



Article

# Improving Biometric-Based Authentication Schemes with Smart Card Revocation/Reissue for Wireless Sensor Networks

Jongho Moon <sup>1</sup>, Donghoon Lee <sup>1</sup>, Youngsook Lee <sup>2</sup> and Dongho Won <sup>3,\*</sup>

- Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea; jhmoon@security.re.kr (J.M.); dhlee@security.re.kr (D.L.)
- Department of Cyber Security, Howon University, 64 Howondae 3-gil, Impi-myeon, Gunsan-si, Jeonrabuk-do 54058, Korea; ysooklee@howon.ac.kr
- Department of Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea
- \* Correspondence: dhwon@security.re.kr; Tel.: +82-31-290-7107

Academic Editors: Subhas Chandra Mukhopadhyay, Hemant Ghayvat and Nagender Kumar Suryadevara Received: 8 March 2017; Accepted: 4 April 2017; Published: 25 April 2017

Abstract: User authentication in wireless sensor networks is more difficult than in traditional networks owing to sensor network characteristics such as unreliable communication, limited resources, and unattended operation. For these reasons, various authentication schemes have been proposed to provide secure and efficient communication. In 2016, Park et al. proposed a secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks. However, we found that their scheme was still insecure against impersonation attack, and had a problem in the smart card revocation/reissue phase. In this paper, we show how an adversary can impersonate a legitimate user or sensor node, illegal smart card revocation/reissue and prove that Park et al.'s scheme fails to provide revocation/reissue. In addition, we propose an enhanced scheme that provides efficiency, as well as anonymity and security. Finally, we provide security and performance analysis between previous schemes and the proposed scheme, and provide formal analysis based on the random oracle model. The results prove that the proposed scheme can solve the weaknesses of impersonation attack and other security flaws in the security analysis section. Furthermore, performance analysis shows that the computational cost is lower than the previous scheme.

Keywords: wireless sensor networks; user authentication; biometric; smart card

# 1. Introduction

Wireless Sensor Networks (WSNs) generally consist of gateways, users, and a large number of sensor nodes. The sensor node is tiny and can be easily deployed in various kinds of severe environments. However, each sensor node has limited resources, is lacking in memory, has low computational capabilities and short radio transmission range [1]. Data collected from sensor nodes in WSNs sometimes include valuable and classified information such as the environmental surrounding the wartime, a patient's personal information, monitoring information of museums, and the power company's voltage variation monitoring data. To ensure the confidentiality and reliability of deployed WSNs, only registered and legitimate users should be able to access them. In addition, establishing secure protocol actively requires a mutual authentication between the user and the sensor node. For these reasons, secure user authentication is one of the major issues in WSNs.

To support message confidentiality and secure authentication for sensor networks, various

Sensors 2017, 17, 940 2 of 24

authentication schemes for WSNs have been proposed. However, a problem that occurs with respect to password-based authentication schemes is that a server must maintain a password table to legitimately verify a login user. Therefore, the server requires additional memory space to store the password table. For this reason, many researchers have proposed a new type of remote user authentication scheme whereby personal biological characteristics are used, such as a fingerprint or an iris. The main advantage biometrics is uniqueness. A smart card can be used as a tool to store biometric information. Since the smart card has its own calculation function, it can operate at more than one level. In 2004, Watro et al. [2] proposed a user authentication scheme using the RSA and DH algorithms for WSNs. After that, Wong et al. [3] proposed a hash-based dynamic user authentication scheme in 2006. However, Tseng et al. [4] found that Wong et al.'s authentication scheme was vulnerable to replay, stolen verifiers, and forgery attacks. In 2009, Das [5] proposed and claimed that his scheme can resist various real-time attacks. Unfortunately, He et al. [6] found that Das's scheme was vulnerable to insider and impersonation attacks, and proposed an improved two factor scheme to solve these security problems. In the same year, Khan and Alghathbar [7] demonstrated that Das's scheme did not provide mutual authentication, and was vulnerable to gateway bypassing and privileged insider attacks, and Chen et al. [8] also found that Das's scheme did not provide mutual authentication between the gateway and the sensor, proposed a robust mutual authentication scheme for WSNs, and claimed that their scheme provides greater security than Das's scheme. In 2010, Yuan et al. [9] proposed a biometric-based user authentication scheme. However, Yoon et al. demonstrated that Yuan et al.'s scheme was vulnerable to insiders, user impersonation, gateway node impersonation and sensor node impersonation attacks. To solve these problems, Yoon et al. [10] proposed an improved user authentication scheme. Unfortunately, He et al. [11] found that Yoon et al.'s scheme was still vulnerable to denial of service (DoS) and sensor impersonation attacks, and then proposed an improved scheme to overcome these security problems.

In 2013, Yoon and Kim [12] pointed out that He et al.'s scheme had various security vulnerabilities such as poor repairability, and was vulnerable to user and sensor node impersonation attacks. They then proposed an advanced biometrics-based user authentication scheme for WSNs. They claimed that their scheme was more effective and had stronger security than other related schemes. However, Choi et al. [13] found that Yoon and Kim's scheme [12] had various security problems, including a biometric recognition error, a user verification problem, lack of anonymity, perfect forward secrecy, session key exposure by the gateway node, vulnerability to DoS attacks, and a revocation problem. To overcome these vulnerabilities, they proposed a biometric-based user authentication scheme using a fuzzy extractor, and claimed that their scheme is more secure than other authentication schemes. Unfortunately, Park et al. [14] demonstrated that Choi et al.'s scheme [13] was still insecure against user impersonation attacks, and had security weakness in the revocation/reissue phase. They then proposed an enhanced biometric-based authentication scheme for WSNs that has improved security functions. After careful analysis, however, we found that Park et al.'s scheme [14] was still insecure against impersonation attack, and also had a problem in the smart card revocation/reissue phase.

In this paper, we show how an adversary can impersonate a legitimate user or a sensor node, and perform an illegal smart card revocation/reissue. After demonstrating these problems, we propose an improved biometric authentication scheme. Finally, we analyze our proposed scheme for security properties and computational cost. The proposed scheme is more secure and efficient than other related authentication schemes.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the elliptic curve cryptosystem, threat assumptions and fuzzy extractors that we adopt in our scheme. In Sections 3 and 4, we review and analyze, respectively, Park et al.'s scheme [14]. In Section 5, we propose an improved authentication scheme for WSNs. In Section 6, we present a security analysis of the proposed scheme. Section 7 shows performance analysis comparing our scheme with previous schemes. Our conclusions are presented in Section 8.

Sensors 2017, 17, 940 3 of 24

#### 2. Preliminaries

In this section, we briefly introduce the elliptic curve cryptosystem, threat assumptions, and fuzzy extractor.

# 2.1. Elliptic Curve Cryptosystem

The elliptic curve cryptosystem (ECC) was first proposed by Koblitz [15] and Miller [16] to design public key cryptosystems, and presently it is widely used in several cryptographic schemes to provide desired levels of security and performance. An elliptic curve  $E_K$  defined over a field K of the characteristic  $\neq 2$  or 3 is the set of solutions  $(x, y) \in K^2$  to the equation:

$$y^2 = x^3 + ax + b$$
,  $a, b \in K, 4a^3 + 27b^2 \neq 0$ . (1)

Cryptosystems based on  $GF(q)^*$  can be translated to systems using the group E, where E is an elliptic curve defined over a GF(q). The point multiplications kP = (P + P + ... + P, k times) that means k times addition of point P. Given an elliptic curve E defined over a GF(q) and two points P,  $Q \in E$ , it finds an integer E such that E0 are sistength. This problem proved to be more intractable than the typically discrete logarithm problem. The details of the ECC definitions can be found in [15]. There are several computational problems based on ECC which are presented below:

**Definition 1.** The elliptic curve discrete logarithm problem (*ECDLP*) is defined as: given two elements Q,  $R \in G_p$ , find an integer  $k \in [1, n-1]$  such that R = kQ.

**Definition 2.** The computational Diffie–Hellman problem (*CDHP*) is defined as: given three elements (P, aP, bP) for any a,  $b \in [1, n-1]$ , computation of abP is very hard to the group  $G_p$ .

**Definition 3.** The elliptic curve factorization problem (*EC FP*) is defined as: given two elements  $P, Q \in G_p$ , where Q = sP + tP and  $(s, t) \in [1, n - 1]$ , then computation of sP and tP is impossible.

**Definition 4.** The decisional Diffie–Hellman problem ( $DD\ HP$ ) is defined as: given four elements (P, aP, bP, cP) for any (a, b, c)  $\in$  [1, n – 1], decide whether or not cP = abP i.e., c =  $ab\ mod\ p$  or not.

**Definition 5.** The weak Diffie–Hellman problem (*WDHP*) is defined as: for  $Q \in G_p$  and some  $k \in [1, n-1]$  from the given three elements (P, Q, kP), computation of kQ is very hard.

# 2.2. Threat Assumptions

We introduce the Dolev–Yao [17] and some threat model [18,19], and consider the risk of side channel attack [20] to construct the threat assumptions, which are described as follows:

- 1. An adversary A can be either a user, gateway, or sensor node. Any registered user can act as an adversary.
- 2. An adversary A can eavesdrop every communication in a public channel, thereby capturing any message exchanged between a user and gateway or sensor node.
- 3. An adversary A has the ability to alter, delete or reroute the captured message.
- 4. Information can be extracted from the smart card by examining the power consumption of the card.

# 2.3. Fuzzy Extractor

In this subsection, we describe the basis for a biometric-based fuzzy extractor, which converts biometric information data into a random value. Based on [21–23], the fuzzy extractor is given by two procedures (*Gen*, *Rep*). The mechanism of a fuzzy extractor consists of two procedures (*Gen*, *Rep*), which is demonstrated as:

Sensors **2017**, *17*, 940 4 of 24

- $Gen(BIO) \rightarrow \langle R, P \rangle$ ,
- $Rep(BIO^*, P) = R$  if  $BIO^*$  is reasonably close to BIO.

The function Gen is a probabilistic generation procedure, which on biometric input BIO outputs an "extracted" string  $R \in \{0,1\}^l$  and an auxiliary string  $P \in \{0,1\}^*$ . Rep is a deterministic reproduction procedure, which recovery of R from the corresponding auxiliary string P, and any vector  $BIO^*$  close to BIO. The detailed information about fuzzy extractors can be founded in the literature [24].

# 3. Review of Park et al.'s Authentication Scheme

In this section, we review Park et al.'s authentication scheme [14]. The notations used in the paper are listed in Table 1. The GW generates two master keys, x and y, and provides a long-term secret key  $h(SID_j||y)$  to the sensor  $S_j$  before the scheme is executed. Their scheme involves three parties, i.e., the user  $U_i$ , the gateway GW and the sensor  $S_j$ , to communicate with each other to perform the following three phases: registration, login/authentication, and revocation/reissue.

Term	Description
$\overline{U_i}$	user i
$\mathcal A$	adversary
$B_i$	biometric template of $U_i$
$E_k(\cdot)/D_k(\cdot)$	encryption or decryption with key $k$
GW	gateway node
$\mathbb{G}_1$	cyclic groups of order q
$h(\cdot)$	hash function
$h(SID_i  y)$	long-term secret of $S_i$ generated by $GW$
$ID_i^{'}$	actual identity of $U_i$
P	generator of $\mathbb{G}_1$
$r_i, r_1, r_2$	random number generated by $U_i$
$r_{\scriptscriptstyle S}$	random number generated by $S_j$
$S_{j}$	sensor node <i>j</i>
SĆ <sub>i</sub>	smart card of user $U_i$
$SID_{j}$	identity of $S_i$
$T_i$	time stamp
$up_i$	<i>i</i> -th update phase
x, y	two master keys of GW
RM	Response to the query message
	concatenation operation
<b></b>	bitwise XOR operation

Table 1. Notations used in this paper.

# 3.1. Registration Phase

In this phase, a user  $U_i$  chooses an identity  $ID_i$ , imprints biometric template  $B_i$ , and then performs the following steps:

- 1.  $U_i$  computes  $\langle R_i, P_i \rangle = Gen(B_i)$  and  $A_i = h(R_i)$ .  $U_i$  then sends  $\langle ID_i, A_i \rangle$  to the gateway GW.
- 2. Upon receiving  $\langle ID_i, A_i \rangle$  from  $U_i$ , GW computes the authentication parameters as:

$$V_i = h(ID_i||A_i)$$

$$M_i = h(x||y||A_i)$$

$$N_i = M_i \oplus A_i$$

$$C_i = E_x(A_i||up_i).$$

- 3. *GW* stores  $h(\cdot)$  and the authentication parameters;  $\langle V_i, N_i, C_i, h(\cdot) \rangle$  in the smart card  $SC_i$ . *GW* then issues  $SC_i$  to  $U_i$  through a secure channel.
- 4. After receiving the smart card from GW,  $U_i$  stores  $P_i$  in the smart card.

Sensors **2017**, *17*, 940 5 of 24

# 3.2. Login and Authentication Phase

If a user  $U_i$  wants to log in to the GW and  $S_j$ ,  $U_i$  performs the login phase, and then  $U_i$ , GW and  $S_j$  verify each other's authenticities. Finally,  $U_i$  and  $S_j$  generate a session key in this phase as follows:

1.  $U_i$  enters  $ID_i$  and imprints biometric template  $B_i^*$ .  $SC_i$  then computes  $R_i^*$ ,  $A_i^*$  and  $V_i^*$  using fuzzy extraction, and compares  $V_i^*$  with  $V_i$  as:

$$\begin{split} R_i^* &= Rep(B_i^*, P_i) \\ A_i^* &= h(R_i^*) \\ V_i^* &= h(ID_i||A_i^*) \\ \textit{verifies} \quad V_i \overset{?}{=} V_i^*. \end{split}$$

2.  $SC_i$  generates a random number  $r_i$  and computes  $X_i$  and  $M_i$  as:

$$X_i = r_i \times P$$
$$M_i = N_i \oplus A_i.$$

3.  $U_i$  picks up  $T_i$  and computes  $AID_i$  and  $W_i$  as:

$$AID_i = ID_i \oplus h(M_i||T_i)$$
  
$$W_i = h(M_i||ID_i||X_i||T_i).$$

 $U_i$  then sends the login request message  $M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle$  to GW.

4. After receiving  $M_1$ , GW retrieves T' and verifies  $T' - T_i \leq \triangle T$ . If this holds, GW computes  $A_i^*$ ,  $M_i^*$ ,  $ID_i^*$  and  $W_i^*$ , and compares  $W_i^*$  with  $W_i$  as:

$$A_{i}^{*}||up_{i} = D_{x}(C_{i})$$
 $M_{i}^{*} = h(x||y||A_{i}^{*})$ 
 $ID_{i}^{*} = AID_{i} \oplus h(M_{i}^{*}||T_{i})$ 
 $W_{i}^{*} = h(M_{i}^{*}||ID_{i}^{*}||X_{i}^{*}||T_{i})$ 
verifies  $W_{i} \stackrel{?}{=} W_{i}^{*}$ .

If this holds, GW verifies the legitimacy of  $U_i$ .

5. *GW* picks up  $T_g$  and computes  $k_g$ ,  $C_g$  and  $W_g$ ,

$$\begin{split} k_g &= h(h(SID_j||y)||T_g) \\ C_g &= E_{k_g}(AID_i||X_i) \\ W_g &= h(h(SID_j||y)||AID_i||C_g||T_g). \end{split}$$

*GW* then sends the authentication message  $M_2 = \langle AID_i, C_g, T_g, W_g \rangle$  to  $S_j$ .

6. When receiving  $M_2$  from GW,  $S_j$  retrieves T'', and verifies  $T'' - T_g \le \triangle T$ . If this holds,  $S_j$  verifies the validity of  $W_g$  by comparing it with  $h(h(SID_j||y)||AID_i||C_g||T_g)$  to check the legitimacy of GW. After that,  $S_j$  computes  $k_g^*$ , and decrypts  $C_g$  using  $k_g^*$ .  $S_j$  then checks the validity of the received  $AID_i$  by comparing  $AID_i^*$  as:

$$k_g^* = h(h(SID_j||y)||T_g)$$
 $D_{k_g^*} = AID_i^*||X_i^*$ 
 $verifies \quad AID_i \stackrel{?}{=} AID_i^*.$ 

Sensors 2017, 17, 940 6 of 24

7.  $S_j$  generates a random number  $r_s$  and computes  $K_{SU}$ ,  $Y_i$  and a session key sk.  $S_j$  then picks up  $T_s$  and computes RM,  $V_s$  as:

$$K_{SU} = r_x \times X_i^*$$
  
 $Y_i = r_s \times P$   
 $sk = h(AID_i||K_{SU}||T_s)$   
 $RM = \text{Response to the query of } U_i$   
 $V_s = h(AID_i||X_i||Y_i||RM||T_s).$ 

 $S_i$  hence sends  $M_3 = \langle RM, Y_i, V_s, T_s \rangle$  to  $U_i$ .

8. Úpon receiving  $M_3$ ,  $U_i$  retrieves  $T_s$ , and checks the sameness of  $V_s$ .  $U_i$  then computes  $K_{US}$  and sk as:

$$V_s^* = h(AID_i||X_i||Y_i||RM||T_s)$$
  
verifies  $V_s^* \stackrel{?}{=} V_s$   
 $K_{US} = r_i \times Y_i$   
 $sk = h(AID_i||K_{US}||T_s).$ 

Only the legitimate  $U_i$  can compute  $K_{US}$  and sk.  $U_i$  then accepts RM. Finally,  $U_i$  and  $S_j$  can communicate securely using the common sk.

3.3. Revocation or Reissue Phase

To make up for smart card loss or long term key disclosure, the smart card should be revoked and reissued in cycles.

- 1.  $U_i$  who wants to revoke and reissue a smart card inputs the previous identity  $ID_i$  and the new identity  $ID_i^*$  to prevent adversaries from registering with the same identity  $ID_i$ . Then,  $U_i$  imprints biometric template  $B_i$  and computes  $R_i = Rep(B_i, P_i)$ ,  $A_i = h(R_i)$  and  $M_i = N_i \oplus A_i$ .
- 2.  $U_i$  computes  $Z_i = ID_i \oplus M_i$  and sends the revocation/reissue request message  $\langle ID_i, ID_i^*, A_i, Z_i \rangle$  to GW.
- 3. *GW* computes  $M_i^*$ ,  $Z_i^*$  and checks the legitimacy of the user as:

$$M_i^* = h(x||y||A_i)$$
  
 $Z_i^* = ID_i \oplus M_i^*$   
 $verifies \quad Z_i^* \stackrel{?}{=} Z_i.$ 

4. If this holds, GW revokes  $ID_i$  and records it on the revocation look-up table. Then, GW computes new authentication parameters  $V_i$ ,  $N_i$  and  $C_i$  as:

$$V_i = h(ID_i^*||A_i)$$

$$N_i = M_i^* \oplus A_i$$

$$C_i = E_x(A_i||up_i^*).$$

- 5. *GW* stores  $h(\cdot)$  and the new authentication parameters;  $\langle V_i, N_i, C_i, h(\cdot) \rangle$  in the smart card  $SC_i$ . *GW* then reissues  $SC_i$  to  $U_i$  through a secure channel.
- 6.  $U_i$  stores  $P_i$  in the smart card.

# 4. Cryptanalysis of Park et al.'s Authentication Scheme

In this section, we analyze the security problems of Park et al.'s scheme [14]. Park et al. cryptanalyzed a scheme of Choi et al. [13] and improved it to support better security functionality. However, we found that Park et al.'s scheme [14] was still insecure against impersonation attack and

Sensors 2017, 17, 940 7 of 24

had a problem with smart card revocation/reissue. The following attacks are based on the threat assumptions that a malicious adversary A was completely monitored through the communication channel connecting  $U_i$ , GW and  $S_i$  in the login and authentication phases, and that the A obtained the information saved in their own smart card [20]. A therefore can eavesdrop, modify, insert, or delete any message transmitted over a public channel. We now reveal the details of these problems.

# 4.1. User Impersonation Attack

- Let A be an active adversary who is he/she legal user and owns a smart card to extract information  $\langle V_a, N_a, C_a, h(\cdot), P_a \rangle$ .
- 2.  $\mathcal{A}$  then imprints one's biometric template  $B_a^*$  and computes  $R_a^* = Rep(B_a^*, P_a)$  and  $A_a^* = h(R_a^*)$ .
- A generates a random number  $r_A$ , and selects any identity  $ID_i$ . A then computes login request message  $M_1$  as:

$$X_a = r_a \times P$$

$$M_a = N_a \oplus A_a^*$$

$$AID_i = ID_i \oplus h(M_a||T_a)$$

$$W_a = h(M_a||ID_i||X_a||T_a).$$

A then sends the login request message  $M_1 = \langle AID_i, X_a, C_a, T_a, W_a \rangle$  to GW.

When receiving  $M_1$ , GW retrieves T' and verifies  $T' - T_a \leq \triangle T$ . If this holds, GW computes  $A_a^*$ ,  $M_a^*$ ,  $ID_i^*$ ,  $W_a^*$  and compares  $W_a^*$  with  $W_a$  as:

$$A_a^*||up_a = D_x(C_a)$$
  
 $M_a^* = h(x||y||A_a^*)$   
 $ID_i^* = AID_i \oplus h(M_a^*||T_a)$   
 $W_a^* = h(M_a^*||ID_i^*||X_a^*||T_a)$   
verifies  $W_a \stackrel{?}{=} W_a^*$ .

If this holds and IDi does exist in the database, the gateway GW continues to proceed the scheme without detected. Otherwise, A selects another identity nominee and performs the same processes until he/she locates the valid identity.

5. GW picks up  $T_g$  and computes  $k_g$ ,  $C_g$  and  $W_g$ :

$$\begin{split} k_g &= h(h(SID_j||y)||T_g) \\ C_g &= E_{k_g}(AID_i||X_a) \\ W_g &= h(h(SID_j||y)||AID_i||C_g||T_g). \end{split}$$

*GW* then sends the authentication message  $M_2 = \langle AID_i, C_g, T_g, W_g \rangle$  to  $S_j$ . Upon receiving  $M_2$  from *GW*,  $S_j$  retrieves T'' and verifies  $T'' - T_g \leq \triangle T$ . If this holds,  $S_j$  verifies the validity of  $W_g$  by comparing it with  $h(h(SID_j||y)||AID_i||C_g||T_g)$  to check the legitimacy of GW. After that,  $S_i$  computes  $k_g^*$  and decrypts  $C_g$  using  $k_g^*$ .  $S_i$  then checks the validity of the received  $AID_i$  by comparing  $AID_i^*$  as

$$k_g^* = h(h(SID_j||y)||T_g)$$
 $D_{k_g^*} = AID_i^*||X_a^*$ 
 $verifies \quad AID_i \stackrel{?}{=} AID_i^*.$ 

 $S_i$  generates a random number  $r_s$  and computes  $K_{SU}$ ,  $Y_i$  and a session key sk.  $S_i$  then computes RM,  $V_s$  as:

Sensors 2017, 17, 940 8 of 24

$$K_{SU} = r_x \times X_a^*$$
  
 $Y_i = r_s \times P$   
 $sk = h(AID_i||K_{SU}||T_s)$   
 $RM = \text{Response to the query of } U_i$   
 $V_s = h(AID_i||X_a||Y_i||RM||T_s)$ ,

where  $T_s$  is current timestamp.  $S_i$  then sends  $M_3 = \langle RM, Y_i, V_s, T_s \rangle$  to A.

8. After receiving  $M_3$ , A retrieves  $T_s$ , and checks the sameness of  $V_s$ . Then, A computes  $K_{US}$  and sk as:

$$V_s^* = h(AID_i||X_a||Y_i||RM||T_s)$$
  
verifies  $V_s^* \stackrel{?}{=} V_s$   
 $K_{US} = r_a \times Y_i$   
 $sk = h(AID_i||K_{US}||T_s).$ 

Lastly, A and  $S_j$  "successfully" agree on a session key sk. Unfortunately, the sensor  $S_j$  mistakenly believes that he/she is communicating with the legitimate user  $U_i$ .

# 4.2. Sensor Node Impersonation Attack

Park et al. [14] claimed that if  $\mathcal{A}$  wants to masquerade as the sensor node  $S_j$ ,  $\mathcal{A}$  is required to compute  $h(SID_j||y)$ . This is because the symmetric key  $k_g = h(h(SID_j||y)||T_g)$ . However, if  $\mathcal{A}$  obtains the login request message  $M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle$  of the legitimate user  $U_i$ ,  $\mathcal{A}$  then can easily impersonate the sensor node  $S_j$ .

1. After receiving  $M_2$  from GW, A generates a random number  $r_a$  and computes  $K_{AU}$ ,  $Y_a$ , RM,  $V_a$  and a session key sk as:

$$K_{AU} = r_a \times X_i$$
  
 $Y_a = r_a \times P$   
 $sk = h(AID_i||K_{AU}||T_a)$   
 $RM = \text{Any response to the query of } U_i$   
 $V_a = h(AID_i||X_i||Y_a||RM||T_a),$ 

where  $T_a$  is current timestamp. A then sends  $M_3 = \langle RM, Y_a, V_a, T_a \rangle$  to  $U_i$ .

2. Upon receiving  $M_3$  from A,  $U_i$  retrieves  $T_a$  and checks the sameness of  $V_a$ . Then,  $U_i$  computes  $K_{UA}$  and sk as:

$$V_a^* = h(AID_i||X_i||Y_a||RM||T_a)$$
  
verifies  $V_a^* \stackrel{?}{=} V_a$   
 $K_{UA} = r_i \times Y_a$   
 $sk = h(AID_i||K_{UA}||T_a)$ .

Lastly,  $U_i$  and A "successfully" agree on a session key sk. Unfortunately, the user  $U_i$  mistakenly believes that he/she is communicating with the legitimate sensor  $S_j$ .

## 4.3. Illegal Smart Card Revocation/Reissue Attack

Park et al. [14] claimed that, although A could get the identity  $ID_i$  in some way, GW checks the legitimacy of the user on the requested identity, and A cannot compute  $M_i$  and the revocation/reissue request message  $Z_i$  without the biometric information of  $U_i$ . However, A can modify the

Sensors 2017, 17, 940 9 of 24

revocation/reissue request message. This is because GW cannot distinguish whether or not the user who wishes to revoke  $ID_i$  is the real user  $U_i$ .

- 1. Suppose A owns a smart card to extract information  $\langle V_a, N_a, C_a, h(\cdot), P_a \rangle$  and obtains the identity  $ID_i$  of the legitimate user  $U_i$  by using a user impersonation attack.
- 2. Next, A imprints the personal biometric information  $B_a$  at the sensor. The sensor hence sketches  $B_a$  and extracts  $\langle R_a, P_a \rangle$  from  $Gen(B_a) \to \langle R_a, P_a \rangle$ .
- 3.  $\mathcal{A}$  computes  $A_a^* = h(R_a)$  and  $Z_i' = ID_i \oplus N_a \oplus A_a^*$  and sends the revocation/reissue request message  $\langle ID_i, ID_i^*, A_a^*, Z_i' \rangle$  to GW.
- 4. *GW* computes  $M_i^*$ ,  $Z_i^*$ , and checks the legitimacy of the user as:

$$M_a^* = h(x||y||A_a^*)$$
  
 $Z_i^* = ID_i \oplus M_a^*$   
verifies  $Z_i' \stackrel{?}{=} Z_i^*$ .

5. If this holds, GW revokes  $ID_i$  and records it on the revocation look-up table. Then, GW computes new authentication parameters  $V_i$ ,  $N_i$  and  $C_i$  as:

$$V_i = h(ID_i^*||A_a^*)$$

$$N_i = M_a^* \oplus A_a^*$$

$$C_i = E_x(A_a^*||up_i^*).$$

- 6. *GW* stores the new authentication parameters  $\langle V_i, N_i, C_i, h(\cdot) \rangle$  in the smart card  $SC_i$ . *GW* then reissues  $SC_i$  to A through a secure channel.
- 7. A stores  $P_a$  in the smart card.

Adversary A can revoke the smart card of an authenticated user who does not wish the said smart card to be revoked without permission because GW has no proper process for checking the legitimacy of the user on the previous identity  $ID_i$ .

# 5. The Proposed Scheme

In this section, we will propose a new biometric-based password authentication scheme using a smart card. In our scheme, there are also three participants, the user  $U_i$ , the gateway GW and the sensor  $S_j$ . The GW generates two master keys x and y, and sends a long-term secret key  $h(SID_j||y)$  to the sensor  $S_j$  before the scheme is executed. After that, GW computes  $x \times P$  where xP is the public key of the gateway. The proposed scheme consists of three phases: registration, login and authentication, and revocation/reissue.

## 5.1. Registration Phase

In this phase, a user  $U_i$  chooses an identity  $ID_i$ , imprints biometric template  $B_i$  at the sensor, and then performs the following steps:

- 1. The sensor sketches  $B_i$ , extracts  $\langle R_i, P_i \rangle$  from  $Gen(B_i) \to \langle R_i, P_i \rangle$ , and stores  $P_i$  in the memory.  $U_i$  then sends  $\langle ID_i, A_i = h(R_i) \rangle$  to GW over a secure channel.
- 2. When receiving the registration request message  $\langle ID_i, A_i \rangle$  from  $U_i$ , the gateway GW computes the authentication parameters as:

$$C_i = h(ID_i||x||y)$$

$$M_i = h(C_i) \oplus A_i$$

$$N_i = x \oplus C_i \oplus y$$

$$V_i = h(ID_i||A_i).$$

Sensors 2017, 17, 940 10 of 24

3. *GW* stores the authentication parameters  $\langle M_i, N_i, V_i, h(\cdot) \rangle$  in the smart card  $SC_i$ . *GW* hence issues  $SC_i$  to  $U_i$  via a secure channel.

4. Lastly,  $U_i$  stores  $P_i$  in the smart card.

Figure 1 illustrates the registration phase of the proposed scheme.

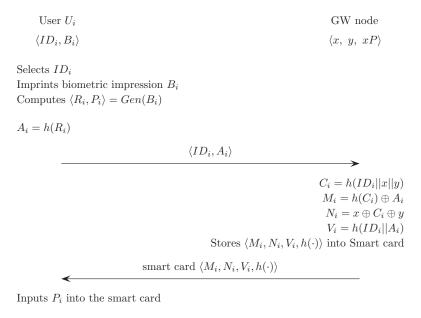


Figure 1. Registration phase of the proposed scheme.

# 5.2. Login and Authentication Phase

In this phase,  $U_i$  performs the login phase, and hence  $U_i$ , GW and  $S_j$  verify each other's authenticity. Finally,  $U_i$  and  $S_j$  generates a common session key in this phase as follows:

- 1.  $U_i$  inserts his/her smart card  $SC_i$  into the card reader, inputs the identity  $ID_i$ , and imprints the personal biometrics  $B_i^*$  at the sensor.
- 2. The sensor then sketches  $B_i^*$  and extracts  $R_i$  from  $Rep(B_i^*, P_i) \to \langle R_i \rangle$ . Then,  $SC_i$  computes  $A_i^*$  and  $V_i^*$  using fuzzy extraction and compares  $V_i^*$  with  $V_i$  as:

$$R_{i}^{*} = Rep(B_{i}^{*}, P_{i})$$
  
 $A_{i}^{*} = h(R_{i}^{*})$   
 $V_{i}^{*} = h(ID_{i}||A_{i}^{*})$   
 $verifies V_{i} \stackrel{?}{=} V_{i}^{*}.$ 

3.  $SC_i$  generates random numbers  $r_1$  and  $r_2$  and hence computes

$$X_i = r_1 \times P$$
  
 $h(C_i) = M_i \oplus A_i^*$   
 $AID_i = ID_i \oplus h(r_2)$   
 $M_1 = r_2 \oplus h(C_i)$   
 $M_2 = h(AID_i||h(C_i)||X_i||r_2||T_i)$   
 $M_3 = N_i \oplus (r_1 \times xP)$ ,

where  $T_i$  is current timestamp. Then,  $U_i$  sends the login request message  $\langle AID_i, X_i, M_1, M_2, M_3, T_i \rangle$  to GW.

Sensors 2017, 17, 940 11 of 24

4. Upon receiving a login request message from  $U_i$ , GW retrieves T' and verifies  $T' - T_i \leq \triangle T$ . If this is true, GW computes  $C_i^*$ ,  $r_2^*$ ,  $ID_i^*$  and  $M_2^*$  and compares  $C_i^*$  with  $h(ID_i^*||x||y)$  and  $M_2^*$  with  $M_2$  as:

$$C_{i}^{*} = M_{3} \oplus (x \times X_{i}) \oplus x \oplus y$$

$$r_{2}^{*} = M_{1} \oplus h(C_{i}^{*})$$

$$ID_{i}^{*} = AID_{i} \oplus h(r_{2}^{*})$$

$$verifies \quad C_{i}^{*} = h(ID_{i}^{*}||x||y)$$

$$M_{2}^{*} = h(AID_{i}||h(C_{i}^{*})||X_{i}||r_{2}^{*}||T_{i})$$

$$verifies \quad M_{2} \stackrel{?}{=} M_{2}^{*}.$$

If this holds, GW verifies the legitimacy of  $U_i$ .

5. *GW* computes  $k_g$ ,  $C_g$  and  $W_g$ ,

$$\begin{split} k_g &= h(h(SID_j||y)||T_g) \\ C_g &= E_{k_g}(AID_i||r_2||X_i) \\ W_g &= h(h(SID_j||y)||AID_i||C_g||T_g), \end{split}$$

where  $T_g$  is current timestamp. GW then sends the authentication message  $\langle AID_i, C_g, T_g, W_g \rangle$  to  $S_i$ .

6. When receiving the authentication message from GW,  $S_j$  retrieves T'' and verifies  $T'' - T_g \le \triangle T$ . If this is true,  $S_j$  verifies the validity of  $W_g$  by comparing it with  $h(h(SID_j||y)||AID_i||C_g||T_g)$  to check the legitimacy of GW. After that,  $S_j$  computes  $k_g^*$ , and decrypts  $C_g$  using  $k_g^*$ . Then,  $S_j$  checks the validity of the received  $AID_i$  by comparing the computed  $AID_i^*$  as

$$\begin{aligned} k_g^* &= h(h(SID_j||y)||T_g) \\ D_{k_g^*} &= AID_i^*||r_2^*||X_i^* \\ verifies &\quad AID_i \stackrel{?}{=} AID_i^*. \end{aligned}$$

7.  $S_i$  generates a random number  $r_s$  and computes  $K_{SU}$ ,  $Y_i$ , RM,  $V_s$  and a session key sk as:

$$K_{SU} = r_s \times X_i^*$$
  
 $Y_i = r_s \times P$   
 $sk = h(AID_i||K_{SU}||T_s)$   
 $RM = \text{Response to the query of } U_i$   
 $V_s = h(AID_i||r_2^*||Y_i||sk||RM||T_s),$ 

where  $T_s$  is current timestamp.  $S_i$  then sends  $\langle RM, Y_i, V_s, T_s \rangle$  to  $U_i$ .

8. After receiving response message  $\langle RM, Y_i, V_s, T_s \rangle$  from  $S_j$ ,  $U_i$  computes sk and checks whether  $V_s^*$  is equal to  $V_s$ :

$$K_{US} = r_1 \times Y_i$$
  
 $sk = h(AID_i||K_{US}||T_s)$   
 $V_s^* = h(AID_i||r_2||Y_i||sk||RM||T_s)$   
 $verifies \quad V_s^* \stackrel{?}{=} V_s.$ 

The legitimate user  $U_i$  can only compute  $K_{US}$  and sk.  $U_i$  and  $S_j$  can communicate securely using the common session key sk.

Figure 2 illustrates the login and authentication phase of the proposed scheme.

Sensors **2017**, *17*, 940

```
User U_i
                                                                                     GW node
                                                                                                                                            Sensor node S_i
   \langle ID_i, \text{ smart card} \rangle
                                                                                                                                             \langle h(SID_j||y)\rangle
                                                                                    \langle x,\ y,\ xP\rangle
Inserts smart card
Inputs ID_i and biometric B_i^*
R_i^* = Rep(B_i^*, P_i), \ A_i^* = h(R_i^*)
V_i^* = h(ID_i || A_i^*)
Verifies V_i \stackrel{?}{=} V_i^*
Generates random number r_1, r_2
X_i = r_1 \times P, \ h(C_i) = M_i \oplus A_i^*
Picks up timestamp \mathcal{T}_i
AID_i = ID_i \oplus h(r_2), \ M_1 = r_2 \oplus h(C_i),
M_2 = h(AID_i||h(C_i)||X_i||r_2||T_i), M_3 = N_i \oplus (r_1 \times xP)
                          \langle AID_i,~X_i,~M_1,~M_2,~M_3,~T_i\rangle
                                                                           Verifies (T' - T_i) \leq \Delta T
                                                                        C_i^* = M_3 \oplus (x \times X_i) \oplus x \oplus y
                                                                               r_2^* = M_1 \oplus h(C_i^*)
                                                                             ID_i^* = AID_i \oplus h(r_2^*)
                                                                         Verifies C_i^* \stackrel{?}{=} h(ID_i^*||x||y)
                                                                     M_2^* = h(AID_i||h(C_i^*)||X_i||r_2^*||T_i)
                                                                               Verifies M_2 \stackrel{?}{=} M_2^*
                                                                            Picks up timestamp T_g
                                                                           k_g = h(h(SID_j||y)||T_g)
                                                                           C_g = E_{k_g}(AID_i||r_2^*||X_i)
                                                                    W_g = h(h(SID_j||y)||AID_i||C_g||T_g)
                                                                                                                \langle AID_i,\ C_g,\ T_g,\ W_g\rangle
                                                                                                                                   Verifies (T'' - T_g) \le \Delta T
                                                                                                           Verifies W_g \stackrel{?}{=} h(h(SID_j||y)||AID_i||C_g||T_g)
                                                                                                                                     k_q^* = h(h(SID_j||y)||T_g)
                                                                                                                                   AID_i^* ||r_2^*|| X_i^* = D_{k_g^*}(C_g)
                                                                                                                                     Verifies AID_i \stackrel{?}{=} AID_i^*
                                                                                                                            Generates random number r_s
                                                                                                                             K_{SU} = r_s \times X_i^*, \ Y_i = r_s \times P
                                                                                                                      Computes sk = h(AID_i || K_{SU} || T_s)
                                                                                                                                     Picks up timestamp T_s
                                                                                                                     RM = Respond to the query of U_i
                                                                                                                            V_s = h(AID_i || X_i^* || Y_i || RM || T_s)
                                                                    \langle RM, Y_i, V_s, T_s \rangle
Verifies (T''' - T_s) \le \Delta T
Verifies V_s \stackrel{?}{=} h(AID_i || X_i || Y_i || RM || T_s)
K_{US} = r_1 \times Y_i
Computes sk = h(AID_i || K_{US} || T_s)
Accepts RM
                                                  Shared sk = h(AID_i||r_1 \times r_s \times P||T_s)
```

Figure 2. Login and authentication phase of the proposed scheme.

# 5.3. Revocation or Reissue Phase

To make up for smart card loss or long term key disclosure, the smart card should be revoked and reissued in cycles.

1. If  $U_i$  wants to revoke and reissue a smart card, he/she inserts his/her smart card  $SC_i$  into the card reader, inputs the previous identity  $ID_i$  and the new identity  $ID_i^*$  to prevent adversaries from registering with the same identity  $ID_i$ , and then imprints the personal biometrics  $B_i^*$  at the sensor.

Sensors 2017, 17, 940 13 of 24

2. The sensor then sketches  $B_i^*$  and extracts  $R_i$  from  $Rep(B_i^*, P_i) \to \langle R_i \rangle$ . Then,  $SC_i$  computes  $A_i^*$  and  $V_i^*$  using fuzzy extraction,

$$R_i = Rep(B_i^*, P_i)$$
  
 $A_i = h(R_i).$ 

- 3.  $U_i$  computes  $Z_i = ID_i \oplus M_i$  and sends the revocation/reissue request message  $\langle ID_i, ID_i^*, A_i, Z_i \rangle$  to GW over a secure channel.
- 4. *GW* first checks whether  $ID_i$  is the same as  $ID_i^*$  or not. If they are different, *GW* computes  $M_i^*$ ,  $Z_i^*$  and checks the legitimacy of the user as:

$$C_i^* = h(ID_i||x||y)$$
  
 $Z_i^* = ID_i \oplus h(C_i^*) \oplus A_i$   
verifies  $Z_i^* \stackrel{?}{=} Z_i$ .

5. If this is true, GW revokes  $ID_i$  and records it on the revocation look-up table. Then, GW computes new authentication parameters  $V_i$ ,  $N_i$  and  $C_i$  as:

$$C_i = h(ID_i^*||x||y)$$

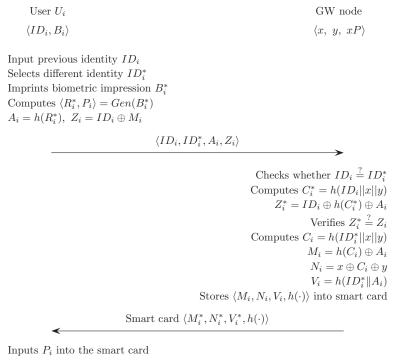
$$M_i = h(C_i) \oplus A_i$$

$$N_i = x \oplus C_i \oplus y$$

$$V_i = h(ID_i^*||A_i).$$

- 6. *GW* stores  $h(\cdot)$  and the new authentication parameters  $\langle M_i, N_i, V_i, h(\cdot) \rangle$  in the smart card  $SC_i$ . *GW* then reissues  $SC_i$  to  $U_i$  through a secure channel.
- 7.  $U_i$  stores  $P_i$  in the smart card.

Figure 3 illustrates the revocation/reissue phase of the proposed scheme.



**Figure 3.** Revocation/reissue phase of proposed scheme.

Sensors 2017, 17, 940 14 of 24

# 6. Security Analysis

In this section, we demonstrate that the proposed scheme, which retains the merits of Park et al.'s scheme [14], can withstand several types of possible attacks; and we also show that the scheme supports several security properties. The security analysis of the proposed scheme was conducted with the threat assumptions made in the Preliminaries.

# 6.1. Formal Security Analysis

In this subsection, we have demonstrated that the proposed scheme is secure through a formal proof using the random oracle model [18,25]. At first, we specify a collision-free one-way hash function as follows.

**Definition 6.** The collision-resistance one-way hash function  $f : \{0, 1\}^* \to \{0, 1\}^n$  pick up an input as a binary string  $x \in \{0, 1\}^*$  that has a random length, produces a binary string  $h(x) \in \{0, 1\}^n$ , and gratifies the following requirements:

- Given the  $y \in Y$ , it is not possible to find out computationally about  $x \in X$  such that y = h(x).
- Given the  $x \in X$ , it is not possible to find out computationally about another  $x' \neq x \in X$ , such that h(x') = h(x).
- It is not possible to find out computationally about a pair  $(x', x) \in X' \times X$ , with  $x' \neq x$ , such that h(x') = h(x).

**Theorem 1.** According to the assumption that if the collision-free one-way hash function  $h(\cdot)$  closely acts like an oracle, the proposed scheme is then distinctly secure against an adversary A for the protection of the sensitive information including the identity  $ID_i$ , nearly random binary string  $r_2$  and master secret key x, y of the gateway node GW.

**Proof.** This random oracle can extract the input value x from the given hash result y = h(x) without fail.  $\mathcal{A}$  now executes the experimental algorithm as shown in Algorithm 1,  $EXP_{HASH,A}^{JHKAS}$ , for the proposed scheme as JHKAS, for example. Let us define the probability of success for  $EXP_{HASH,A}^{JHKAS}$  as  $Success_{HASH,A}^{JHKAS} = |Pr[EXP_{HASH,A}^{JHKAS}] = 1] - 1|$ , where  $Pr(\cdot)$  means the probability of  $EXP_{HASH,A}^{JHKAS}$ . The advantage function for this algorithm then becomes  $Adv_{HASH,A}^{JHKAS}$  (t,  $q_R$ ) =  $max_{Success}$ , where the t is the execution time and  $q_R$  is the number of queries. Discuss the algorithm as shown in Algorithm 1 for the A. If A has the capability to solve the hash function problem given in Definition 6, then he/she can immediately retrieve the identity  $ID_i$ , nearly random binary string  $r_2$  and master secret key x, y of the gateway node GW. In that case, the A will detect the complete connections between the  $U_i$  and the GW; however, the inversion of the input from a given hash value is not possible computationally, i.e.,  $Adv_{HASH,A}^{JHKAS}$  (t)  $\leq \epsilon$ , for all  $\epsilon > 0$ . Thus,  $Adv_{HASH,A}^{JHKAS}$  (t,  $q_R$ )  $\leq \epsilon$ , since  $Adv_{HASH,A}^{JHKAS}$  (t, t). In conclusion, there is no way for A to detect the complete connections between the  $U_i$  and the GW, and the proposed scheme is distinctly secure to an adversary A for retrieving  $ID_i$ ,  $ID_i$ 

Sensors **2017**, *17*, 940

# Algorithm 1 EXP<sub>HASH, A</sub>

```
1. Eavesdrop login request message \langle AID_i, X_i, M_1, M_2, M_3, T_i \rangle during the login phase.
2. Call the Reveal oracle. Let (AID'_i, h(C_i)', X'_i, r'_2, T'_i) \leftarrow \text{Reveal}(M_2)
3. if (AID'_i == AID_i) then
        Accept h(C_i)', X'_i, r'_2, T'_i as the correct of user U_i
       Call the Reveal oracle. Let (C'_i) \leftarrow \text{Reveal}(h(C_i)')
Call the Reveal oracle. Let (C''_i) \leftarrow \text{Reveal}(M_1 \oplus r_2)
5.
6.
7.
        if (C'_i == C''_i) then
8.
           Accept the C_i as the correct of user U_i
9.
           Call the Reveal oracle. Let (ID'_i, x', y') \leftarrow \text{Reveal}(C_i)
10.
           Compute ID_i'' = AID_i \oplus h(r_2)
          if (I\hat{D}_i == I\hat{D}_i'') then
11.
12.
              Accept x', y' as the correct secret key x, y of gateway node GW
13.
             return 1(Success)
14.
          else
15.
             return 0
16.
        else
17.
           return 0
18.
       end if
19. else
20.
       return 0
21. end if
```

# 6.2. Simulation for Formal Security Verification Using the AVISPA Tool

In this subsection, we simulate the proposed scheme using the widely accepted AVISPA for the formal security verification. The main purpose of the simulation is to verify whether the proposed scheme is secure to replay and man-in-the middle attacks. The AVISPA tool consists of four back-ends: (i) On-the-fly Model-Checker (OFMC); (ii) Constraint-Logic-based Attack Searcher; (iii) SAT-based Model Checker; and (iv) Tree Automata based on Automatic Approximations for the Analysis of Security Protocols. In the AVISPA, the protocol is implemented in HLPSL [26], which is based on: the basic roles for representing each participant role and composition roles for representing the scenarios of the basic roles. The basic types available in the HLPSL are [27]:

- agent: The agent denotes a principal name. The intruder always has the special identifier i.
- symmetric\_key: The symmetric\_key is the key for a symmetric-key cryptosystem.
- text: The text values are often used as nonces. They can also be applied for messages.
- nat: The nat is used for denoting the natural numbers in non-message contexts.
- const: This type represents constants.
- hash\_func: The base type hash\_func represents cryptographic one-way hash functions.

The role of the initiator, the user  $U_i$ , is provided in Algorithm 2. The  $U_i$  first receives the start signal and updates its state value from 0 to 1. The state value is maintained by the variable State. In a similar way, the roles of the gateway GW and sensor node  $S_j$  of the proposed scheme are implemented and shown in Algorithm 3 and 4, respectively. The specifications in HLPSL language for the role of session, goal, and environment are specified in Algorithm 5. The simulation result for the formal security verification of the proposed scheme using CL-AtSe is shown in Algorithm 6. It is clear that the proposed scheme is secure to passive and active attacks including the replay and man-in-the middle attacks.

Sensors 2017, 17, 940

# **Algorithm 2** Role specification for user $U_i$ role user (Ui, GW, Si: agent, SKug, SKus: symmetric\_key, H, F: function, SND, RCV: channel (dy)) played\_by Ui def= local State: nat, IDi, Ri, P, Ai, Mi, Ni, Vi: text, AIDi, R1, R2, Xi, Ci, Di, M1, M2, M3, Ti, Rs: text, RM, Yi, Vs, Ts, Gx, Gy, Kus: text init State := 0transition 0. State = $0 \land RCV(start) = | >$ State':= $5 \wedge Ai' := H(Ri)$ ∧ secret(Ri, scrt0, Ui) ∧ secret(IDi, scrt1, {Ui, GW}) $\land$ SND({IDi.Ai'}\_SKug) 5. State = $2 \land RCV(\{Mi'.Ni'.Vi'\}\_SKug.P') = |>$ State':= $8 \wedge R1'$ :=new() $\wedge$ R2':=new() $\wedge$ Ti':=new() $\wedge Xi' := F(R1'.P')$ $\wedge$ Di':=xor(Mi', Ai') $\wedge$ AIDi':=xor(IDi, H(R2')) $\wedge$ M1':=xor(R2', Di') $\wedge$ M2':=H(AIDi'.Di'.Xi'.R2'.Ti') $\wedge$ M3':=xor(Ni', F(R1'.F(Gx'.P))) $\land$ secret({Gx', Gy'}, scrt2, GW) ∧ SND(AIDi', Xi', M1', M2', M3', Ti') $\land$ witness(Ui, Sj, ui\_sj\_r1, R1') 8. State = $8 \land RCV(\{RM'.Yi'.Vs'.Ts'\}\_SKug.P') = |>$ State':= $9 \land Vs'$ :=H(AIDi'.Xi'.Yi'.RM'.Ts')

∧ Kus':= F(R1'.F(Rs'.P))
∧ SKus':=H(AIDi'.Kus'.Ts')
∧ witness(Ui, GW, ui\_gw\_r2, R2')
∧ request(Sj, Ui, sj\_ui\_rs, Rs')

end role

Sensors 2017, 17, 940 17 of 24

# **Algorithm 3** Role specification for gateway GW role gateway (Ui, GW, Sj: agent, SKug, SKgs, Kg: symmetric\_key, H, F: function, SND, RCV: channel (dy)) played\_by GW def= local State: nat, IDi, R1, R2, Ri, P, Ai, Mi, Ni, Vi, Rs: text, AIDi, SIDj, Ks, Xi, Ci, Di, M1, M2, M3, Ti: text, Cg, Tg, Wg, Gx, Gy: text init State := 1 transition 1. State = $1 \land RCV(\{IDi.Ai'\}\_SKug) = |>$ State':= $6 \wedge Ci'$ := H(IDi.Gx'.Gy') $\wedge$ P':=new() $\wedge$ Di':=H(Ci') $\land$ Mi':=xor(Di', H(Ri)) $\wedge$ Ni':=xor(Gx', Ci', Gy') $\wedge$ Vi':=H(IDi.H(Ri)) $\land$ secret({Gx', Gy'}, scrt2, GW) $\land$ SND({Mi'.Ni'.Vi'}\_SKug.P') 3. State = $3 \land RCV(\{SIDj\}\_SKgs) = 1 >$ State':= $6 \land Ks'$ :=H(SIDj.Gy') $\land$ SND({Ks'}\_SKgs.P') 6. State = 6 ∧ RCV(AIDi'.Xi'.M1'.M2'.M3'.Ti') = | > State':= $9 \wedge Tg' = new()$ $\wedge$ Di':=H(xor(M3', F(R1'.F(Gx'.P)), Gx', Gy')) $\land$ R2':=xor(M1', Di') $\wedge$ IDi':=xor(AIDi', H(R2')) $\land$ Kg':=H(H(SIDj'.Gy').Tg') $\land Cg' := \{AIDi'.R2'.Xi'\}_Kg'$ $\land$ Wg':=H(H(SIDj'.Gy').AIDi'.Cg'.Tg')

 $\land$  secret({Gx', Gy'}, scrt2, GW)  $\land$  secret(H(SIDj'.Gy'), scrt4, {GW, Sj})

 $\land$  SND(AIDi'.Cg'.Tg'.Wg')  $\land$  request(Ui, Sj, ui\_sj\_r1, R1')

end role

Sensors 2017, 17, 940 18 of 24

# **Algorithm 4** Role specification for sensor $S_i$

role sensor (Ui, GW, Sj: agent, SKgs, Kg, SKus: symmetric\_key, H, F: function, SND, RCV: channel (dy))

played\_by Sj def=

local State: nat, IDi, Ri, P, Ai, Mi, Ni, Vi, R1, R2: text, AIDi, SIDj, Ks, Xi, Ci, M1, M2, M3, Ti: text, Cg, Tg, Wg: text, RM, Yi, Vs, Ts, Gx, Gy, Rs, Kus:text

init State := 2

# transition

2. State =  $2 \land RCV(start) = | > State' := 4 \land SND(\{SIDj\}\_SKgs.P')$ 

4. State =  $4 \land RCV(\{H(SIDj.Gy')\}\_SKgs.P') = | > State' := 7 \land secret(\{Gx', Gy'\}, scrt2, GW)$ 

7. State =  $7 \land RCV(AIDi'.Xi'.M1'.M2'.M3'.Ti') = | > State' := 10 \land Kg' := H(H(SIDj'.Gy').Tg')$ 

 $\land$  Kus':=F(R1'.F(Rs'.P))

 $\wedge$  Yi':=F(Rs'.P)

 $\land Ts':=new()$ 

∧ SKus':=H(AIDi'.Kus'.Ts')

 $\wedge$  RM':=new()

 $\land Vs':=H(AIDi'.Xi'.Yi'.RM'.Ts')$ 

 $\land \, SND(RM', Yi', Vs', Ts')$ 

∧ secret(H(SIDj.Gy'), scrt4, {GW, Sj})

∧ witness(Sj, Ui, sj\_ui\_rs, Rs')

∧ request(Ui, Sj, ui\_sj\_r1, R1')

# end role

Sensors 2017, 17, 940

```
Algorithm 5 Role specification for session, goal and environment
role session(Ui, GW, Sj: agent, SKug, SKus, SKgs, Kg: symmetric_key, H, F: function) def=
local Z1, Z2, Z3, S1, S2, S3: channel (dy)
composition
user(Ui, GW, Sj, SKug, SKus, H, F, Z1, S1)
∧ gateway(Ui, GW, Sj, SKug, SKgs, Kg, H, F, Z2, S2)
∧ sensor(Ui, GW, Sj, SKgs, Kg, SKus, H, F, Z3, S3)
end role
role environment() def=
const ui, gw, sj: agent,
skug, skgs, skus, kg: symmetric_key,
h, f: function,
aidi, sidj, p, xi: text,
xi, m1, m2, m3, ti: text,
cg, tg, wg: text,
rm, yi, vs, ts: text,
ui_sj_r1, ui_gw_r2, sj_ui_rs: protocol_id,
scrt0, scrt1, scrt2, scrt3, scrt4 : protocol_id
intruder_knowledge = {ui, gw, sj, h, f, p, aidi, sidj, xi, m1, m2, m3, ti, cg, tg, wg, rm, yi, vs,
ts}
composition
session(ui, gw, sj, skug, skgs, kg, skus, h, f)
end role
goal
secrecy_of scrt0
secrecy_of scrt1
secrecy_of scrt2
secrecy_of scrt3
secrecy_of scrt4
authentication_on ui_sj_r1
authentication_on ui_gw_r2
authentication_on sj_ui_rs
end goal
```

Sensors 2017, 17, 940 20 of 24

# Algorithm 6 Role specification for session, goal and environment

**SUMMARY** 

**SAFE** 

**DETAILS** 

BOUNDED\_NUMBER\_OF\_SESSIONS

**PROTOCOL** 

/home/span/span/testsuite/results/testrv.if

**GOAL** 

As Specified

**BACKEND** 

CL-AtSe

STATISTICS

Analysed: 1 states Reachable: 0 states Translations: 0.03 s Computation: 0.00 s

# 6.3. Informal Security Analysis

Table 2 compares the security features provided by the proposed scheme with other related schemes.

**Table 2.** Comparison of security features.

Features	Yoon and Kim [12]	Choi et al. [13]	Park et al. [14]	The Proposed
Provides user anonymity	N/A	×	0	$\circ$
Provides mutual authentication	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$
Provides message confidentiality	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$
Provides perfect forward secrecy	N/A	$\bigcirc$	$\bigcirc$	$\bigcirc$
Resists insider attack	$\bigcirc$	×	$\bigcirc$	$\bigcirc$
Resists impersonation attack	$\bigcirc$	×	×	$\bigcirc$
Resists illegal smart card revocation/reissue attack	×	×	×	$\bigcirc$
Resists biometric recognition error	×	$\bigcirc$	$\bigcirc$	$\circ$
Resists session key exposure by gateway	×	$\bigcirc$	$\bigcirc$	$\bigcirc$
Resists denial of service attack	×	$\bigcirc$	$\bigcirc$	$\bigcirc$
Resists user verification problem	×	$\bigcirc$	$\bigcirc$	$\bigcirc$
Resists stolen verifier attack	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$
Resists replay attack	$\circ$	$\bigcirc$	$\bigcirc$	$\bigcirc$
Security factor	Two-factor	Two-factor	Two-factor	Two-factor

 $<sup>\</sup>bigcirc$ : scheme provides the property;  $\times$ : scheme does not provide the property; N/A: scheme does not consider the property.

# 6.3.1. User Anonymity

Suppose an adversary A intercepts the login request message  $\langle AID_i, X_i, M_1, M_2, M_3, T_i \rangle$  of a legitimate user  $U_i$ . However,  $ID_i$  cannot be derived from  $AID_i$  without the knowledge of a random number  $r_2$ ; furthermore, the  $r_2$  cannot be derived from  $M_1$  without a hash value

Sensors 2017, 17, 940 21 of 24

 $h(C_i) = h(h(ID_i||x||y))$ . The  $U_i$  and GW can only compute  $h(C_i)$ . The proposed scheme therefore provides user anonymity.

# 6.3.2. Mutual Authentication

The proposed scheme not only guarantees secrecy as the other authentication scheme, but also  $U_i$ ,  $S_j$  and GW authenticate each other. GW authenticates  $U_i$  by checking whether  $M_2$  is valid or not because only a legitimate user can compute a valid  $h(C_i)$  using a biometric template. Then,  $S_j$  authenticates GW by checking  $W_g$ , which only GW can compute using the shared long-term key  $h(SID_j||y)$  and the time stamp  $T_g$ . Finally, the  $U_i$  authenticates  $S_j$  by checking the validity of  $V_s$  because only  $U_i$  and  $S_j$  can compute the session key sk.

# 6.3.3. Message Confidentiality

Message confidentiality is an important security aspect that provides secrecy by limiting the adversary's access to the message. Communication messages in the public channel do not affect the disclosure of secret values, such as  $ID_i$ ,  $r_1$ ,  $r_2$ ,  $r_s$  and sk. A cannot compute important information from  $AID_i$ ,  $X_i$ ,  $M_1$ ,  $M_2$ ,  $M_3$ ,  $C_g$ ,  $W_g$ ,  $Y_i$  and  $V_s$ . Furthermore,  $T_i$ ,  $T_g$ ,  $T_s$  and RM are basically public information, so they do not need to be protected.

# 6.3.4. Perfect Forward Secrecy

The perfect forward secrecy means that if one long-term key is compromised, a session key that is derived from these long-term keys will not be compromised in the future [28]. In our scheme, a session key sk between user  $U_i$  and sensor  $S_i$  is calculated as follows:

$$X_i = r_1 \times P$$
  
 $Y_i = r_s \times P$   
 $K_{US} = K_{SU} = r_s \times X_i = r_1 \times Y_i$   
 $sk = h(AID_i||K_{US}||T_s).$ 

Even if the gateway GW's long-term key (x, y) is compromised, adversary  $\mathcal{A}$  cannot retrieve  $r_1$  and  $r_s$  to generate the session keys between  $U_i$  and  $S_j$ . The session key of our proposed scheme is based on a elliptic curve discrete logarithm problem (ECDLP). An adversary  $\mathcal{A}$  cannot obtain  $r_1 \times r_s \times P$  from  $r_1 \times P$  and  $r_s \times P$ . Our scheme therefore provides the perfect forward secrecy.

# 6.3.5. User Impersonation Attack

Suppose  $\mathcal{A}$  owns a smart card to extract information  $\langle V_a, N_a, C_a, h(\cdot), P_a \rangle$  and intercepts the login request message  $\langle AID_i, X_i, M_1, M_2, M_3, T_i \rangle$  of legitimate user  $U_i$ .  $\mathcal{A}$  can then try modifying a login request message. Even if  $\mathcal{A}$  guesses or obtains  $U_i$ 's identity  $ID_i$ , GW verifies whether  $C_i^*$  is equal to  $h(ID_i^*||x||y)$ . The  $\mathcal{A}$  cannot compute  $C_i$  and  $h(C_i)$ , and then fails to impersonate a legitimate user  $U_i$ . The proposed scheme therefore can resist user impersonation attack.

# 6.3.6. Gateway or Sensor Node Impersonation Attack

If  $\mathcal{A}$  wants to masquerade as the gateway node GW or a sensor node  $S_j$ , the hash value  $h(SID_j||y)$  is needed. However, it is computationally difficult to guess  $h(SID_j||y)$  or  $k_g$  correctly. Furthermore, even if  $\mathcal{A}$  obtains the login request message  $\langle AID_i, X_i, M_1, M_2, M_3, T_i \rangle$ ,  $\mathcal{A}$  does not know  $r_2$ . Thus,  $V_s = h(AID_i||r_2||Y_i||sk||RM||T_s)$  cannot be computed. The proposed scheme therefore can resist gateway or sensor node impersonation attack.

Sensors 2017, 17, 940 22 of 24

# 6.3.7. Illegal Smart Card Revocation/Reissue Attack

Even if  $\mathcal{A}$  obtains the identity  $ID_i$  of a legitimate user  $U_i$ ,  $\mathcal{A}$  cannot compute  $h(C_i)$  without the value  $A_i = h(R_i)$ . Furthermore, GW checks the legitimacy of the user on the request identity by computing  $C_i^* = h(ID_i||x||y)$  and  $Z_i^* = ID_i \oplus h(C_i^*) \oplus A_i$ . Therefore, even if  $\mathcal{A}$  sends the revocation/reissue request message  $\langle ID_i, ID_i^*, A_a = h(R_a), Z_a = ID_i \oplus h(C_a) \oplus A_a \rangle$  to GW,  $\mathcal{A}$  fails to revoke  $ID_i$  and reissue the smart card with  $ID_i$ . The proposed scheme therefore can resist an illegal smart card revocation/reissue attack.

# 6.3.8. Session Key Exposure by GW

The gateway GW can intercept communication messages and obtain both  $X_i = r_1 \times P$  and  $Y_i = r_s \times P$ . However, GW cannot derive  $r_1$  and  $r_s$  and therefore cannot compute the common session key sk. This is because our proposed scheme based on the elliptic curve discrete logarithm problem (ECDLP).

# 6.3.9. Denial of Service Attack

In the proposed scheme,  $U_i$ ,  $S_j$  and GW basically check for freshness of timestamp in each authentication step. Each message for verification such as  $M_2$ ,  $W_g$  and  $V_s$  includes the current timestamp T. Furthermore, each entity checks whether the calculated value is equal to the received value. The proposed scheme can resist denial of service attack.

# 6.3.10. User Verification Problem

GW checks for the sameness in the identity  $ID_i$  to verify the status a legitimate user  $U_i$  by computing  $C_i^* = h(ID_i^*||x||y)$ . Furthermore,  $U_i$  can compute constant values including  $A_i = h(R_i)$  as a result of the fuzzy extractor. GW can authenticate a legal user even if the user inputs slightly different biometric information  $B_i^*$ . Our proposed scheme therefore can prevent user verification problems.

# 6.3.11. Stolen Verifier Attack

In the proposed scheme, GW and  $S_j$  do not store any identification, password table or user biometrics. GW stores only the master secret key (x, y), and  $S_j$  stores only  $h(SID_j||y)$ . The proposed scheme therefore can resist stolen verifier attacks.

# 6.3.12. Replay Attack

Even if the adversary A obtains the communication message, and sends them again with the current timestamps  $T_i$ ,  $T_g$ , and  $T_s$ , A cannot compute  $M_2$ ,  $W_g$ ,  $V_s$  using the current timestamps. The proposed scheme therefore can resist replay attacks.

# 7. Performance Analysis

In this section, we compare the computational costs of the proposed scheme with the other related schemes [13,14,29,30]. Table 3 shows a comparison of the computational costs of the proposed scheme with the other related schemes. In the comparisons, XOR operations are not considered because these also can be ignored. Compared to Park et al.'s scheme [14], the proposed scheme performs three further hash operations and two elliptic curve computations. However, we reduce three encryption/decryption operations. Additionally, the proposed scheme provides the revocation and reissue phase, and can resist well-known attacks.

Sensors 2017, 17, 940 23 of 24

Phases		Choi et al. [13]	Park et al. [14]	Nam et al. [29]	Park et al. [30]	The Proposed
Registration	$U_i$ $GWN$ $S_j$	$T_H + T_F$ $3T_H$	$T_H + T_F$ $2T_H + T_E$	$T_H$ $T_H + T_E + T_e$ -	$T_H + T_F$ $5T_H$	$T_H + T_F$ $T_e + 3T_H$
Login and authentication	$U_i$ $GWN$ $S_j$		$7T_H+2T_E$	$3T_H + 3T_e + T_E + T_M$ $T_H + 2T_E + T_e + 3T_M$ $T_H + 2T_e + 2T_M$	$11T_H$	$6T_H + T_F + 3T_e$ $6T_H + T_e + T_E$ $4T_H + T_E + 2T_e$
Revocation and reissue	$U_i$ $GWN$ $S_j$	$T_H + T_F$ $3T_H$	$T_H + T_F$ $2T_H + T_E$	- - -	- - -	$T_H + T_F$ $5T_H$
Total cos	t	$34T_{H} + 3T_{F} + 4T_{E} + 4T_{e}$	$23T_H + 3T_F + 5T_E + 4T_e$	$7T_H + 4T_E + 7T_e + 6T_M$	$31T_H + 2T_F + 4T_e$	$26T_{H} + 3T_{F} + 2T_{E} + 6T_{e}$

**Table 3.** Comparison of computational costs.

 $T_e$ : computational time for elliptic curve computation;  $T_E$ : computational time for encryption/decryption;  $T_F$ : computational time for fuzzy extraction;  $T_H$ : computational time for hash function;  $T_M$ : computational time for massage authentication code.

#### 8. Conclusions

The various authentication schemes for WSNs have been proposed. Recently, Park et al. demonstrated the security vulnerabilities of Choi et al.'s scheme and proposed an enhanced authentication scheme. However, in this paper, we have identified vulnerabilities in Park et al.'s scheme in terms of impersonation and revocation/reissue. To overcome these vulnerabilities, we proposed a new biometric-based authentication scheme with improved security. Security and performance analysis shows that our proposed scheme is more secure and efficient than other related schemes.

Acknowledgments: This work was supported by the Institute for Information & Communications Technology Promotion(IITP) grant funded by the Korea government (Ministry of Science, ICT and Future Planning) (No. R0126-15-1111, the Development of Risk-Based Authentication Access Control Platform and Compliance Technique for Cloud Security).

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

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

# References

- 1. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. Comput. Netw. 2008, 52, 2292–2330.
- 2. Watro, R.; Kong, D.; Cuti, S.; Gardiner, C.; Lynn, C.; Kruus, P. TinyPK: Securing sensor networks with public key technology. In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington, DC, USA, 25 October 2004; pp. 59–64.
- 3. Wong, K.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, Taiwan, 5–7 June 2006; Volume 1, pp. 1–8.
- Tseng, H.; Jan, R.; Yang, W. An improved dynamic user authentication scheme for wireless sensor networks. In Proceedings of the Global Telecommunications Conference, Washington, DC, USA, 26–30 November 2007; pp. 986–990.
- 5. Das, M. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
- 6. He, D.; Gao, Y.; Chan, S.; Chen, C.; Bu, J. An enhanced two-factor user authentication scheme in wireless sensor networks. *Ad Hoc Sens. Wirel. Netw.* **2010**, *10*, 361–371.
- 7. Khan, H.; Alghathbar, K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors* **2010**, *10*, 2450–2459.

Sensors **2017**, *17*, 940 24 of 24

8. Chen, T.; Shih, W. A robust mutual authentication protocol for wireless sensor networks. *ETRI J.* **2010**, 32, 704–712.

- 9. Yuan, J.; Jiang, C.; Jiang, Z. A biometric-based user authentication for wireless sensor networks. *Wuhan Univ. J. Nat. Sci.* **2010**, *15*, 272–276.
- 10. Yoon, E.; Yoo, K. A new biometric-based user authentication scheme without using password for wireless sensor networks. In Proceedings of the 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative, Paris, France, 27–29 June 2011; pp. 279–284.
- 11. He, D.; Zhang, Y.; Chen, J. Robust biometric-based user authentication scheme for wireless sensor networks. *IACR Cryptol. ePrint Arch.* **2012**, 2012, 203–217.
- 12. Yoon, E.; Kim, C. Advanced biometric-based user authentication scheme for wireless sensor networks. *Sens. Lett.* **2013**, *11*, 1836–1843.
- 13. Choi, Y.; Lee, Y.; Won, D. Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. *Int. J. Distrib. Sens. Netw.* **2016**, 2016, 1–16.
- 14. Park, Y., Lee, S.; Kim, C.; Park, Y. Secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1–11.
- 15. Koblitz, N. Elliptic curve cryptosystems. Math. Comput. 1987, 48, 203-209.
- 16. Miller, V. Use of elliptic curves in cryptography. Adv. Cryptol. 1985, 218, 417–426.
- 17. Dolev, D.; Yao, A. On the security of public key protocols. IEEE Trans. Inf. Theory 1983, 29, 198-208.
- 18. Moon, J.; Choi, Y.; Jung, J.; Won, D. An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards. *PLoS ONE* **2015**, *10*, 1–15.
- 19. Choi, Y.; Lee, D.; Kim, J.; Jung, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2014**, *14*, 10081–10106.
- 20. Kocher, P.; Jaffe, J.; Jun, B.; Rohatgi, P. Introduction to differential power analysis. *J. Cryptogr. Eng.* **2011**, *1*, 1–23.
- 21. Das, A. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *Int. J. Commun. Syst.* **2015**, 2015, 1–25.
- 22. Wang, C.; Zhang, X.; Zheng, Z. Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme. *PLoS ONE* **2016**, *11*, 1–25.
- 23. Dodis, Y.; Kanukurthi, B.; Katz, J.; Smith, A. Robust fuzzy extractors and authenticated key agreement from close secrets. *IEEE Trans. Inf. Theory* **2013**, *58*, 6207–6222.
- 24. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 523–540.
- 25. Das, A. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communication. *Netw. Sci.* **2013**, *2*, 12–27.
- 26. von Oheimb, D. The high-level protocol specification language hlpsl developed in the eu project avispa. In Proceedings of the Applied Semantics 2005 Workshop, Frauenchiemsee, Germany, 12–15 September 2005; pp. 1–17.
- 27. Avispa Tool Documentation. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 28 March 2016).
- 28. Zhu, H.; Hao, X. A provable authenticated key agreement protocol with privacy protection using smart card based on chaotic maps. *Nonlinear Dyn.* **2015**, *81*, 311–321.
- 29. Nam, J.; Kim, M.; Park, J.; Lee, Y.; Won, D. A provably-secure ECC-based authentication scheme for wireless sensor networks *Sensors* **2014**, *14*, 21023–21044.
- 30. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).