# Cryptanalysis of Kumar et al.'s authentication protocol for wireless sensor networks

Sewan Ha[1][0000-0002-0024-5716], Jihyeon Ryu[1][0000-0001-8124-3853], Hyoungshick Kim[1][0000-0002-1605-3866], Dongho Won[1][0000-0002-5208-1338], and Youngsook Lee[2]

[1] Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 16419, Korea
[2] Howon University, 64 Howon University 3 Gil, Impi-Myeon, Gunsan-Si, Jeonrabuk-Do, 54058, Korea
ysooklee@howon.ac.kr

**Abstract.** Wireless sensor networks are popularly used for many applications to monitor environmental changes and track events. In 2018, Kumar et al. proposed a secure and efficient authentication scheme for coal mine monitoring using wireless sensor networking technology. In this paper, we analyze the security issues of Kumar et al.'s authentication scheme. We found that Kumar et al.'s has four security weaknesses: (1) sensor nodes' critical information can be disclosed; (2) session keys can be compromised; (3) user impersonation is possible, and (4) users' identity and password can be leaked with a stolen smart card.

**Keywords:** Wireless sensor network, User authentication, Key agreement, Sensor network

## 1      Introduction

Wireless sensor networks (WSNs) consist of many compact, low-powered, autonomous sensor nodes and collect environmental and physical data from the surrounding environment. With the deployment of Internet of Things (IoT) applications, WSNs have received more attention recently. WSNs are particularly useful for industrial control applications (e.g., smart city, smart factory, and smart grid). Also, WSNs would be used in hazardous environments (e.g., ocean and coal mine) for safety monitoring because people can periodically check the status of hazardous environments with environmental data collected from wireless sensors that are deployed in such an environment.

Unsurprisingly, it is important to provide the integrity of sensor outputs because modified sensor data may cause significant financial losses and/or affect human safety. Therefore, an important issue is to provide a secure user authentication mechanism to prevent unauthorized access to sensor nodes. There have been several attempts to provide secure and efficient user authentication schemes [8 – 10]. Recently, Kumar et al. [9] also proposed an authentication protocol for WSNs, looking promising. However, we found that Kumar et al.'s authentication protocol has four vulnerabilities. First, sensor nodes' authentication information can be misused. Second, the attacker

can compromise all session keys between the user, gateway, and sensor node. Third, the scheme cannot resist user impersonation attack. Last, from a stolen smart card, the user's identification information can be leaked.

### 1.1    Threat Model

In this paper, we assume that an attacker is a Dolev-Yao attacker [3]. In particular, the attacker has the following capabilities:

1  The attacker can intercept all messages transmitted through public channels.

2  The attacker can modify, delete, and resend the intercepted messages.

3  (Section 4.4 Smart card loss attack) Devices are not tamper-resistant, so the attacker can extract all parameters in the device by applying side-channel attacks.

4  (Section 4.4 Smart card loss attack) User's identity and password are weak secrets, so the attacker can derive them if there are no other values to be changed except for identity or password.

### 1.2    Organization of the Paper

The remainder of the paper is organized as follows. In Section 2, we discuss various related studies about authentication schemes for WSNs. In Section 3, we review the Kumar et al.'s scheme in detail. Furthermore, we do cryptanalysis on Kumar et al.'s scheme and demonstrate four vulnerabilities of the scheme in Section 4. Lastly, we conclude our study and provide future works of this paper in Section 5.

## 2    Related Work

An authentication for WSNs has begun with Wong et al.'s scheme [14]. In 2006, Wong et al. introduced a symmetric-key based lightweight authentication for WSNs. However, in 2009, Das [2] showed that Wong et al.'s scheme has security weaknesses such as same login identity, replay, and stolen-verifier attacks; thus, Das proposed an enhanced version of Wong et al.'s. Unfortunately, Huang et al. [5] found that Das' scheme is susceptible to the sensor node impersonation and many logged-in-users attacks, so they proposed an improved version of Das'. However, He et al. [4] reported that Huang et al.'s scheme has impersonation and privileged insider attacks besides the absence of a password change phase. He et al. proposed a new scheme, but Vaidya et al. [13] found security weaknesses of the scheme and presented a two-factor authentication scheme with key agreement for WSNs. In 2014, Kim et al. [7] discussed the security problems of Vaidya et al.'s scheme such as user impersonation and gateway node bypass attacks, and they introduced an improved version of Vaidya et al.'s scheme. Unfortunately, in 2015, Chang et al. [1] showed that Kim et al.'s scheme has security weaknesses (e.g., man-in-the-middle attack, impersonation attack), so they proposed a new two-factor authentication using dynamic identities. However, Park et al. [11] and Jung et al. [6] proved that Chang et al.'s scheme has vulnerabilities and presented new authentication schemes to overcome the weakness-

es. In 2019, Shin et al. [12] pointed out that Jung et al.'s scheme is vulnerable to tracing, information leakage, and session key attacks.

In 2016, Kumari and Om [10] proposed an authentication scheme for WSNs for safety monitoring in coal mines. Unfortunately, in 2017, Kumar et al. [9] found that Kumari and Om's scheme is vulnerable to smart card loss, stolen verifier, and denial of service attacks. As a result, they introduced an improved version of Kumari and Om's scheme; however, we found that Kumar et al.'s scheme still has four vulnerabilities such as sensor node's critical information leakage, session key compromise, user impersonation attack, and smart card loss attack.

## 3 Review of Kumar Et Al.'s Scheme

In this section, we review the Kumar et al. [9]'s scheme. The scheme is two-factor key agreement authentication scheme, and there are four entities: registration center, user, gateway, and sensor node. A registration center involves only when a gateway and sensor nodes are set up, and when a user registers. During login and authentication phase, a user logs in to a gateway using his/her smart card; then, the gateway and the sensor nodes verify the user using timestamp and secret parameters. Comprehensive notations are included in Table 1.

**Table 1.** Notations used in Kumar et al.'s scheme.

| Notation | Description |
|---|---|
| RC | A registration center |
| $U_i$ | A user |
| $GW_j$ | A gateway |
| $SN_k$ | A sensor node |
| $MSK_{RC}$ | A registration center's master secret key |
| $GID_j$ | $GW_j$'s identity |
| $SID_k$ | $SN_k$'s identity |
| $ID_i$ | $U_i$'s identity |
| $PW_i$ | $U_i$'s password |
| $SK_{ijk}$ | A session key between $U_i$, $GW_j$ and $SN_k$ |
| $V_i$ | Secret nonces for $U_i'$s authentication |
| $l$ | the length of fuzzy verifier |
| $h(\cdot)$ | An one-way hash function |
| $\oplus, \parallel$ | Xor and Concatenation operation |
| $\mathcal{A}$ | An attacker |

### 3.1 Sensor-gateway Node Registration Phase

A registration center $RC$ deploys gateways $GW_j$ and sensors $SN_k$ as follows:

1. $RC$ chooses $MSK_{RC}$ as a master secret key and assigns an identity $GID_j$ for $GW_j$.

2. $RC$ gets $GW_j$'s hashed identity by calculating $HGID_j = h(GID_j \parallel MSK_{RC})$.

3. $RC$ selects $SN_k$'s identity $SID_k$ and computes the hashed identity $HSID_k = h(SID_k \parallel MSK_{RC})$.

4. $RC$ stores the gateway $GW_j$'s authentication information and the connected sensors $SN_k$'s authentication information $\{GID_j, HGID_j, SID_k, HSID_k\}$ in $GW_j$ and deploys $GW_j$ in a desirable place.

5. Similarly, $RC$ saves the sensor $SN_k$'s information $\{SID_k, HSID_k\}$ in $SN_k$, and sets the sensor node $SN_k$ at appropriate place.

### 3.2 User Registration Phase

In case a user $U_i$ registers to the gateway $GW_j$, the following steps will proceed:

1. $U_i$ chooses his/her identity $ID_i$ and password $PW_i$ and picks a nonce $b_i$. Then, $U_i$ calculates hashed identity $HID_i = h(ID_i \parallel b_i)$ and hashed password $HPW_i = h(PW_i \parallel b_i)$, and transmits $\{HID_i, HPW_i\}$ to $GW_j$ via a private channel.

2. After receiving the $U_i$'s registration request, $GW_j$ chooses a integer $l$ between $2^4$ and $2^8$. After that, $GW_j$ generates a nonce $V_i$ and computes $A_{ij} = h(V_i \parallel HGID_j) \oplus h(HID_i \parallel HPW_i)$, $B_{ij} = V_i \oplus h(HGID_j)$.

3. $GW_j$ puts parameters and hash function $\{A_{ij}, h(\cdot), B_{ij}, l\}$ into the smart card $SC$ that will be $U_i$'s; then, $GW_j$ securely delivers the smart card to $U_i$.

4. After obtaining smart card from $GW_j$, $U_i$ computes and stores the following parameters into the smart card:

$$HB_i = h(HPW_i \parallel HID_i \parallel b_i) \quad \bmod l$$
$$c_i = b_i \oplus h(ID_i \parallel PW_i) \qquad \bmod l$$

5. The $U_i$'s smart card will contain $\{A_{ij}, h(\cdot), B_{ij}, l, HB_i, c_i\}$.

### 3.3 Login Phase

When a user $U_i$ wants to log in to $GW_j$, $U_i$ performs the following steps:

1. $U_i$ inputs his/her identity $ID_i$ and password $PW_i$ when inserting his/her smart card $SC$ into a card reader.

2. $SC$ computes the following parameters which are needed for the login process:

$$b_i = c_i \oplus h(ID_i \parallel PW_i) \qquad \bmod l$$
$$HID_i = h(ID_i \parallel b_i)$$
$$HPW_i = h(PW_i \parallel b_i)$$
$$HB'_i = h(HPW_i \parallel HID_i \parallel b_i) \quad \bmod l$$

3. $SC$ compares $HB'_i$ with the stored parameter $HB_i$. If two parameters are different, $SC$ terminates the login session. Otherwise, $SC$ generates a nonce $R_i$.

4. $SC$ calculates the following parameters and transmits $\{L, V_2, V_3, T_1\}$ to $GW_j$ via a public channel:

$$L = B_{ij} \oplus T_1, \text{ where } T_1 \text{ is a current timestamp}$$
$$V_1 = A_{ij} \oplus h(HID_i \parallel HPW_i)$$
$$V_2 = h(T_1 \parallel R_i \parallel V_1)$$

$$V_3 = (R_i \parallel T_1) \oplus V_1$$

## 3.4 Authentication and Key Agreement Phase

After receiving login request $\{L, V_2, V_3, T_1\}$ from the user $U_i$, the gateway $GW_j$ and the sensor node $SN_k$ perform the following steps to establish mutual authentication between $U_i$, $GW_j$, and $SN_k$:

1. When $GW_j$ obtains the login request $\{L, V_2, V_3, T_1\}$ from $U_i$, $GW_j$ computes the following equations and gets $V'_i$, $V'_1$, and $R_i$:
$$V'_i = L \oplus h(HGID_j) \oplus T_1$$
$$V'_1 = h(V'_i \parallel HGID_i)$$
$$(R'_i \parallel T'_1) = V'_1 \oplus V_3$$

2. $GW_j$ checks the validity of $T_1$ by checking $T'_1 = T_1$. Moreover, $GW_j$ checks the freshness of $T_1$. If the timestamp $T_1$ is not valid or not fresh, $GW_j$ terminates the authentication session. Otherwise, $GW_j$ calculates $V'_2 = h(T_1 \parallel R'_i \parallel V'_1)$.

3. $GW_j$ checks whether two parameters $V_2$ and $V'_2$ are equal or not. If two parameters are the same, $GW_j$ can infer that $U_i$ is authenticated; otherwise, $GW_j$ terminates the authentication session.

4. $GW_j$ generates a nonce $R_j$ and calculates the following parameters:
$$C_1 = h(SID_k \parallel V'_i \parallel HSID_k \parallel R_j \parallel T_2), \text{ where } T_2 \text{ is a current timestamp}$$
$$C_2 = (R'_i \parallel R_j \parallel T_2) \oplus HSID_k$$
$$C_3 = V'_i \oplus h(SID_k \parallel h(R_j) \parallel R'_i)$$

5. $GW_j$ sends the message $\{C_1, C_2, C_3\}$ to $SN_k$.

6. When $SN_k$ receives the authentication message $\{C_1, C_2, C_3\}$ from $GW_j$, $SN_k$ computes $(R''_i \parallel R'_j \parallel T'_2) = C_2 \oplus HSID_k$ and obtains $R''_i$, $R'_j$, and $T'_2$.

7. $SN_k$ validates the freshness of the authentication message using calculated timestamp $T'_2$. If $T'_2$ fails the freshness test, $SN_k$ terminates the authentication session; otherwise, $SN_k$ computes the following parameters:
$$V'_i = C_3 \oplus h(SID_k \parallel h(R'_j) \parallel R''_i)$$
$$C'_1 = h(SID_k \parallel V'_i \parallel HSID_k \parallel R'_j \parallel T'_2)$$

8. $SN_k$ checks $C'_1 = C_1$. If $C'_1$ and $C_1$ differ, $SN_k$ terminates the authentication session. Otherwise, $SN_k$ can infer that $G_j$ is validated.

9. $SN_k$ selects a nonce $R_K$, computes parameters, and transmits $\{D_1, D_2\}$ to $GW_j$:
$$SK_{ijk} = h(R''_i \parallel R'_j \parallel R_k)$$
$$D_1 = h(T_3 \parallel R_k \parallel SK_{ijk} \parallel HSID_k \parallel SID_k \parallel T'_2)$$
$$D_2 = (R_k \parallel T_3) \oplus R'_j$$

10. After receiving the message from $SN_k$, $GW_j$ calculates $(R'_k \parallel T'_3) = D_2 \oplus R_j$ and obtains $R'_k$ and $T'_3$. After that, $GW_j$ performs a freshness test on $T'_3$. If $T'_3$ is not fresh, $GW_j$ terminates the authentication session.

11. $GW_j$ computes the following parameters and compares $D'_1$ with $D_1$:
$$SK_{ijk} = h(R'_i \parallel R_j \parallel R'_k)$$
$$D'_1 = h(T'_3 \parallel R'_k \parallel SK_{ijk} \parallel HSID_k \parallel SID_k \parallel T_2)$$

12. If they are different, $GW_j$ terminates the session; otherwise, $SN_k$ is verified.

13. $GW_j$ calculates the following parameters and sends $\{C_1, C_4, C_5\}$ to $U_i$:

$C_4 = h(SK_{ijk} \parallel R_j \parallel T_4 \parallel C_1)$, where $T_4$ is current timestamp

$C_5 = (R'_k \parallel R_j \parallel T_4) \oplus R'_i$

14. After acquiring the authentication message $\{C_1, C_4, C_5\}$ from $GW_j$, $U_i$'s smart card $SC$ computes $(R''_k \parallel R'_j \parallel T'_4) = C_5 \oplus R_i$ and obtains $R''_k$, $R'_j$, and $T'_4$. Then, $SC$ checks the freshness of $T'_4$. If $T'_4$ is not fresh, $SC$ halts the authentication process; otherwise, $SC$ proceeds to the next step.

15. $SC$ calculates the following parameters and checks $C'_4 = C_4$:

$SK_{ijk} = h(R_i \parallel R'_j \parallel R''_k)$

$C'_4 = h(SK_{ijk} \parallel R'_j \parallel T'_4 \parallel C_1)$

16. Lastly, if $C'_4$ and $C_4$ are equal, $SC$ can prove both $GW_j$ and $SN_k$ are verified; otherwise, $SC$ stops the process. Now, $U_i$, $GW_j$, and $SN_k$ can mutually authenticate and securely communicate using the session key $SK_{ijk} = h(R_i \parallel R_j \parallel R_k)$.

## 3.5 Password Change Phase

When a user $U_i$ wants to change his/her password, $U_i$ performs the following process:

1. After inserting his/her smart card into the card reader, $U_i$ inputs his/her identity $ID_i$ and password $PW_i$.

2. $SC$ calculates the following parameters:

$b_i = c_i \oplus h(ID_i \parallel PW_i) \qquad \mod l$

$HID'_i = h(ID_i \parallel b_i)$

$HPW'_i = h(PW_i \parallel b_i)$

$HB'_i = h(HPW'_i \parallel HID'_i \parallel b_i) \qquad \mod l$

3. $SC$ checks if $HB'_i$ is the same as $HB_i$, which is stored in $SC$. If both parameters are different, $SC$ terminates the session; otherwise, $SC$ proceeds to the next step.

4. $SC$ requests a new $PW_{new}$ and a nonce $b_{new}$; then, $U_i$ inputs $PW_{new}$ and $b_{new}$.

5. $SC$ computes the following new parameters:

$HID_{new} = h(ID_i \parallel b_{new})$

$HPW_{new} = h(PW_{new} \parallel b_{new})$

$HB_{new} = h(HPW_{new} \parallel HID_{new} \parallel b_{new}) \mod l$

$A_{new} = A_{ij} \oplus h(HID_i \parallel HPW_i) \oplus h(HID_{new} \parallel HPW_{new})$

$c_{new} = b_{new} \oplus h(ID_i \parallel PW_{new}) \qquad \mod l$

6. Finally, $SC$ replaces $HB_i \leftarrow HB_{new}$, $c_i \leftarrow c_{new}$, and $A_{ij} \leftarrow A_{new}$.

## 3.6 Sensor Node Addition Phase

Whenever a new sensor node $SN_k$ is required to be added after setting a WSNs, the registration center $RC$ follows the steps listed below:

1. $RC$ chooses $SN_k$'s identity $SID_k$ and calculates the hashed identity $HSID_k = h(SID_k \parallel MSK_{RC})$; then, $RC$ stores $\{SID_k, HSID_k\}$ in $SN_k$'s memory and deploys it at the appropriate position.

2. $RC$ sends $SN_k$'s identification information $\{SID_k, HSID_k\}$ to the closest gateway $GW_j$ for registration.

3. After receiving $\{SID_k, HSID_k\}$, $GW_j$ saves the parameters in its database.

# 4 Cryptanalysis of Kumar Et Al.'s Scheme

In this section, we analyze Kumar et al.'s scheme and prove that the scheme is still vulnerable. Overall, the scheme has four security weaknesses: critical information leakage, session key compromise, user impersonation, and smart card loss attack.

## 4.1 Critical Information Leakage

In Kumar et al.'s scheme, sensors nodes' secret parameter can be leaked. Specifically, an attacker $\mathcal{A}$ can collect hashed identity $HSID_k$ of all sensors $SN_k$ connected to a gateway $GW_j$ and determine $SN_k$ from the parameter $C_2$ using the collected $HSID_k$ and timestamp $T_1$. The attack progress is as follows:

 1. Let suppose the attacker $\mathcal{A}$ is registered user and sends his/her login request $\{L, V_2, V_3, T_1\}$ to the victim gateway $GW_j$ with his/her nonce $R_i$.

 2. During authentication phase, $\mathcal{A}$ overhears $\{C_1, C_2, C_3\}$ between $GW_j$ and $SN_k$.

 3. After authentication process, $\mathcal{A}$ can acquire $GW_j$ and $SN_k$'s nonces $R_j$ and $R_k$.

 4. $\mathcal{A}$ computes the possible hashed identity $HSID'_k = C_2 \oplus (R_i \parallel R_j \parallel T_2^*)$ using $R_i$, $R_j$ and $T_2^*$, where $T_2^*$ is $T_2$ candidate, and $T_2$ is the next timestamp of $T_1$ ($T_1 \leq T_2^* \leq T_1 + \Delta T$).

 5. $\mathcal{A}$ resends another his/her login request $\{L', V'_2, V'_3, T'_1\}$ to $GW_j$, and eavesdrops the authentication message $\{C'_1, C'_2, C'_3\}$ between $GW_j$ and $SN'_k$.

 6. Same as step 3, using $R'_i$ and the obtained $R'_j$, $\mathcal{A}$ can partially determine whether $SN'_k$ and $SN_k$ are different by checking prefix of $HSID''_k (= C'_2 \oplus (R'_i \parallel R'_j \parallel 0 \cdots 0))$, where $0 \cdots 0$ is the same length as $T_2$. If $SN_k$ and $SN'_k$ are expected to be the same, $\mathcal{A}$ calculates other possible $HSID'''_k = C'_2 \oplus (R'_i \parallel R'_j \parallel T_2^{**})$ using $R'_i$, $R'_j$ and $T_2^{**}$.

 7. $\mathcal{A}$ can get potential $HSID_k^*$ for $SN_k$ by intersecting $\{HSID'_k\}$ and $\{HSID'''_k\}$.

 8. After repeating the process several times, $\mathcal{A}$ can find a desirable $HSID_k^*$ of $SN_k$.

 Note 1) Timestamp $T_2$ can be inferred easily because the base timestamp $T_1$ is known, and the maximum transmission delay ($\Delta T$) is usually small. If $\Delta T$ is big, then the system cannot resist against a replay attack, so the system has to set $\Delta T$ small; thus, the number of $T_2^*$ candidates is very small, which can be easily predicted.

## 4.2 Session Key Compromise

Kumar et al.'s scheme is vulnerable to session key compromise attack. Especially, if the length of the data type is fixed, an attacker $\mathcal{A}$ can easily forge the session key.

### 4.2.1 The Length of Variable Is the Same

Usually, nonces are set to 32-bit or 64-bit. However, if we set $R_i$, $R_j$, and $R_k$ to the same bits, crucial information is leaked, and an attacker can use this information to compromise session key. This is because a concatenation operation increases the parameter's size. The detailed attack trace is described as follows:

1. The attacker $\mathcal{A}$ overhears overall login request of $U_i$; then, $\mathcal{A}$ can acquire at least $\{L, V_2, V_3, T_1, C_1, C_4, C_5\}$.

2. Since all nonces have fixed length, $C_5$ can be interpreted as $C_5 = (R_k \parallel R_j \parallel T_4) \oplus R_i = (R_k \parallel R_j \parallel T_4 \oplus R_i)$. This is because the timestamp is usually 32-bit or 64-bit unsigned integer. Therefore, $\mathcal{A}$ can acquire $R_k$ and $R_j$ very easily.

3. Let assume $\text{T}_r = T_4 \oplus R_i$. Now, $\mathcal{A}$ can efficiently calculate $R_i$ because $T_1 \leq T_4 \leq T_1 + 3\Delta T$, where $\Delta T$ is the maximum transmission delay. For each $T_4$ candidate $T'_4$, $\mathcal{A}$ can test whether $h(h(\text{T}_r \oplus T'_4 \parallel R_j \parallel R_k) \parallel R_j \parallel T'_4 \parallel C_1) = C_4$. If $\mathcal{A}$ finds a desirable $T'_4$, $\mathcal{A}$ can derive $R_i$ by computing $R_i = \text{T}_r \oplus T'_4$.

4. Finally, $\mathcal{A}$ can forge a session key $SK_{ijk} = h(R_i \parallel R_j \parallel R_k)$ between $U_i$, $GW_j$, and $SN_k$, so $\mathcal{A}$ can eavesdrop all secret messages among them by using the forged session key.

### 4.2.2 The Length of Variable Is Set Properly

Even if all nonces' sizes $(R_i, R_j, R_k)$ are set to the appropriate length, the attacker $\mathcal{A}$ can obtain these nonces according to the following attack method:

1. Through the same method as the Section 4.1, the attacker $\mathcal{A}$ obtains $SN_k$'s hashed id $HSID_k$; then, $\mathcal{A}$ overhears the login request between $U_i$, $GW_j$, and $SN_k$.

2. $\mathcal{A}$ acquires $T'_2$ by calculating $(R'_i \parallel R'_j \parallel T'_2) = C_2 \oplus HSID_k$, where $HSID_k$ is candidate of the possible hashed ID of $SN_k$. By checking $T_1 \leq T'_2 \leq T_1 + \Delta T$, $\mathcal{A}$ finds a desirable $R'_i$ and $R'_j$.

3. Then, $\mathcal{A}$ can find $R_k$ by computing $(R_k \parallel R'_j \parallel T_4) = C_5 \oplus R'_i$.

4. Finally, with the computed $R'_i$, $R'_j$, and $R_k$, $\mathcal{A}$ forges a session key $SK_{ijk} = h(R'_i \parallel R'_j \parallel R_k)$ between $U_i$, $GW_j$, and $SN_k$, so $\mathcal{A}$ can eavesdrop all secret messages among them by using the forged session key.

## 4.3 User Impersonation Attack

During Section 4.2, after obtaining $U_i$'s nonce $R_i$, $\mathcal{A}$ can fabricate a fake login request $\{L', V'_1, V'_2, V'_3\}$ and impersonate $U_i$. The attack procedure is as below:

1. The attacker $\mathcal{A}$ overhears overall login request of $U_i$; then, $\mathcal{A}$ can acquire at least $\{L, V_2, V_3, T_1, C_1, C_4, C_5\}$, and by implementing session key compromise attack, $\mathcal{A}$ can obtain $R_i$. Then, $\mathcal{A}$ computes and acquires $V_1 = V_3 \oplus (R_i \parallel T_1)$.

2. Now, $\mathcal{A}$ can impersonate $U_i$ because $\mathcal{A}$ can make $L' = L \oplus T_1 \oplus T'_1$, $V'_1 = V_1$, $V'_2 = h(T'_1 \parallel R'_i \parallel V_1)$, and $V'_3 = h(R'_i \parallel T'_1) \oplus V_1$, where $R'_i$ is chosen by $\mathcal{A}$ and $T'_1$ is a current timestamp. Then, $\mathcal{A}$ transmits the login request $\{L', V'_2, V'_3, T'_1\}$ to the gateway $GW_j$.

3. $GW_j$ considers the message from $\mathcal{A}$ as a login request from the $U_i$ because $T'_1$ is fresh and the message is authenticated by the shared information $V_1$.

4. When authentication message $\{C'_1, C'_4, C'_5\}$ comes, $\mathcal{A}$ can create a session key $SK'_{ijk} = h(R'_i \parallel R'_j \parallel R'_k)$, where $R'_j$ and $R'_k$ can be derived from $(R'_k \parallel R'_j \parallel T'_4) = C'_5 \oplus R'_i$, and successfully play the role of $U_i$.

### 4.4 Smart Card Loss Attack

When an attacker acquires a $U_i$'s smart card, $\mathcal{A}$ can eventually learn $U_i$'s identity $ID_i$ and password $PW_i$. To be specific, after obtaining $U_i$'s smart card, $\mathcal{A}$ extracts $B_{ij}$ and finds a message in the log of all login-authentication messages whose result of $L \oplus T_1$ matches $B_{ij}$. After that, $\mathcal{A}$ performs a session key compromise attack to get $R_i$ and $V_1$. With the derived $V_1$, $\mathcal{A}$ can find $U_i$'s identity $ID_i$ and password $PW_i$ by performing the following steps:

1. After obtaining $U_i$'s smart card ($SC$), $\mathcal{A}$ extracts $\{A_{ij}, c_i, l, HB_i\}$ from $SC$.

2. $\mathcal{A}$ chooses a random $ID_{\mathcal{A}}$ and $PW_{\mathcal{A}}$ and computes the following:
$$b_{\mathcal{A}} = c_i \oplus h(ID_{\mathcal{A}} \parallel PW_{\mathcal{A}}) \mod l$$
$$HID_{\mathcal{A}} = h(ID_{\mathcal{A}} \parallel b_{\mathcal{A}})$$
$$HPW_{\mathcal{A}} = h(PW_{\mathcal{A}} \parallel b_{\mathcal{A}})$$
$$HB_{\mathcal{A}} = h(HPW_{\mathcal{A}} \parallel HID_{\mathcal{A}} \parallel b_{\mathcal{A}}) \mod l$$

3. $\mathcal{A}$ compares $HB_{\mathcal{A}}$ with $HB_i$. If two parameters are different, then chooses another id $ID'_{\mathcal{A}}$ and password $PW'_{\mathcal{A}}$; otherwise, $\mathcal{A}$ inputs $ID_{\mathcal{A}}$ and $PW_{\mathcal{A}}$ after inserting $U_i$'s SC into the card reader.

4. After inputting $ID_{\mathcal{A}}$ and $PW_{\mathcal{A}}$, $SC$ requests a new password and nonce. Then $\mathcal{A}$ inputs the new password $PW_{\mathcal{A}}^*$ and a nonce $b_{\mathcal{A}}^*$, and $SC$ performs password change phase.

5. After password change phase is done, $\mathcal{A}$ extracts a new parameter $A_{ij}^* = A_{ij} \oplus h(HID_{\mathcal{A}} \parallel HPW_{\mathcal{A}}) \oplus h(HID_{new} \parallel HPW_{new})$

6. Since $\mathcal{A}$ knows $HID_{\mathcal{A}}$, $HPW_{\mathcal{A}}$, $PW_{\mathcal{A}}^*$, and $b_{\mathcal{A}}^*$, $\mathcal{A}$ can find $h(HID_{new} \parallel HPW_{new}) = A_{ij}^* \oplus A_{ij} \oplus h(HID_{\mathcal{A}} \parallel HPW_{\mathcal{A}})$

7. $\mathcal{A}$ can find $U_i$'s identity $ID_i$ by applying identity guessing attack on $h(HID_{new} \parallel HPW_{new}) = h(h(ID_i \parallel b_{\mathcal{A}}^*) \parallel h(PW_{\mathcal{A}}^* \parallel b_{\mathcal{A}}^*))$, where $PW_{\mathcal{A}}^*$ and $b_{\mathcal{A}}^*$ are chosen by $\mathcal{A}$.

8. Now, $\mathcal{A}$ can also find $U_i$'s password $PW_i$ by applying password guessing attack on $h(HID_i \parallel HPW_i) = h(h(ID_i \parallel b_i) \parallel h(PW_i \parallel b_i)) = A_{ij} \oplus V_1$, where $ID_i$ is already derived, $V_1$ is derived from the previous log of SC, and $b_i$ is relatively small ($2^4 \le l \le 2^8$).

Note 2) Normally, this attack cannot be realized. This can be done in Kumar et al.'s scheme because the size of fuzzy verifier $l$ is very small. If $l$ is small, the probability of finding a collision $HB_{\mathcal{A}}$ of $HB_i$ is very high. In addition, $b_i \in Z_l$ is also small, so $\mathcal{A}$ can apply password guessing attack on $HPW_i$ (which is up to 256 times slower than original password guessing attack); thus, $\mathcal{A}$ can derive $ID_i$ and $PW_i$. For this reason, it is required to increase the size of $l$ sufficiently.

## 5 Conclusion

In this paper, we examined Kumar et al.'s two-factor user authentication scheme for WSNs and found that Kumar et al.'s scheme has four security vulnerabilities such as critical information leakage, session key compromise, user impersonation attack, and smart card loss attack. As a result of these attacks, sensor nodes' identification

information is disclosed, the session key can be intentionally exposed, the user can be impersonated, and the user's identity and password can also be stolen. Therefore, we do not recommend using the current Kumar et al.'s authentication scheme.

In future work, we plan to develop an improved version of Kumar et al.'s scheme that is resistant to these attacks. Kumar et al.'s scheme is vulnerable due to the predictability of timestamp and the leakage of critical information. Therefore, our future scheme will be based on nonce-based and added additional technique on hiding critical information.

# References

1. Chang, I.P., Lee, T.F., Lin, T.H., Liu, C.M.: Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. Sensors 15(12), 29841–29854 (2015)
2. Das, M.L.: Two-factor user authentication in wireless sensor networks. IEEE transactions on wireless communications 8(3), 1086–1090 (2009)
3. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on information theory 29(2), 198–208 (1983)
4. He, D., Gao, Y., Chan, S., Chen, C., Bu, J.: An enhanced two-factor user authentication scheme in wireless sensor networks. Ad hoc & sensor wireless networks 10(4), 361–371 (2010)
5. Huang, H.F., Chang, Y.F., Liu, C.H.: Enhancement of two-factor user authentication in wireless sensor networks. In: Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. pp. 27–30. IEEE (2010)
6. Jung, J., Moon, J., Lee, D., Won, D.: Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. Sensors 17(3) (2017)
7. Kim, J., Lee, D., Jeon, W., Lee, Y., Won, D.: Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks. Sensors 14(4), 6443–6462 (2014)
8. Kumari, S., Om, H.: Authentication protocol for wireless sensor networks applications like safety monitoring in coal mines. Computer Networks 104, 137–154 (2016)
9. Kumar, A., Kingson, T., Verma, R., Mandal, R., Dutta, S., Chaulya, S., Prasad, G.: Application of gas monitoring sensors in underground coal mines and hazardous areas. International Journal of Computer Technology and Electronics Engineering 3(3), 9–23 (2013)
10. Kumar, D., Chand, S., Kumar, B.: Cryptanalysis and improvement of an authentication protocol for wireless sensor networks applications like safety monitoring in coal mines. Journal of Ambient Intelligence and Humanized Computing 10(2), 641–660 (2019)
11. Park, Y., Park, Y.: Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors 16(12) (2016)
12. Shin, S., Kwon, T.: A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes. Sensors 19(9) (2019)
13. Vaidya, B., Makrakis, D., Mouftah, H.: Two-factor mutual authentication with key agreement in wireless sensor networks. Security and Communication Networks 9(2), 171–183 (2016)
14. Wong, K.H., Zheng, Y., Cao, J., Wang, S.: A dynamic user authentication scheme for wireless sensor networks. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (2006)