Cryptanalysis of Kim Jiye et al.’s Two-Factor Mutual Authentication with Key Agreement in WSNs

Jiping Li¹, Yaoming Ding¹, Zenggang Xiong¹, Shouyin Liu², and Honglai Li¹
¹ School of Computer and Information Science, Hubei Engineering University, Xiaogan 430079, China
² College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China

Email: oucljp2012@yahoo.com; {xgdym2015, xglhl2015}@aliyun.com; jkxxzg2003@163.com; syliu@phy.ccnu.edu.cn

Abstract — User authentication and key management play an important role in the security of WSNs (Wireless Sensor Networks). In WSNs, for some applications, the user needs to obtain real-time data directly from dedicated sensors. For this case, several user authentication schemes have been proposed in recent years. Among these schemes, Kim Jiye et al.’s scheme is very novel. However, in the current work, we find that Kim Jiye et al.’s scheme is still vulnerable to some attacks such as offline password guessing attack, user impersonation attack using his/her own smart card, sensor node impersonation attack and gateway node bypassing attack. In this paper, we give detailed cryptanalysis of Kim Jiye et al.’s two-factor mutual authentication with key agreement in WSNs.

Index Terms — WSN, mutual authentication, key-agreement, smart card, password

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are composed of a number of sensors (tens or thousands) that are deployed to collect data in a target area [1], [2]. WSN has been recently applied in various fields, including environmental monitoring, healthcare, agriculture, manufacturing, military sensing and tracking, and disaster alert [1]-[5]. The design of specific WSNs is dependent on the given application and the environment under which it operates [1]. In addition, different from traditional wireless networks, sensors in WSNs operate with resource constraints such as limited power, low computing and communication ability and small storage capability [1]-[3], [6]-[8]. In WSNs, user queries are generally transmitted to and received from the GW (gateway node). However, in some special applications, user needs to obtain real-time data directly from sensors [1], [3], [8], [9].

In recent years, several two-factor user authentication schemes in WSNs have been proposed. In 2006, Wong et al. [10] proposed a dynamic user authentication scheme using only one-way hash functions for computation efficiency on sensor nodes. However, Das [3] in 2009 pointed out that Wong et al.’s scheme cannot prevent some attacks such as many logged-in users with the same login-id threats and stolen-verifier attacks. Das [3] proposed a two-factor user authentication in WSNs using a smart card and a password instead of maintaining a password/verifier table. In the subsequent years, several researchers, however, pointed out that Das’s scheme still has certain security flaws. In 2010, Chen and Shih [11] pointed out that Das’s scheme does not provide mutual authentication, and proposed a mutual authentication scheme between the user, the gateway, and the sensor nodes. In the same year, He et al. [9] insisted that Das’s scheme has security weaknesses against insider attacks and impersonation attacks. Khan and Alghathbar [4] pointed out that Das’s scheme is vulnerable to gateway node bypassing attacks and privileged-insider attacks. 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. In [12], Vaidya et al. proposed a novel two-factor mutual authentication with key agreement scheme to prevent these attacks. In 2014, Kim Jiye et al. [13] pointed out that Vaidya et al.’s scheme [12] 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. To remedy the security flaws in Vaidya et al.’s scheme [12], Kim Jiye et al. proposed an improved two-factor mutual authentication with key agreement in wireless sensor networks by storing secret data in unique cipher text form in each node. However, we found that Kim Jiye et al.’s scheme still has some security flaws such as offline password guessing attack, user impersonation attacks using an attacker’s own smart card, sensor node impersonation attacks and gateway node bypassing attacks. In this paper, we give detailed cryptanalysis of Kim Jiye et al.’s two-factor mutual authentication with key agreement in WSNs.

The remainder of the paper is organized as follows. Section 2 presents review of Kim Jiye et al.’s scheme. Section 3 gives detailed cryptanalysis of Kim Jiye et al.’s scheme. Section 4 finally concludes this paper.

©2016 Journal of Communications
II. REVIEW OF KIM JIYE ET. AL.’S SCHEME

Three communication parties are involved in Kim Jiye et al.’s scheme [13]: a user, a gateway node, and a sensor node. The scheme is composed of four phases: registration phase, login phase, authentication-key agreement phase, and password change phase. We describe each phase in detail from section A to section D. The notations used in the remainder of this paper are shown in Table I.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>$i$-th user</td>
</tr>
<tr>
<td>$S_j$</td>
<td>$j$-th sensor node</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway node</td>
</tr>
<tr>
<td>ID$_U$</td>
<td>Identity of $U_i$</td>
</tr>
<tr>
<td>pw$_U$</td>
<td>Password of $U_i$</td>
</tr>
<tr>
<td>SID$_U$</td>
<td>Identity of $S_j$</td>
</tr>
<tr>
<td>ID$_S$</td>
<td>Identity of smart card</td>
</tr>
<tr>
<td>$h(*)$</td>
<td>One-way hash function</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>XOR operation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>$K_s$</td>
<td>Session key</td>
</tr>
<tr>
<td>$f(x,k)$</td>
<td>Pseudo-random function of variable $x$ with key $k$</td>
</tr>
<tr>
<td>$T_i,T_j$</td>
<td>Current timestamp of $U_i, GW$</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>The maximum of transmission delay time permitted</td>
</tr>
</tbody>
</table>

A. Registration Phase

In the registration phase, $U_i$ selects ID$_U$ and pw$_U$, generates a random nonce RN$_i$, and computes $H\_PW_i = h(pw_U \| RN_U)$, then sends the registration request $(ID_U, pw_U)$ to GW. Then, GW personalizes a smart card for $U_i$.

R-1: $U_i$ selects ID$_U$ and pw$_U$.

R-2: $U_i$ generates random nonce RN$_i$, and computes $H\_PW_i = h(pw_U \| RN_U)$. Then $U_i$ sends the registration request $(ID_U, H\_PW_i)$ to GW in secure channels.

R-3: GW computes the following when it receives the registration request from $U_i$. $H\_ID_U = h(ID_U \| K)$,

$X_S = h(H\_ID_U \| x_S)$, $A_i = h(H\_PW_i \| X_S) \oplus h(ID_U \| K)$, $B_i = h(H\_PW_i \| X_S)$, $C_i = X_S \oplus h(ID_U \| H\_PW_i)$. GW personalizes the smart card with ID$_S$, H\_ID$_S$, b$(\cdot)$, A, B, and C, then GW sends the smart card to $U_i$ in secure channels.

R-4: $U_i$ computes $X\_PW_i = h(pw_U) \oplus RN_i$ and adds $X\_PW_i$ to the smart card.

B. Login Phase

The login phase begins when $U_i$ inserts his/her smart card into a terminal and inputs ID$_U$ and pw$_U$. In this phase, $U_i$ sends the authentication request to GW. The detailed login phase is shown as follows.

L-1: $U_i$ inserts its smart card into a terminal and inputs ID$_U$ and pw$_U$. $U_i$ sends the registration request $(ID_U, pw_U)$ to GW. Then, GW compares

$L-2$: The smart card computes RN$_s$ = $h(pw_U) \oplus X\_PW_i$, $H\_PW_i = h(pw_U \| RN'_s)$, $X_s = C_i \oplus h(ID_U \| H\_PW_i)$, $B'_i = h(H\_PW_i \| X_s)$. Then the smart card compares B with B$'_i$. If B$'_i = B$, the next step proceeds, otherwise, this phase is aborted.

L-3: The smart card generates a random nonce RN, and computes $DID = h(H\_PW_i \| X_s) \oplus h(X_s H\_PW_i \| RN_i \| T_i)$, $M_{v_i-g} = h(A_i \| X_s \| RN_i \| T_i)$, $v_i = RN_i \| X_s \| T_i$ is the current timestamp of $U_i$ system. The smart card sends the authentication request $(DID, M_{v_i-g}, v_i, T_i, H\_ID_U)$ to GW.

C. Authentication-Key Agreement Phase

The authentication-key agreement phase begins when GW receives an authentication request from $U_i$. In this phase, $U_i, S$, send and receive authentication request from one another. The following describes the process in detail.

A-1: GW checks if $(T_g - T_i) \leq \Delta T$, where $T_g$ is the current time-stamp of GW system. If $(T_g - T_i) \leq \Delta T$ holds, the next step proceeds; otherwise, this phase is aborted.

A-2: GW computes $X_S = h(H\_ID_U \| X_S)$, RN$_i = v_i \oplus X_S$, $X' = DID \oplus h(X_S \| RN_i \| T_i)$, $M_{v_i-g} = h(X' \oplus h(H\_ID_U \| K))$. GW compares $X_s \| RN_i \| T_i$ to $M_{v_i-g}$. If $M_{v_i-g} = M_{v_i-g}$ holds true, the next step proceeds; otherwise, this phase is aborted.

A-3: GW computes $X_s = h(SID_U \| x_S)$, $M_{g-s} = h(DID \| SID_U \| X_S \| T_g)$, where $T_g$ is the current timestamp of GW system, $S$ is the nearest sensor node that can respond to $U, S$ request. GW sends the authentication request $(DID, M_{g-s}, T_g)$ to $S$. 

©2016 Journal of Communications

59
A-4: $S_j$ checks if $(T_e - T_o) \leq \Delta T$, where $T_e$ is the current timestamp of $S_j$. If $(T_e - T_o) \leq \Delta T$ holds true, then the next step proceeds; otherwise, this phase is aborted.

A-5: $S_j$ computes $M'_{g,s_j} = h(DID || SID_j \parallel X_{s_j} || T_e)$. $S_j$ compares $M'_{g,s_j}$ with $M_{g,s_j}$. If $M'_{g,s_j} = M_{g,s_j}$ holds true, then the next step proceeds; otherwise, this phase is aborted.

A-6: $S_j$ generates a random nonce $RN_j$, and computes $y_j = RN_j \oplus X_{s_j}$, $z_i = M'_{g,s_j} \oplus RN_j$, $M_{s_j,a} = h(z_i \parallel X_{s_j} \parallel T_e)$, then sends the authentication request $(y_j, M_{s_j,a}, T_e)$ to GW.

A-7: GW checks if $(T_e - T_o) \leq \Delta T$, where $T_o$ is the current timestamp of GW. If $(T_e - T_o) \leq \Delta T$ holds true, then the next step proceeds; otherwise, this phase is aborted.

A-8: GW computes $RN_j = y_j \oplus X_{s_j}$, $z_i = M_{g,s_j} \oplus RN_j$, $M'_{s_j,a} = h(z_i \parallel X_{s_j} \parallel T_e)$. GW compares $M'_{s_j,a}$ with $M_{s_j,a}$. If $M'_{s_j,a} = M_{s_j,a}$ holds true, then the next step proceeds; otherwise, this phase is aborted.

A-9: GW computes $M_{g,v_i} = h(DID || M_{g,s_j} \parallel X_{v_i} || T_e)$, $w_i = z_i \oplus X_{s_j}$, $y_i = RN_j \oplus X_{v_i}$, $q_i = X_{s_j} \oplus RN_j$, then sends the authentication request $(y_i, w_i, M_{g,v_i}, q_i, T_e)$ to $U_i$.

A-10: $U_i$ checks if $(T_e - T_o) \leq \Delta T$, where $T_o$ is the current timestamp of $U_i$. If $(T_e - T_o) \leq \Delta T$ holds true, then the next step proceeds; otherwise, this phase is aborted.

A-11: The smart card computes $RN_j = y_i \oplus X_{s_j}$, $z_i = w_i \oplus X_{s_j}$, $M'_{g,v_i} = M'_{g,s_j} \oplus RN_j$, $M_{v_i,a} = h(DID || M'_{g,v_i} || X_{v_i} || T_e)$. The smart card compares $M'_{v_i,a}$ with $M_{v_i,a}$. If $M'_{v_i,a} = M_{v_i,a}$ holds true, the mutual authentication between $U_i$ and $S_j$ 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 $K_s = f(DID || RN_j, X_{s_j})$ to share a session key with $U_i$, where $X_{s_j} = q_i \oplus RN_j$.

D. Password Change Phase

The password change phase proceeds when $U_i$ changes his/her existing password to a new one. In the password change phase, $U_i$ does not have to communicate with GW. The password change phase is shown as follows in detail.

P-1: $U_i$ inserts its smart card into a terminal and inputs $ID_{U_i}^*, pw_{U_i}^*$ and $pw_{m}^*$, where $pw_{m}^*$ is $U_i$'s new password.

P-2: The smart card computes $RN_j^* = h(pw_{U_i}^*) \oplus X_{U_i} \parallel PW_i$, $h_{-PW} = h(pw_{U_i}^* || RN_j^*)$, $X_{s_j} = C_s \oplus h(ID_s \parallel H_{-PW})$, $B_j = h(H_{-PW} \parallel X_{s_j})$. The smart card compares $B_j$ with $B_i$. If $B_j = B_i$ holds true, then the next step proceeds; otherwise, this phase is aborted.

P-3: The smart card computes $H_{-PW_m} = h(pw_{m}^* || RN_j^*)$, $A_m = A \oplus h(H_{-PW_m} || X_{s_j}) \oplus h(H_{-PW_m} || X_{s_j})'$, $B_j = h(H_{-PW_m} \parallel X_{s_j})$. $C_m = X_{s_j}' \oplus h(ID_s \parallel H_{-PW_m})$. The smart card replaces the existing values $A, B$ and $C$ with the new values $A_m, B_m$ and $C_m$.

III. SECURITY ANALYSIS OF KIM JIYE ET AL.’S SCHEME

In this section, we analyze the security of Kim Jiye et al.’s scheme. In the following sections, we describe possible attacks in detail. We assume that an attacker can eavesdrop on or intercept all message sent or received between communication parties. We also assume that an attack can read data stored in a smart card in any manner as found in the related works [2], [3], [14]-[17]. In addition, we have to note that data stored in sensor nodes are not secure since attackers can capture sensor nodes that are deployed in unattended environments and then can extract data from them.

A. Offline Password Guessing Attack

Since $B_i$ and $C_i$ are stored in $U_i$’s smart card, the attacker can obtain $U_i$’s password by using offline password guessing attack. Besides the $U_i$’s password $PW_i$ and identity $ID_s$, some important secrets such as $x_i$ and $K$ can also be derived. The detailed process is shown as follows.

Step 1: The attacker $U_a$ reads $ID_s$, $h(x_i) \parallel H_{-ID_s}$, $X_{-PW}$, $A_i$, $B$ and $C$ in $U_i$’s smart card in any manner which is used in the related works [2], [3], [14]-[17].

Step 2: $U_a$ chooses a random nonce $H_{-PW}$, and verifies if $B_i = h(H_{-PW} \oplus C_i \parallel h(ID_s \parallel H_{-PW}))$ holds true. The operation repeats until $B_i = h(H_{-PW} \oplus C_i \parallel h(ID_s \parallel H_{-PW}))$ holds true.

Step 3: $U_a$ chooses a random nonce $PW$, and verifies if $H_{-PW} = h(pw_i || (X_{-PW} \oplus h(pw)))$ holds true. The operation repeats until $H_{-PW} = h(pw_i || (X_{-PW} \oplus h(pw)))$ holds true.

Step 4: The $X_{s_j}$ can be computed as $X_{s_j} = C_s \parallel h(ID_s \parallel H_{-PW})$.

Step 5: The $x_i$ can be guessed as follows. $U_a$ chooses a random nonce $x_{s}$, and verifies if $X_{s} = h(H_{-ID_s} \parallel x_{s})$ holds true. The operation repeats until $X_{s} = h(H_{-ID_s} \parallel x_{s})$ holds true.

Step 6: The secret $K$ can be guessed as follows. $U_a$ chooses a random nonce $K$, and verifies if $A_i = h(H_{-PW} || X_{s_j} \parallel h(ID_s \parallel K))$ holds true. The operation repeats until the equation holds true.
Step 7: $U_i$ identity can be guessed as follows. $U_i$ chooses a random nonce $ID_i$, and verifies if $H_{- ID_i}=h(ID_i(K))$ holds. The operation repeats until the equation holds true.

B. User Impersonation Attacks Using an Attacker’s Own Smart Card

If attacker $U_a$ has registered with GW, he/she can receive the smart card personalized with his/her own identity $ID_a$ and password $pw_a$. The detailed registration and login processes can be shown as follows.

Step 1: $U_a$ selects $ID_a$ and $pw_a$.

Step 2: $U_a$ generates a random nonce $RN_a$ and computes $H_{- PW_a}=h(pw_a \oplus RN_a)$, then sends the registration request $\{ID_a, H_{- PW_a}\}$ to GW in a secure channel.

Step 3: When GW receives a registration request from $U_a$, it computes $H_{- ID_a}=h(ID_a(K))$. $X_a=h(ID_a)$, $A_a=h(H_{- PW_a}[X_a]) \oplus h(H_{- ID_a}(K))$, $B_a=h(H_{- PW_a} \oplus X_a)$, $C_a=X_a \oplus h(ID_a \oplus H_{- PW_a})$. GW personalizes the smart card with $ID_a$, $H_{- ID_a}$, $w_a$, $A_a$, $B_a$, and $C_a$, and sends the smart card to $U_a$ in a secure channel.

Step 4: $U_a$ computes $X_{- PW_a}=h(pw_a) \oplus RN_a$ and adds $X_{- PW_a}$ to the smart card.

Step 5: $U_a$ sends the authentication request to GW and inputs $ID_a$ and $pw_a$.

Step 6: The smart card computes $RN_{a}'=h(pw_a) \oplus X_{- PW_a}$, $H_{- PW_{a}'}=h(pw_a \oplus RN_{a}')$, $X_{a}'=C_a \oplus h(ID_a \oplus H_{- PW_{a}'})$, $B_{a}'=h(H_{- PW_{a}'} \oplus X_{a}')$. The smart card compares $B_{a}'$ with $B_a$. Obviously $B_{a}'=B_a$ holds true, then next step proceeds.

Step 7: The smart card generates random nonce $RN_{a}'$ and computes the following.

$$DID_{a}=h(H_{- PW_{a}'})[X_{a}'] \oplus h(X_{a}')[RN_{a}'][T_{a}]$$

where $T_{a}$ is the current timestamp of $U_a$ system, $M_{\nu_{a}, \nu}=h \{X_{a}' \parallel RN_{a}' \parallel T_{a}\}$, $v_{a}=RN_{a}' \oplus X_{a}'$. The smart card sends the authentication request $\{DID_{a}, M_{\nu_{a}, \nu_{a}}, v_{a}, T_{a}, H_{- ID_a}\}$ to GW.

Step 8: When GW receives the authentication request $\{DID_{a}, M_{\nu_{a}, \nu_{a}}, v_{a}, T_{a}, H_{- ID_a}\}$ from $U_a$, it checks if $(T_{a}-T_{e}) \leq \Delta T$, where $T_{e}$ is the current timestamp of the GW system. If it holds true, the next step proceeds; otherwise, this phase is aborted.

Step 9: GW computes $X_{a}=h(H_{- ID_a}) \parallel X_{a}'$, $RN_{a}=v_{a} \oplus X_{a}$, $\chi'=DID_{a}[h(X_{a})[RN_{a}'][T_{a}]]$, $M_{\nu_{a}, \nu_{a}}'=h(\chi'[h(H_{- ID_a}(K))][X_{a}'][RN_{a}'][T_{a}])$, and then compares $M_{\nu_{a}, \nu_{a}}'=M_{\nu_{a}, \nu_{a}}$. Obviously, if $M_{\nu_{a}, \nu_{a}}'=M_{\nu_{a}, \nu_{a}}$ holds true, the attacker $U_a$ is authenticated by the GW. Once $U_a$ is authenticated by GW, with the help of GW, a mutual authentication between $U_a$ and $S_j$ is completed successfully. In addition, the smart card and $S_j$ both compute a session key $K_x=f(DID_{a}[\parallel RN_{a}], X_{a})$ and share it when communication.

C. Sensor Node Impersonation Attacks

In Kim Jiye et al.’s scheme, if an attacker $U_a$ captures $S_j$ deployed in unattended environments, he/she can extract $X_{s_j}=h(SID_{j}[x_j])$ from it. Once $U_a$ eavesdrops on or intercepts $U_a$'s login request $\{DID_{a}, M_{G-a}, v_{a}, T_{a}, H_{- ID_a}\}$, he/she can forge a valid sensor node $S_j$ and complete mutual authentication between $U_a$ and $U_a$. With the help of session key $K_x=f(DID_{a}[\parallel RN_{a}], X_{a})$, $U_a$ can send fake message to $U_j$. The detailed process can be shown as follows.

Step 1: $U_a$ strives to capture $S_j$, and then extracts $SID_{j}$ and $X_{s_j}=h(SID_{j}[x_j])$ stored in $S_j$.

Step 2: $U_a$ eavesdrops on or intercepts $U_a$'s login request $\{DID_{a}, M_{G-a}, v_{a}, T_{a}, H_{- ID_a}\}$ sent to GW, and then gets $U_a$'s dynamic identity $DID_{a}$.

Step 3: When intercepting the authentication request $\{DID_{a}, M_{G-a}, T_{a}\}$ from GW to $S_j$, $U_a$ checks if $(T_{a}-T_{e}) \leq \Delta T$, where $T_{e}$ is the current timestamp of $U_a$ system. If $(T_{a}-T_{e}) \leq \Delta T$ holds true, the next step proceeds; otherwise, this phase is aborted.

Step 4: $U_a$ computes $M_{\nu_{a}, \nu_{a}}'=h(DID_{a}[\parallel RN_{a}], X_{a}), U_a$ compares $M_{\nu_{a}, \nu_{a}}'$ with $M_{\nu_{a}, \nu_{a}}$, since it holds the next step.

Step 5: $U_a$ generates a random nonce $RN_{a}'$ and uses the extracted $X_{s_j}$ stored in $S_j$ to compute $y_{a}=RN_{a} \oplus X_{s_j}$, $z_{a}=M_{G-a} \oplus RN_{a}', M_{\nu_{a}, \nu_{a}}'=h(z_{a}[\parallel X_{s_j} \parallel T_{e}])$, $U_a$ sends the authentication request $\{y_{a}, M_{\nu_{a}, \nu_{a}}', T_{e}\}$ to GW.

Step 6: GW checks if $(T_{e}-T_{a}) \leq \Delta T$, where $T_{e}$ is the current timestamp of GW. If $(T_{e}-T_{a}) \leq \Delta T$ holds true, the next step proceeds; otherwise, this phase is aborted.

Step 7: GW computes $RN_{s_j}=\gamma_{a} \oplus X_{s_j}$, $z_{a}=M_{G-a} \oplus RN_{a}'$, $M_{\nu_{a}, \nu_{a}}'=h(z_{a}[\parallel X_{s_j} \parallel T_{e}])$. GW compares $M_{\nu_{a}, \nu_{a}}'$ with $M_{\nu_{a}, \nu_{a}}$. Since $M_{\nu_{a}, \nu_{a}}'=M_{\nu_{a}, \nu_{a}}$ holds true, the next step proceeds.

Step 8: GW computes $M_{\nu_{a}, \nu_{a}}'=h(DID_{a}[\parallel M_{\nu_{a}, \nu_{a}}'[ \parallel X_{a}[\parallel T_{a}])$. $w_{a}=\gamma_{a} \oplus X_{s_j}$, $y_{a}=RN_{a} \oplus X_{s_j}$, $q_{a}=X_{s_j} \oplus RN_{a}'$. GW sends the authentication request $\{y_{a}, w_{a}, M_{\nu_{a}, \nu_{a}}'[ \parallel q_{a} \parallel T_{a}\}$ to $U_a$.

Step 9: $U_a$ checks if $(T_{e}-T_{a}) \leq \Delta T$, where $T_{e}$ is the current timestamp of $U_a$ system. If $(T_{e}-T_{a}) \leq \Delta T$ holds, then the next step proceeds; otherwise, this phase is aborted.

Step 10: The smart card computes $RN_{a}=\gamma_{a} \oplus X_{s_j}$, $z_{a}=M_{G-a} \oplus RN_{a}'$, $M_{\nu_{a}, \nu_{a}}'=h(DID_{a}[\parallel M_{\nu_{a}, \nu_{a}}'[ \parallel X_{s_j} \parallel T_{e}])$. The smart card compares $M_{\nu_{a}, \nu_{a}}'$ with $M_{\nu_{a}, \nu_{a}}$. Since
\[ M'_{a-q} = M_{a-q} \] holds true, then the mutual authentication between \( U_i \) and \( U_x \) is completed successfully.

### D. Gateway Node Bypassing Attacks

In Kim Jiye et al.’s scheme, if an attacker \( U_x \) derives \( X_s \) from smart card and extracts \( SID \) from a captured sensor node \( S_i \), and then eavesdrops on the authentication request \((DDI, M_{a-q}, v, T, H - ID)\) from \( U_i \) to GW, he/she can mount gateway node bypassing attacks. The detailed process is as follows.

**Step 1:** \( U_x \) gets \( X_s \) from \( U_i \)’s smart card and extracts \( SID \) from a captured sensor node \( S_i \).

**Step 2:** When \( U_i \) sends the authentication request \((DDI, M_{a-q}, v, T, H - ID)\) to GW, \( U_x \) eavesdrops on it.

**Step 3:** \( U_i \) computes the following using \( X_s, SID \) and \((DDI, M_{a-q}, v, T, H - ID)\), \( T_s \) and \( T_e \) are the current timestamp of \( U_i \) system, and \( T_s \leq T_e \). \( RN \) is a random nonce generated by \( U_a \cdot y_a = RN \otimes X_s \), \( M_{g-j} = h(DID) \| SID \| X_s \| T_e \) , \( z' = M_{g-j} \otimes RN \), \( w = z' \otimes X_s \), \( M_{g-j} = h(DID) \| M_{g-j} \| X_s \| T_e \) .

\( U_i \) forges the authentication request sent from GW to \( U_i \) in the authentication-key agreement phase using \((y, w, M_{a-g}, T_e)\).

**Step 4:** When receiving \((y, w, M_{a-g}, T_a)\) from \( U_x \), \( U_i \) checks if \((T_e - T_a) \leq \Delta T\), where \( T_e \) is the current timestamp of \( U_i \) system. If \((T_e - T_a) \leq \Delta T\) holds true, the next step proceeds; otherwise, this phase is aborted.

**Step 5:** The smart card computes \( RN = y_a \otimes X_s \), \( z'' = w \otimes X_s \), \( M_{g-j} = z'' \otimes RN \), \( M_{g-j} = h(DID) \| M_{g-j} \| X_s \| T_e \) .

The smart card compares \( M_{a-g} \) with \( M'_{a-g} \). Since \( M'_{a-g} = M_{a-g} \cdot U_i \) regards \((y, w, M_{a-g}, T_e)\) as being transmitted from GW. Therefore, \( U_i \) can communicate with \( U_x \) using session key \( K = f((DDI) \| RN_a \| X_s) \).

### IV. Conclusions

In this study, we have cryptanalyzed a two-factor mutual authentication with key agreement in wireless sensor networks proposed by Kim Jiye et al., and pointed out the scheme’s vulnerability to offline password guessing attack, user impersonation attacks using an attacker’s own smart card, sensor node impersonation attacks and gateway node bypassing attacks. In the future work, we will propose an improved authentication scheme to remedy the security weakness of Kim Jiye et al.’s scheme.

### ACKNOWLEDGMENT

The authors gratefully thanks for the helpful and suggestions of reviewers. This work is funded by Natural Science Foundation of Hubei Province of China under Grant No.2014CFB577, and partly supported both by the National Natural Science Foundation of China under Grant No.61370223 and by Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation under Grant No.T201410.

### REFERENCES


Jiping Li was born in Hubei Province, China, in 1972. He received M.S. degree in application of computer from Ocean University of China, Qingdao in 2006, and the Ph.D. degree in radio physics from Central China Normal University, Wuhan in 2012. He is presently an associate professor in computer science of Hubei Engineering University. His research interests include network security, wireless resource management and application of internet of things.

Yaoming Ding was born in Hubei Province, China, in 1974. He received B.S. and M.S. degree in physics from Central China Normal University, Wuhan in 1986 and in 2000 respectively. He received Ph.D. degree in Huazhong University of Science and Technology in 2011. He is presently a professor in physics science at Hubei Engineering University. His research interests include optical communication and security of wireless communication.

Zenggang Xiong was born in Hubei Province, China, in 1974. He received M.S degree in computer application from Hubei University in 2005 and Ph.D. degree in computer application from University of Science and Technology Being in 2009 respectively. He is presently a professor in computer science at Hubei Engineering University. His research interest includes cloud computing and big data.

Shouyin Liu was born in Henan Province, China, in 1963. He received the BS degree in physics and the MS degree in radio electronics from Central China Normal University, Wuhan, China, in 1985 and in 1988, respectively. He received the Ph.D. degree from Hanyang University, Korea in 2005 in electronic communication engineering. From 2004, he has been a professor at Central China Normal University. His current research interests include digital communication, WSN and location techniques.

Honglai Li was born in Hubei Province, China, in 1964. He received B.S. degree in information management from Zhongnan University of Economics and Law. He is presently a senior Experimentalist at Hubei Engineering University. His research interest includes zigbee network.