Research Article

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

Younsung Choi,1 Youngsook Lee,2 and Dongho Won1

1Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 440-746, Republic of Korea
2Department 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

Copyright © 2016 Younsung Choi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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.'s 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 (ECDHP). No polynomial time algorithm can solve the ECDLP, ECCDHP, and ECDHP with nonnegligible probability.

Let $p > 3$ be a large prime and choose two field elements $a, b \in F_p$ satisfying $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ to define the equation of a nonsuper-singular elliptic curve $E: y^2 = x^3 + ax + b \mod p$ over $F_p$. Choose a generator point $P = (x, y) \in E(F_p)$ whose order is a large prime number $q$ over $E(F_p)$. The subgroup $G$ of the elliptic curve group $E(F_p)$ 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 $G$, find an integer $x \in Z_q^*$ 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 $a, b, c \in Z_q^*$, given two point elements $a \times P, b \times P$ in $G$, compute $a \times b \times P$ in $G$.

(iii) ECDHP: for $a, b, c \in Z_q^*$, given three point elements $a \times P, b \times P$, and $c \times P$ in $G$, 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 ECDHP for protecting $K_{US}$ and $K_{SU}$ due to ECDHP even though 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 $K_{SU} = r_s \times X_i = r_s \times r_i \times P$. 
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 Gen(Bi) = (Ri, Pi) and Rep(Bi, Pr) are used. The fuzzy extractor Gen generates Ri and Pi by using a user’s biometric information during the registration phase. Ri is a uniform and random string, and Pi is the helper string. Thus, Ri can be the same under the assistance of auxiliary information Pr even if 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 Ri using the biometric information that has been newly input BI and PI in the login phase. To reproduce the same RI, the metric space distances between BI and BI must 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(IDj || y) to sensor node Sj. x is then used for Uj. During the registration phase, the gateway issues a smart card stored as h(IDj || x) to Uj [11].

3.1. Registration Phase. In the registration phase, a user Uj communicates securely with the gateway. Uj sends important information regarding the user’s identification and biometrics. Figure 1 describes the registration phase, and detailed steps are given as follows. First, Uj chooses IDj and imprints his biometrics Bi on the specific sensor device. Then, Uj computes Ei = h(IDj, Bi) and sends IDj and Ei to GW node by using a secure channel. Then, GW node computes two values: Ri = h(IDj || x) ⊕ Ei and Vi = h(IDj || Ei). And the GW node inputs (IDj, Ri, Vj, h(⋅), d(⋅), τ) into a smart card and sends the smart card to user Uj. h(⋅) is a one-way hash function. d(⋅) 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 Uj enters IDj and Bi into a smart card terminal, the smart card must validate the legitimacy of Uj. Then, Uj, Sj, and GW authenticate each other. This scheme uses three messages ⟨M1, M2, M3⟩ during authentication, as shown in Figure 2. Finally, Uj and Sj share the session key sk after the login and authentication phases; Uj and Sj communicate with each other using the session key sk.

(1) Uj inserts his smart card into the card reader and imprints his biometric Bi on a specific device to verify the user’s biometrics.

(2) The smart card computes two values: Ei = h(IDj, Bi) and Vj = h(IDj || Ei).

(3) The smart card compares the computed Vj and the Vi that is stored in the smart card. If d(Vj, Vj) ≤ τ, the user’s smart card stops the login phase. Otherwise, the smart card generates a random number rj.

(4) The smart card computes three values: D1 = Ri ⊕ Ei, kj = h(D1 || Tj) and Cj = Eks(IDj || rj), where Tj is the current timestamps.

(5) Then, the smart card sends the login message M1 = ⟨IDj, Cj, Tj⟩ to the GW node.

GW receives M1 and executes the following actions.

(1) The GW node checks the freshness of Tj by using (Tj′′ − Tj) ≤ ΔT. ΔT is the expected time for the transmission delay. If Tj is not fresh, the GW node rejects the user’s request.

(2) The GW node computes three values: D1 = h(IDj || x), k1j = h(D1 || Tj), and IDj || r1j = Dks(Cj).

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

(4) The GW node computes two values: kg = h(h(SIDj || y) || Tg) and Cg = Ekg(IDj || r1j).

(5) The GW node then computes M2 = ⟨IDj, Cg, Tg⟩ and sends it to the sensor node Sj.

Sj receives M2 and performs the following actions.

(1) Sj checks the freshness of Tg using (Tg′′ − Tg) ≤ ΔT.

(2) Sj computes two values: k′g = h(h(SIDj || y) || Tg) and IDj || r′g = Dks(Cg).

(3) Sj checks whether IDj and IDj′ are equal and Sj picks up the current timestamps Tg′. 
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 \( U_i \); \( V_s = h(\text{ID}_i^{\prime\prime} \parallel r_i^{\prime\prime} \parallel \text{RM} \parallel T_s) \).

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

\( U_i \) receives \( M_s \) and executes the following actions.

(1) \( U_i \) checks the freshness of \( T_s \) using \( (T^{\prime\prime\prime} - T_s) \leq \Delta T \).

(2) \( U_i \) computes \( V'_s = h(\text{ID}_i \parallel r_i \parallel \text{RM} \parallel T_s) \) 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 \( sk = h(\text{ID}_i \parallel r_i \parallel T_s) \).
Hash function $h(\cdot)$; small differences of input data make very big differences of output data

(i) In login phase of Yoon and Kim’s scheme, login problem relevant to biometrics can occur as follows:

| Normal login phase | Even if $U_i$ inputs same biometrics, |
|--------------------|-------------------------------------|
| (1) $U_i$ inputs biometrics $B_i^*$ |
| (2) $U_i$ computes $E_i^* = h(ID_i \parallel B_i^*)$ |
| (3) $U_i$ computes $V_i^* = h(ID_i \parallel E_i^*)$ |
| (4) $U_i$ verifies $d(V_i, V_i^*) \leq \tau$ | \implies It has possibility to make different biometrics $\sim B_i^*$ |
| | \implies Different $B_i^*$ makes very big differences in $\sim E_i^*$ due to $h(\cdot)$ |
| | \implies $\sim E_i^*$ makes more very big differences in $\sim V_i^*$ due to $h(\cdot)$ |
| | \implies Because of very big differences between $V_i$ and $\sim V_i^*$, $d(V_i, \sim V_i^*)$ exceeds limit of predetermined threshold $\tau$ |

(ii) Though $U_i$ inputs own same biometrics, login failure can occur due to biometric recognition error

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 in $M_1$ and ID$^j$ in the output of the decrypted $C_i$. Specifically, the user computes $C_i$ using a symmetric encryption algorithm; $C_i = E_k(ID, r_i)$. ID and $r_i$ do not matter but $E_k$ has a problem in that there is a possibility to obtain unexpected results. This is the reason why $E_k$ is made up by $D_i$ and $T_i$. The GW and $M_1$ consist of $R_i$, $E_i^*$:

$$E_i^* = h(ID_i \parallel B_i^*)$$

$$D_i = R_i \oplus E_i^*$$

$$k_i = h(D_i \parallel T_i)$$

(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 from $C_i$ so the user is not authenticated by GW even when the user uses its own normal ID 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’s 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 in the WSNs communication instead of sending a normal ID [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 $C_{ij}, T_{ij}$, and $T_i$ in previous public channel.
(2) Attacker knew one of user’s long-term secret: $D_i$.

$$\implies$$ Attacker has $C_{ij}, T_{ij}, T_i$, and $D_i$ and computes $k_i$ and $D_{k_i}$ as follows:

$$k_i = h(D_i \parallel T_i)$$
(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 $r_i$
2. $U_i$ computes $D_i = R_i \oplus E_i^*$
3. $U_i$ generates timestamp $T_i$
4. $U_i$ computes $k_i = h(D_i || T_i)$
5. $U_i$ computes $C_i = E_{k_i}(ID_i || r_i)$
6. $U_i$ sends $M_i = (ID_i, C_i, T_i)$
7. GW receives $M_i = (ID_i, C_i, T_i)$
8. GW verifies $(T_i - T) \leq \Delta T$
9. GW computes $D'_i = h(TM || x)$
10. GW computes $k'_i = h(D'_i || T_i)$
11. GW computes $ID'_i || r'_i = D_{k'_i}(C_i)$
12. GW verifies $ID'_i = ID_i$

(ii) So, it is possible that GW cannot verify user’s legal ID, even if $U_i$ sends own $M_i$ because of inaccurate $-B^*$

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 $S_j$

$M_1 = (ID_i, C_i, T_i) \rightarrow$ Attacker can know how many user are registered in GW

$M_1 = (ID_i, C_i, T_i) \rightarrow$ Attacker can know which user wants to access to sensor $S_j$

Therefore, Yoon and Kim’s scheme basically does not provide user anonymity

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

$D_{k'_i}(C_i) = ID_i || r_i$.

(3) Attacker acquires ID$^i$ and $r_i$ and then computes sk.

$sk = h(ID_i || r_i || T_i)$.

(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 $D$. Then, the attacker can compute $k_i = h(D || T_i)$ 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 $ID_i, r_j$, and $T_c$.

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 $T_i$ over a public channel. Then, the GW node can compute all sk = $h(ID_i || r_i || T_i)$ 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 $T_i$ in public channel:

$\Rightarrow$ GW has ID$^i$, $r_i$ and $T_i$.

$sk = h(ID_i || r_i || T_i)$.

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 $(ID_j, C_i, T_c)$ 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 $(ID_j, C_i, T_c)$, they first check for the freshness of the timestamps $T_c$. However, the $T_c$ generated by the attacker is current.
and the GW node and sensor cannot determine that IDᵢ and Ci are from previous messages. So they execute various functions, such as a hash function, decryption function, and verification function before checking whether the IDi sent by the attacker and the computed IDi 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 IDᵢ and biometrics Bi. 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 (IDᵢ, Ri, Vi, h(), d(), τ) if the Ui sends the same IDᵢ and Bi to the GW node. So even though Ui reissues a new smart card, Uᵢ cannot discard the lost smart card because the reissued smart card and the lost smart card are the same. Therefore, the user Uᵢ 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].

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 Uᵢ 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 Ekᵢ, small differences of kᵢ 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 Ekᵢ = h(IDᵢ, Biᵢ), Vⁱᵢ = h(IDᵢ ⊕ Eⁱᵢ), and d(Vᵢ, Vᵢ') ≤ τ to prevent a biometric recognition error. Moreover, the difference in Bi and Biᵢ results in a different encryption key kᵢ. 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 \( D_j = R_j \oplus E_j^* \) and \( k_i = h(D_j \parallel T_j) \). We thus improve Yoon and Kim’s scheme using fuzzy extraction as follows.

During the registration phase, instead of \( E_j^* = h(ID_j, B_j^*) \), the smart card computes \( R_j \) and \( P_j \) using a fuzzy extraction function \( Gen(B_j) \) such as \( \langle R_j, P_j \rangle = Gen(B_j) \). It also computes \( F_j = h(R_j) \) and sends both \( ID_j \) and \( F_j \) to the GW node. The GW node modifies the computation of \( A_j \) and \( V_j \) from \( A_j = h(ID_j \parallel x) \oplus E_i \) and \( V_j = h(ID_j \parallel E_i) \) to
\[
A_j = h(R_j),
\]
\[
M_j = h(ID_j \parallel x) \oplus A_j,
\]
\[
V_j = h(ID_j \parallel A_j).
\]

During the login and authentication phase, instead of \( E_j^* = h(ID_j, B_j^*), V_j^* = h(ID_j \parallel E_j^*), \) \( d(V_j, V_j^*) \leq r \), and \( D_j = A_j \oplus E_j^* \), smart card computes the following:
\[
R_j^* = \text{Rep}(B_j^* \parallel P_j),
\]
\[
A_j^* = h(R_j^*),
\]
\[
V_j^* = h(ID_j \parallel A_j^*),
\]
\[
V_j = V_j^*,
\]
\[
D_j = M_j \oplus A_j^*.
\]

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) \( ID_j \) 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 \( ID_j \), the attacker can acquire a variety of information on the user, GW, and the sensor. Therefore, we propose to use an anonymity \( AID_j \) to provide anonymity. Instead of sending a normal \( ID_j \), we suggest using \( AID_j \) in the communication as follows:
\[
AID_j = ID_j \oplus h(x \parallel y \parallel T_j).
\]

GW sends \( h(x \parallel y) \) to a user using \( N_j = h(x \parallel y) \oplus A_j \) 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 \( AID_j \) changes every session due to \( T_j \) so we can provide user anonymity.

To provide the perfect forward secrecy in our proposed scheme, we modify the computation of \( sk \) from \( sk = h(ID_j \parallel r_j \parallel T_j) \) to
\[
sk = h(AID_j \parallel K_{US} \parallel T_j).
\]

\( AID_j = ID_j \oplus h(x \parallel y \parallel T_j) \) 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 \( S_j \):
\[
K_{US} = r_j \times r_i \times P.
\]

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 \( K_{US} \):
\[
X_i = r_i \times P,
\]
\[
K_{US} = r_i \times Y_i = r_i \times r_x \times P.
\]

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:
\[
Y_i = r_x \times P,
\]
\[
K_{SU} = r_x \times X_i = r_x \times r_i \times P.
\]

\( K_{US} \) and \( K_{SU} \) can be used by the user and sensor to compute \( sk = h(AID_j \parallel K_{US} \parallel T_j) \) 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 \( M_1 = \langle ID_j, C_y, T_y \rangle \) and \( M_2 = \langle ID_j, C_y, T_y, W_y \rangle \) to
\[
M_1 = \langle AID_j, X_i, C_i, T_i, W_i \rangle,
\]
\[
M_2 = \langle AID_j, C_y, T_y, W_y \rangle.
\]

In advance, the GW node and sensor check \( W_i \) and \( W_y \) to verify an incoming message:
\[
W_i = h(h(x \parallel y) \parallel AID_j \parallel X_i \parallel C_i \parallel T_i),
\]
\[
W_y = h(h(SID_j \parallel y) \parallel AID_j \parallel C_y \parallel T_y).
\]
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 \( h(SID_j \parallel y) \) to the sensor \( S_j \).

6.1. Registration Phase. The registration phase of the proposed scheme is described in Figure 8. The \( U_i \) performs 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 \( \langle R_i, P_i \rangle = \text{Gen}(B_i) \) 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:
   \[
   M_i = h(ID_i \parallel x) \oplus A_i, \\
   N_i = h(x \parallel y) \oplus A_i, \\
   V_i = h(ID_i \parallel A_i).
   \] (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:
   \[
   R_i^* = \text{Rep}(B_i^*, P), \\
   A_i^* = h(R_i^*), \\
   V_i^* = h(ID_i \parallel A_i^*),
   \] (12)
   verifies \( V_i \) in \( V_i^* \).
2. The smart card generates a random number \( r_i \) and computes \( X_i, D_i \), and \( h(x \parallel y) \). \( X_i \) is used for the session key between \( U_i \) and \( S_j \). This scheme uses \( h(x \parallel y) \) to provide perfect forward secrecy:
   \[
   X_i = r_i \times P, \\
   D_i = M_i \oplus A_i^*, \\
   h(x \parallel y) = N_i \oplus A_i^*.
   \] (13)
3. \( U_i \) picks up the current timestamps \( T_i \) and computes \( k_j, C_i, AID_j \), 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:
   \[
   k_j = h(D_i \parallel T_i), \\
   C_i = E_{k_j}(ID_j \parallel X_i), \\
   AID_j = ID_j \oplus h(h(x \parallel y) \parallel T_i), \\
   W_i = h(h(x \parallel y) \parallel AID_j \parallel X_i \parallel C_i \parallel T_i), \\
   M_1 = \langle AID_j, X_i, C_i, T_i, W_i \rangle.
   \] (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 \( (T^* - T_i) \leq \Delta T \). Then, GW verifies the received \( W_i \) using \( W_i^* = h(h(x \parallel y) \parallel AID_j \parallel X \parallel C_i \parallel T_i) \).
User $U_i$

\langle ID_i, \text{smart card} \rangle

Inserts smart card

Inputs $ID_i$ and biometric $R_i^*$

$R_i^* = \text{Rep}(B_i^*, P_i)$, $A_i^* = h(R_i^*)$

$V_i^* = h(ID_i \parallel A_i^*)$

Verifies $V_i^* = V_i^*$

Generates random number $r_i$

$X_i = r_i \times P$

$D_i = M_i \oplus A_i^*$

$h(x \parallel y) = N_i \oplus A_i^*$

Picks up timestamp $T_i$

$k_i = h(D_i \parallel T_i)$

$C_i = E_k(ID_i \parallel X_i)$

$AID_i = ID_i \oplus h(h(x \parallel y) \parallel T_i)$

$W_i = h(h(x \parallel y) \parallel AID_i \parallel X_i \parallel C_i \parallel T_i)$

$M_i = \langle AID_i, X_i, C_i, T_i, W_i \rangle$

\[ \text{Verifies} \ (T' - T_i) \leq \Delta T \]

\[ \text{Verifies} \ W_i = h(h(x \parallel y) \parallel AID_i \parallel X_i \parallel C_i \parallel T_i) \]

\[ ID'_i = AID_i \oplus h(h(x \parallel y) \parallel T_i) \]

\[ D'_i = h(ID'_i \parallel x) \]

\[ k'_i = h(D'_i \parallel T_i) \]

\[ ID''_i \parallel X''_i = D'_i \parallel (C_i) \]

\[ \text{Verifies} \ ID''_i = ID'_i \]

Picks up timestamp $T_g$

$k_g = h(h(SID_j \parallel y) \parallel T_g)$

$C_g = E_k(AID_i \parallel X'_i)$

$W_g = h(h(SID_j \parallel y) \parallel AID_i \parallel C_g \parallel T_g)$

$M_g = \langle AID_i, C_g, T_g, W_g \rangle$

\[ \text{Verifies} \ (T' - T_g) \leq \Delta T \]

\[ \text{Verifies} \ W_g = h(h(SID_j \parallel y) \parallel AID_i \parallel C_g \parallel T_g) \]

\[ K'_g = h(h(SID_j \parallel y) \parallel T_g) \]

$AID'_j \parallel X' = D'_g \parallel (C_g)$

\[ \text{Verifies} \ AID'_j = AID'_i \]

Generates random number $r_s$

$K_{RS} = r_s \times X'_i$, $Y_i = r_s \times P$

Computes $sk = h(AID'_i \parallel K_{RS} \parallel T_i)$

Picks up timestamp $T_s$

$RM = \text{respond to the query of } U_i$

$V_s = h(AID'_i \parallel X'_i \parallel Y_i \parallel RM \parallel T_s)$

$M_3 = \langle RM, Y_i, V_s, T_s \rangle$

\[ \text{Verifies} \ (T'' - T_i) \leq \Delta T \]

\[ \text{Verifies} \ V_s = h(AID'_i \parallel X'_i \parallel Y_i \parallel RM \parallel T_s) \]

\[ K_{RS} = r_s \times X'_i, Y_i = r_s \times P \]

Computes $sk = h(AID'_i \parallel K_{RS} \parallel T_i)$

Accepts $RM$

\[ \text{Shared} \ sk = h(AID_i \parallel r_i \times r_j \times P \parallel T_s) \]

\[ \text{Figure 9: Login and authentication phase of the proposed scheme.} \]
(5) GW computes ID'\_i, D'\_i, and k'\_i and decrypts the C\_i in M\_i. The output of the decryption is used by GW to authenticate U\_i by comparing ID'\_i and ID''\_i:

\[
\text{ID'}\_i = \text{AID} \oplus h (h(x \parallel y) \parallel T_i), \\
D'\_i = h (\text{ID}'\_i \parallel x), \\
k'\_i = h (D'\_i \parallel T_i), \\
\text{ID''\_i} = X'\_i = Dk\_i (C\_i),
\]

(15)

verifies ID'\_i = ID''\_i.

(6) GW picks up the current timestamps T\_g and k\_g, C\_g, and W\_g to authenticate S\_i. Then, GW sends an authentication message M\_2 to S\_i:

\[
k\_g = h (h (\text{SID} \parallel y) \parallel T\_g), \\
C\_g = E_k\_g (\text{AID} \parallel X'\_i), \\
W\_g = h (h (\text{SID} \parallel y) \parallel \text{AID} \parallel C\_g \parallel T\_g), \\
M\_2 = \langle \text{AID}, C\_g, T\_g, W\_g \rangle.
\]

(16)

(7) With M\_2 in hand, S\_i retrieves the current timestamps T''\_g and checks if T''\_g - T\_g \leq \Delta T. If the freshness check for T''\_g fails, S\_i stops the authentication phase. Otherwise, S\_i computes W\_g and compares it with h(h(SID \parallel y) \parallel \text{AID} \parallel C\_g \parallel T\_g). After that, S\_i computes k'\_g and decrypts C\_g using k'\_g. Then, S\_i checks the sameness between the received AID\_i and the computed AID'\_i:

\[
k'\_g = h (h (\text{SID} \parallel y) \parallel T\_g), \\
\text{AID'}\_i \parallel X' = Dk'\_g (C\_g),
\]

(17)

verifies AID\_i = AID'\_i.

(8) To compute sk, S\_i generates a random number r\_i and computes K\_S\_i, Y\_i. This sk is computed only by U\_i and S\_i due to a mathematical problem inherent to ECC. Then, S\_i picks up the current timestamps T\_i and computes RM, V\_i for the authentication. After the operation is finished, S\_i sends M\_3 to U\_i:

\[
K\_S\_i = r\_i \times X'\_i, \\
Y\_i = r\_i \times P, \\
\text{computes sk} = h (\text{AID} \parallel K\_S\_i \parallel T\_i), \\
\text{RM} = \text{Response to the query of U\_i}, \\
V\_i = h (\text{AID} \parallel X'\_i \parallel Y\_i \parallel \text{RM} \parallel T\_i), \\
M\_3 = \langle \text{RM}, Y\_i, V\_i, T\_i \rangle.
\]

(18)

(9) First, U\_i checks the freshness in T\_i and the sameness of V\_i. Then, U\_i computes K\_US using the received Y\_i, and sk by using AID\_i, K\_US, and T\_i. Only a legal U\_i computes K\_US using Y\_i so anyone cannot compute sk including GW. Then, U\_i accepts RM:

\[
\text{verifies } (T''\_g - T\_g) \leq \Delta T, \\
\text{verifies } V\_i = h (\text{AID} \parallel X\_i \parallel Y\_i \parallel \text{RM} \parallel T\_i), \\
K\_US = r\_i \times Y\_i,
\]

(19)

computes sk = h (AID \parallel K\_US \parallel T\_i), accepts RM.

(10) From now on, U\_i can communicate securely with S\_i using sk:

\[
\text{shared sk} = h (\text{AID} \parallel r\_i \times r\_i \times P \parallel T\_i), \\
h (\text{AID} \parallel r\_i \times X\_i \parallel T\_i) = h (\text{AID} \parallel r\_i \times Y\_i \parallel T\_i).
\]

(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 extractor. Yoon and Kim’s scheme uses a hash function to check for conformity in the biometrics. Even if they use a threshold r, 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.
\begin{align*}
\text{User } U_i \quad \langle \text{ID}^*_i, B_i \rangle & \quad \text{GW node} \quad \langle x, y \rangle \\
\text{Input previous ID ID}_i \\
\text{Selects different ID ID}^*_i \\
\text{Imprints biometric impression } B_i \\
\text{Computes } \langle R_i, P_i \rangle = \text{Gen}(B_i) \\
A_i = h(R_i) \\
\langle \text{ID}^*_i, \text{ID}^*_i, A_i \rangle & \\
\text{Checking ID}^*_i \text{ and ID}^*_i \\
\text{Computes } M_i = h(\text{ID}^*_i || x) \oplus A_i \\
\text{Computes } N_i = h(x \oplus y) \oplus A_i \\
\text{Computes } V_i = h(\text{ID}^*_i || A_i) \\
\text{Stores } \langle \text{ID}^*_i, M_i, N_i, V_i, h(\cdot) \rangle \text{ into smart card} \\
\text{Smart card } \langle \text{ID}^*_i, M_i, N_i, V_i, h(\cdot) \rangle & \\
\text{Inputs } P_i \text{ into the smart card} \\
\text{Figure 10: Revocation and reissue phase of the proposed scheme.}
\end{align*}

The fuzzy extractor \( \text{Gen()} \) 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:

\[
\langle R_i, P_i \rangle = \text{Gen}(B_i), \\
A_i = h(R_i).
\]

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

\[
R_i^* = \text{Rep}(B_i^*, P_i), \\
A_i^* = h(R_i^*), \\
V_i^* = h(\text{ID}^*_i || A_i^*), \quad \text{verify } V_i = V_i^*. 
\]

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 \( D_j, h(x \parallel y), k_j, C_j, \text{AID}_j, \) and send \( M_j \) to GW. GW computes \( \text{ID}_j^\prime \) from \( \text{AID}_j \) and then computes \( D_j^\prime \) and \( k_j^\prime \). \( U_i \) decrypts \( C_j \) and confirms \( \text{ID}_j^\prime \). Finally, GW authenticates \( U_i \) as follows if \( \text{ID}_j^\prime \) and \( \text{ID}_i^\prime \) are the same. \( U_i \) imprints \( B_i \) for authentication. And then GW computes \( M_1 \) as follows:

\[
\langle R_i, P_i \rangle = \text{Gen}(B_i), \\
A_i = h(R_i), \\
D_j = M_j \oplus A_i^*, \\
h(x \parallel y) = N_j \oplus A_i^*, \\
k_j = h(D_j \parallel T_j), \\
C_j = E_k(D_j \parallel X_j), \\
\text{AID}_j = \text{ID}_j \oplus h(h(x \parallel y) \parallel T_j), \\
M_1 = \langle \text{AID}_j, X_j, C_j, T_j, W_j \rangle.
\]

GW verifies sameness of ID as follows:

\[
\text{ID}'_j = \text{AID}_j \oplus h(h(x \parallel y) \parallel T_j), \\
D_j' = h(D_j' \parallel x), \\
k_j' = h(D_j' \parallel T_j), \\
\text{ID}_j'' \parallel X_j' = D_k'(C_j), \\
\text{verify } \text{ID}_j'' = \text{ID}_j'.
\]

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 \( h(x \parallel y) \) because the real identification of the user is always protected using \( \text{AID}_j = \text{ID}_j \oplus h(h(x \parallel y) \parallel T_j) \). Therefore, only the legal user \( U_i \) and GW can compute ID\(_j\) from \( \text{AID}_j \). GW stores \( x \) and \( y \), so GW can easily compute \( h(x \parallel y) \). \( U_i \) can compute \( h(x \parallel y) \) from the \( N_j \) stored in the smart card as follows:

\[
N_j = h(x \parallel y) \oplus A_i, \\
\langle R_i, P_i \rangle = \text{Gen}(B_i), \\
A_i = h(R_i), \\
h(x \parallel y) = N_j \oplus A_i^*.
\]

To compute \( h(x \parallel y) \), 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:

\[
\text{sk} = h(AID_j \parallel K_{US} \parallel T_j),
\]

\[
AID_j = ID_j \oplus h(h(x \parallel y) \parallel T_j).
\] (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:

\[
\text{sk} = h(AID_i \parallel r_i \times Y_i \parallel T_i) = h(AID_i \parallel r_i \times X_i \parallel T_i)
\]

\[
= h(AID_i \parallel r_i \times r_j \times P \parallel T_j).
\] (27)

GW can know AID, \( T_j \), \( X_i \), and \( Y_j \) but cannot acquire \( r_i \) and \( r_j \). Even though \( X_i = r_i \times P \) and \( Y_j = r_j \times P \), GW cannot compute \( r_i \) and \( r_j \) from \( X_i \) and \( Y_j \) 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 \( W_i \), \( W_g \), and \( V_s \) in \( M_1 \), \( M_2 \), and \( M_3 \), respectively:

\[
W_i = h(h(x \parallel y) \parallel AID_i \parallel X_i \parallel C_i \parallel T_i),
\]

\[
W_g = h(h(SID_j \parallel y) \parallel AID_i \parallel C_g \parallel T_g),
\] (28)

\[
V_s = h(AID_j \parallel X_j \parallel Y_j \parallel RM \parallel T_j).
\]

\( W_i, W_g \), 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, 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, 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 same reason. Therefore, the proposed scheme suggests for the GW to check for differences between the previous ID, and new ID 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 \( ID_1, AID_j, \) 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 \( ID_1 \). Only the legal case computes \( C_i \) using \( ID_1 \) 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 \( ID_1 \). Only a legal \( S_j \) can compute \( V_s \) within the current time because only \( S_j \) can know \( X_j \) and \( Y_j \).

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:

\[
M_1 = \langle AID_j, X_i, C_i, T_i, W_i \rangle,
\]

\[
M_2 = \langle AID_j, C_g, T_g, W_g \rangle,
\] (29)

\[
M_3 = \langle RM, Y_j, V_s, T_s \rangle.
\]

\( T_i, T_g, T_s, \) 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, \( X_i, C_i, W_i, C_g, W_g, Y_j, \) 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 \( h(\text{password}) \). 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 \( h(\text{password}) \) 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 \( h(\text{biometrics}) \) 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 \( h(\text{SID}_j \parallel y) \). 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 provided 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 \( \text{ID}_i, M_j, N_j, V_i, P_i, h(\cdot) \) from the smart card. However, since \( M_j, N_j, V_i \) are hashed values, the attacker cannot obtain any secret information from them. \( \text{ID}_i \) 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 \( T_{i1}, T_{i2}, T_i \) into authentication messages \( W_i, W_g, V_i \) 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 \( W_i, W_g, V_i \) using the current timestamps:

\[
M_1 = \langle \text{AID}_i, X_i, C_i, T_i, W_i \rangle,
\]

\[
W_i = h(h(x \parallel y) \parallel \text{AID}_i \parallel X_i \parallel C_i \parallel T_i),
\]

\[
M_2 = \langle \text{AID}_i, C_g, T_g, W_g \rangle,
\]

\[
W_g = h(h(\text{SID}_j \parallel y) \parallel \text{AID}_i \parallel C_g \parallel T_g),
\]

\[
M_3 = \langle \text{RM}, Y_i, V_i, T_i \rangle,
\]

\[
V_i = h(h(\text{AID}_i' \parallel X_i' \parallel Y_i' \parallel \text{RM} \parallel T_i).)
\]

7.14. Impersonation Attack. Even if an attacker intercepts the authentication message \( M_1 = \langle \text{AID}_i, X_i, C_i, T_i, W_i \rangle \) to impersonate a legitimate user, the attacker cannot normally extract \( D_i \) and \( k_i \) from \( \text{AID}_i, C_i \) 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 \( M_2 = \langle \text{AID}_i, C_g, T_g, W_g \rangle \), the attacker cannot compute \( k_i \) or \( h(h(\text{SID}_j \parallel y) \parallel T_g) \) 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 \( (R_i, P_i) \) using \( \text{Gen}(B_i) \) 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 $i$th user
- $I_D_i$: The identification of $U_i$
- $AID_i$: The anonymous identification of user $i$
- $B_j$: The biometric template of $U_i$
- GW: The gateway of WSNs
- $x, y$: Two master keys of GW
- $S_j$: The $j$th sensor node
- SID: $S_j$ identification
- $h(\cdot)$: A secure one-way hash function
- $d(\cdot)$: Symmetric parametric function
- $\tau$: 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
- $r_{ij}$: 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.
Submit your manuscripts at http://www.hindawi.com