni}||RN^*_r)$, $A_{ni} = A_i \oplus h(HPW^*_i||X^*_{S_i}) \oplus h(HPW_{ni}||X^*_{S_i})$, $B_{ni} = h(HPW_{ni} \oplus X^*_{S_i})$, $C_{ni} = X^*_{S_i} \oplus h(ID_S||HPW_{ni}$. Then, SC_i replaces (A_i, B_i, C_i) with (A_{ni}, B_{ni}, C_{ni}).

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

In this section, we analyze the security weaknesses of Chang et al.'s scheme [14]. Chang et al. cryptanalyzed Kim et al.'s scheme [13] and improved it by providing enhanced security properties. They claimed that their protocol could withstand various attacks. However, we show that their protocol is vulnerable to off-line password guessing attacks and does not provide perfect forward secrecy. We also show that their protocol cannot satisfy accurate password change. The capabilities of an adversary $\mathcal{A}$ [25] throughout this paper are as follows:

- An adversary $\mathcal{A}$ can be either a user or a sensor node, but not a gateway node [26].
- An adversary $\mathcal{A}$ has total control over the public communication channel. Thus, the adversary can intercept, insert, delete or modify any message transmitted via a public channel.
- An adversary $\mathcal{A}$ may steal a user's smart card and extract the information stored in it by means of analyzing the power consumption [27].
- An adversary $\mathcal{A}$ can easily guess low-entropy passwords in an off-line manner, but the guessing of two secret parameters is computationally infeasible in polynomial time [28].

4.1. Off-Line Password Guessing Attack

Previous works [27] demonstrated that smart cards could be vulnerable to side channel attack, i.e., $\mathcal{A}$ could extract the information stored in the smart card SC_i. $\mathcal{A}$ chooses an arbitrary password pw^*_i, then computes to guess a correct password as follows:

$$\begin{aligned}
RN_r^* &= XPW_i \oplus h(pw_i^*)\\
HPW_i^* &= h(pw_i^*||RN_r^*)\\
X_{S_i}^* &= C_i \oplus h(ID_S||HPW_i^*)\\
B_i^* &= h(HPW_i^* \oplus X_{S_i}^*)\\
\text{verifies } B_i^* &\overset{?}{=} B_i
\end{aligned}$$

If they are equal, $\mathcal{A}$ finds the correct password. Otherwise, $\mathcal{A}$ guesses another pw_i^* and repeats the steps listed above until the correct password is found. In practical applications, people usually choose an easy-to-remember password for convenience, thus passwords could come from a very small dictionary. Therefore, $\mathcal{A}$ could find the correct password using a brute-force attack.

Even though Chang et al. has claimed that it is secure, once $\mathcal{A}$ guesses the password correctly, $\mathcal{A}$ can launch various attacks, such as impersonation, stolen verifier and lost smart card attacks. This is due to the fact that the scheme uses only a password to check the validity of users. Therefore, it is crucial to protect password guessing attacks and use various authentication factors to check the validity of users.

4.2. Lack of Perfect Forward Secrecy

In Chang et al.'s scheme, session key K_S is computed as $h(DID_i, k_j)$. Once a long-term key of S_j, X_{S_j}, is disclosed to $\mathcal{A}$, $\mathcal{A}$ can compute previous session keys as follows:

Step 1: $\mathcal{A}$ intercepts and stores all messages exchanged in previous sessions, such as DID_i and T_i.
Step 2: $\mathcal{A}$ computes $k_j = h(X_{S_j}||T_j)$, then finally retrieves a previous session key $K_S = f(DID_i, k_j)$.

This result indicates that Chang et al.'s scheme does not provide perfect forward secrecy. Furthermore, $\mathcal{A}$ who knows X_{S_j} also can compute present and future session keys by intercepting messages via the public channel, indicating that Chang et al.'s scheme does not provide backward secrecy.

4.3. Incorrectness of Password Change

Chang et al.'s adopted Kim et al.'s password change phase; however, we found out that Kim et al.'s password update is not suitable for Chang et al.'s scheme. We demonstrate the incorrectness of the password change phase as follows:

Step 1: Once the user performs the password change phase, the previous password pw_i is changed into pw_{ni}, and information in the smart card, (A_i, B_i, C_i), is replaced with (A_{ni}, B_{ni}, C_{ni}).
Step 2: Then, the user performs the login phase using the new password pw_{ni}; however, U_i is not allowed to access for not computing the proper RN_r from XPW_i. XPW_i is not updated in the password change phase; therefore, $RN_r^* = XPW_i \oplus h(pw_{ni}^*) \neq RN_r$ and, finally, $B_i^* \neq B_i$.

In addition, it is of no use to update the password if the password is revealed even one time because no other information, such as identity, is required to login and change the password. Therefore, regardless of whether a user changes the password, $\mathcal{A}$ can also change the password and be verified by the smart card.

5. The Proposed Three-Factor Authentication and Key Agreement Scheme

In this section, we propose a secure three-factor authentication and key agreement scheme for WSNs to overcome the security weaknesses in Chang et al.'s scheme. Based on Kim et al. and Chang et al.'s schemes, the proposed scheme provides better security functionality by using biometric information of the user and makes up for the password update inaccuracy. The proposed scheme consists of four phases: registration, login, authentication and key agreement and password change. The details of each phase are presented as follows.

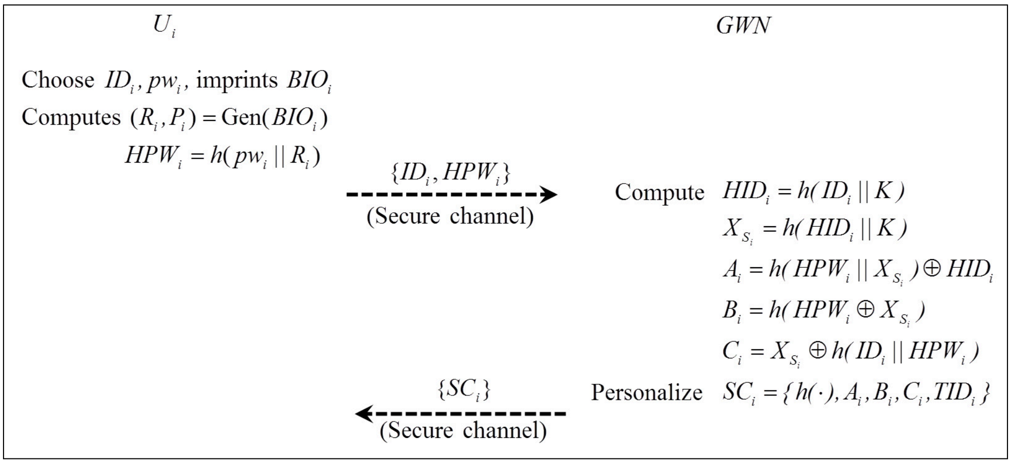
5.1. Registration Phase

A user U_i registers the identity and password to GWN, then GWN generates a smart card SC_i for U_i and sends it to U_i through a secure channel. Likewise, a sensor node S_j is distributed with (SID_j, X_{S_j}), where $X_{S_j} = h(SID_j||K)$. Figure 1 illustrates the registration phase, which is performed as follows:

Step 1: $U_i \Rightarrow GWN : \{ID_i, HPW_i\}$
U_i chooses ID_i and pw_i and imprints BIO_i, then U_i computes $(R_i, P_i) = \text{Gen}(BIO_i)$ and $HPW_i = h(pw_i||R_i)$ and sends $\{ID_i, HPW_i\}$ to GWN through a secure channel.

Step 2: $GWN \Rightarrow U_i : SC_i = \{h(\cdot), A_i, B_i, C_i, TID_i\}$
GWN computes $HID_i = h(ID_i||K)$, $X_{S_i} = h(HID_i||K))$, $A_i = h(HPW_i||X_{S_i}) \oplus HID_i$, $B_i = h(HPW_i \oplus X_{S_i})$, $C_i = X_{S_i} \oplus h(ID_i||HPW_i)$.

Step 3: GWN stores parameters $(TID_i, TID_i^\circ, HID_i)$, where $TID_i = RN_G$ (RN_G is a nonce); $TID_i^\circ = ""$. TID_i° is empty at first time because TID_i has not been updated; however, this parameter is required to check the correctness of the received TID_i and retrieve HID_i safely when GWN does not find a proper updated TID_i in the case of an unsuccessful update process.
Then, GWN issues the smart card $SC_i = \{h(\cdot), A_i, B_i, C_i, TID_i\}$ and sends it to U_i through a secure channel.



U_i
GWN
Choose ID_i, pw_i, imprints BIO_i
Computes $(R_i, P_i) = \text{Gen}(BIO_i)$
$HPW_i = h(pw_i || R_i)$
$\{ID_i, HPW_i\}$
(Secure channel)
Compute $HID_i = h(ID_i || K)$
$X_{S_i} = h(HID_i || K)$
$A_i = h(HPW_i || X_{S_i}) \oplus HID_i$
$B_i = h(HPW_i \oplus X_{S_i})$
$C_i = X_{S_i} \oplus h(ID_i || HPW_i)$
$\{SC_i\}$
(Secure channel)
Personalize $SC_i = \{h(\cdot), A_i, B_i, C_i, TID_i\}$

Figure 1. Registration phase.

5.2. Login Phase

When U_i tries to access the S_j, the login request is launched at first by U_i with SC_i. Figure 2 illustrates the login phase, which is performed as follows:

Step 1: U_i inserts SC_i, inputs ID_i^*, pw_i^* and imprints BIO_i^*.

Step 2: SC_i computes $R_i^* = \text{Rep}(BIO_i^*, P_i)$, $HPW_i^* = h(pw_i^*||R_i^*)$, $X_{S_i}^* = C_i \oplus h(ID_i^*||HPW_i^*)$, $B_i^* = h(HPW_i^* \oplus X_{S_i}^*)$. Then, SC_i verifies $B_i^* \overset{?}{=} B_i$. If it is correct, SC_i generates a random number $a \in \mathbb{Z}_p^*$ and computes $X_i = aP$, $k_i = h(X_{S_i}^*||T_i)$, $DID_i = h(HPW_i^*||X_{S_i}^*) \oplus k_i$, $M_{U_i,G} = h(A_i||X_{S_i}^*||X_i||T_i)$, where T_i is the current timestamp.

Step 3: U_i sends the login request message $\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$ to GWN.

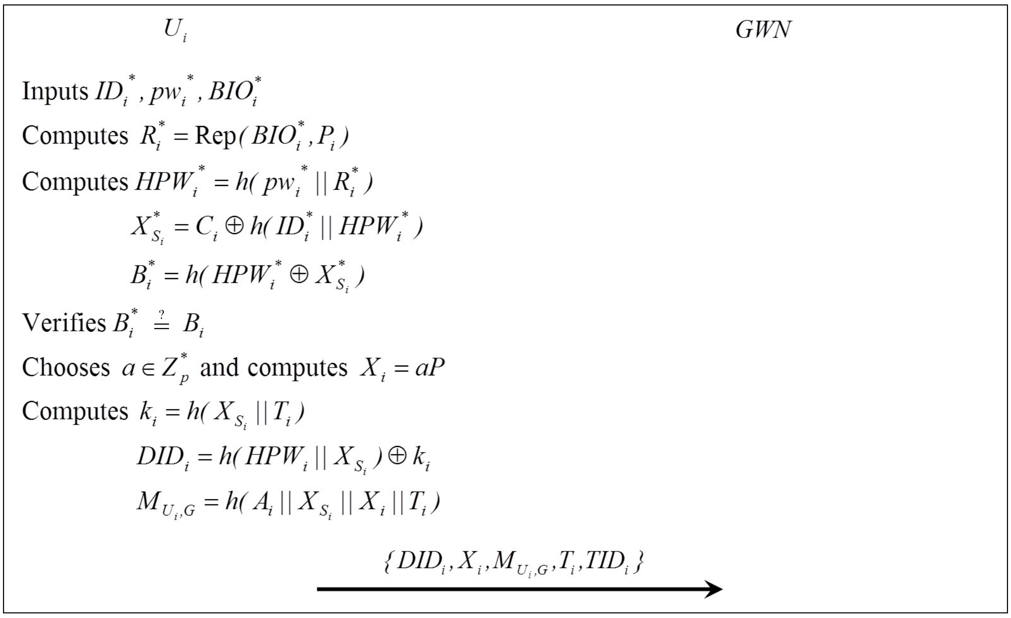

U_i
GWN
Inputs ID_i^*, pw_i^*, BIO_i^*
Computes $R_i^* = \mathrm{Rep}(BIO_i^*, P_i)$
Computes $HPW_i^* = h(pw_i^* || R_i^*)$
$X_{S_i}^* = C_i \oplus h(ID_i^* || HPW_i^*)$
$B_i^* = h(HPW_i^* \oplus X_{S_i}^*)$
Verifies $B_i^* \overset{?}{=} B_i$
Chooses $a \in Z_p^*$ and computes $X_i = aP$
Computes $k_i = h(X_{S_i} || T_i)$
$DID_i = h(HPW_i || X_{S_i}) \oplus k_i$
$M_{U_i,G} = h(A_i || X_{S_i} || X_i || T_i)$
$\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$

Figure 2. Login phase.

5.3. Authentication and Key Agreement Phase

In this phase, U_i and S_j authenticate each other and generate a common session key SK by the help of GWN. The trusted party GWN is interconnected with U_i and S_j, respectively, and helps to establish a session key between U_i and S_j; however, GWN is not able to derive the session key. Figure 3 illustrates the authentication and key agreement phase, which is performed as follows:

Step 1: $GWN \Rightarrow S_j : \{DID_i, X_i, M_{G,S_j}, T_G\}$
After receiving $\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$, GWN checks the validity of T_i and retrieves HID_i from TID_i. If no TID_i is found, GWN checks TID_i°. If it still is not found, GWN rejects the login request; otherwise, GWN computes $X_{S_i} = h(HID_i||K)$ and $k_i = h(X_{S_i}||T_i)$. Then, GWN verifies $M_{U_i,G} \overset{?}{=} h((DID_i \oplus k_i \oplus HID_i)||X_{S_i}||X_i||T_i)$. If it is valid, GWN authenticates U_i and computes $M_{G,S_j} = h(DID_i||SID_j||X_{S_j}||X_i||T_G)$, then sends $\{DID_i, X_i, M_{G,S_j}, T_G\}$ to S_j, where T_G is the current timestamp.

Step 2: $S_j \Rightarrow GWN : \{M_{S_j,G}, Y_j, T_j\}$
After receiving $\{DID_i, X_i, M_{G,S_j}, T_G\}$, S_j checks the validity of T_G and verifies $M_{G,S_j} \overset{?}{=} h(DID_i||X_i||X_{S_j}^*||T_G)$ using its stored secret value $X_{S_j}^* = h(SID_j||K)$. If it is valid, S_j authenticates GWN and computes $k_j = h(X_{S_j}^*||T_j)$, $Z_i = M_{G,S_j} \oplus k_j$, where T_j is the current timestamp. Then, S_j generates a random number $b \in \mathbb{Z}_p^*$ and computes $Y_j = bP$ and a session key $SK = k_{ji} = h(DID_i||k_j||bX_i)$. Finally, S_j computes $(M_{S_j,G} = h(Z_i||X_{S_j}^*||X_i||Y_j||T_j))$ and sends $\{M_{S_j,G}, Y_j, T_j\}$ to GWN.

Step 3: $GWN \Rightarrow U_i : \{e_i, M_{G,U_i}, Y_i, T_G'\}$
After receiving $\{M_{S_j,G}, Y_i, T_j\}$, GWN checks the validity of T_j, computes $k_j = h(X_{S_j}||T_j)$, $Z_i^* = M_{G,S_j}^* \oplus k_j$ and verifies $M_{S_j,G} \overset{?}{=} h(Z_i^*||X_{S_j}||X_i||Y_j||T_j)$. If it is valid, GWN authenticates S_j and computes $e_i = k_j \oplus h(k_i)$, $(M_{G,U_i} = h(DID_i||M_{U_i,G}||k_j||X_{S_i}||X_i||Y_j||T_G'))$, $TID_{i_{new}} = h(HID_i||T_i)$, where T_G' is the current timestamp. Then, GWN sends $\{e_i, M_{G,U_i}, Y_i, T_G'\}$ to U_i and updates (TID_i, TID_i°) as $(TID_{i_{new}}, TID_i)$ in its storage.

Step 4: After receiving $\{e_i, M_{G,U_i}, Y_i, T_G'\}$, U_i checks the validity of T_G', computes $k_j^* = e_i \oplus h(k_i^*)$ and verifies $M_{G,U_i} \overset{?}{=} h(DID_i||M_{U_i,G}||k_j^*||X_{S_i}||X_i||Y_j||T_G')$. If it is valid, U_i computes the session key $SK = k_{ij} = h(DID_i||k_j||aY_i)$. Finally, U_i updates TID_i as $h(HID_i||T_i)$.

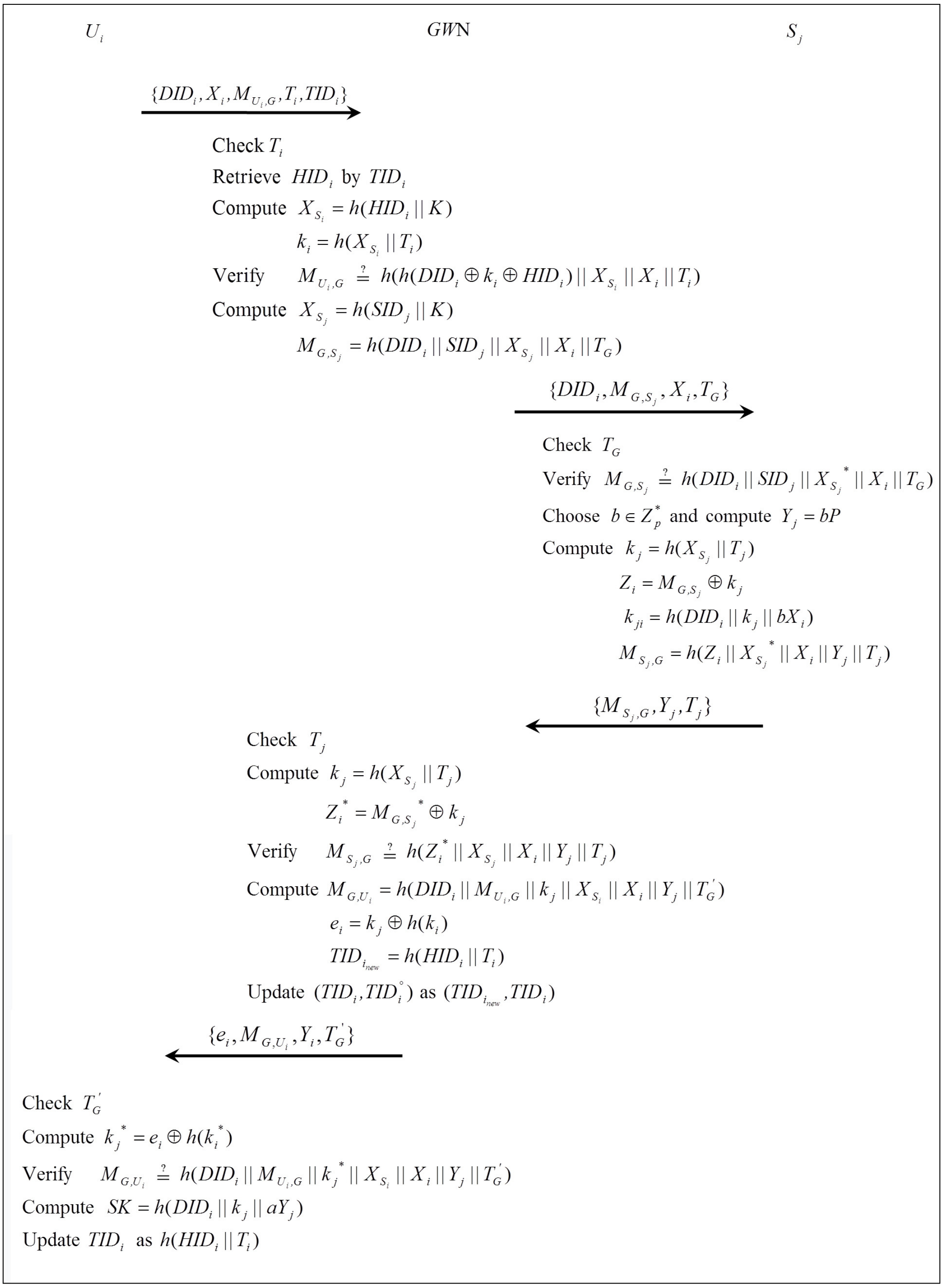

U_i
GWN
S_j
$\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$
Check T_i
Retrieve HID_i by TID_i
Compute $X_{S_i} = h(HID_i \,||\, K)$
$k_i = h(X_{S_i} \,||\, T_i)$
Verify $M_{U_i,G} \overset{?}{=} h(h(DID_i \oplus k_i \oplus HID_i) \,||\, X_{S_i} \,||\, X_i \,||\, T_i)$
Compute $X_{S_j} = h(SID_j \,||\, K)$
$M_{G,S_j} = h(DID_i \,||\, SID_j \,||\, X_{S_j} \,||\, X_i \,||\, T_G)$
$\{DID_i, M_{G,S_j}, X_i, T_G\}$
Check T_G
Verify $M_{G,S_j} \overset{?}{=} h(DID_i \,||\, SID_j \,||\, X_{S_j}{}^* \,||\, X_i \,||\, T_G)$
Choose $b \in Z_p^*$ and compute $Y_j = bP$
Compute $k_j = h(X_{S_j} \,||\, T_j)$
$Z_i = M_{G,S_j} \oplus k_j$
$k_{ji} = h(DID_i \,||\, k_j \,||\, bX_i)$
$M_{S_j,G} = h(Z_i \,||\, X_{S_j}{}^* \,||\, X_i \,||\, Y_j \,||\, T_j)$
$\{M_{S_j,G}, Y_j, T_j\}$
Check T_j
Compute $k_j = h(X_{S_j} \,||\, T_j)$
$Z_i^* = M_{G,S_j}{}^* \oplus k_j$
Verify $M_{S_j,G} \overset{?}{=} h(Z_i^* \,||\, X_{S_j} \,||\, X_i \,||\, Y_j \,||\, T_j)$
Compute $M_{G,U_i} = h(DID_i \,||\, M_{U_i,G} \,||\, k_j \,||\, X_{S_i} \,||\, X_i \,||\, Y_j \,||\, T_G^{'})$
$e_i = k_j \oplus h(k_i)$
$TID_{i_{new}} = h(HID_i \,||\, T_i)$
Update (TID_i, TID_i°) as $(TID_{i_{new}}, TID_i)$
$\{e_i, M_{G,U_i}, Y_i, T_G^{'}\}$
Check $T_G^{'}$
Compute $k_j^* = e_i \oplus h(k_i^*)$
Verify $M_{G,U_i} \overset{?}{=} h(DID_i \,||\, M_{U_i,G} \,||\, k_j^* \,||\, X_{S_i} \,||\, X_i \,||\, Y_j \,||\, T_G^{'})$
Compute $SK = h(DID_i \,||\, k_j \,||\, aY_j)$
Update TID_i as $h(HID_i \,||\, T_i)$

Figure 3. Authentication and key agreement phase.

5.4. Password Change Phase

When U_i wants to change pw_i with the new pw_{ni}, U_i performs the password change phase. Figure 4 illustrates the password change phase, which is performed as follows:

Step 1: U_i imprints BIO_i^* and computes $R_i^* = \text{Rep}(BIO_i^*, P_i)$, then inputs $\{ID_i^*, R_i^*, pw_i^*, pw_{ni}\}$ into SC_i.

Step 2: SC_i computes $HPW_i^* = h(pw_i^*||R_i^*)$, $X_{S_i}^* = C_i \oplus h(ID_i^*||HPW_i^*)$, $B_i^* = h(HPW_i^* \oplus X_{S_i}^*)$. Then, SC_i verifies $B_i^* = B_i$ to check the validity of U_i. If it is correct, SC_i computes updated values $HPW_{ni} = h(pw_{ni}||R_i^*)$, $A_{ni} = A_i \oplus h(HPW_i||X_{S_i}^*) \oplus h(HPW_{ni}||X_{S_i}^*)$, $B_{ni} = h(HPW_{ni} \oplus X_{S_i}^*)$, $C_{ni} = X_{S_i}^* \oplus h(ID_i^*||HPW_{ni})$. Then, SC_i replaces (A_i, B_i, C_i) with (A_{ni}, B_{ni}, C_{ni}).

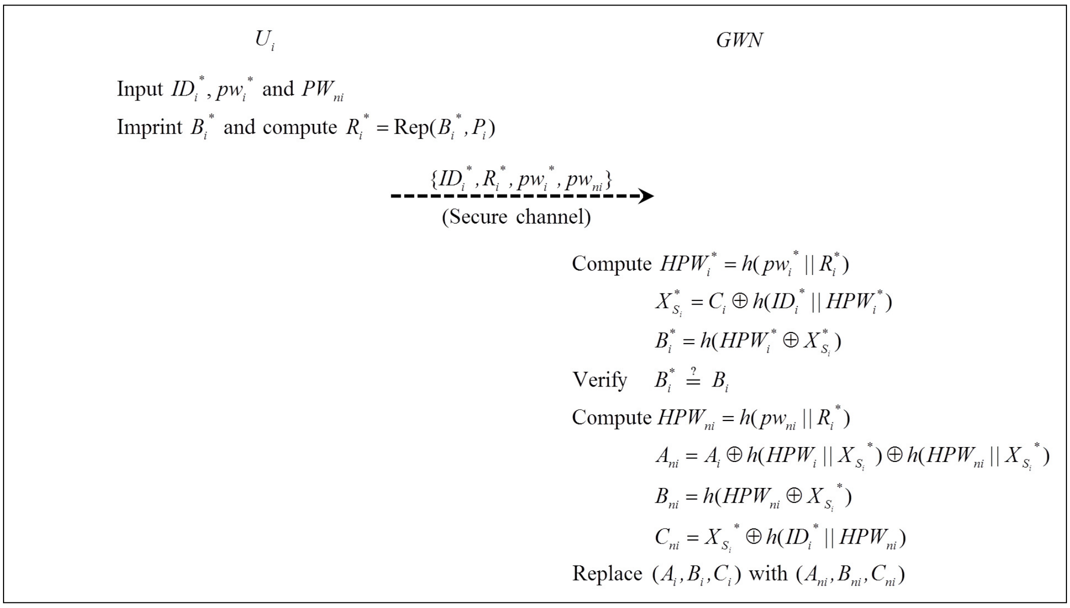

U_i
GWN
Input ID_i^*, pw_i^* and PW_{ni}
Imprint B_i^* and compute $R_i^* = \text{Rep}(B_i^*, P_i)$
$\{ID_i^*, R_i^*, pw_i^*, pw_{ni}\}$
(Secure channel)
Compute $HPW_i^* = h(pw_i^* || R_i^*)$
$X_{S_i}^* = C_i \oplus h(ID_i^* || HPW_i^*)$
$B_i^* = h(HPW_i^* \oplus X_{S_i}^*)$
Verify $B_i^* \overset{?}{=} B_i$
Compute $HPW_{ni} = h(pw_{ni} || R_i^*)$
$A_{ni} = A_i \oplus h(HPW_i || X_{S_i}^*) \oplus h(HPW_{ni} || X_{S_i}^*)$
$B_{ni} = h(HPW_{ni} \oplus X_{S_i}^*)$
$C_{ni} = X_{S_i}^* \oplus h(ID_i^* || HPW_{ni})$
Replace (A_i, B_i, C_i) with (A_{ni}, B_{ni}, C_{ni})

Figure 4. Password change phase.

6. Analysis

In this section, we describe an analysis of our proposed authentication and key agreement scheme with respect to security and efficiency. We assume that the capabilities of the adversary are the same as those from our cryptanalysis of Chang et al.'s scheme in Section 4. We first prove the security of our scheme with BAN logic [21], then analyze the proposed scheme based on the security requirements for WSNs.

6.1. Proof of Authentication and Key Agreement Based on BAN Logic

Recently, security analyses about authentication and key agreement schemes in WSNs have been conducted using the BAN logic, which is a method to prove the security of mutual authentication and a session key [25,29]. In this section, we analyze the security of our proposed authentication scheme with BAN logic [21]. Table 2 illustrates notations used in BAN logic.

Table 2. BAN logic notations.

Notations	Meaning
$P \mid\equiv X$	P believes X
$P \lhd X$	P sees X
$P \mid\sim X$	P once said X
$P \Rightarrow X$	P has jurisdiction over X
$\#(X)$	X is fresh
$P \overset{K}{\leftrightarrow} Q$	P and Q may use the shared key K
SK	The session key shared between two principals
$\langle X \rangle_Y$	X combined with the formula Y
$(X)_K$	X hashed under the key K
$\{X\}_K$	X encrypted under the key K

1. The BAN logic postulates:

   (a) Message meaning rule:

   $$\frac{P \text{ believes } Q \overset{K}{\longleftrightarrow} P, P \text{ sees } \{X\}_K}{P \text{ believes } Q \text{ said } X}$$

   (b) Nonce-verification rule:

   $$\frac{P \text{ believes fresh } (X), P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ believes } X}$$

   (c) Jurisdiction rule:

   $$\frac{P \text{ believes } Q \text{ controls } X, P \text{ believes } Q \text{ believes } X}{P \text{ believes } X}$$

   (d) Freshness-conjuncatenation rule:

   $$\frac{P \text{ believes fresh}(X)}{P \text{ believes fresh}(X, Y)}.$$

2. Security goals:

   The proposed scheme should satisfy the following goals:

   g_1. $U_i| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$
   g_2. $S_j| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$
   g_3. $U_i| \equiv S_j| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$
   g_4. $S_j| \equiv U_i| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$

3. Idealized scheme:

   We transform our scheme into the idealized form as follows:

   Msg_1. $U_i \rightarrow GWN : (DID_i, K, X_i, T_i)_{HID_i}$
   Msg_2. $GWN \rightarrow S_j : (DID_i, SID_j, K, X_i, T_G)_{X_{S_j}}$
   Msg_3. $S_j \rightarrow GWN : (DID_i, SID_j, K, X_i, Y_i, T_j)_{X_{S_j}}$
   Msg_4. $GWN \rightarrow U_i : (DID_i, k_j, K, X_i, Y_i, T'_G)_{HID_i}$

4. Initiative premises:

   We make the assumptions about the initial state of the scheme to analyze the proposed scheme as follows.

   p_1. $GWN| \equiv \#(T_i)$
   p_2. $GWN| \equiv \#(T_j)$
   p_3. $S_j| \equiv \#(T_G)$
   p_4. $U_i| \equiv \#(T'_G)$
   p_5. $GWN| \equiv GWN \overset{X_{S_j}}{\longleftrightarrow} S_j$
   p_6. $S_j| \equiv GWN \overset{X_{S_j}}{\longleftrightarrow} S_j$
   p_7. $U_i| \equiv U_i \overset{HID_i}{\longleftrightarrow} GWN$
   p_8. $GWN| \equiv U_i \overset{HID_i}{\longleftrightarrow} GWN$

p_9. $U_i| \equiv S_j \Rightarrow U_i \stackrel{SK}{\longleftrightarrow} S_j$

p_{10}. $S_j| \equiv U_i \Rightarrow U_i \stackrel{SK}{\longleftrightarrow} S_j$

(The meanings of p_9 and p_{10} are different from g_3 and g_4. p_9 and p_{10} are not the goals that we want to deduce. These are widely-used premises as done in [29–32].)

5. Security analysis of the idealized form of the proposed scheme:

a_1. According to Msg_1, we could get:

$$s_1 : GWN \lhd (DID_i, K, X_i, T_i)_{HID_i}$$

a_2. According to p_8, we apply the message-meaning rule to obtain:

$$s_2 : GWN| \equiv U_i| \sim (DID_i, K, X_i, T_i)_{HID_i}$$

a_3. According to p_1, we apply the freshness-conjuncatenation rule to obtain:

$$s_3 : GWN| \equiv \#(DID_i, K, X_i, T_i)_{HID_i}$$

Then, from s_2 and s_3, we apply the nonce-verification rule to obtain:

$$s_4 : GWN| \equiv U_i| \equiv (DID_i, K, X_i, T_i)_{HID_i}$$

a_4. According to Msg_2, we could get:

$$s_5 : S_j \lhd (DID_i, SID_j, K, X_i, T_G)_{X_{S_j}}$$

a_5. According to p_6, we apply the message-meaning rule to obtain:

$$s_6 : S_j| \equiv GWN| \sim (DID_i, SID_j, K, X_i, T_G)_{X_{S_j}}$$

a_6. According to p_3, we apply the the freshness-conjuncatenation rule to obtain:

$$s_7 : S_j| \equiv \#(DID_i, SID_j, K, X_i, T_G)_{X_{S_j}}$$

Then, from s_6 and s_7, we apply the nonce-verification rule to obtain:

$$s_8 : S_j| \equiv GWN| \equiv (DID_i, SID_j, K, X_i, T_G)_{X_{S_j}}$$

a_7. According to Msg_3, we could get:

$$s_9 : GWN \lhd (DID_i, SID_j, K, X_i, Y_i, T_j)_{X_{S_j}}$$

a_8. According to p_5, we apply the message-meaning rule to obtain:

$$s_{10} : GWN| \equiv S_j| \sim (DID_i, SID_j, K, X_i, Y_i, T_j)_{X_{S_j}}$$

a_9. According to p_2, we apply the the freshness-conjuncatenation rule to obtain:

$$s_{11} : GWN| \equiv \#(DID_i, SID_j, K, X_i, Y_i, T_j)_{X_{S_j}}$$

Then, from s_{10} and s_{11}, we apply the nonce-verification rule to obtain:

$$s_{12} : GWN| \equiv U_i| \equiv (DID_i, SID_j, K, X_i, Y_i, T_j)_{X_{S_j}}$$

a_{10}. According to Msg_4, we could get:

$$s_{13} : U_i \lhd (DID_i, k_j, K, X_i, Y_i, T'_G)_{HID_i}$$

a_{11}. According to p_7, we apply the message-meaning rule to obtain:

$$s_{14} : U_i| \equiv GWN| \sim (DID_i, k_j, K, X_i, Y_i, T'_G)_{HID_i}$$

a_{12}. According to p_4, we apply the the freshness-conjuncatenation rule to obtain:

$$s_{15} : U_i| \equiv \#(DID_i, k_j, K, X_i, Y_i, T'_G)_{HID_i}$$

Then, from s_{14} and s_{15}, we apply the nonce-verification rule to obtain:

$$s_{16} : U_i| \equiv GWN| \equiv (DID_i, k_j, K, X_i, Y_i, T'_G)_{HID_i}$$

a_{13}. Because $SK = h(DID_i||k_j||bX_i)$, according to s_{16} and s_{12}, we could produce:

$$s_{17} : U_i| \equiv S_j| \equiv U_i \overset{SK}{\longleftrightarrow} S_j \qquad \text{(Goal 3)}$$

Likewise, $SK = h(DID_i||k_j||aY_i)$, according to s_8 and s_4, we could produce:

$$s_{18} : S_j| \equiv U_i| \equiv U_i \overset{SK}{\longleftrightarrow} S_j \qquad \text{(Goal 4)}$$

a_{14}. According to s_{17} and p_9, we apply the jurisdiction rule to produce:

$$s_{19} : U_i| \equiv U_i \overset{SK}{\longleftrightarrow} S_j \qquad \text{(Goal 1)}$$

Likewise, according to s_{18} and p_{10}, we apply the jurisdiction rule to produce:

$$s_{20} : S_j| \equiv U_i \overset{SK}{\longleftrightarrow} S_j \qquad \text{(Goal 2)}$$

According to Goal 1, Goal 2, Goal 3 and Goal 4, we conclude that both U_i and S_j believe they share the session key.

6.2. Security Analysis against Various Attacks

- User anonymity and untraceability: Our scheme provides anonymity of users. The user U_i does not reveal a real identity ID_i in open channels; instead, GWN generates and sends a pseudonym identity $TID_i = HID_i = RN_G$ to U_i in the registration phase and updates it as $TID_i = h(HID_i||T_i)$ before finalizing the session. The identity is dynamic for every session; thus, an adversary $\mathcal{A}$ cannot obtain the user's true identity. The proposed scheme also provides untraceability by having all messages used in the session satisfy a freshness requirement. Therefore, $\mathcal{A}$ cannot trace the user.
- Perfect forward secrecy: A session key SK is computed as $h(DID_i||k_j||abP)$. Even though the long-term private keys X_{S_i} and X_{S_j} are disclosed to $\mathcal{A}$, he/she cannot compute previous session keys, because it is hard to compute abP using X_i and Y_i due to the difficulty of ECDH.

Thus, $\mathcal{A}$ cannot compute previous session keys using long-term private keys. Therefore, our scheme provides forward secrecy.

- Mutual authentication: In our scheme, U_i and GWN authenticate each other, and GWN and S_j authenticate each other, respectively. GWN authenticates U_i by checking $M_{U_i,G} \overset{?}{=} h((DID_i \oplus k_i \oplus HID_i)||X_{S_i}||X_i||T_i)$. $\mathcal{A}$ needs to compute X_{S_i} and k_i to reconstruct $M_{U_i,G}$; however, only a legal user can compute those values. U_i authenticates GWN by checking $(M_{G,U_i} = h(DID_i||M_{U_i,G}||k_j||X_{S_i}||X_i||Y_j||T'_G))$. $\mathcal{A}$ needs to compute k^*_j and X_{S_i} to reconstruct $(M_{G,U_i}$; however, only a legal GWN can compute those values. Therefore, U_i and GWN mutually authenticate. Similarly, S_j authenticates GWN by checking M_{G,S_j}, and GWN authenticates S_j by checking $M_{S_j,G}$. Additionally, only legal S_j and GWN can reconstruct them, then authenticate mutually. Therefore, our scheme provides proper mutual authentication.
- Off-line password guessing attack: $\mathcal{A}$ may attempt to guess the password pw_i by extracting the values stored in the smart card SC_i. $\mathcal{A}$ could guess correctly if he/she generates a series of equations and computes the valid B_i using guessing passwords. However, $\mathcal{A}$ is required to know the biometric information of the user, which cannot be forged, for generating equations. Therefore, it is infeasible to correctly guess the user's password in our scheme.
- Smart card loss attack: $\mathcal{A}$ can extract values in the smart card by means of power analysis and other techniques. Suppose $\mathcal{A}$ obtains the user's smart card and extracts stored parameters $\{h(\cdot), A_i, B_i, C_i, TID_i\}$. From these values, $\mathcal{A}$ cannot obtain any useful information because the parameters are safeguarded with a one-way hash function, and TID_i is just a nonce. Furthermore, $\mathcal{A}$ may attempt to log in by generating a login request message. However, $\mathcal{A}$ cannot even pass the login phase and generate a valid login request message without proper ID_i, pw_i and B_i. Therefore, the proposed scheme withstands smart card loss attacks.
- User impersonation attack: $\mathcal{A}$ who somehow possesses a valid smart card SC_i of U_i and wants to access S_j is required to generate and send a valid login request message $\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$ to GWN. $\mathcal{A}$ must know HPW_i and X_{S_i} to compute these values. However, in our scheme, ID_i, pw_i and R_i are not revealed. Thus, $\mathcal{A}$ cannot compute the temporal key k_i and generate a valid login request message. Therefore, our scheme is secure against the user impersonation attack.
- Man-in-the-middle attack and replay attack: $\mathcal{A}$ who knows public channel information and has the smart card SC_i of U_i may attempt to establish a secure channel with S_j. However, $\mathcal{A}$ cannot authenticate with GWN because $\mathcal{A}$ cannot generate a valid login request message, as mentioned above. In addition, those messages captured in a public channel are refreshed in every session, so that $\mathcal{A}$ cannot use them repeatedly. Therefore, our scheme withstands man-in-the-middle and replay attacks.
- Stolen verifier attack: $\mathcal{A}$ who obtains the verifier table of GWN may attempt to attack users to gain some advantages. However, $\mathcal{A}$ still cannot compute HPW_i, X_{S_i} and k_i and will fail to pass the login phase. Of course, $\mathcal{A}$ will fail to compute a login request message without pw_i and R_i. Therefore, even if $\mathcal{A}$ has the verifier table, our protocol withstands stolen verifier attacks.
- Known-key attack: A session key SK is computed as $h(DID_i||k_j||abP)$, and DID_i, k_j and abP are independent in each session. Though $\mathcal{A}$, who somehow possesses each value, attempts to generate other session keys, he/she will find that they cannot successfully derive valid session keys. Therefore, our proposed scheme withstands known-key attacks.

We compare the functionality features of the proposed scheme with related user authentication schemes for WSNs in Table 3. $\circ$ denotes that the scheme provides the property; $\times$ denotes that the scheme does not provide the property; $\triangle$ denotes that the scheme does not provide the property when off-line password guessing attacks succeed; $-$ denotes that the scheme does not concern the property.

Table 3. Comparisons of the functionality features. ECC, elliptic curve cryptosystem.

	Kim et al.′ Scheme [13]	**Chang et al.′ Scheme** [14]	**Yoon and Yoo's Scheme** [15]	**Choi et al.′ Scheme** [18]	**Proposed Scheme**
Provides user anonymity	×	○	×	×	○
Provides user untraceability	×	△	×	×	○
Provides forward secrecy	×	×	○	○	○
Provides secure password update	○	×	—	—	○
Provides mutual authentication	○	○	○	○	○
Resists off-line password guessing attack	×	×	—	—	○
Resists user impersonation attack	×	△	○	×	○
Resists lost smart card attack	×	△	○	○	○
Resists stolen verifier attack	×	△	—	—	○
Resists man-in-the-middle attack	×	△	○	○	○
Resists replay attack	○	○	○	○	○
Resist biometric recognition error	—	—	×	○	○
Usage of biometrics	×	×	○	○	○
Usage of ECC	×	×	○	○	○

6.3. Performance Comparisons

In Table 4, we compare the computational cost with related schemes. T_h denotes the computation time for the hash function; T_x denotes the XOR operation; T_F denotes the fuzzy extraction; T_E denotes the ECC multiplication; T_{enc} denotes the encryption/decryption. The computation cost of ours is a bit higher than [13,14] because of the usage of biometrics and ECC, but it is considered to be operationally viable in WSNs [15,18]. Additionally, our proposed scheme provides the enhanced security functionalities and is secure against various attacks.

Table 4. Comparisons of the computation costs.

Scheme		**Computation Cost**		
		Registration	**Login & Authentication**	**Total**
Kim et al.'s [13]	User	$2T_h + T_x$	$9T_h + 9T_x$	$11T_h + 10T_x$
	GWN	$6T_h + 3T_x$	$8T_h + 8T_x$	$14T_h + 11T_x$
	Sensor	0	$2T_h + 2T_x$	$2T_h + 2T_x$
Chang et al.'s [14]	User	$2T_h + T_x$	$9T_h + 5T_x$	$11T_h + 6T_x$
	GWN	$5T_h + 3T_x$	$10T_h + 4T_x$	$15T_h + 7T_x$
	Sensor	0	$4T_h + T_x$	$4T_h + T_x$
Yoon and Yoo's [15]	User	T_h	$3T_h + 2T_x + 2T_E$	$4T_h + 2T_x + 2T_E$
	GWN	$2T_h + 2T_x$	$4T_h$	$6T_h + 2T_x$
	Sensor	0	$3T_h + 2T_E$	$3T_h + 2T_E$
Choi et al.'s [18]	User	$T_h + T_F$	$10T_h + 2T_x + T_F + T_{enc} + 2T_E$	$11T_h + 2T_x + 2T_F + T_{enc} + 2T_E$
	GWN	$3T_h + 3T_x$	$10T_h + T_x + 2T_{enc}$	$13T_h + 4T_x + 2T_{enc}$
	Sensor	0	$6T_h + T_{enc} + 2T_E$	$6T_h + T_{enc} + 2T_E$
Proposed	User	$T_h + T_F$	$9T_h + 4T_x + T_F + 2T_E$	$10T_h + 4T_x + 2T_F + 2T_E$
	GWN	$5T_h + 3T_x$	$11T_h + 4T_x$	$16T_h + 7T_x$
	Sensor	0	$4T_h + T_x + 2T_E$	$4T_h + T_x + 2T_E$

7. Conclusions

To provide improved security functionality for mobile services in WSNs, several user authentication and key agreement schemes have been proposed in the last few years. However, most of them cannot provide secure authentication and are vulnerable to security attacks.

In this paper, we analyzed the security weaknesses of Chang et al.'s scheme and found that it is vulnerable to off-line password guessing attacks and does not provide forward secrecy and accurate password updates. To address the security problems, we proposed a biometric-based user authentication and key agreement scheme. The proposed scheme withstands the security attacks described above and provides better security functionality than previous schemes by using biometric information and ECC. In addition, we provided security and efficiency analyses, which demonstrated that the proposed protocol is more secure than the previous schemes and operationally viable in WSNs.

Acknowledgments: This study was supported by the BK21 Plus project funded by the Ministry of Education, Korea (21A20131600011).

Author Contributions: YoHan Park and YoungHo Park found the problems in the related schemes for WSNs, analyzed the vulnerabilities of the related schemes, designed the improved scheme, proved the security of proposed scheme and wrote the manuscript.

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

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114.
2. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330.
3. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gene Comput. Syst.* **2013**, *29*, 1645–1660.
4. Pathan, A.S.K.; Lee, H.W.; Hong, C.S. Security in wireless sensor networks: Issues and challenges. In Proceedings of the 8th International Conference Advanced Communication Technology (ICACT), Phoenix Park, Korea, 20–22 February 2006; pp. 1043–1048.
5. Perrig, A.; Stankovic, J.; Wagner, D. Security in wireless sensor networks. *ACM Commun.* **2004**, *47*, 53–57.
6. Al Ameen, M.; Liu, J.; Kwak, K. Security and privacy issues in wireless sensor networks for healthcare applications. *J. Med. Syst.* **2012**, *36*, 93–101.
7. 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; pp. 1–8.
8. Das, M.L. Two-factor user authentication scheme in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
9. 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.
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.* **2016**, *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. Yoon, E.J.; Yoo, K.Y. A biometric-based authenticated key agreement scheme using ECC for wireless sensor networks. In Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, 24–28 March 2014; pp. 699–705.
16. Das, A.K. A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wirel. Pers. Commun.* **2015**, *82*, 1377–1404.
17. Das, A.K. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *Int. J. Commun. Syst.* **2015**, *2015*, 1–25.
18. Choi, Y.; Lee, Y.; Won, D. Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. *Int. J. Dist. Sens. Netw.* **2016**, *8572410*, 1–16.
19. Park, Y.; Lee, S.; Kim, C.; Park, Y. Secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks. *Int. J. Dist. Sens. Netw.* **2016**, *12*, 1–11.
20. Li, C.T.; Hwang, M.S. An efficient biometric-based remote authentication scheme using smart cards. *J. Netw. Comp. Appl.* **2010**, *33*, 1–5.

21. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. Lond. A Math. Phys. Eng. Sci.* **1989**, *426*, 233–271.
22. Lu, R.; Cao, Z.; Chai, Z.; Liang, X. A Simple User Authentication Scheme for Grid Computing. *IJ Netw. Sec.* **2008**, *7*, 202–206.
23. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 523–540.
24. Tan, Z. A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2014**, *38*, 1–9.
25. Jung, J.; Kim, J.; Choi, Y.; Won, D. An Anonymous User Authentication and Key Agreement Scheme Based on a Symmetric Cryptosystem in Wireless Sensor Networks. *Sensors* **2016**, *16*, 1299.
26. 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.
27. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Advances in Cryptology-CRYPTO'99, Santa Barbara, CA, USA, 15–19 August 1999; Volume 1666, pp. 388–397.
28. Amin, R.; Biswas, G.P. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80.
29. He, D.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inf. Sci.* **2015**, *321*, 263–277.
30. Jiang, Q.; Kumar, N.; Ma, J.; Shen, J.; He, D.; Chilamkurti, N. A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks. *Int. J. Netw. Manag.* **2016**, doi:10.1002/nem.1937.
31. Lu, Y.; Li, L.; Yang, X.; Yang, Y. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. *PLoS ONE* **2015**, *10*, e0126323.
32. Liu, J.; Li, Q.; Yan, R.; Sun, R. Efficient authenticated key exchange protocols for wireless body area networks. *EURASIP J. Wirel. Commun. Netw.* **2015**, *2015*, 1–11.