Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 8572410, 16 pages
http://dx.doi.org/10.1155/2016/8572410

Research Article

Security Improvement on Biometric Based Authentication Scheme for Wireless Sensor Networks Using Fuzzy Extraction

Younsung Choi,[1] Youngsook Lee,[2] and Dongho Won[1]

[1]*Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 440-746, Republic of Korea*
[2]*Department of Cyber Investigation Police, Howon University, 64 Howon University 3Gil, Impi-Myeon, Gunsan-Si, Jeonrabuk-Do 573-718, Republic of Korea*

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 25 June 2015; Accepted 4 October 2015

Academic Editor: Hongxin Hu

Wireless sensor networks are used to monitor physical or environmental conditions. However, authenticating a user or sensor in wireless sensor networks is more difficult than in traditional networks owing to sensor network characteristics such as unreliable communication networks, resource limitation, and unattended operation. As a result, various authentication schemes have been proposed to provide secure and efficient communication. He et al. suggested a robust biometrics-based user authentication scheme, but Yoon and Kim indicated that their scheme had several security vulnerabilities. The latter then proposed an advanced biometrics-based user authentication scheme; in this paper, we analyze this advanced scheme and perform a cryptanalysis. Our analysis shows that Yoon and Kim's scheme has various security weaknesses such as a biometric recognition error, a user verification problem, lack of anonymity and perfect forward secrecy, session key exposure by the gateway node, vulnerability to denial of service attacks, and a revocation problem. Therefore, we suggest countermeasures that can be implemented to solve these problems and then propose a security-enhanced biometrics-based user authentication scheme using fuzzy extraction that conforms to the proposed countermeasures. Finally, we conduct a security analysis for the proposed biometrics-based user authentication scheme.

1. Introduction

Nowadays, numerous physical, chemical, and biological sensors are deployed in wireless sensor network (WSN) environments for various applications. These sensors can monitor a variety of conditions, including temperature, pressure, military surveillance, and real-time traffic conditions. One benefit of WSNs is that the sensors can be easily deployed in various kinds of harsh environments. Therefore, there has been a remarkable increase in the interest in WSNs [1]. WSNs generally consist of gateways, users, and sensors, and communication security is a momentous concern in real-world applications. Various authentication schemes for WSNs have been proposed for ensuring secure communication.

To support confidentiality and authentication for sensor networks, Watro et al. introduced a user authentication scheme employing the RSA and DH algorithms for WSNs in 2004. Wong et al. proposed a dynamic user authentication scheme that used a hash function [2]. But Tseng et al. indicated that Wong et al.'s authentication scheme has vulnerability to replay, stolen-verifier, and forgery attacks [3–7]. Das proposed a two-factor user authentication scheme based on a password and smart card to improve the security in 2009. Das demonstrated his scheme to be secure against various real-time attacks [6]. However, He et al. indicated that Das's scheme has vulnerability to insider attacks and impersonation attacks and that no provision was available for users to change their passwords. And also He et al. proposed an improved two-factor scheme to solve these security problems [8]. Khan and Alghathbar demonstrated that Das's scheme did not provide mutual authentication, and it has vulnerability to gateway bypassing and privileged-insider attacks [9]. Chen and Shih indicated that Das's scheme did not provide mutual authentication between the gateway and the sensor, and Chen and Shih proposed a robust mutual authentication scheme for WSNs and claimed that their scheme provides greater security than Das's scheme [10]. In 2010, Yuan et al. [11] proposed a biometric-based user

authentication scheme, but it was found to have various security problems. Yoon and Yoo pointed out that Yuan et al.'s scheme has vulnerability to insider, user impersonation, gateway node impersonation, and sensor node impersonation attacks. To address these problems, Yoon and Yoo proposed an improved user authentication scheme [12]. However, in 2012, He demonstrated that Yoon and Yoo scheme was still vulnerable to denial of service (DoS) and sensor impersonation attacks. The former then proposed an improved scheme to overcome these security problems [13].

In 2013, Yoon and Kim [14] indicated that even He et al.' scheme had various security vulnerabilities such as poor repair-ability and vulnerability to user and sensor node impersonation attacks. The former then proposed an advanced biometrics-based user authentication scheme for WSNs. They demonstrated that their scheme was more effective and had stronger security than other related schemes [13, 14]. To verify the security of Yoon and Kim's advanced scheme, we analyzed their scheme and performed a security cryptanalysis. We found that it has various security problems, including a biometric recognition error, a user verification problem, lack of anonymity and perfect forward secrecy, session key exposure by the gateway node, vulnerability to DoS attacks, and a revocation problem. To solve these problems, we first suggest appropriate countermeasures and then propose a biometrics-based user authentication scheme using fuzzy extraction with improved security that conforms to the proposed countermeasures. Moreover, we also conduct a security analysis of 16 security properties for the proposed biometrics-based user authentication scheme.

The remainder of this paper is organized as follows. Section 2 describes some related work to understand this paper. Section 3 explains Yoon and Kim's authentication scheme, and Section 4 analyzes their scheme to discuss the inherent security problems. Section 5 explains countermeasures to solve these problems. Section 6 proposes the biometric-based authentication using fuzzy extraction with improved security, and Section 7 presents a security analysis about 16 security properties for the proposed scheme. Section 8 concludes the paper.

2. Related Works

2.1. Attacker's Capability. Throughout this paper, we make the following assumptions about the capabilities of a probabilistic, polynomial-time attacker $\mathcal{A}$ in order to properly capture the security requirements of the two-factor authentication scheme that uses smart cards in WSNs [15].

(i) $\mathcal{A}$ has complete control over all message exchanges between the protocol participants, including a user, a sensor, and the gateway. That is, $\mathcal{A}$ can eavesdrop, insert, modify, intercept, and delete messages exchanged among the three parties at will.

(ii) $\mathcal{A}$ is able to (1) extract sensitive information from the smart card of a user through a power analysis attack or (2) determine the user's password possibly via shoulder-surfing or by employing a malicious card reader. However, it is assumed that $\mathcal{A}$ is unable to compromise both the information of the smart card and the password of the user. It is otherwise clear that there is no way to prevent $\mathcal{A}$ from impersonating the user if both factors have been compromised.

2.2. Elliptic Curves Cryptography. Elliptic Curves Cryptography (ECC) is a form of public-key cryptography that is based on the use of algebraic structures of elliptic curves over finite fields. Elliptic curves are also used in several integer factorization algorithms. The important benefit of ECC is that it provides a smaller key size, so ECC can maintain the same degree of security with a smaller key size than other public-key forms of cryptography, such as Rivest Shamir Adleman (RSA), Diffie-Hellman (DH), and Digital Signature Algorithm (DSA). Therefore, ECC is especially useful for wireless devices that are typically limited in terms of their computational ability, power, and network connectivity.

ECC has three related mathematical problems: an Elliptic Curve Discrete Logarithm Problem (ECDLP), Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP), and Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP). No polynomial time algorithm can solve the ECDLP, ECCDHP, and ECDDHP with nonnegligible probability.

Let $p > 3$ be a large prime and choose two field elements $a, b \in \mathbf{F}_p$ satisfying $4a^3 + 27b^2 \neq 0 \bmod p$ to define the equation of a nonsuper-singular elliptic curve $\mathbf{E}$: $y^2 = x^3 + ax + b \bmod p$ over $\mathbf{F}_p$. Choose a generator point $P = (x \times P, y \times P)$ whose order is a large prime number q over $\mathbf{E}(\mathbf{F}_p)$. The subgroup $\mathbf{G}$ of the elliptic curve group $\mathbf{E}(\mathbf{F}_p)$ with order q is constructed in the same way. Then, the three mathematical problems in ECC that are defined in several studies [16–18] are given as follows:

(i) ECDLP: given a point element Q in $\mathbf{G}$, find an integer $x \in \mathbf{Z}_q^*$ such that $Q = x \times P$, where $x \times P$ indicates that point P is added to itself x times through an operation with elliptic curves.

(ii) ECCDHP: for $a, b \in \mathbf{Z}_q^*$, given two point elements $a \times P, b \times P$ in $\mathbf{G}$, compute $a \times b \times P$ in $\mathbf{G}$.

(iii) ECDDHP: for $a, b, c \in \mathbf{Z}_q^*$, given three point elements $a \times P, b \times P$, and $c \times P$ in $\mathbf{G}$, decide whether $c \times P = a \times b \times P$.

In the proposed scheme, we use ECDLP for protecting r_i and r_s. In detail, a user sends $X_i = r_i \times P$ to the gateway, and the sensor node sends $Y_i = r_s \times P$ to the user for authentication and session key agreement. If an attacker knows r_i and r_s, he can attempt various attacks. However, the attacker cannot compute r_i and r_s due to ECDLP even if he steals X_i and Y_i from public communication. And in proposed scheme, we use ECCDHP for protecting $K_{US} = K_{SU} = r_i \times r_s \times P$. In other words, An attacker cannot compute K_{US} and K_{SU} due to ECCDHP even thought he know X_i and Y_i. Only legal user and sensor node can compute K_{US} and K_{SU}, respectively, using X_i and Y_i, and own random number. The user computes $K_{US} = r_i \times Y_i = r_i \times r_s \times P$ and the sensor node computes $K_{SU} = r_s \times X_i = r_s \times r_i \times P$.

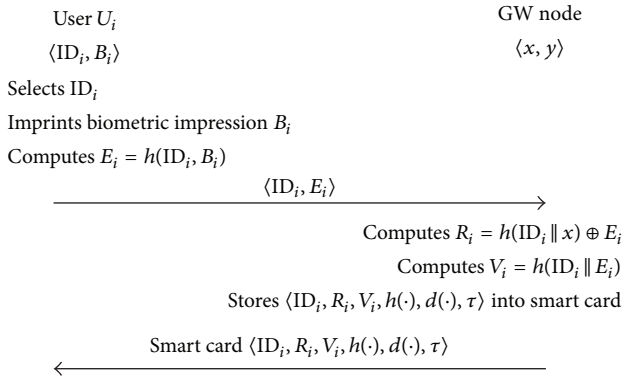


Figure 1: Registration phase of Yoon and Kim's scheme.

2.3. Fuzzy Extraction. The fuzzy extractor converts biometric information into a uniformly random string. Therefore, it is possible to apply cryptographic techniques for biometric security. The extractor consists of a pair of efficient randomized procedures, Gen (generate) and Rep (reproduce). Li et al.'s scheme uses $\text{Gen}(B_i) = (R_i, P_i)$ and $R_i = \text{Rep}(B_i', P_i)$ are used. The fuzzy extractor Gen generates R_i and P_i by using a user's biometric information during the registration phase. R_i is a uniform and random string, and P_i is the helper string; thus, R_i can be the same under the assistance of auxiliary information P_i even if the the biometric information that has been input changes, so long it maintains a reasonably similar status as the original biometric information. As a result, the fuzzy extraction is error-tolerant, and Rep reproduces the R_i using the biometric information that has been newly input B_i' and P_i in the login phase. To reproduce the same R_i, the metric space distances between B_i and B_i' have to meet the given verification threshold [19, 20].

The basic notation that is used consistently throughout this paper is shown in "Notations."

3. Review of Yoon and Kim's Authentication Scheme

Yoon and Kim's authentication scheme includes a registration phase and login and authentication phases. This scheme does not require making changes to a user's password because this scheme only uses biometrics. The gateway node has two master keys (x and y) and before starting the system, the gateway issues a long-term secret key $h(SID_j \| y)$ to sensor node S_j. x is then used for U_i. During the registration phase, the gateway issues a smart card stored as $h(ID_i \| x)$ to U_i [11].

3.1. Registration Phase. In the registration phase, a user U_i communicates securely with the gateway. U_i sends important information regarding the user's identification and biometrics. Figure 1 describes the registration phase, and detailed steps are given as follows. First, U_i chooses ID_i and imprints his biometrics B_i on the specific sensor device. Then, U_i computes $E_i = h(ID_i, E_i)$ and sends ID_i and E_i to GW node by using a secure channel. Then, GW node computes two values: $R_i = h(ID_i \| x) \oplus E_i$ and $V_i = h(ID_i \| E_i)$. And the GW node inputs $\langle ID_i, R_i, V_i, h(\cdot), d(\cdot), \tau \rangle$ into a smart card and sends the smart card to user U_i. $h(\cdot)$ is a one-way hash function. $d(\cdot)$ is a symmetric parametric function and τ is a predetermined threshold for the biometric verification.

3.2. Login and Authentication Phases. During the login and authentication phases, when U_i enters ID_i and B_i into a smart card terminal, the smart card must validate the legitimacy of U_i. Then, U_i, S_j, and GW authenticate each other. This scheme uses three messages $\langle M_1, M_2, M_3 \rangle$ during authentication, as shown in Figure 2. Finally, U_i and S_j share the session key sk after the login and authentication phases; U_i and S_j communicate with each other using the session key sk.

(1) U_i inserts his smart card into the card reader and imprints his biometric B_i on a specific device to verify the user's biometrics.

(2) The smart card computes two values: $E_i^* = h(ID_i, B_i)$ and $V_i^* = h(ID_i \| E_i^*)$.

(3) The smart card compares the computed V_i^* and the V_i that is stored in the smart card. If $d(V_i, V_i^*) \leq \tau$, the user's smart card stops the login phase. Otherwise, the smart card generates a random number r_i.

(4) The smart card computes three values: $D_i = R_i \oplus E_i^*$, $k_i = h(D_i \| T_i)$, and $C_i = E_{k_i}(ID_i \| r_i)$, where T_i is the current timestamps.

(5) Then, the smart card sends the login message $M_1 = \langle ID_i, C_i, T_i \rangle$ to the GW node.

GW receives M_1 and executes the following actions.

(1) The GW node checks the freshness of T_i by using $(T' - T_i) \leq \Delta T$. ΔT is the expected time interval for the transmission delay. If T_i is not fresh, the GW node rejects the user's request.

(2) The GW node computes three values: $D_i' = h(ID_i \| x)$, $k_i' = h(D_i' \| T_i)$, and $ID_i' \| r_i' = D_{k_i'}(C_i)$.

(3) The GW node checks whether ID_i and ID_i' are equal. If they are not equal, the GW node stops the session. Otherwise, the GW node picks up the current timestamps T_g.

(4) The GW node computes two values: $k_g = h(h(SID_j \| y) \| T_g)$ and $C_g = E_{k_g}(ID_i' \| r_i')$.

(5) The GW node then computes $M_2 = \langle ID_i, C_g, T_g \rangle$ and sends it to the sensor node S_j.

S_j receives M_2 and performs the following actions.

(1) S_j checks the freshness of T_g using $(T'' - T_g) \leq \Delta T$.

(2) S_j computes two values: $k_g' = h(h(SID_j \| y) \| T_g)$ and $ID_i'' \| r_i'' = D_{k_g'}(C_g)$.

(3) S_j checks whether ID_i and ID_i'' are equal and S_j picks up the current timestamps T_s.

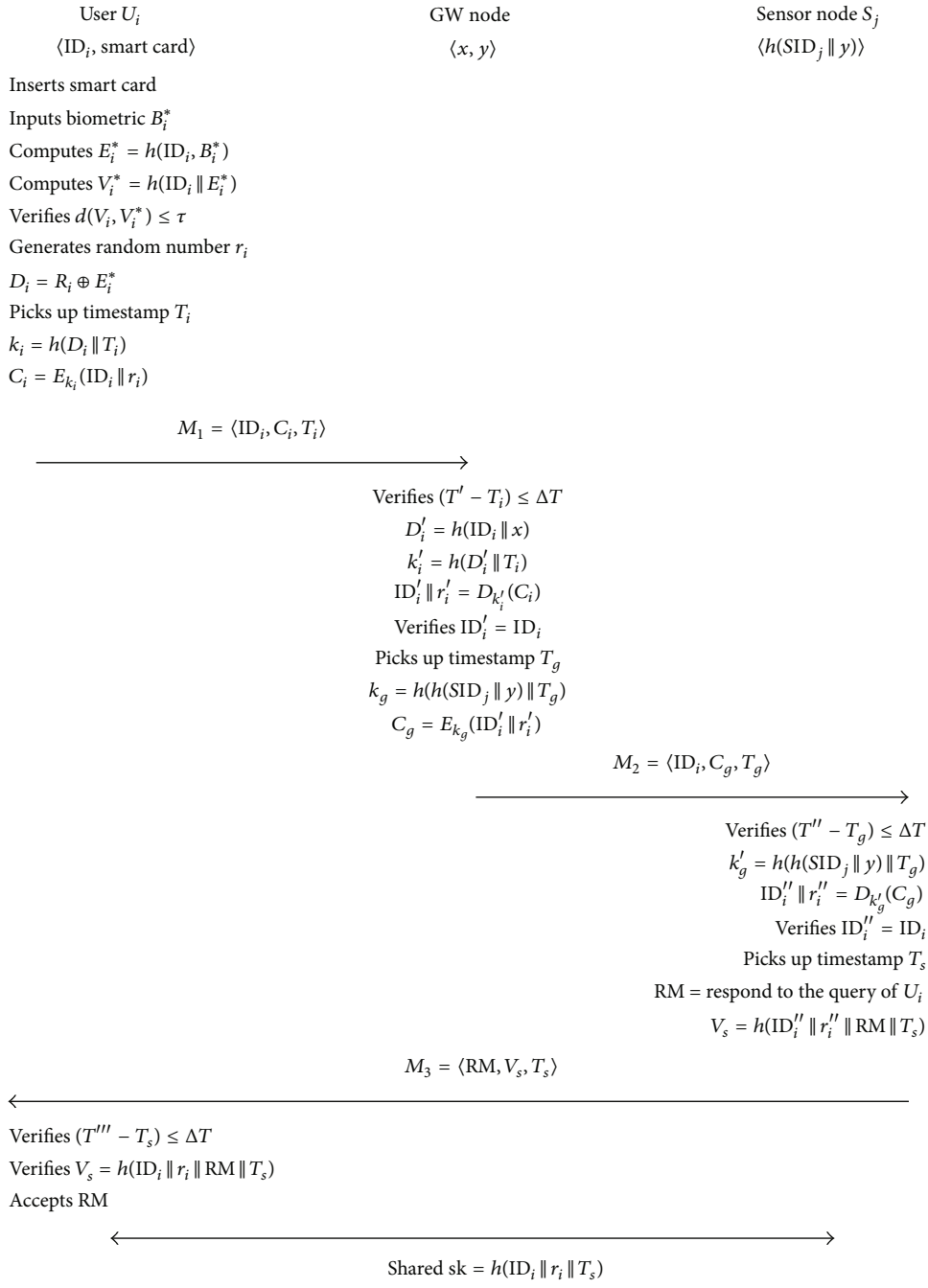


Figure 2: Login and authentication phase of Yoon and Kim's scheme.

(4) S_j generates RM and computes V_s. RM is response to the query of U_i; $V_s = h(\mathrm{ID}_i'' \| r_i'' \| \mathrm{RM} \| T_s)$.

(5) S_j computes $M_3 = \langle \mathrm{RM}, V_s, T_s \rangle$ and sends it to the user U_i.

U_i receives M_3 and executes the following actions.

(1) U_i checks the freshness of T_s using $(T''' - T_s) \le \Delta T$.

(2) U_i computes $V_s' = h(\mathrm{ID}_i \| r_i \| \mathrm{RM} \| T_s)$ and checks whether V_s' and V_s are the same.

(3) If the entire authentication phase finishes without any problems, U_i accepts RM.

(4) U_i and S_j communicate with each other securely using the session key sk, and U_i and S_j compute $\mathrm{sk} = h(\mathrm{ID}_i \| r_i \| T_s)$.

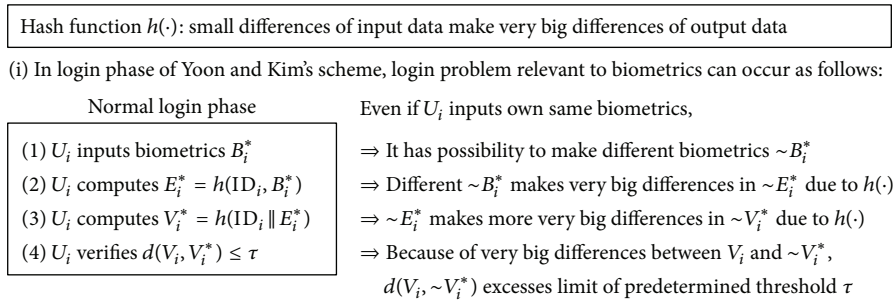


Figure 3: Biometric recognition error on Yoon and Kim's scheme.

4. Cryptanalysis of Yoon and Kim's Authentication Scheme

4.1. Biometric Recognition Error. Yoon and Kim's authentication scheme uses a one-way hash function to provide biometric verification. This hash function can be used to map the data of an arbitrary size to data of a fixed size with slight differences in the input data producing very large differences in the output data. Figure 3 describes the biometric recognition error in Yoon and Kim's scheme. Biometrics have general limitations such as false acceptance and false rejection. This means that the output of the imprinted biometrics is not always constant. Although U_i inputs its own biometrics to the scanning device, it is possible to output a different $\sim B_i^*$. Therefore, the same biometrics can generate different output, such as the B_i^* and $\sim B_i^*$. A different $\sim B_i^*$ causes slight differences in E_i^* and $\sim E_i^*$. Therefore, this difference produces a very large difference between V_i^* and $\sim V_i^*$ due to the property of hash function. The large difference between V_i^* and $\sim V_i^*$ causes a biometric recognition error, so a legal user can fail to accept the smart card verification. As a result, advanced techniques are needed to improve the success rate of a legal user's verification [5].

4.2. User Verification Problem. In Yoon and Kim's authentication scheme, GW verifies a legal user by comparing ID_i in M_1 and ID_i' in the output of the decrypted C_i. Specifically, the user computes C_i using a symmetric encryption algorithm; $C_i = E_{k_i}(\mathrm{ID}_i \| r_i)$. ID_i and r_i do not matter but E_{k_i} has a problem in that there is a possibility to obtain unexpected results. This is the reason why E_{k_i} is made up by D_i and T_i and D_i consist of R_i, E_i^*:

$$\begin{aligned} E_i^* &= h\left(\mathrm{ID}_i, B_i\right), \\ D_i &= R_i \oplus E_i^*, \\ k_i &= h\left(D_i \| T_i\right), \\ C_i &= E_{k_i}\left(\mathrm{ID}_i \| r_i\right). \end{aligned} \tag{1}$$

Even if biometrics are the same, the output of the scanning device is not constant. Therefore, the same biometrics can generate a different output, like $\sim B_i^*$. The different output of the biometrics causes slight differences in E_i^* and $\sim E_i^*$. Due to these slight differences, different $\sim D_i$ and $\sim k_i$ are produced. As a result, the user and GW en/decrypt the C_i using different keys. GW cannot get a normal ID_i from C_i so the user is not authenticated by GW even when the user uses its own normal ID_i and B_i. This is the reason why the hash function and the symmetric key encryption algorithm have a property that results in large differences due to a slight difference of input. Figure 4 specifically describes the user verification problem in Yoon and Kim's scheme.

4.3. Lack of Anonymity. Figure 5 describes how Yoon and Kim's scheme does not provide the anonymity. In this scheme, the user sends its own U_i to GW over public communication, and GW sends U_i to the sensor without any protection. Therefore, an attacker can easily acquire U_i from those communications. This results in an information exposure problem. For the GW' incoming communication, an attacker can obtain information of the approximate number of registered users to GW. Also an attacker can acquire information on which user communicates with S_j. Therefore, the lack of anonymity in Yoon and Kim's scheme raises some problems that need to be addressed by providing user anonymity through a protection technique. To solve this problem, it is necessary to use anonymity identification AID_i in the WSNs communication instead of sending a normal ID_i [21–23].

4.4. Lack of Perfect Forward Secrecy. Perfect forward secrecy means that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised at some point in the future [24]. Unfortunately, Yoon and Kim's authentication scheme does not provide perfect forward secrecy. Therefore, an attacker can compute the session key sk between the U_i and S_j if the attacker knows one of the long-term keys D_i in the future. The following list describes how Yoon and Kim's scheme does not provide perfect forward secrecy [5]:

(1) Attacker got C_i, T_i, and T_s in previous public channel.

(2) Attacker knew one of user's long-term secret: D_i.

$\Rightarrow$ Attacker has C_i, T_i, T_s, and D_i and computes k_i and D_{k_i} as follows:

$\Rightarrow$ $k_i = h(D_i \| T_i)$,

Hash function $h(\cdot)$: small differences of input data make very big differences of output data

Symmetric key algorithm E_{k_i}: small differences of K_i make big differences in the output of E_{k_i}

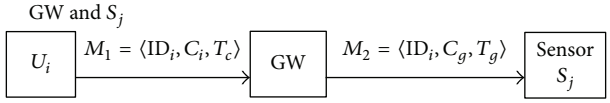
(i) In authentication phase of Yoon and Kim's scheme, user verification problem can occur as follows:

Normal authentication phase	
(1) U_i generates random number r_i	Even if U_i inputs own same biometrics, it has possibility to make different biometrics $\sim B_i^*$
(2) U_i computes $D_i = R_i \oplus E_i^*$	$\Rightarrow$ Different $\sim B_i^*$ make some different $\sim E_i^*$ ($\sim E_i^* = h(\mathrm{ID}_i, \sim B_i^*)$)
(3) U_i generates timestamp T_i	So, different $\sim E_i^*$ make some different $\sim D_i$ ($\sim D_i = R_i \oplus \sim E_i^*$)
(4) U_i computes $k_i = h(D_i \| T_i)$	$\Rightarrow$ Different $\sim D_i$ makes very big differences in $\sim k_i$ due to $h(\cdot)$
(5) U_i computes $C_i = E_{k_i}(\mathrm{ID}_i \| r_i)$	$\Rightarrow$ $\sim k_i$ makes more very big differences in $\sim C_i^*$ due to E_{k_i}
(6) U_i sends $M_1 = \langle \mathrm{ID}_i, C_i, T_i \rangle$	$\Rightarrow$ U_i sends $\sim C_i$ to GW but $\sim C_i$ is very different to normal C_i
(7) GW receives $M_1 = \langle \mathrm{ID}_i, C_i, T_i \rangle$	
(8) GW verifies $(T' - T_i) \leq \Delta T$	
(9) GW computes $D_i' = h(\mathrm{ID}_i \| x)$	$\Rightarrow$ There are differences between D_i' and $\sim D_i$
(10) GW computes $k_i' = h(D_i' \| T_i)$	$\Rightarrow$ There are more differences between k_i' and $\sim k_i$
(11) GW computes $\mathrm{ID}_i' \| r_i' = D_{k_i}(C_i)$	$\Rightarrow$ Due to differences k_i' and $\sim k_i$, $\sim D_{\sim k_i}(C_i)$ is not same to $\mathrm{ID}_i' \| r_i'$
(12) GW verifies $\mathrm{ID}_i' = \mathrm{ID}_i$	$\Rightarrow$ $\sim \mathrm{ID}_i' \| \sim r_i' = \sim D_{\sim k_i}(C_i)$; output $\sim \mathrm{ID}_i'$ is not same to received ID_i

(ii) So, it is possible that GW cannot verify user's legal ID_i even if U_i sends own M_1 because of inaccurate $\sim B_i^*$

Figure 4: User verification problem on Yoon and Kim's scheme.

(i) Attacker can acquire ID_i in communication between U_i and GW, GW and S_j



$\Rightarrow$ Attacker can know how many user are registered in GW

$\Rightarrow$ Attacker can know which user wants to access to sensor S_j

(ii) Therefore, Yoon and Kim's scheme basically does not provide user anonymity

Figure 5: Lack of anonymity on Yoon and Kim's scheme.

$$\Rightarrow D_{k_i}(C_i) = \mathrm{ID}_i \parallel r_i.$$

(3) Attacker acquires ID_i and r_i and then computes sk.

$$\Rightarrow \mathrm{sk} = h(\mathrm{ID}_i \parallel r_i \parallel T_s).$$

(4) Attacker can compute all of previous session key sk.

In advance, the attacker obtains C_i, T_i, and T_s from previous communication between U_i and S_j. The attacker obtains one of the user's long-term secrets D_i. Then, the attacker can compute $k_i = h(D_i \parallel T_i)$ and decrypt the C_i using the computed k_i. So the attacker can figure out ID_i and random number r_i. Finally, the attacker can compute the session key sk using ID_i, r_i, and T_s.

4.5. Session Key Exposure by the Gateway Node. The session key sk is used to provide secure communications between U_i and S_j after the authentication phase is successfully finished. Even if the GW node is a trusted node, it is not necessary for the GW node to know sk because U_i usually wants to communicate secretly with S_j without the observation of the GW node. However, in Yoon and Kim's authentication scheme, the GW node can compute sk without difficulty. GW node can collect previous ID_i and r_i in U_i's authentication phase and thus can obtain all T_s over a public channel. Then, the GW node can compute all $\mathrm{sk} = h(\mathrm{ID}_i \parallel r_i \parallel T_s)$ between U_i and S_j. Therefore, the GW node can decrypt the encrypted message between U_i and S_j, and can figure out all U_i's secret messages that are protected by session key sk. The session key exposure by GW on Yoon and Kim's scheme [5] is described as follows:

(1) GW knew ID_i and r_i in communication with U_i.

(2) GW got T_s in public channel:

$$\Rightarrow \text{GW has } \mathrm{ID}_i, r_i \text{ and } T_s,$$
$$\Rightarrow \mathrm{sk} = h(\mathrm{ID}_i \parallel r_i \parallel T_s).$$

(3) GW can compute all session key sk between U_i and S_j.

(4) GW can decrypt the secret messages between U_i and S_j.

(5) GW can acquire important information between U_i and S_j.

4.6. Vulnerability to Denial of Service Attack. Figure 6 shows the potential for a DoS attack on Yoon and Kim's authentication scheme. The attacker can send malicious messages $\langle \mathrm{ID}_i, C_i, T_c \rangle$ that have been generated to consume the battery power of the GW node and sensor node. The attacker obtains ID_i and C_i from the previous public channel communication and generates a current timestamps T_c. When the GW node and the sensor node receive the malicious messages $\langle \mathrm{ID}_i, C_i, T_c \rangle$, they first check for the freshness of the timestamps T_c. However, the T_c generated by the attacker is current,

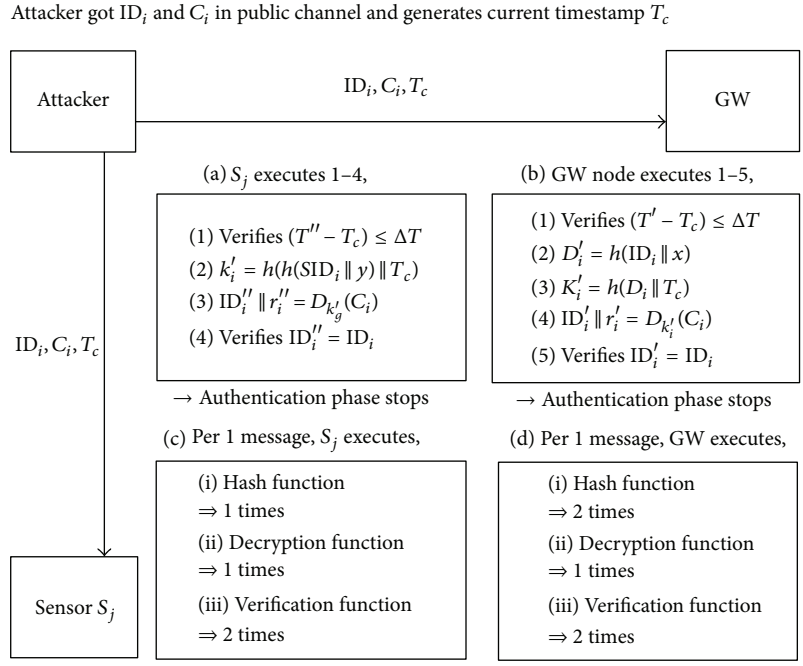


Figure 6: Vulnerability to denial of service attack on Yoon and Kim's scheme.

and the GW node and sensor cannot determine that ID_i and C_i are from previous messages. So they execute various functions, such as a hash function, decryption function, and verification function before checking whether the ID_i sent by the attacker and the computed ID_i are the same. Therefore, the attacker is able to execute a DoS attack without difficulty [5]. The GW node has sufficient resources that can be used in the system, but the sensors are different. The sensor nodes have a limited computational ability, low battery, low bandwidth, and a small amount of memory. The computational cost of a sensor node is a critical consideration in the design of WSNs since this increases the consumption of the battery power of the sensor [15]. Quite often it is economically preferable to discard a sensor rather than recharge it, and for this reason, the battery power of a sensor is usually important for wireless devices, with its lifetime determining the sensor lifetime. Therefore, it is significant issue for the sensor node to be protected from a DoS attack.

4.7. Revocation Problem. Yoon and Kim's authentication scheme does use the user's password but only uses the user's ID_i and biometrics B_i. Therefore, a password change phase is not necessary. For this reason, when an attacker steals or picks up the user's smart card, a revocation problem occurs. When the GW node issues the user's smart card, it always produces the same $\langle ID_i, R_i, V_i, h(\cdot), d(\cdot), \tau \rangle$ if the U_i sends the same ID_i and B_i to the GW node. So even though U_i reissues a new smart card, U_i cannot discard the lost smart card because the reissued smart card and the lost smart card are the same. Therefore, the user U_i has to change his ID in order to reissue a different smart card. Figure 7 describes the potential problem due to lack of revocation phase on Yoon and Kim's scheme [5].

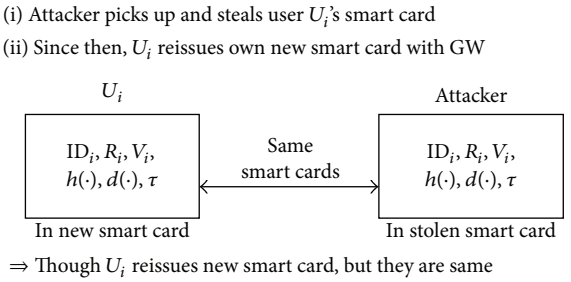


Figure 7: Revocation problem on Yoon and Kim's scheme.

5. Countermeasures

The vulnerability of Yoon and Kim's scheme to a biometric recognition error and a user verification problem is due to the fact that

(i) though the same U_i inputs its own biometrics to the scanning device, a different output can be generated;

(ii) the hash function makes slight differences in the input data by producing very large differences in the output data;

(iii) in a symmetric key encryption algorithm E_{k_i}, small differences of k_i produce large differences in the output.

This design flaw causes normal users to fail the login phase using smart card. Therefore, we suggest modifying the $E_i^* = h(ID_i, B_i^*)$, $V_i^* = h(ID_i \| E_i^*)$, and $d(V_i, V_i^*) \le \tau$ to prevent a biometric recognition error. Moreover, the difference in B_i and B_i^* results in a different encryption key k_i. So, this can cause a user verification problem because

the differences in K_i and K_i' produce a different C_i that is used for authentication between U_i and the GW node. To prevent an authentication error, we also suggest modifying $D_i = R_i \oplus E_i^*$ and $k_i = h(D_i \parallel T_i)$. We thus improve Yoon and Kim's scheme using fuzzy extraction as follows.

During the registration phase, instead of $E_i^* = h(\mathrm{ID}_i, B_i^*)$, the smart card computes R_i and P_i using a fuzzy extraction function $\mathrm{Gen}(B_i)$ such as $\langle R_i, P_i \rangle = \mathrm{Gen}(B_i)$. It also computes $F_i = h(R_i)$ and sends both ID_i and F_i to the GW node. The GW node modifies the computation of A_i and V_i from $A_i = h(\mathrm{ID}_i \parallel x) \oplus E_i$ and $V_i = h(\mathrm{ID}_i \parallel E_i)$ to

$$\begin{aligned} A_i &= h\left(R_i\right), \\ M_i &= h\left(\mathrm{ID}_i \parallel x\right) \oplus A_i, \\ V_i &= h\left(\mathrm{ID}_i \parallel A_i\right). \end{aligned} \tag{2}$$

During the login and authentication phase, instead of $E_i^* = h(\mathrm{ID}_i, B_i^*)$, $V_i^* = h(\mathrm{ID}_i \parallel E_i^*)$, $d(V_i, V_i^*) \le \tau$, and $D_i = A_i \oplus E_i^*$, smart card computes the following:

$$\begin{aligned} R_i^* &= \mathrm{Rep}\left(B_i^*, P_i\right), \\ A_i^* &= h\left(R_i^*\right), \\ V_i^* &= h\left(\mathrm{ID}_i \parallel A_i^*\right), \\ V_i &\stackrel{?}{=} V_i^*, \\ D_i &= M_i \oplus A_i^*. \end{aligned} \tag{3}$$

As a result of this modification carried out using a fuzzy extraction function, the accuracy of verification using biometrics improves. Consequently, the biometric recognition error and user verification problem can be solved.

We next present a possible mechanism for eliminating the vulnerability in Yoon and Kim's scheme due to the lack of anonymity. This vulnerability is due to the fact that

(i) ID_i is used in public communication without any protection;

(ii) the attacker can know how many users are registered in GW and which user wants to access S_j.

Using the user's ID_i, the attacker can acquire a variety of information on the user, GW, and the sensor. Therefore, we propose to use an anonymity AID_i to provide anonymity. Instead of sending a normal ID_i, we suggest using AID_i in the communication as follows:

$$\mathrm{AID}_i = \mathrm{ID}_i \oplus h\left(h\left(x \parallel y\right) \parallel T_i\right). \tag{4}$$

GW sends $h(x \parallel y)$ to a user using $N_i = h(x \parallel y) \oplus A_i$ in registration phase. $h(x \parallel y)$ uses only a previous secret x and y so it does not need to add a new secret. The attacker and sensor cannot know $h(x \parallel y)$ and AID_i changes every session due to T_i so we can provide user anonymity.

To provide the perfect forward secrecy in our proposed scheme, we modify the computation of sk from $\mathrm{sk} = h(\mathrm{ID}_i \parallel r_i \parallel T_s)$ to

$$\mathrm{sk} = h\left(\mathrm{AID}_i \parallel K_{US} \parallel T_s\right). \tag{5}$$

$\mathrm{AID}_i = \mathrm{ID}_i \oplus h(h(x \parallel y) \parallel T_i)$ has a secret $h(x \parallel y)$. Therefore, sk has two secret $h(x \parallel y)$ and K_{US}; moreover, they are independent on each other, and so the session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future. K_{US} is thus important information between U_i and S_j:

$$K_{US} = r_i \times r_s \times P. \tag{6}$$

Since K_{US} can be used to eliminate the vulnerability in Yoon and Kim's scheme to session key exposure by the GW node, this vulnerability is due to the fact that

(i) the GW node can know all elements of sk including r_i;

(ii) it is hard to share secret information between U_i and S_j in advance.

To prevent this problem, we suggest a key exchange using elliptic curve encryption. The user generates r_i and computes X_i and K_{US}:

$$\begin{aligned} X_i &= r_i \times P, \\ K_{US} &= r_i \times Y_i = r_i \times r_s \times P. \end{aligned} \tag{7}$$

Then, the user sends X_i to the sensor through the GW and receives Y_i from the sensor, so the sensor can compute sk as follows:

$$\begin{aligned} Y_i &= r_s \times P, \\ K_{SU} &= r_s \times X_i = r_s \times r_i \times P. \end{aligned} \tag{8}$$

K_{US} and K_{SU} can be used by the user and sensor to compute $\mathrm{sk} = h(\mathrm{AID}_i \parallel K_{US} \parallel T_s)$ in a manner that is concealed from the GW node. Therefore, we resolve the session key exposure by the GW node.

However, even after implementing the modifications described above, Yoon and Kim's scheme is vulnerable to DoS attacks. This type of attack results from the fact that

(i) GW and the sensor perform all operations without checking the freshness of the incoming messages;

(ii) in particular, the sensor has limited energy but this scheme verifies messages after performing various operations.

To prevent the vulnerability to DoS attack, we suggest adding verification to M_1 and M_2 to check incoming message. We modify the computations for M_1 and M_3 from $M_1 = \langle \mathrm{ID}_i, C_i, T_i \rangle$ and $M_2 = \langle \mathrm{ID}_i, C_g, T_g \rangle$ to

$$\begin{aligned} M_1 &= \left\langle \mathrm{AID}_i, X_i, C_i, T_i, W_i \right\rangle, \\ M_2 &= \left\langle \mathrm{AID}_i, C_g, T_g, W_g \right\rangle. \end{aligned} \tag{9}$$

In advance, the GW node and sensor check W_i and W_g to verify an incoming message:

$$\begin{aligned} W_i &= h\left(h\left(x \parallel y\right) \parallel \mathrm{AID}_i \parallel X_i \parallel C_i \parallel T_i\right), \\ W_g &= h\left(h\left(\mathrm{SID}_j \parallel y\right) \parallel \mathrm{AID}_i \parallel C_g \parallel T_g\right). \end{aligned} \tag{10}$$

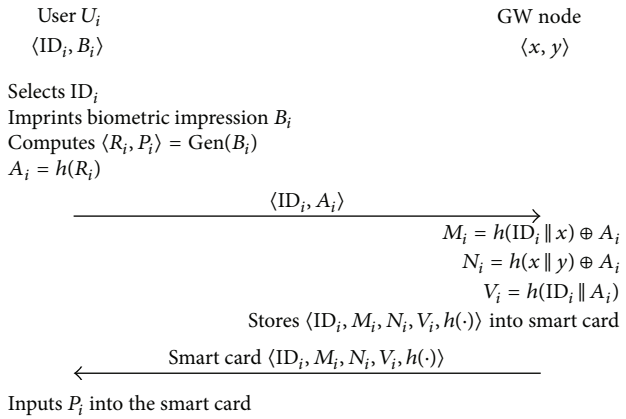


Figure 8: Registration phase of the proposed scheme.

W_i and W_g are thus only computed for a legal user and the GW node. Due to this modification, the GW node and sensor can prevent a DoS attack by checking W_i and W_g in M_1 and M_2, respectively.

Finally, the revocation problem is prevented by implementing a revocation and reissue phase. This phase should also be modified for consistency, particularly to induce a user to select identification different from previous identifications (see Section 6.3 for details). All the modifications suggested above are combined to propose an improved authentication scheme that is described in the following section.

6. Proposed Scheme

Our proposed scheme is divided into three phases: a user registration phase, login and authentication phase, and revocation and reissue phase. Before our scheme is executed, GW generates two master keys, x and y, and provides a long-term secret key $h(SID_j \| y)$ to the sensor S_j.

6.1. Registration Phase. The registration phase of the proposed scheme is described in Figure 8. The U_i perform a user registration phase with GW by using a secure channel. U_i computes B_i by using a biometrics scanning device and R_i and P_i using fuzzy extraction. Then, the user's information is sent to the GW for registration. However, the GW cannot store the user's biometrics information. The detailed steps are as follows.

(1) U_i selects ID_i and scans its own biometrics to compute B_i.

(2) U_i computes $\langle R_i, P_i \rangle = Gen(B_i)$ and $A_i = h(R_i)$. And then U_i sends $\langle ID_i, A_i \rangle$ to the GW node.

(3) After receiving $\langle ID_i, A_i \rangle$, the GW node computes the authentication parameters for U_i as follows:

$$\begin{aligned} M_i &= h(ID_i \| x) \oplus A_i, \\ N_i &= h(x \| y) \oplus A_i, \\ V_i &= h(ID_i \| A_i). \end{aligned} \tag{11}$$

(4) GW stores ID_i, $h(\cdot)$, and the authentication parameters; $\langle ID_i, M_i, N_i, V_i, h(\cdot) \rangle$ in a smart card. And GW issues the smart card to U_i through a secure channel.

(5) U_i receives the smart card and inputs P_i to the smart card.

6.2. Login and Authentication Phases. The login and authentication phase of the proposed scheme is described in Figure 9. During the login phase, the smart card checks the user's legality by using the user's U_i and biometrics B_i^*. GW authenticates the user by checking ID_i through the detailed steps of the login phase as follows:

(1) U_i inserts his smart card into a card reader. Then, U_i inputs his ID_i and computes the biometric information B_i^* using a scanning device. The smart card computes R_i^*, A_i^*, and V_i^* using fuzzy extraction and compares V_i^* with V_i stored in smart card as follows:

$$\begin{aligned} R_i^* &= \mathrm{Rep}(B_i^*, P_i), \\ A_i^* &= h(R_i^*), \\ V_i^* &= h(ID_i \| A_i^*), \\ \text{verifies } V_i &\stackrel{?}{=} V_i^*. \end{aligned} \tag{12}$$

(2) The smart card generates a random number r_i and computes X_i, D_i, and $h(x \| y)$. X_i is used for the session key between U_i and S_j. This scheme uses $h(x \| y)$ to provide perfect forward secrecy:

$$\begin{aligned} X_i &= r_i \times P, \\ D_i &= M_i \oplus A_i^*, \\ h(x \| y) &= N_i \oplus A_i^*. \end{aligned} \tag{13}$$

(3) U_i picks up the current timestamps T_i and computes k_i, C_i, AID_i, and W_i for authentication with GW. W_i is used to prevent the DoS attack. Then, U_i sends the authentication message M_1 to GW:

$$\begin{aligned} k_i &= h(D_i \| T_i), \\ C_i &= E_{k_i}(ID_i \| X_i), \\ AID_i &= ID_i \oplus h(h(x \| y) \| T_i), \\ W_i &= h(h(x \| y) \| AID_i \| X_i \| C_i \| T_i), \\ M_1 &= \langle AID_i, X_i, C_i, T_i, W_i \rangle. \end{aligned} \tag{14}$$

(4) Upon receiving M_1 from U_i, GW retrieves the current timestamps T' and verifies the freshness of the U_i' timestamps T_i using $(T' - T_i) \le \Delta T$. Then, GW verifies the received W_i using $W_i \stackrel{?}{=} h(h(x \| y) \| AID_i \| X \| C_i \| T_i)$.

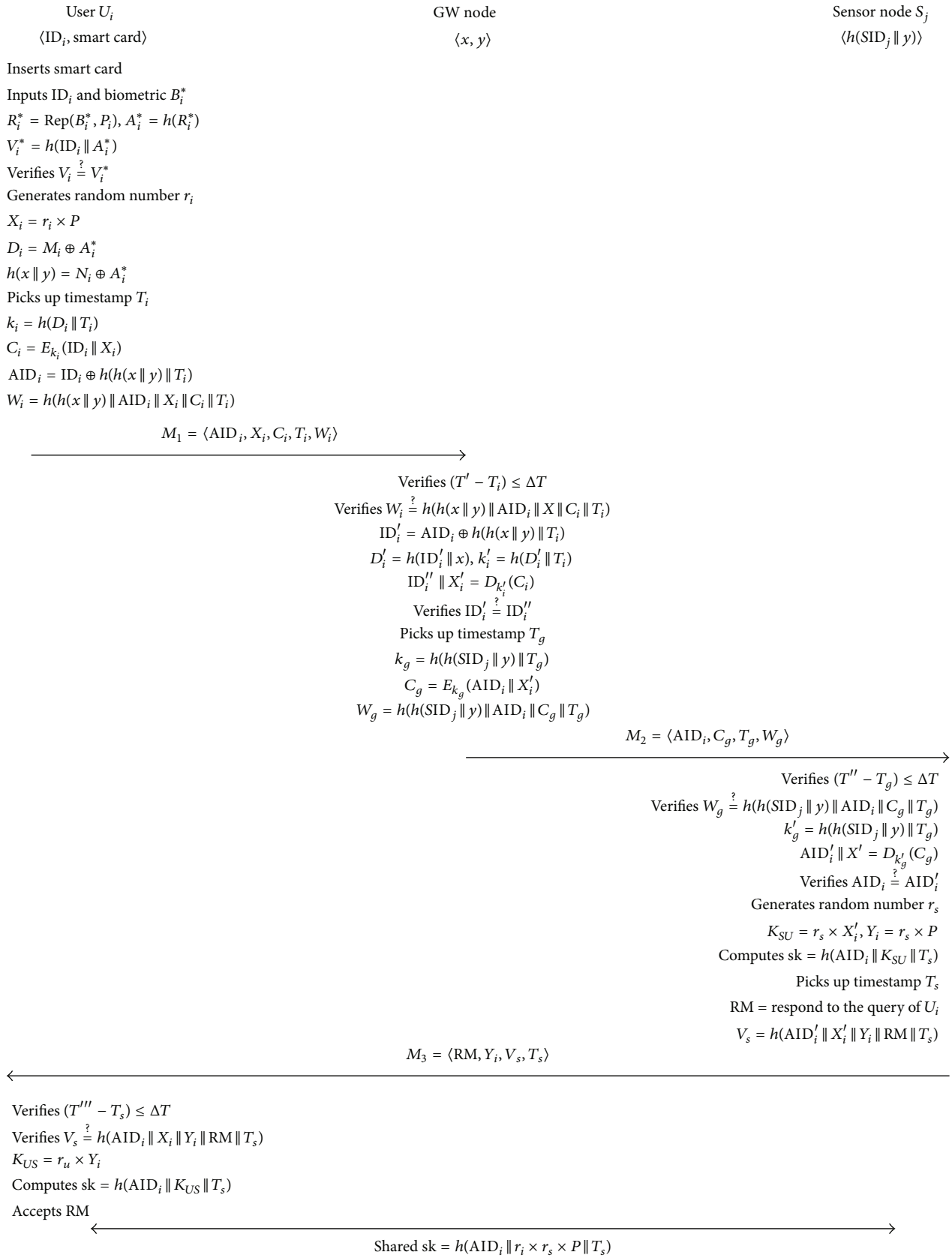


Figure 9: Login and authentication phase of the proposed scheme.

(5) GW computes ID_i', D_i', and k_i' and decrypts the C_i in M_1. The output of the decryption is used by GW to authenticate U_i by comparing ID_i' and ID_i'':

$$\begin{aligned} \mathrm{ID}_i' &= \mathrm{AID}_i \oplus h\left(h\left(x \parallel y\right) \parallel T_i\right), \\ D_i' &= h\left(\mathrm{ID}_i' \parallel x\right), \\ k_i' &= h\left(D_i' \parallel T_i\right), \\ \mathrm{ID}_i'' \parallel X_i' &= D_{k_i'}\left(C_i\right), \\ \text{verifies } \mathrm{ID}_i' &\stackrel{?}{=} \mathrm{ID}_i''. \end{aligned} \tag{15}$$

(6) GW picks up the current timestamps T_g and k_g, C_g, and W_g to authenticate S_j. Then, GW sends an authentication message M_2 to S_j:

$$\begin{aligned} k_g &= h\left(h\left(\mathrm{SID}_j \parallel y\right) \parallel T_g\right), \\ C_g &= E_{k_g}\left(\mathrm{AID}_i \parallel X_i'\right), \\ W_g &= h\left(h\left(\mathrm{SID}_j \parallel y\right) \parallel \mathrm{AID}_i \parallel C_g \parallel T_g\right), \\ M_2 &= \left\langle \mathrm{AID}_i, C_g, T_g, W_g \right\rangle. \end{aligned} \tag{16}$$

(7) With M_2 in hand, S_j retrieves the current timestamps T'' and checks if $T'' - T_g \le \Delta T$. If the freshness check for T'' fails, S_j stops the authentication phase. Otherwise, S_j computes W_g and compares it with $h(h(\mathrm{SID}_j \parallel y) \parallel \mathrm{AID}_i \parallel C_g \parallel T_g)$. After that, S_j computes k_g' and decrypts C_g using k_g'. Then, S_j checks the sameness between the received AID_i and the computed AID_i':

$$\begin{aligned} k_g' &= h\left(h\left(\mathrm{SID}_j \parallel y\right) \parallel T_g\right), \\ \mathrm{AID}_i' \parallel X' &= D_{k_g'}\left(C_g\right), \\ \text{verifies } \mathrm{AID}_i &\stackrel{?}{=} \mathrm{AID}_i'. \end{aligned} \tag{17}$$

(8) To compute sk, S_j generates a random number r_s and computes K_{SU}, Y_i. This sk is computed only by U_i and S_j due to a mathematical problem inherent to ECC. Then, S_j picks up the current timestamps T_s and computes RM, V_s for the authentication. After the operation is finished, S_j send M_3 to U_i:

$$\begin{aligned} K_{SU} &= r_s \times X_i', \\ Y_i &= r_s \times P, \\ \text{computes sk} &= h\left(\mathrm{AID}_i \parallel K_{SU} \parallel T_s\right), \\ \mathrm{RM} &= \text{Response to the query of } U_i, \\ V_s &= h\left(\mathrm{AID}_i' \parallel X_i' \parallel Y_i \parallel \mathrm{RM} \parallel T_s\right), \\ M_3 &= \left\langle \mathrm{RM}, Y_i, V_s, T_s \right\rangle. \end{aligned} \tag{18}$$

(9) First, U_i checks the freshness in T_s and the sameness of V_s. Then, U_i computes K_{US} using the received Y_i, and sk by using AID_i, K_{US}, and T_s. Only a legal U_i computes K_{US} using Y_i so anyone cannot compute sk including GW. Then, U_i accepts RM:

$$\begin{aligned} &\text{verifies } \left(T''' - T_s\right) \le \Delta T, \\ &\text{verifies } V_s \stackrel{?}{=} h\left(\mathrm{AID}_i \parallel X_i \parallel Y_i \parallel \mathrm{RM} \parallel T_s\right), \\ &K_{US} = r_i \times Y_i, \\ &\text{computes sk} = h\left(\mathrm{AID}_i \parallel K_{US} \parallel T_s\right), \\ &\text{accepts RM.} \end{aligned} \tag{19}$$

(10) From now on, U_i can communicate securely with S_j using sk:

$$\begin{aligned} \text{shared sk} &= h\left(\mathrm{AID}_i \parallel r_i \times r_s \times P \parallel T_s\right), \\ h\left(\mathrm{AID}_i \parallel r_s \times X_i \parallel T_s\right) &= h\left(\mathrm{AID}_i \parallel r_i \times Y_i \parallel T_s\right). \end{aligned} \tag{20}$$

6.3. Revocation and Reissue Phase. The revocation problem can result in serious attacks, so a revocation phase should be provided when U_i wants to reissue a smart card due to loss. To prevent the same user identification from being selected, U_i inputs the previous ID_i. Then, U_i selects a different identification ID_i^* and sends ID_i, ID_i^* to the GW with hashed biometrics information A_i. After the GW receives these, GW revokes ID_i and reissues the smart card using ID_i^*. Then, GW continues into a phase that is equal to the registration phase. Revocation and reissue phase of the proposed scheme is described in Figure 10.

7. Security Analysis

This section describes the security analysis to confirm the security of the proposed scheme. We need to provide the following definitions to then compare the proposed scheme to other authentication schemes, including that proposed by Yoon and Kim.

Definition 1. A strong secret key (B_i, x, y) has a high value of entropy $S(k)$ that cannot be guessed in polynomial time.

Definition 2. A secure one-way hash function $y = h(x)$ is the following. Given x to compute y is easy but y to compute x is hard.

7.1. Biometric Recognition Error. The proposed scheme prevents a biometric recognition error by using fuzzy extraction. Yoon and Kim's scheme uses a hash function to check for conformity in the biometrics. Even if they use a threshold τ, since the hash function makes slight differences in the input data that produces very large differences in the output data, it is possible for biometric recognition errors to occur. However, the proposed scheme using fuzzy extractor prevents biometric recognition errors.

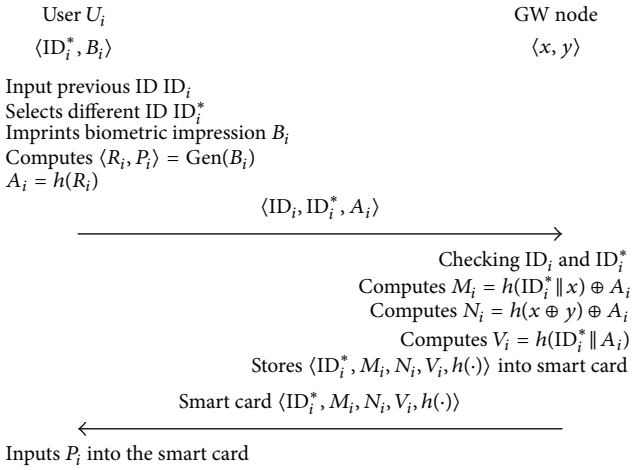


Figure 10: Revocation and reissue phase of the proposed scheme.

The fuzzy extractor Gen() generates R_i and P_i using the user's biometric B_i during the registration phase. R_i is a uniform and random string, and P_i is a helper string, so R_i can be the same with the assistance of auxiliary information P_i even if the user's inputs slightly different biometrics B_i^*, which thus maintains a reasonably similar status as that of the original biometric information. U_i imprints B_i for registration and computes R_i, P_i, and A_i as follows:

$$\begin{aligned} \langle R_i, P_i \rangle &= \mathrm{Gen}(B_i), \\ A_i &= h(R_i). \end{aligned} \tag{21}$$

U_i imprints B_i^* for login and computes R_i^*, P_i^*, and V_i^* and compares V_i with V_i^* as follows:

$$\begin{aligned} R_i^* &= \mathrm{Rep}(B_i^*, P_i), \\ A_i^* &= h(R_i^*), \\ V_i^* &= h(ID_i \| A_i^*), \\ \text{verifies } V_i &\stackrel{?}{=} V_i^*. \end{aligned} \tag{22}$$

With the assistance of P_i, the fuzzy extractor can compute a constant V_i^* even if the user inputs slightly different biometrics, so the proposed scheme is secure against a biometric recognition error.

7.2. User Verification Problem. The proposed scheme checks for the sameness in the ID_i to verify the status a legal user. Concretely, U_i makes B_i using the device and computes R_i and P_i. These values and information stored in smart card are used by U_i to compute $D_i, h(x \| y), k_i, C_i, AID_i$ and send M_1 to GW. GW computes ID_i' from AID_i and then computes D_i' and k_i'. U_i decrypts C_i and confirms ID_i''. Finally, GW authenticates U_i as follows if ID_i' and ID_i'' are the same. U_i imprints B_i for authentication. And then GW computes M_1 as follows:

$$\begin{aligned} \langle R_i, P_i \rangle &= \mathrm{Gen}(B_i), \\ A_i &= h(R_i), \\ D_i &= M_i \oplus A_i^*, \\ h(x \| y) &= N_i \oplus A_i^*, \\ k_i &= h(D_i \| T_i), \\ C_i &= E_{k_i}(ID_i \| X_i), \\ AID_i &= ID_i \oplus h(h(x \| y) \| T_i), \\ M_1 &= \langle AID_i, X_i, C_i, T_i, W_i \rangle. \end{aligned} \tag{23}$$

GW verifies sameness of ID as follows:

$$\begin{aligned} ID_i' &= AID_i \oplus h(h(x \| y) \| T_i), \\ D_i' &= h(ID_i' \| x), \\ k_i' &= h(D_i' \| T_i), \\ ID_i'' \| X_i' &= D_{k_i'}(C_i), \\ \text{verifies } ID_i' &\stackrel{?}{=} ID_i''. \end{aligned} \tag{24}$$

Unlike in Yoon and Kim's scheme, U_i can compute constant values including A_i as a result of the fuzzy extractor. Therefore, GW can authenticate a legal user even if the user inputs a slightly different biometric information B_i^*. Therefore, the proposed scheme can prevent a user verification problem.

7.3. Anonymity. In the proposed scheme, an attacker cannot compute a user's real identification ID_i without $h(x \| y)$ because the real identification of the user is always protected using $AID_i = ID_i \oplus h(h(x \| y) \| T_i)$. Therefore, only the legal user U_i and GW can compute ID_i from AID_i. GW stores x and y, so GW can easily compute $h(x \| y)$. U_i can compute $h(x \| y)$ from the N_i stored in the smart card as follows:

$$\begin{aligned} N_i &= h(x \| y) \oplus A_i, \\ \langle R_i, P_i \rangle &= \mathrm{Gen}(B_i), \\ A_i &= h(R_i), \\ h(x \| y) &= N_i \oplus A_i^*. \end{aligned} \tag{25}$$

To compute $h(x \| y)$, the attacker has to obtain both the user's smart card and B_i. However, even if the attacker can obtain a smart card, he cannot compute B_i. As a result, the attacker cannot obtain the user's real identification ID_i. Therefore, the proposed scheme provides user anonymity.

7.4. Perfect Forward Secrecy. Proposed scheme computes the session key between U_i and S_j as follows:

$$\begin{aligned} \mathrm{sk} &= h\left(\mathrm{AID}_i \parallel K_{US} \parallel T_s\right), \\ \mathrm{AID}_i &= \mathrm{ID}_i \oplus h\left(h\left(x \parallel y\right) \parallel T_i\right). \end{aligned} \tag{26}$$

Therefore, to compute all of the session keys of a user, an attacker has to know both $h(x \parallel y)$ and K_{US}. However, the attacker cannot compute two values using another long-term key because $h(x \parallel y)$ and K_{US} are independent of each other. In other words, if an attacker knows one of $h(x \parallel y)$ and K_{US}, he cannot compute the other one, so the session key that is derived from a set of long-term keys will not be compromised, even if one of the long-term keys is compromised in the future. Therefore, the proposed scheme achieves the perfect forward secrecy.

7.5. Session Key Exposure by the Gateway Node. In the proposed scheme, GW also knows most of the information related to the scheme but cannot compute sk between U_i and S_j. we suggest sk as follows:

$$\begin{aligned} \mathrm{sk} &= h\left(\mathrm{AID}_i \parallel r_i \times Y_i \parallel T_s\right) = h\left(\mathrm{AID}_i \parallel r_s \times X_i \parallel T_s\right) \\ &= h\left(\mathrm{AID}_i \parallel r_i \times r_s \times P \parallel T_s\right). \end{aligned} \tag{27}$$

GW can know AID_i, T_s, X_i, and Y_i but cannot acquire r_i and r_s. Even though $X_i = r_i \times P$ and $Y_i = r_s \times P$, GW cannot compute r_i and r_s from X_i and Y_i because it is mathematical problem about ECC. Therefore, sk is not exposed by GW in proposed scheme.

7.6. Vulnerability to Denial of Service Attack. In the proposed scheme, U_i, GW, and S_j basically check for freshness in the message using timestamps. Therefore, the scheme is considered to be able to endure a DoS attack if an attacker sends a previous message to the server with previous timestamps. Moreover, the DoS attack can be better prevented since the proposed scheme uses W_i, W_g, V_s in M_1, M_2, and M_3, respectively:

$$\begin{aligned} W_i &= h\left(h\left(x \parallel y\right) \parallel \mathrm{AID}_i \parallel X_i \parallel C_i \parallel T_i\right), \\ W_g &= h\left(h\left(\mathrm{SID}_j \parallel y\right) \parallel \mathrm{AID}_i \parallel C_g \parallel T_g\right), \\ V_s &= h\left(\mathrm{AID}_i' \parallel X_i' \parallel Y_i \parallel \mathrm{RM} \parallel T_s\right). \end{aligned} \tag{28}$$

W_i, W_g, and V_s include the current timestamps T_i. So, U_i, GW, and S_j can check for the freshness and legality of M_1, M_2, and M_3 because the timestamps of W_i, W_g, and V_s do not match the timestamps of M_1, M_2, and M_3 even if the attacker sends the previous W_i, W_g, and V_s with the current timestamps. Therefore, the proposed scheme is more secure against the DoS attack than Yoon and Kim's scheme.

7.7. Revocation Problem. The proposed scheme does not use the user's password but only uses the user's ID_i and biometrics B_i for registration. As a result, the proposed scheme needs to provide a revocation and reissue phase when the U_i wants to reissue a smart card due to loss. If a user reissues their own smart card with the same ID_i as the previous ID_i, the reissued smart card is going to be completely the same as the previous smart card. An attacker could thus make use of the lost smart card due to the sameness. Therefore, the proposed scheme suggests for the GW to check for differences between the previous ID_i and new ID_i^* during the revocation and reissue phase. In other words, we induce a user to select a different identification from the previous identification, so the GW reissues a new smart card with different information when the user loses his smart card, and the revocation problem is solved in this manner.

7.8. Mutual Authentication. In the proposed scheme, U_i, GW, and S_j authenticate each other by checking the accuracy of ID_i, AID_i, and V_s. Specifically, GW can authenticate whether the U_i that sent M_1 is a legal user or not by checking the validity of $\mathrm{ID}_i \stackrel{?}{=} \mathrm{ID}_i^*$. Only the legal case computes C_i using ID_i and B_i, so GW can confirm the user's legitimacy by decrypting C_i using k_i. S_j can authenticate GW by checking if it sends a correct hash k_g by verifying $\mathrm{ID}_i \stackrel{?}{=} \mathrm{ID}_i^*$. k_g can be computed within the current time only by a legal GW and S_j because $k_g = h(h(\mathrm{SID}_j \parallel y) \parallel T_g)$. Finally, U_i can authenticate S_j and GW by checking $V_s \stackrel{?}{=} h(\mathrm{AID}_i \parallel X_i \parallel Y_i \parallel \mathrm{RM} \parallel T_s)$. Only a legal S_j can compute V_s within the current time because only S_j can know X_i and Y_i.

7.9. Message Confidentiality. The proposed scheme uses 3 messages M_1, M_2, and M_3 in the login and authentication phase as follows:

$$\begin{aligned} M_1 &= \left\langle \mathrm{AID}_i, X_i, C_i, T_i, W_i \right\rangle, \\ M_2 &= \left\langle \mathrm{AID}_i, C_g, T_g, W_g \right\rangle, \\ M_3 &= \left\langle \mathrm{RM}, Y_i, V_s, T_s \right\rangle. \end{aligned} \tag{29}$$

T_i, T_g, T_s, and RM are basically public information, so they do not need to be protected. Other information can provide confidentiality because an attacker cannot compute important information from AID_i, X_i, C_i, W_i, C_g, W_g, Y_i, and V_s.

7.10. Password Change Phase. In a password-based authentication scheme, the user should be able to change his own password when he forgets his password or loses his smart card. In detail, to change a password freely, a smart card has to store information related to the user's password, such as h(password). However, when an attacker steals a user's smart card, he can gain all the information stored in the smart card by using a simple and differential power analysis. Therefore, the attacker can obtain a user's password even when it is protected by h(password) because a few characters are necessary to use the password. Therefore, a password change phase is important but poses a serious risk in that information (such as the password) for login and authentication can

be exposed. However, the proposed scheme uses only a user's biometric information with high entropy; therefore, the attacker cannot obtain the original biometric information, even if h(biometrics) is known. Moreover, a user does not forget his biometric information and so does not need to change it.

7.11. Stolen Verifier Attack. If the GW or S_j stores verifier information, an attacker can attempt a stolen verifier attack. However, the proposed scheme is resistant to a stolen verifier attacker because GW and S_j do not store a user's identification/password table and the user's biometrics. In the proposed scheme, GW only stores the secret key x, y, and S_j store only $h(\mathrm{SID}_j \parallel y)$. Therefore, the GW cannot obtain authentication information from a legal user even if the attacker has the authority to access the database of the GW and S_j.

7.12. Guessing Attack. Since the proposed scheme does not use the user's password, this scheme is not vulnerable to a guessing attack. Moreover, the user's biometrics is always protected by the one-way hash function. Since the biometrics information has a high level of entropy, unlike a password, the attacker cannot calculate the user's biometric information from the hashed value. When the attacker steals a user's smart card, the attacker can obtain ID_i, M_i, N_i, V_i, P_i, $h(\cdot)$ from the smart card. However, since M_i, N_i, V_i are hashed values, the attacker cannot obtain any secret information from them. ID_i and P_i are not secret information, so the attacker cannot acquire secret information using a guessing attack. Therefore, the proposed scheme is not vulnerable to a guessing attack [25–27].

7.13. Replay Attack. The proposed scheme is secure against a replay attack by adding timestamps T_i, T_g, T_s into authentication messages W_i, W_g, V_s in M_1, M_2, M_3, respectively. Even if the attacker obtains M_1, M_2, M_3 and sends them again with the current timestamps, the attacker cannot compute W_i, W_g, V_s using the current timestamps:

$$\begin{aligned} M_1 &= \langle \mathrm{AID}_i, X_i, C_i, T_i, W_i \rangle, \\ W_i &= h\left(h\left(x \parallel y\right) \parallel \mathrm{AID}_i \parallel X_i \parallel C_i \parallel T_i\right), \\ M_2 &= \left\langle \mathrm{AID}_i, C_g, T_g, W_g \right\rangle, \\ W_g &= h\left(h\left(\mathrm{SID}_j \parallel y\right) \parallel \mathrm{AID}_i \parallel C_g \parallel T_g\right), \\ M_3 &= \langle \mathrm{RM}, Y_i, V_s, T_s \rangle, \\ V_s &= h\left(\mathrm{AID}_i' \parallel X_i' \parallel Y_i \parallel \mathrm{RM} \parallel T_s\right). \end{aligned} \tag{30}$$

7.14. Impersonation Attack. Even if an attacker intercepts the authentication message $M_1 = \langle \mathrm{AID}_i, X_i, C_i, T_i, W_i \rangle$ to impersonate a legitimate user, the attacker cannot normally extract D_i and k_i from AID_i, C_i, and W_i since the one-way hash function is implemented according to Definition 2. Without D_i and k_i, the attacker cannot produce a legitimate login and authentication message in the attacker's current time. Therefore, the proposed scheme is secure from impersonation attacks. Likewise, the attacker cannot impersonate a legitimate GW and S_j. Even if the attacker obtains $M_2 = \langle \mathrm{AID}_i, C_g, T_g, W_g \rangle$, the attacker cannot compute k_i or $h(h(\mathrm{SID}_j \parallel y) \parallel T_g)$ from such due to Definition 2.

7.15. Insider Attack. Typically, malicious insiders want to acquire private user information, such as their biometrics. In the proposed registration phase, a user's smart card device imprints the biometric impression B_i and computes $\langle R_i, P_i \rangle$ using $\mathrm{Gen}(B_i)$ and then sends A_i to GW; $A_i = h(R_i)$. Therefore, GW cannot obtain B_i using the incoming A_i because of the properties of the one-way hash function. Therefore, the proposed scheme is secure against insider attacks.

7.16. Security Factor. Two- or three-factor authentication methods are implemented by means of a combination of two or three different components. In WSNs, most authentication schemes use a user's password, smart card, and biometric information as components. We propose a two-factor authentication scheme that uses the smart card and biometric information without a password but can provide a similar secure authentication environment comparable to those provided by three-factor authentication schemes.

Table 1 provides a summary and comparison of the security provided by the proposed scheme and that provided by other schemes, including the one by Yoon and Kim.

8. Conclusions

To provide security to wireless sensors and users, various authentication schemes for WSNs have been proposed recently. The security problem in He et al.'s scheme was addressed by Yoon and Kim, who proposed an advanced biometrics-based user authentication scheme for WSNs. In this paper, we have identified vulnerabilities in Yoon and Kim's scheme in terms of a biometric recognition error, a user verification problem, lack of anonymity and perfect forward secrecy, session key exposure by the GW node, vulnerability to a DoS attack, and a revocation problem. To solve these security vulnerabilities, we have suggested specific countermeasures, including the use of fuzzy extraction to imprint biometrics during the registration phase consisting of error-tolerant cryptographic primitives for biometric security. We recommend the use of the sensor node's random number and ECC to exchange a random number between a user and the sensor node during the authentication phase. ECC can maintain the same degree of security with a smaller key size than other forms of public-key cryptography. Therefore, ECC is suitable for use with wireless devices that have limited resources. In accordance with these countermeasures, we propose a biometrics-based authentication scheme based on fuzzy extraction with improved security. In addition, we conduct a security analysis to show that the proposed scheme is more secure than other authentication schemes.

Table 1: Security comparison.

Security property	Yuan et al. [11]	He et al. [8]	Yoon and Kim [14]	Proposed scheme
Biometric recognition error	Insecure	Secure	Insecure	Secure
User verification problem	Insecure	Secure	Insecure	Secure
Anonymity	Provide	Do not provide	Do not provide	Provide
Perfect forward secrecy	No sk	No sk	Do not provide	Provide
sk exposure by GW	No sk	No sk	Insecure	Secure
Vulnerable to DoS attack	Secure	Insecure	Insecure	Secure
Revocation problem	Secure	Secure	Insecure	Secure
Mutual authentication	Do not provide	Provide	Provide	Provide
Message confidentiality	Do not provide	Do not provide	Provide	Provide
Password change phase	Required	Required	Not required	Not required
Stolen verifier attack	Secure	Secure	Secure	Secure
Guessing attack	Secure	Insecure	Secure	Secure
Replay attack	Secure	Secure	Secure	Secure
Impersonation attack	Insecure	Insecure	Secure	Secure
Insider attack	Insecure	Insecure	Secure	Secure
Security factor	Three-factor	Two-factor	Two-factor	Two-factor

Notations

U_i: The ith user
ID_i: The identification of U_i
AID_i: The anonymous identification of user$_i$
B_i: The biometric template of U_i
GW: The gateway of WSNs
x, y: Two master keys of GW
S_j: The jth sensor node
SID_j: S_j identification
$h(\cdot)$: A secure one-way hash function
$d(\cdot)$: Symmetric parametric function
τ: Predetermined threshold for biometric verification
$E_k(\cdot)$: A symmetric encryption function with key k
$D_k(\cdot)$: The decryption function corresponding to $E_k(\cdot)$
T: Timestamps
r_i: User's random nonce
r_s: Sensor's random nonce
sk: Session key between user and sensor
$\|$: A string concatenation operation
$\times$: Multiplication operation
$\oplus$: A string XOR operation
RM: Response to the query message.

Conflict of Interests

The authors declare no conflict of interests.

Authors' Contribution

Younsung Choi, Youngsook Lee, and Dongho Won contributed to the security analysis, design of the proposed scheme, and preparation of the paper.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1003) supervised by the IITP (Institute for Information and communications Technology Promotion).

References

[1] J. Nam, M. Kim, J. Paik, Y. Lee, and D. Won, "A provably-secure ECC-based authentication scheme for wireless sensor networks," *Sensors (Switzerland)*, vol. 14, no. 11, pp. 21023–21044, 2014.

[2] K. H. M. Wong, Z. Yuan, C. Jiannong, and W. Shengwei, "A dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 1, pp. 244–251, Taichung, Taiwan, June 2006.

[3] J. Kim, D. Lee, W. Jeon, Y. Lee, and D. Won, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6443–6462, 2014.

[4] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 59–64, ACM, Washington, DC, USA, October 2004.

[5] Y. Choi, J. Nam, Y. Lee, S. Jung, and D. Won, "Cryptanalysis of advanced biometric-based user authentication scheme for wireless sensor networks," in *Computer Science and Its Applications*, vol. 330 of *Lecture Notes in Electrical Engineering*, pp. 1367–1375, Springer, Berlin, Germany, 2015.

[6] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[7] H.-R. Tseng, R.-H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, vol. 1, pp. 986–990, Washington, DC, USA, November 2007.

[8] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad-Hoc and Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.

[9] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[10] T.-H. Chen and W.-K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.

[11] J. Yuan, C. Jiang, and Z. Jiang, "A biometric-based user authentication for wireless sensor networks," *Wuhan University Journal of Natural Sciences*, vol. 15, no. 3, pp. 272–276, 2010.

[12] E.-J. Yoon and K.-Y. Yoo, "Cryptanalysis of robust mutual authentication protocol for wireless sensor networks," in *Proceedings of the 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC '11)*, pp. 392–396, IEEE, Alberta, Canada, August 2011.

[13] D. He, "Robust biometric-based user authentication scheme for wireless sensor networks," *IACR Cryptology ePrint Archive*, vol. 2012, article 203, 2012.

[14] E.-J. Yoon and C. Kim, "Advanced biometric-based user authentication scheme for wireless sensor networks," *Sensor Letters*, vol. 11, no. 9, pp. 1836–1843, 2013.

[15] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10081–10106, 2014.

[16] J. Kar and B. Majhi, "An efficient password security of multiparty key exchange protocol based on ECDLP," *International Journal of Computer Science and Security*, vol. 3, no. 5, pp. 405–413, 2009.

[17] R. Lu, Z. Cao, Z. Chai, and X. Liang, "A simple user authentication scheme for grid computing," *International Journal of Network Security*, vol. 7, no. 2, pp. 202–206, 2008.

[18] V. C. Giruka, S. Chakrabarti, and M. Singhal, "A distributed multiparty key agreement protocol for dynamic collaborative groups using ECC," *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 959–970, 2006.

[19] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology—EUROCRYPT 2004*, pp. 523–540, Springer, Berlin, Germany, 2004.

[20] X. Boyen, "Reusable cryptographic fuzzy extractors," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pp. 82–91, Washington, DC, USA, October 2004.

[21] W. Jeon, J. Kim, J. Nam, Y. Lee, and D. Won, "An enhanced secure authentication scheme with anonymity for wireless environments," *IEICE Transactions on Communications*, vol. E95-B, no. 7, pp. 2505–2508, 2012.

[22] J. Nam, K. R. Choo, M. Park, J. Paik, and D. Won, "On the security of a simple three-party key exchange protocol without server's public keys," *The Scientific World Journal*, vol. 2014, Article ID 479534, 7 pages, 2014.

[23] K. R. Choo, J. Nam, and D. Won, "A mechanical approach to derive identity-based protocols from Diffie-Hellman-based protocols," *Information Sciences*, vol. 281, pp. 182–200, 2014.

[24] Y. Choi, J. Nam, D. Lee, J. Kim, J. Jung, and D. Won, "Security enhanced anonymous multiserver authenticated key agreement scheme using smart cards and biometrics," *The Scientific World Journal*, vol. 2014, Article ID 281305, 15 pages, 2014.

[25] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO'99*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer, Berlin, Germany, 1999.

[26] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[27] J. Muthukuru and B. Sathyanarayana, "A survey of elliptic curve cryptography implementation approaches for efficient smart card processing," *Global Journal of Computer Science and Technology*, vol. 12, no. 1, 2012.