Sensors **2014**, *14*, 6443-6462; doi:10.3390/s140406443

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Security Analysis and Improvements of Two-Factor Mutual Authentication with Key Agreement in Wireless Sensor Networks

Jiye Kim [1], Donghoon Lee [1], Woongryul Jeon [2], Youngsook Lee [3] and Dongho Won [4,*]

[1] College of Information and Communication Engineering, Sungkyunkwan University, 2066 Seobu-Ro, Jangan-Gu, Suwon-Si, Gyeonggi-Do 440-746, Korea; E-Mails: jykim@security.re.kr (J.K.); dhlee@security.re.kr (D.L.)

[2] Department of Cyber Security and Police, Gwangju University, 277 Hyodeok-Ro, Nam-Gu, Gwangju-Si 503-703, Korea; E-Mail: wrjeon@gwangju.ac.kr

[3] Department of Cyber Investigation Police, Howon University, 64 Howon University 3Gil, Impi-Myeon, Gunsan-Si, Jeonrabuk-Do 573-718, Korea; E-Mail: ysooklee@howon.ac.kr

[4] College of Information and Communication Engineering, Sungkyunkwan University, 2066 Seobu-Ro, Jangan-Gu, Suwon-Si, Gyeonggi-Do 440-746, Korea

***** Author to whom correspondence should be addressed; E-Mail: dhwon@security.re.kr; Tel.: +82-31-290-7213; Fax: +82-31-290-7686.

Received: 22 January 2014; in revised form: 30 March 2014 / Accepted: 31 March 2014 / Published: 9 April 2014

Abstract: User authentication and key management are two important security issues in WSNs (Wireless Sensor Networks). In WSNs, for some applications, the user needs to obtain real-time data directly from sensors and several user authentication schemes have been recently proposed for this case. We found that a two-factor mutual authentication scheme with key agreement in WSNs is vulnerable to gateway node bypassing attacks and user impersonation attacks using secret data stored in sensor nodes or an attacker's own smart card. In this paper, we propose an improved scheme to overcome these security weaknesses by storing secret data in unique ciphertext form in each node. In addition, our proposed scheme should provide not only security, but also efficiency since sensors in a WSN operate with resource constraints such as limited power, computation, and storage space. Therefore, we also analyze the performance of the proposed scheme by comparing its computation and communication costs with those of other schemes.

Keywords: wireless sensor networks (WSNs); mutual authentication; key agreement; two-factor authentication; passwords; smart cards

1. Introduction

A wireless sensor network (WSN) is composed of a number of sensors (tens to thousands) that are deployed to collect data in a target area [1,2]. The number of potential applications for WSNs is increasing in various fields, including environmental monitoring, healthcare, agriculture, manufacturing, military sensing and tracking, and disaster alert [1–5]. The design of a specific WSN is dependent on the given application and the environment under which it operates [1]. In addition, sensors in a WSN operate with resource constraints such as limited power, computation, and storage space [1,3,6–8]. In WSNs, user queries are generally transmitted to the gateway [1,3,8,9]. However, for some applications, the user needs to obtain real-time data directly from sensors [1,3,8,9]. In this case, only legitimate users should be able to access the WSN.

Several schemes for user authentication in WSNs have been proposed recently. Wong *et al.* [10] proposed a user authentication scheme that uses only one-way hash functions for computation efficiency on sensor nodes [10]. However, Das [3] pointed out that Wong *et al.*'s scheme does not prevent many logged-in users with the same login-ID threats and stolen-verifier attacks [3]. Das [3] proposed a two-factor user authentication in WSNs using a smart card and a password instead of maintaining a password/verifier table [3]. Other researchers, however, pointed out that Das' scheme still has security flaws. Chen and Shih [11] insisted that Das' scheme does not provide mutual authentication, and proposed a mutual authentication scheme between the user, the gateway, and the sensor node [11]; He *et al.* [9] said that Das' scheme has security weaknesses against insider attacks and impersonation attacks [9]; and Khan and Alghathbar [4] pointed out that Das' scheme is vulnerable to gateway node bypassing attacks and privileged-insider attacks [4]. In 2012, Vaidya *et al.* [12] pointed out that the schemes proposed by Das [3], Kan and Alghathbar [4] and Chen and Shih [11] are all insecure against stolen smart card attacks and sensor node impersonation attacks with node capture attacks and do not provide key agreement [12]. Therefore, they proposed a novel two-factor mutual authentication and key agreement scheme to prevent these attacks. In addition, they insisted that computational costs for gateway and sensor nodes in their proposed scheme are not so high. However, we found that their proposed scheme still has security flaws.

In this paper, we present that gateway node bypassing attacks and user impersonation attacks are possible using secret data stored in a sensor or an attacker's own smart card in Vaidya *et al.*'s scheme. Additionally, we propose an improved scheme that eliminates such security weaknesses from Vaidya *et al.*'s scheme. We verify that the proposed scheme is secure against possible attacks. We also analyze the performance of the proposed scheme by comparing its computation cost and communication cost with those of other schemes.

The remainder of the paper is organized as follows. Section 2 presents a review of Vaidya *et al.*'s scheme. Section 3 is devoted to analyzing the security of Vaidya *et al.*'s scheme. Section 4 proposes the improved scheme. Section 5 analyzes the security of the proposed scheme against possible attacks.

Section 6 is devoted to analyzing the performance of the proposed scheme and Section 7 concludes this paper.

2. Review of Vaidya *et al.*'s Scheme

There are three communication parties in Vaidya *et al.*'s scheme [12]: a user, a gateway node, and a sensor node. This scheme is composed of four phases: registration phase, login phase, authentication-key agreement phase, and password change phase. We describe each phase in detail in Sections 2.1–2.4, and Table 1 shows the notations used in the remainder of the paper.

Table 1. Notations [12].

Symbol	Description
U_i	i-th user
S_j	j-th sensor node
GW	Gateway node
ID_i	Identity of U_i
pw_i	Password of U_i
SID_j	Identity of S_j
ID_s	Identity of smart card
K	Secret key known to only GW
x_s	Secret value generated by GW and shared between only GW and S_j
$h(\cdot)$	One-way hash function
RN_i	Random nonce of U_i
RN_j	Random nonce of S_j
$\oplus$	XOR operation
$\|$	Concatenation operation
$=?, \leq?$	Verification operation
K_s	Session key
$f(x,k)$	Pseudo-random function of variable x with key k
T_i, T_i'	Current timestamp of U_i
T_G, T_G'	Current timestamp of GW
T_j	Current timestamp of S_j
ΔT	The maximum of transmission delay time permitted
$\xrightarrow{/}$	Secure channel
$\longrightarrow$	Insecure channel

Registration phase begins when the user sends a registration request with his/her identity and a hashed password to the gateway node. Then, the gateway node personalizes a smart card for the user and sends it to him/her as a response to the registration request. In the registration phase, all these communication messages are transmitted in secure channels.

Login phase begins when the user inserts his/her smart card into the terminal and inputs his/her identity and password. After the verification of the user's input value, the smart card computes and

sends the authentication request to the gateway node. When the gateway node receives the authentication request from the user side, the authentication-key agreement phase begins. The gateway node verifies whether the authentication request comes from a legitimate user. If the verification is successful, the gateway node sends the authentication request to a sensor node which can respond to a request or a query from the user. In this phase, three authentication requests are transmitted. The first request is from the gateway node to the sensor node, the second is from the sensor node to the gateway node, and the final is from the gateway node to the user. As stated, when one party receives an authentication request, the party verifies its validity and sends a new authentication request to the other party. In login phase and authentication-key agreement phase, these request messages are transmitted in insecure channels. If all verifications are passed successfully, the user and the sensor node then share the session key for communication. The password change phase begins whenever the user wants to change his/her password. In the password change phase, the user side does not have to communicate with other parties.

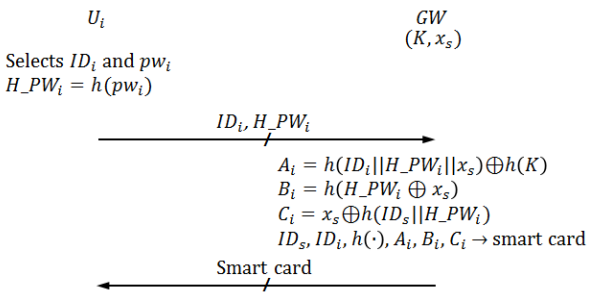
2.1. Registration Phase

We describe the registration phase in this subsection. U_i selects ID_i and pw_i, computes $H\_PW_i = h(pw_i)$ and sends the registration request $\{ID_i, h(pw_i)\}$ to GW. Then, GW personalizes a smart card for U_i and sends it to U_i. Figure 1 shows the registration phase of Vaidya *et al.*'s scheme.

R-1	U_i selects ID_i and pw_i.
R-2	U_i computes $H\_PW_i = h(pw_i)$. U_i sends a registration request $\{ID_i, H\_PW_i\}$ to GW in secure channels (it was not mentioned whether the registration request from U_i to GW is sent by secure channels [12], but we guess that it is sent this way).
R-3	GW computes the following when it receives the registration request from U_i. $A_i = h(ID_i \| \| H\_PW_i \| \| x_s) \oplus h(K)$ $B_i = h(H\_PW_i \oplus x_s)$ $C_i = x_s \oplus h(ID_s \| \| H\_PW_i)$ GW personalizes the smart card with ID_s, ID_i, $h(\cdot)$, A_i, B_i and C_i. GW sends the smart card to U_i in secure channels.

Meanwhile, SID_j and a secret value x_s generated by GW are stored in S_j before it is deployed into a target field.

Figure 1. Registration phase of Vaidya *et al.*'s scheme [12].



2.2. Login Phase

The login phase begins when U_i inserts U_i's smart card into a terminal and inputs ID_i^* and pw_i^*. In this phase, U_i sends the authentication request to GW. Figure 2 illustrates the login phase of Vaidya *et al.*'s scheme.

L-1 U_i inserts U_i's smart card into a terminal and inputs ID_i^* and pw_i^*.

L-2 The smart card computes the following.

$$H\_PW_i^* = h(pw_i^*)$$
$$x_s = C_i \oplus h(ID_s||H\_PW_i^*)$$
$$B_i^* = h(H\_PW_i^* \oplus x_s)$$

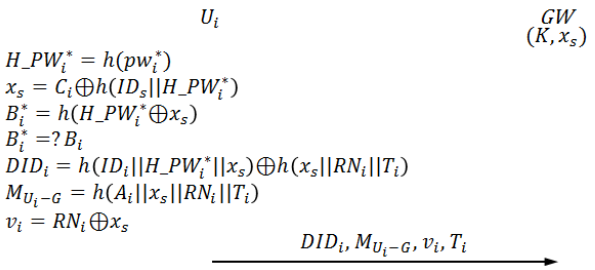
The smart card compares B_i^* with B_i. If $B_i^* = B_i$, then the next step proceeds; otherwise, this phase is aborted.

L-3 The smart card generates a random nonce RN_i and computes the following. T_i is the current timestamp of U_i system.

$$DID_i = h(ID_i||H\_PW_i^*||x_s) \oplus h(x_s||RN_i||T_i)$$
$$M_{U_i-G} = h(A_i||x_s||RN_i||T_i)$$
$$v_i = RN_i \oplus x_s$$

The smart card sends the authentication request $\{DID_i, M_{U_i-G}, v_i, T_i\}$ to GW.

Figure 2. Login phase of Vaidya *et al.*'s scheme [12].



2.3. Authentication-Key Agreement Phase

When GW receives the authentication request from U_i, the authentication-key agreement phase begins. In this phase, U_i, GW, and S_j send and receive authentication requests from one another. Figure 3 depicts the authentication-key agreement phase of Vaidya *et al.*'s scheme. The following describes this process in detail.

A-1 GW checks if $(T_G - T_i) \leq \Delta T$, where T_G is the current timestamp of GW system, and ΔT is the maximum permitted transmission delay time. If $(T_G - T_i) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-2 GW computes the following.

$$RN_i = v_i \oplus x_s$$
$$X^* = DID_i \oplus h(x_s||RN_i||T_i)$$
$$M_{U_i-G}{}^* = h\left(\left(X^* \oplus h(K)\right)||x_s||RN_i||T_i\right)$$

GW compares $M_{U_i-G}{}^*$ with M_{U_i-G}. If $M_{U_i-G}{}^* = M_{U_i-G}$, then the next step proceeds; otherwise, this phase is aborted.

A-3 GW computes $M_{G-S_j} = h(DID_i||SID_j||x_s||T_G)$. T_G is the current timestamp of GW system. S_j is the nearest sensor node that can respond to U_i's request.
GW sends the authentication request $\{DID_i, M_{G-S_j}, T_G\}$ to S_j.

A-4 GW checks if $(T_j - T_G) \leq \Delta T$, where T_j is the current timestamp of S_j system.
If $(T_j - T_G) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-5 S_j computes $M_{G-S_j}{}^* = h(DID_i||SID_j||x_s||T_G)$.
S_j compares $M_{G-S_j}{}^*$ with M_{G-S_j}. If $M_{G-S_j}{}^* = M_{G-S_j}$, then the next step proceeds; otherwise, this phase is aborted.

A-6 S_j generates a random nonce RN_j and computes the following.

$$y_i = RN_j \oplus x_s$$
$$z_i = M_{G-S_j}{}^* \oplus RN_j$$
$$M_{S_j-G} = h(z_i||x_s||T_j)$$

S_j sends the authentication request $\{y_i, M_{S_j-G}, T_j\}$ to GW.

A-7 GW checks if $(T_G{}' - T_j) \leq \Delta T$, where $T_G{}'$ is the current timestamp of GW system.
If $(T_G{}' - T_j) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-8 GW computes the following.

$$RN_j = y_i \oplus x_s$$
$$z_i^* = M_{G-S_j} \oplus RN_j$$
$$M_{S_j-G}{}^* = h(z_i^*||x_s||T_j)$$

GW compares $M_{S_j-G}{}^*$ with M_{S_j-G}. If $M_{S_j-G}{}^* = M_{S_j-G}$, then the next step proceeds; otherwise, this phase is aborted.

A-9 GW computes the following.

$$M_{G-U_i} = h(DID_i||M_{G-S_j}||M_{U_i-G}||x_s||T_G{}')$$
$$w_i = z_i^* \oplus x_s$$

GW sends the authentication request $\{y_i, w_i, M_{G-U_i}, T_G{}'\}$ to U_i.

A-10 U_i checks if $(T_i{}' - T_G{}') \leq \Delta T$, where $T_i{}'$ is the current timestamp of U_i system.
If $(T_i{}' - T_G{}') \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

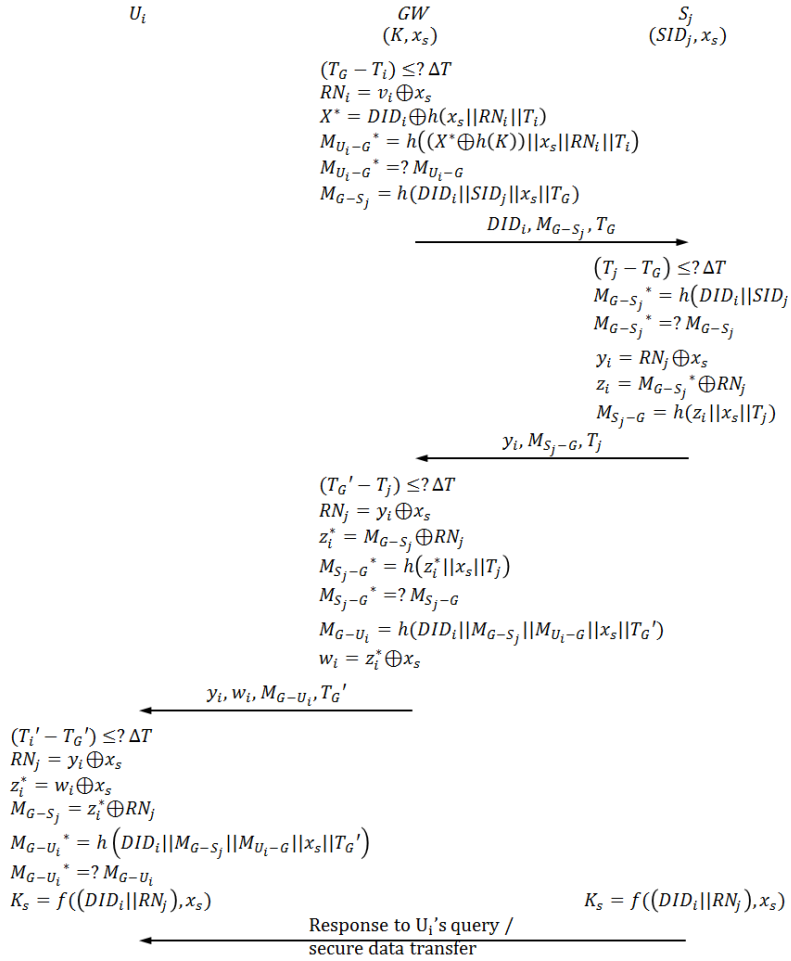
A-11 The smart card computes the following.

$$RN_j = y_i \oplus x_s$$
$$z_i^* = w_i \oplus x_s$$
$$M_{G-S_j} = z_i^* \oplus RN_j$$
$$M_{G-U_i}{}^* = h\left(DID_i||M_{G-S_j}||M_{U_i-G}||x_s||T_G{}'\right)$$

The smart card compares $M_{G-U_i}{}^*$ with M_{G-U_i}. If $M_{G-U_i}{}^* = M_{G-U_i}$, then mutual authentication between U_i and S_j is completed successfully; otherwise, this phase is aborted.

A-12 The smart card computes $K_s = f((DID_i||RN_j), x_s)$ to obtain a session key for communication with S_j. Meanwhile, S_j also computes $K_S = f((DID_i \,||\, RN_j), x_s)$ to share a session key with U_i.

Figure 3. Authentication-key agreement phase of Vaidya *et al.*'s scheme [12].



2.4. Password Change Phase

The password change phase proceeds when U_i changes U_i's existing password to a new one. In the password change phase, U_i does not communicate with GW.

P-1 U_i inserts U_i's smart card into a terminal and inputs ID_i^*, pw_i^*, and pw_{ni}. pw_{ni} is U_i's new password.

P-2 The smart card computes the following.

$$H\_PW_i^* = h(pw_i^*)$$
$$x_s = C_i \oplus h(ID_s||H\_PW_i^*)$$
$$B_i^* = h(H\_PW_i^* \oplus x_s)$$

The smart card compares B_i^* with B_i. If $B_i^* = B_i$, then the next step proceeds; otherwise, this phase is aborted.

P-3 The smart card computes the following.

$$H\_PW_{ni} = h(pw_{ni})$$
$$A_{ni} = A_i \oplus h(ID_i||H\_PW_i^*||x_s) \oplus h(ID_i||H\_PW_{ni}||x_s)$$
$$B_{ni} = h(H\_PW_{ni} \oplus x_s)$$
$$C_{ni} = x_s \oplus h(ID_s||H\_PW_{ni})$$

The smart card replaces the existing values A_i, B_i, and C_i with the new values A_{ni}, B_{ni}, and C_{ni}.

3. Security Analysis of Vaidya *et al.*'s Scheme

In this section, we analyze the security of Vaidya *et al.*'s scheme. We found that gateway node bypassing attacks are possible in Vaidya *et al.*'s scheme if an attacker captures a sensor node and extracts secret values stored in it. Additionally, an attacker can know secret values x_s and $h(K)$ from the attacker's own smart card and use them for user impersonation attacks or gateway node bypassing attacks.

In Sections 3.1–3.3, we describe possible attacks in Vaidya *et al.*'s scheme in detail. We assume that an attacker can eavesdrop on or intercept all messages sent or received between communication parties. We also assume that an attacker can read data stored in a smart card in any manner like in the related works [2,6,13–16]. In addition, we have to note that data stored in sensor nodes are not secure since an attacker can capture sensor nodes that are deployed in unattended environments and can then extract data from them.

3.1. Gateway Node Bypassing Attacks Using Secret Data Stored in a Sensor Node

In Vaidya *et al.*'s scheme, if an attacker extracts the secret data x_s from a sensor node, he/she can impersonate GW and communicate with U_i. These attacks proceed as explained below. U_α denotes an attacker here.

Step 1 U_α extracts x_s and SID_j from a sensor node captured in the WSN.

Step 2 Login phase begins when U_i wants to access to the WSN as in Section 2.2. When U_i sends the authentication request $\{DID_i, M_{U_i-G}, v_i, T_i\}$ to GW, U_α eavesdrops on it.

Step 3 U_α computes the following using x_s, SID_j and $\{DID_i, M_{U_i-G}, v_i, T_i\}$. T_α and T_α' denote the current timestamp of U_α system, and $T_\alpha < T_\alpha'$. U_α generates a random nonce RN_α.

$$y_i = RN_\alpha \oplus x_s$$
$$M_{G-S_j} = h(DID_i||SID_j||x_s||T_\alpha)$$
$$z_i^* = M_{G-S_j} \oplus RN_\alpha$$
$$w_i = z_i^* \oplus x_s$$
$$M_{G-U_i} = h(DID_i||M_{G-S_j}||M_{U_i-G}||x_s||T_\alpha')$$

U_α forges the authentication request sent from GW to U_i in authentication-key agreement phase using $\{y_i, w_i, M_{G-U_i}, T_\alpha'\}$.

Step 4 When U_i receives $\{y_i, w_i, M_{G-U_i}, T_\alpha'\}$ from U_α, U_i checks if $(T_U' - T_\alpha') \leq \Delta T$, where T_U' is the current timestamp of U_i system. If $(T_U' - T_\alpha') \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

Step 5 The smart card computes the following.

$$RN_\alpha = y_i \oplus x_s$$
$$z_i^* = w_i \oplus x_s$$
$$M_{G-S_j} = z_i^* \oplus RN_\alpha$$
$$M_{G-U_i}{}^* = h\left(DID_i||M_{G-S_j}||M_{U_i-G}||x_s||T_\alpha'\right)$$

	The smart card compares M_{G-U_i} with $M_{G-U_i}{}^*$. Since $M_{G-U_i} = M_{G-U_i}{}^*$, U_i regards $\{y_i, w_i, M_{G-U_i}, T_\alpha\}$ as being transmitted from GW. Therefore, U_α can communicate with U_i using the session key $K_s = f((DID_i \| RN_\alpha), x_s)$.

3.2. User Impersonation Attacks Using an Attacker's Own Smart Card

If an attacker U_α registers with GW, U_α receives the smart card personalized with U_α's own identity and password, ID_α and pw_α. U_α can compute x_s and $h(K)$ using ID_α, pw_α, and secret values stored in the smart card.

Step 1	As shown in the Section 2.1, U_α selects ID_α and pw_α.
Step 2	U_α computes $H\_PW_\alpha = h(pw_\alpha)$. U_α sends the registration request $\{ID_\alpha, h(pw_\alpha)\}$ to GW.
Step 3	GW computes the following when it receives the registration request from U_α. $A_\alpha = h(ID_\alpha \| H\_PW_\alpha \| x_s) \oplus h(K)$ $B_\alpha = h(H\_PW_\alpha \oplus x_s)$ $C_\alpha = x_s \oplus h(ID_s \| H\_PW_\alpha)$ GW personalizes the smart card with ID_s, ID_α, $h(\cdot)$, A_α, B_α and C_α. GW sends the smart card to U_α.
Step 4	U_α reads ID_s, ID_α, A_α, B_α, and C_α from the smart card. U_α can know x_s and $h(K)$ by computing the following. $x_s = C_\alpha \oplus h(ID_s \| H\_PW_\alpha)$ $h(K) = A_\alpha \oplus h(ID_\alpha \| H\_PW_\alpha \| x_s)$

U_α can impersonate a legitimate user who has registered with GW using x_s and $h(K)$. In addition, U_α can also log in with any temporary identity that does not actually exist.

3.2.1. Logging in with Any Temporary Identity

We describe the process where U_α logs in with any temporary identity that does not actually exist using x_s and $h(K)$.

Step 1	U_α selects any temporary identity and password ID_β and pw_β. U_α computes the authentication request as follows. T_α denotes the current timestamp of U_α system, and RN_α is a random nonce generated by U_α. $H\_PW_\beta^* = h(pw_\beta)$ $A_\beta = h(ID_\beta \| H\_PW_\beta^* \| x_s) \oplus h(K)$ $DID_\beta = h(ID_\beta \| H\_PW_\beta^* \| x_s) \oplus h(x_s \| RN_\alpha \| T_\alpha)$ $M_{U_\beta - G} = h(A_\beta \| x_s \| RN_\alpha \| T_\alpha)$ $v_\beta = RN_\alpha \oplus x_s$ U_α sends the authentication request $\{DID_\beta, M_{U_\beta - G}, v_\beta, T_\alpha\}$ to GW.
Step 2	When GW receives the authentication request, GW checks if $(T_G - T_\alpha) \leq \Delta T$, where T_G is the current timestamp of GW system. If $(T_G - T_\alpha) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

Step 3	GW computes the following. $RN_\alpha = v_\beta \oplus x_s$ $X^* = DID_\beta \oplus h(x_s \| \| RN_\alpha \| \| T_\alpha)$ $M_{U_\beta - G}{}^* = h\left(\left(X^* \oplus h(K)\right) \| \| x_s \| \| RN_\alpha \| \| T_\alpha\right)$ GW compares $M_{U_\beta - G}$ with $M_{U_\beta - G}{}^*$. GW regards $\{DID_\beta, M_{U_\beta - G}, v_\beta, T_\alpha\}$ as being sent from a legitimate user because $M_{U_\beta - G} = M_{U_\beta - G}{}^*$.

3.2.2. Logging in with the Identity of a Legitimate User

We describe when U_α impersonates a legitimate user U_i who has registered with GW using x_s and $h(K)$.

Step 1	In the previous session, when U_i sends the authentication request $\{DID_i, M_{U_i - G}, v_i, T_i\}$ to GW as shown in Section 2.2, U_α eavesdrops on it.
Step 2	U_α computes the following. RN_α is a random nonce generated by U_α. T_α is the current timestamp of U_α system. x_s and $h(K)$ are already known to U_α, as mentioned above. $RN_i = v_i \oplus x_s$ $h(ID_i \| \| H\_PW_i^* \| \| x_s) = DID_i \oplus h(x_s \| \| RN_i \| \| T_i)$ $DID_i = h(ID_i \| \| H\_PW_i^* \| \| x_s) \oplus h(x_s \| \| RN_\alpha \| \| T_\alpha)$ $A_i = h(ID_i \| \| H\_PW_i^* \| \| x_s) \oplus h(K)$ $M_{U_i - G} = h(A_i \| \| x_s \| \| RN_\alpha \| \| T_\alpha)$ $v_i = RN_\alpha \oplus x_s$ U_α sends the authentication request $\{DID_i, M_{U_i - G}, v_i, T_\alpha\}$ to GW.
Step 3	When GW receives $\{DID_i, M_{U_i - G}, v_i, T_\alpha\}$, GW checks if $(T_G - T_\alpha) \leq \Delta T$, where T_G is the current timestamp of GW system. If $(T_G - T_\alpha) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.
Step 4	GW computes the following. $RN_\alpha = v_i \oplus x_s$ $X^* = DID_i \oplus h(x_s \| \| RN_\alpha \| \| T_\alpha)$ $M_{U_i - G}{}^* = h\left(\left(X^* \oplus h(K)\right) \| \| x_s \| \| RN_\alpha \| \| T_\alpha\right)$ GW compares $M_{U_i - G}$ with $M_{U_i - G}{}^*$. GW regards $\{DID_i, M_{U_i - G}, v_i, T_\alpha\}$ as being sent from a legitimate user because $M_{U_i - G} = M_{U_i - G}{}^*$.

3.3. Gateway Node Bypassing Attacks Using an Attacker's Own Smart Card

As discussed in Section 3.2, if an attacker U_α obtains x_s and $h(K)$ using data stored in his/her own smart card, he/she can impersonate GW. The following shows the attack process in detail. U_α denotes an attacker here.

Step 1	Login phase begins when U_i wants to access the WSN as described in Section 2.2. When U_i sends the authentication request $\{DID_i, M_{U_i - G}, v_i, T_i\}$ to GW, U_α eavesdrops on the transmission.

Step 2 U_α computes the following using x_s and $\{DID_i, M_{U_i-G}, v_i, T_i\}$. T_α and $T_\alpha{}'$ denote the current timestamp of T_α system, and $T_\alpha < T_\alpha{}'$. U_α generates a random nonce RN_α. SID_α is created by U_α.

$$y_i = RN_\alpha \oplus x_s$$
$$M_{G-S_j} = h(DID_i||SID_\alpha||x_s||T_\alpha)$$
$$z_i^* = M_{G-S_j} \oplus RN_\alpha$$
$$w_i = z_i^* \oplus x_s$$
$$M_{G-U_i} = h(DID_i||M_{G-S_j}||M_{U_i-G}||x_s||T_\alpha{}')$$

U_α forges the authentication request sent from GW to U_i in authentication-key agreement phase using $\{y_i, w_i, M_{G-U_i}, T_\alpha{}'\}$.

Step 3 When U_i receives $\{y_i, w_i, M_{G-U_i}, T_\alpha{}'\}$ from U_α, U_i checks if $(T_U{}' - T_\alpha{}') \leq \Delta T$, where $T_U{}'$ is the current timestamp of U_i system. If $(T_U{}' - T_\alpha{}') \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

Step 4 The smart card computes the following.

$$RN_\alpha = y_i \oplus x_s$$
$$z_i^* = w_i \oplus x_s$$
$$M_{G-S_j} = z_i^* \oplus RN_\alpha$$
$$M_{G-U_i}{}^* = h\left(DID_i||M_{G-S_j}||M_{U_i-G}||x_s||T_\alpha{}'\right)$$

The smart card compares M_{G-U_i} with $M_{G-U_i}{}^*$. Since $M_{G-U_i} = M_{G-U_i}{}^*$, U_i regards $\{y_i, w_i, M_{G-U_i}, T_\alpha{}'\}$ as being transmitted from GW. Therefore, U_α can communicate with U_i using the session key $K_s = f((DID_i||RN_\alpha), x_s)$.

4. The Proposed Scheme

In this section, we propose an improved scheme that can overcome the security weaknesses presented in Section 3. The reason why Vaidya *et al.*'s scheme is vulnerable to sensor node capture attacks is that x_s is stored in plaintext form in S_j though it is a secret value. To make matters worse, x_s is shared between all sensor nodes in the WSN. Also, in Vaidya *et al.*'s scheme, an attacker can compute and use x_s and $h(K)$ for attacks because they are stored in all users' smart cards. Therefore, the main ideas of our proposed scheme are as follows:

- When GW personalizes a smart card for U_i in the registration phase, GW uses $Xs_i = h(H\_ID_i||x_s)$ and $h(H\_ID_i||K)$ instead of x_s and $h(K)$ to prevent an attacker from computing x_s or $h(K)$. Since Xs_i and $h(H\_ID_i||K)$ are unique for each user, an attacker cannot reuse them to impersonate a legitimate user.
- In the proposed scheme, $Xs_j^* = h(SID_j||x_s)$ instead of x_s is stored in S_j to prevent an attacker from extracting x_s from S_j. Since Xs_j^* is unique for each sensor node, we can attenuate the effects of sensor node capture attacks as much as possible.

We describe each phase in detail in Sections 4.1 through 4.4. Before describing the proposed scheme in detail, we present the security requirements for the proposed scheme.

- The proposed scheme has to be secure against possible attacks such as replay, password guessing, user impersonation, gateway node bypassing and parallel session attacks.

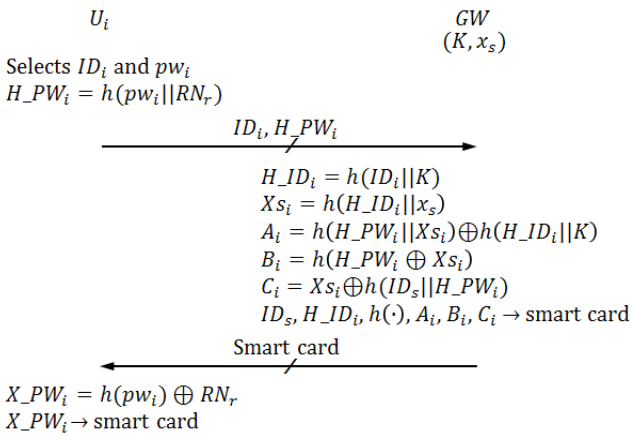
- The proposed scheme has to minimize the damage caused by sensor node capture attacks. The authentication scheme cannot be a perfect solution that blocks sensor node capture attacks completely. Nevertheless, the proposed scheme should attenuate the effects of sensor node capture attacks as much as possible.
- We assume an attacker can obtain all data from a smart card. Therefore, our proposed scheme has to be devised considering stolen smart card attacks, lost smart card problems, and attacks that use an attacker's own smart card, as shown in Section 3.
- The proposed scheme must be secure against privileged-insider attacks or stolen-verifier attacks.
- The proposed scheme has to provide methods for mutual authentication, key agreement between U_i and S_j, and password change.

4.1. Registration Phase

In the registration phase, U_i selects ID_i and pw_i. U_i computes and sends the registration request $\{ID_i, h(pw_i||RN_r)\}$ to the gateway node, where RN_r is a random nonce. Then, GW personalizes a smart card for U_i. Figure 4 illustrates the registration phase of the proposed scheme. Meanwhile, SID_j and Xs_j^* are stored in S_j, where $Xs_j^* = h(SID_j||x_s)$ before S_j is deployed into a target field.

R-1	U_i selects ID_i and pw_i.
R-2	U_i generates a random nonce RN_r and computes $H\_PW_i = h(pw_i\|\|RN_r)$. U_i sends the registration request $\{ID_i, H\_PW_i\}$ to GW in secure channels.
R-3	GW computes the following when it receives a registration request from U_i. $H\_ID_i = h(ID_i\|\|K)$ $Xs_i = h(H\_ID_i\|\|x_s)$ $A_i = h(H\_PW_i\|\|Xs_i) \oplus h(H\_ID_i\|\|K)$ $B_i = h(H\_PW_i \oplus Xs_i)$ $C_i = Xs_i \oplus h(ID_s\|\|H\_PW_i)$ GW personalizes the smart card with ID_s, H_ID_i, $h(\cdot)$, A_i, B_i and C_i. GW sends the smart card to U_i in secure channels.
R-4	U_i computes $X\_PW_i = h(pw_i) \oplus RN_r$ and adds X_PW_i to the smart card.

Figure 4. Registration phase of the proposed scheme.



4.2. Login Phase

The login phase begins when U_i inserts U_i's smart card into a terminal and inputs ID_i^* and pw_i^*. In this phase, U_i sends the authentication request to GW. Figure 5 depicts the login phase of the proposed scheme.

L-1 U_i inserts U_i's smart card into a terminal and inputs ID_i^* and pw_i^*.

L-2 The smart card computes the following.

$$RN_r^* = h(pw_i^*) \oplus X\_PW_i$$
$$H\_PW_i^* = h(pw_i^* || RN_r^*)$$
$$Xs_i^* = C_i \oplus h(ID_s || H\_PW_i^*)$$
$$B_i^* = h(H\_PW_i^* \oplus Xs_i^*)$$

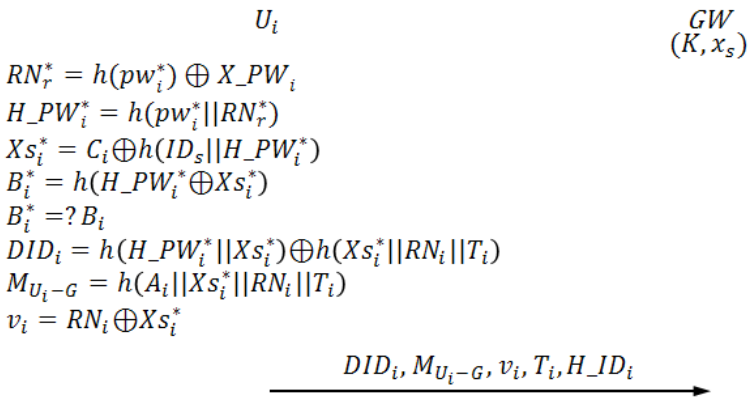
The smart card compares B_i^* with B_i. If $B_i^* = B_i$, then the next step proceeds; otherwise, this phase is aborted.

L-3 The smart card generates a random nonce RN_i and computes the following. T_i is the current timestamp of U_i system.

$$DID_i = h(H\_PW_i^* || Xs_i^*) \oplus h(Xs_i^* || RN_i || T_i)$$
$$M_{U_i-G} = h(A_i || Xs_i^* || RN_i || T_i)$$
$$v_i = RN_i \oplus Xs_i^*$$

The smart card sends the authentication request $\{DID_i, M_{U_i-G}, v_i, T_i, H\_ID_i\}$ to GW.

Figure 5. Login phase of the proposed scheme.



4.3. Authentication-Key Agreement Phase

When GW receives an authentication request from U_i, the authentication-key agreement phase begins. In this phase, U_i, GW, and S_j send and receive authentication requests from one another. Figure 6 shows the authentication-key agreement phase of the proposed scheme. The following describes this process in detail.

A-1 GW checks if $(T_G - T_i) \leq \Delta T$, where T_G is the current timestamp of GW system.
If $(T_G - T_i) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-2 GW computes the following.

$$Xs_i = h(H\_ID_i||x_s)$$
$$RN_i = v_i \oplus Xs_i$$
$$X^* = DID_i \oplus h(Xs_i||RN_i||T_i)$$
$$M_{U_i-G}{}^* = h\left(\left(X^* \oplus h(H\_ID_i||K)\right)||Xs_i||RN_i||T_i\right)$$

GW compares $M_{U_i-G}{}^*$ with M_{U_i-G}. If $M_{U_i-G}{}^* = M_{U_i-G}$, then the next step proceeds; otherwise, this phase is aborted.

A-3 GW computes the following. T_G is the current timestamp of GW system. S_j is the nearest sensor node that can respond to U_i's request.

$$Xs_j = h(SID_j||x_s)$$
$$M_{G-S_j} = h(DID_i||SID_j||Xs_j||T_G)$$

GW sends the authentication request $\{DID_i, M_{G-S_j}, T_G\}$ to S_j.

A-4 GW checks if $\left(T_j - T_G\right) \leq \Delta T$, where T_j is the current timestamp of S_j.
If $\left(T_j - T_G\right) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-5 S_j computes $M_{G-S_j}{}^* = h\left(DID_i||SID_j||Xs_j^*||T_G\right)$.
S_j compares $M_{G-S_j}{}^*$ with M_{G-S_j}. If $M_{G-S_j}{}^* = M_{G-S_j}$, then the next step proceeds; otherwise, this phase is aborted.

A-6 S_j generates a random nonce RN_j and computes the following.

$$y_j = RN_j \oplus Xs_j^*$$
$$z_i = M_{G-S_j}{}^* \oplus RN_j$$
$$M_{S_j-G} = h(z_i||Xs_j^*||T_j)$$

S_j sends the authentication request $\{y_i, M_{S_j-G}, T_j\}$ to GW.

A-7 GW checks if $(T_G{}' - T_j) \leq \Delta T$, where $T_G{}'$ is the current timestamp of GW.
If $(T_G{}' - T_j) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-8 GW computes the following.

$$RN_j = y_j \oplus Xs_j$$
$$z_i^* = M_{G-S_j} \oplus RN_j$$
$$M_{S_j-G}{}^* = h\left(z_i^*||Xs_j||T_j\right)$$

GW compares $M_{S_j-G}{}^*$ with M_{S_j-G}. If $M_{S_j-G}{}^* = M_{S_j-G}$, then the next step proceeds; otherwise, this phase is aborted.

A-9 GW computes the following:

$$M_{G-U_i} = h(DID_i||M_{G-S_j}||M_{U_i-G}||Xs_i||T_G{}')$$
$$w_i = z_i^* \oplus Xs_i$$
$$y_i = RN_j \oplus Xs_i$$
$$q_j = Xs_j \oplus RN_j$$

GW sends the authentication request $\{y_i, w_i, M_{G-U_i}, q_j, T_G{}'\}$ to U_i.

A-10 U_i checks if $(T_i{}' - T_G{}') \leq \Delta T$, where $T_i{}'$ is the current timestamp of U_i. If $(T_i{}' - T_G{}') \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

A-11 The smart card computes the following:

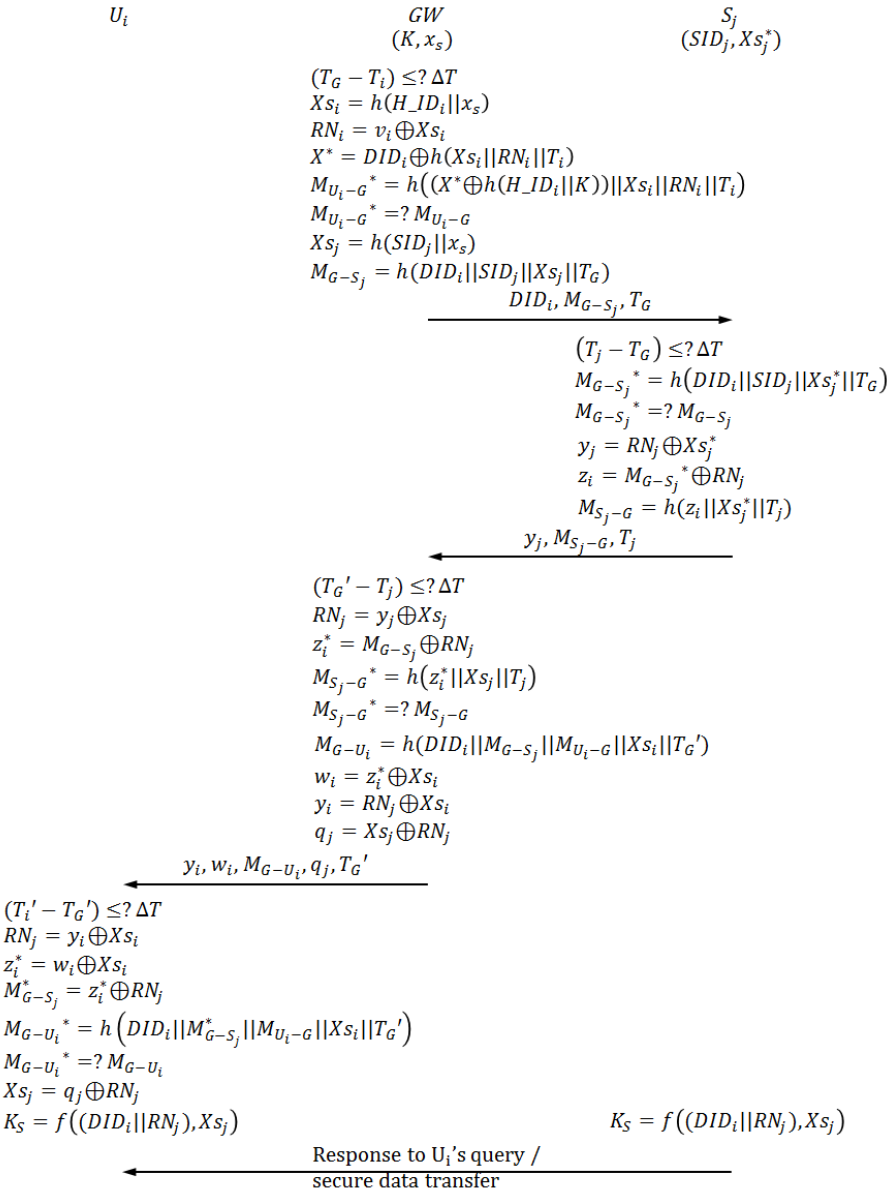
$$RN_j = y_i \oplus Xs_i$$
$$z_i^* = w_i \oplus Xs_i$$
$$M_{G-S_j}^* = z_i^* \oplus RN_j$$
$$M_{G-U_i}{}^* = h\left(DID_i||M_{G-S_j}^*||M_{U_i-G}||Xs_i||T_G{}'\right)$$

The smart card compares $M_{G-U_i}{}^*$ with M_{G-U_i}. If $M_{G-U_i}{}^* = M_{G-U_i}$, then mutual authentication between U_i and SN_j is completed successfully; otherwise, this phase is aborted.

A-12 The smart card computes the following to get a session key for communication with S_j. Meanwhile, S_j also computes $K_S = f\left((DID_i||RN_j), Xs_j\right)$ to share a session key with U_i.

$$Xs_j = q_j \oplus RN_j$$
$$K_S = f\left((DID_i||RN_j), Xs_j\right)$$

Figure 6. Authentication-key agreement phase of the proposed scheme.



4.4. Password Change Phase

The password change phase proceeds when U_i changes U_i's existing password to a new one. In the password change phase, U_i does not have to communicate with GW.

P-1 U_i inserts its smart card into a terminal and inputs ID_i^*, pw_i^* and pw_{ni}. pw_{ni} is U_i's new password.

P-2 The smart card computes the following.

$$RN_r^* = h(pw_i^*) \oplus X\_PW_i$$
$$H\_PW_i^* = h(pw_i^* || RN_r^*)$$
$$Xs_i^* = C_i \oplus h(ID_s || H\_PW_i^*)$$
$$B_i^* = h(H\_PW_i^* \oplus Xs_i^*)$$

The smart card compares B_i^* with B_i. If $B_i^* = B_i$, then the next step proceeds; otherwise, this phase is aborted.

P-3 The smart card computes the following.

$$H\_PW_{ni} = h(pw_{ni} || RN_r^*)$$
$$A_{ni} = A_i \oplus h(H\_PW_i^* || Xs_i^*) \oplus h(H\_PW_{ni} || Xs_i^*)$$
$$B_{ni} = h(H\_PW_{ni} \oplus Xs_i^*)$$
$$C_{ni} = Xs_i^* \oplus h(ID_s || H\_PW_{ni})$$

The smart card replaces the existing values A_i, B_i and C_i with the new values A_{ni}, B_{ni} and C_{ni}.

5. Security Analysis of the Proposed Scheme

This section is devoted to the security analysis of our proposed scheme. We discuss the security of our proposed scheme in terms of the security requirements presented in Section 4. Table 2 shows a security comparison of the proposed scheme.

Table 2. Security comparison of the proposed scheme.

Security Features	Das' Scheme [3]	Khan and Alghathbar's Scheme [4]	Vaidya *et al.*'s Scheme[12]	The Proposed Scheme
Replay attacks	Yes	Yes	Yes	Yes
User impersonation attacks	No	No	No	Yes
Gateway node bypassing attacks	No	No	No	Yes
Parallel session attacks	No	No	Yes	Yes
Password guessing attacks	No	No	Yes	Yes
Sensor node capture attacks	No	No	No	Yes
Stolen smart card attacks	No	No	Yes	Yes
Lost smart card problems	No	No	Yes	Yes
Privileged-insider attacks	No	Yes	Yes	Yes
Stolen-verifier attacks	Yes	Yes	Yes	Yes
Mutual authentication	No	No	Yes	Yes
Key agreement	No	No	Yes	Yes
Password change phase	No	Yes	Yes	Yes

(Yes: The scheme resists the attacks or provides the functionality; No: The scheme does not resist the attacks or provide the functionality).

- **Replay attacks**: The proposed scheme resists replay attacks because all authentication requests include current timestamps, such as T_i of $\{DID_i, M_{U_i-G}, v_i, T_i, H\_ID_i\}$.
- **User impersonation attacks and gateway node bypassing attacks**: In the proposed scheme, an attacker cannot create valid authentication requests $\{DID_i, M_{U_i-G}, v_i, T_i, H\_ID_i\}$ and $\{y_i, w_i, M_{G-U_i}, q_j, T_G'\}$ because he/she cannot compute the secret data x_s and $h(K)$. Therefore, user impersonation attacks and gateway node bypassing attacks are impossible.
- **Parallel session attacks**: The proposed scheme is secure against parallel session attacks because all authentication requests include random nonces such as DID_i, M_{U_i-G} and v_i of $\{DID_i, M_{U_i-G}, v_i, T_i, H\_ID_i\}$.
- **Password guessing attacks**: pw_i cannot be guessed by an attacker because it is transmitted as the results which are concatenated with some secret values and one-way hashed. Even a privileged-insider cannot guess U_i's password from the registration request $\{ID_i, H\_PW_i\}$ because RN_r in $H\_PW_i = h(pw_i||RN_r)$ is a unknown value to him/her.
- **Sensor node capture attacks**: Though an attacker captures a sensor node and obtains secret data SID_j and Xs_j^* from it, the attacker cannot impersonate U_i, GW, or other sensor nodes. Since Xs_j^* is the unique secret data only for S_j, an attacker cannot compute Xs_i for U_i or x_s for GW. In addition, he/she cannot compute the secret data of other sensor nodes except S_j.
- **Stolen smart card attacks and lost smart card problems**: Though an attacker extracts ID_s, H_ID_i, $h(\cdot)$, A_i, B_i, C_i, and X_PW_i from U_i's smart card, he/she cannot compute any secret data $h(K)$ or x_s for attacks. Therefore, the proposed scheme is secure against stolen smart card attacks or lost smart card problems. In addition, though an attacker extracts ID_s, H_ID_α, $h(\cdot)$, A_α, B_α, C_α, and X_PW_α from his/her own smart card, he/she cannot compute any secret data $h(K)$ or x_s for attacks. Therefore, the proposed scheme prevents attacks using an attacker's own smart card.
- **Privileged-insider attacks**: The proposed scheme resists privileged-insider attacks because pw_i is transmitted as a digest of some other secret components.
- **Stolen-verifier attacks**: The proposed scheme is secure against stolen-verifier attacks, since GW does not maintain a verifier table.
- **Mutual authentication, key agreement, and password change phase**: The proposed scheme provides mutual authentication, key agreement between U_i and S_j, and password change phase.

6. Performance Analysis of the Proposed Scheme

Table 3 shows the computation cost comparison of the proposed scheme. Das' scheme [3], Khan and Alghathbar's scheme [4], Vaidya *et al.*'s scheme [12], and the proposed scheme use only hash and XOR operations. We compare these schemes in terms of the number of hash and XOR operations. The proposed scheme needs seven hash operations more than Vaidya's *et al.*'s [12]. Nevertheless, one of our main concerns is the computation cost of a sensor node rather than that of the entire scheme, because sensor nodes are resource-constrained. The computation cost of S_j in the proposed scheme is the same as that of Vaidya *et al.*'s [12]. This means that the computation cost increase of the entire scheme is negligible considering the enhanced security. Meanwhile, with respect to communication

cost, the number of messages transmitted in the proposed scheme is four, which is the same as that of Vaidya *et al.*'s scheme.

7. Conclusions

We have proposed an improved mutual authentication and key agreement scheme to overcome the security weaknesses of Vaidya *et al.*'s scheme. The proposed scheme resists user impersonation attacks and gateway node bypassing attacks using secret data stored in an attacker's own smart card or a sensor. In addition, the proposed scheme prevents possible attacks such as replay attacks, parallel session attacks, password guessing attacks, sensor node capture attacks, stolen smart card attacks, lost smart card problems, privileged-insider attacks, and stolen-verifier attacks. The proposed scheme is also efficient in terms of computation and communication cost considering the limited resources of sensors.

Table 3. Computation cost comparison of the proposed scheme.

Phases		Das' Scheme [3]	Khan and Alghathbar's Scheme[4]	Vaidya *et al.*'s Scheme [12]	The Proposed Scheme
Registration phase	U_i	0	1H	1H	2H + 1X
	GW	3H + 1X	2H + 1X	4H + 3X	6H + 3X
	S_j	0	0	0	0
Login phase	U_i	3H + 1X	3H + 1X	6H + 4X	7H + 5X
	GW	0	0	0	0
	S_j	0	0	0	0
Authentication and key agreement phase	U_i	0	0	1H + 3X	1H + 4X
	GW	4H + 2X	5H + 2X	6H + 6X	8H + 8X
	S_j	1H	2H	2H + 2X	2H + 2X
Password change phase	U_i	-	3H + 2X	8H + 6X	9H + 7X
	GW	-	0	0	0
	S_j	-	0	0	0
Total		11H + 4X	16H + 6X	28H + 24X	35H + 30X

(H: The number of hash operations; X: The number of XOR operations).

Acknowledgments

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1004) supervised by the NIPA(National IT Industry Promotion Agency).

Author Contributions

Jiye Kim, Donghoon Lee, Woongryul Jeon, Youngsook Lee, and Dongho Won have contributed to security analysis, design of the proposed scheme, and manuscript preparation.

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. Yoon, E.J.; Yoo, K.Y. 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), Banff, AB, Canada, 18–20 August 2011.
3. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
4. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of "two-factor user authentication in wireless sensor networks". *Sensors* **2010**, *10*, 2450–2459.
5. Nyang, D.H.; Lee, M.K. Improvement of Das's Two-Factor Authentication Protocol in Wireless Sensor Networks. IACR Cryptology ePrint Archive. Available online: http://eprint.iacr.org/2009/631.pdf (accessed on 20 January 2014).
6. Li, C.T.; Weng, C.Y.; Lee, C.C. An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. *Sensors* **2013**, *13*, 9589–9603.
7. Yoo, S.G.; Lee, H.; Kim, J. A Performance and Usability Aware Secure Two-Factor User Authentication Scheme for Wireless Sensor Networks. Available online: http://www.hindawi.com/journals/ijdsn/2013/543950/ (accessed on 20 January 2014).
8. Tseng, H.R.; Jan, R.H.; Yang, W. An Improved dynamic user authentication scheme for wireless sensor networks. In Proceedings of the Global Telecommunications Conference (GLOBECOM'07), Washington, DC, USA, 26–30 November 2007.
9. He, D.; Gao, Y.; Chan, S.; Chen, C.; Bu, J. An enhanced two-factor user authentication scheme in wireless sensor networks. *Ad Hoc Sens. Wirel. Netw.* **2010**, *10*, 361–371.
10. Wong, K.H.M.; 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.
11. Chen, T.H.; Shih, W.K. A robust mutual authentication protocol for wireless sensor networks. *Electron. Telecommun. Res. Inst.* **2010**, *32*, 704–712.
12. Vaidya, B.; Makrakis, D.; Mouftah, H. Two-Factor Mutual Authentication with Key Agreement in Wireless Sensor Networks. Available online: http://onlinelibrary.wiley.com/doi/10.1002/sec.517/full (accessed on 20 January 2014).
13. Das, A.K.; Sharma, P.; Chatterjee, S.; Sing, J.K. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1646–1656.
14. Xu, J.; Zhu, W.T.; Feng, D.G. An improved smart card based password authentication scheme with provable security. *Comput. Stand. Interfaces* **2009**, *31*, 723–728.
15. Turkanovic, M.; Holbl, M. An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *Electron. Electr. Eng.* **2013**, *19*, 109–116.

16. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323.