Research Article

Revisiting a Multifactor Authentication Scheme in Industrial IoT

Ding Wang,1,2 Shuhong Hong,1,2 and Qingxuan Wang1,2

1College of Cyber Science, Nankai University, Tianjin 300350, China
2Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China

Correspondence should be addressed to Qingxuan Wang; wangqingxuan@mail.nankai.edu.cn

Received 23 March 2021; Accepted 22 May 2021; Published 4 June 2021

1. Introduction

Internet of Things (IoT) has developed rapidly in recent years, which generally penetrates into people’s life, and there are many IoT devices applied to various domains [1, 2]. Due to the superiority in automatic monitoring, efficient control, and intelligent manufacturing, Industry IoT (IIoT) is widely concerned among these domains. In the IIoT environment, sensing devices can be accessed and controlled by users remotely. During the process of production, sensing devices collect the real-time data, and the data can be obtained by users. The network model for IIoT is described in Figure 1. As a security critical system, IIoT has higher requirements in the secure transmission and communication of data [3, 4]. However, it is vulnerable to an attacker to perform attacks because the collected data is often transmitted through a public channel, and this brings security problems in the IIoT environment. It is possible for an adversary to launch attacks and impersonate an authorized user to obtain the data by accessing sensing devices. The unsatisfactory situation mentioned above will lead to destruction of the industrial production.

Therefore, in order to ensure the safe data transmission between users and sensing devices, many authenticated key agreement schemes [5–7] in IIoT are proposed. In these schemes, users are only allowed to access one sensing device at a time. When the user accesses multiple devices, his identity must be validated repeatedly. While supporting the critical security features, such as shared session key establishment and user authentication, an authentication scheme for IIoT environments should also be able to reduce the communication and computational costs due to the resource-constrained nature of IoT devices.

Recently, Vinoth et al. [8] proposed a multifactor authenticated key agreement scheme for the IIoT environment, aiming to support the authorized user remotely accessing multiple sensing devices. They claimed that their scheme is suitable for the resource-constrained IIoT and has less cost during communication and computation processes. Besides, they demonstrated the security of the proposed scheme through a formal security analysis, which indicates that their scheme is resistant to various known attacks. Unfortunately, some subtleties are overlooked. In this paper, we find that their scheme cannot resist the smart card attack and the sensing device capture attack. Furthermore, we point out that their scheme cannot support forward secrecy. Although the scheme is a multifactor authentication mechanism
2. Revisiting Vinoth et al.’s Scheme

In this section, we first revisit Vinoth et al.’s scheme [8] briefly and list some intuitive notations and abbreviates in Table 1 for the convenience of description. Their scheme includes six phases, while we only review the first three phases, which are related to our proposed attacks.

2.1. Offline Sensing Devices’ Registration Phase. Each sensing device SDj is registered by GWN in offline and is distributed a unique identity IDSDj. In order to calculate the secret, GWN chooses a secret value $S$ and two vectors Vector1 and Vector2. Assume that $S = \text{Vector}_1 \cdot x_0$ and $S^2 = \text{Vector}_2 \cdot x_0$. GWN then computes $s_j = \text{Vector}_1 \cdot x_j$ and $f_j = \text{Vector}_2 \cdot x_j$ and picks pair-wise relative positive numbers $k_1, \ldots, k_n$ for each sensing device SDj. GWN computes $\text{Mul}_j = \prod_{j=1}^{n} k_j$ and $\text{Mul}_j = (\text{Mul}/k_j)$. Then, GWN generates a random nonce $\text{Nonce}_j$, which satisfies $\text{Mul}_j \times \text{Nonce}_j \equiv 1 \mod k_j$. GWN calculates $y$ as $y = \sum_{j=1}^{n} \text{Var}_j = \sum_{j=1}^{n} \text{Mul}_j \times \text{Nonce}_j$ and stores it. GWN sends $\langle \text{IDSD}_j, s_j, f_j, k_j \rangle$ to each sensing device.

2.2. User Registration Phase

(1) Step URP1: $U_i$ chooses a high-entropy password $PW_i$ and an identity $ID_i$. $U_i$ imprints the biometrics $B_i$ and uses the generation algorithm to calculate $(\text{BI}_i, v_j) = \text{Gen}(B_i)$. It notes that the algorithm is built into the fuzzy extractor. $U_i$ generates a 128-bit random nonce $a$ and computes $\text{TPW}_i$ as $\text{TPW}_i = h(\text{ID}_i||\text{PW}_i||\text{BI}_i) \oplus a$. Finally, $U_i$ sends a message $\{\text{ID}_i, \text{TPW}_i\}$ to GWN.

(2) Step URP2: after receiving the message $\{\text{ID}_i, \text{TPW}_i\}$, GWN generates a 1024-bit random secret key $\text{KEYGWN}$ and further calculates $\text{KEYGWN}\cdot_{U_i} = h(\text{ID}_i||\text{KEYGWN})$. Then, GWN calculates $A_i$ as $A_i = \text{KEYGWN}\cdot_{U_i}\text{TPW}_i$ and $C_i$ as $C_i = \text{IDGWN}\cdot_{U_i}\text{TPW}_i$. In addition, for each user $U_i$, GWN generates a 128-bit identity $\text{TID}_i$. Finally, GWN generates the smart card $SC_i$ and sends $SC_i$ to $U_i$.

(3) Step URP3: after receiving $SC_i$, in order to protect $A_i$, $U_i$ calculates $\text{RPW}_i = h(\text{ID}_i||\text{PW}_i||\text{BI}_i)$ and $A_i' = A_i \oplus \text{RPW}_i$. $U_i$ then computes $D_i = a \oplus h(\text{ID}_i||\text{BI}_i)$ and $C_i = C_i \oplus \text{RPW}_i \oplus h(\text{ID}_i||\text{BI}_i)$. $U_i$ further calculates $V_i = h(\text{RPW}_i||A_i'||\text{h}(\text{ID}_i||\text{BI}_i)) \mod \omega$. Finally, $U_i$ needs to store $\{\text{TID}_i, A_i', C_i', D_i, V_i, \text{Gen}(\cdot), r_i, \omega, \text{Rep}(\cdot), h(\cdot)\}$ into the memory.

2.3. Authenticated Key Agreement Phase. This phase includes the following steps. This phase along with the login phase is summarized in Table 2.

(1) Step AKAP1: after receiving the message $\{\text{TID}_i, M_1, M_2, TS_i\}$, GWN firstly verifies whether $|TS_i - TS| \leq \Delta T$ to check the freshness of login request. If it is true, GWN obtains $ID_i$ and $\text{KEYGWN}\cdot_{U_i}$ corresponding to $\text{TID}_i$ by retrieving the database. GWN calculates $r_1 = M_1 \oplus \text{KEYGWN}\cdot_{U_i}$. $M_3 = h(TID_i||M_1||\text{IDGWN}||r_1||TS_i)$. In order to authenticate the authenticity of $U_i$, GWN checks whether $M_3$ is equal to $M_2$. If it holds, GWN generates the current timestamp $TS_2$ and a random nonce $r_2 = \text{KEYGWN}\cdot_{U_i}$ $\{r_{GWN}, r_{GWN} \leq \min\{k_j\}\}$, GWN calculates $M_4$ as $M_4 = r_{GWN} \times y$ to securely send $r_{GWN}$ to each sensing device. Then, GWN calculates $M_5 = \text{Enc}_{U_i}(\text{ID}_i||\text{IDGWN}||r_1||r_{GWN}||\text{KEYGWN}\cdot_{U_i})$ to encrypt the parameters. After that, GWN computes $M_6 = h(\text{ID}_i||\text{IDGWN}||r_1||M_4||TS_2\text{VAR}||\text{KEYGWN}\cdot_{U_i})$.

Figure 1: Network model for IIoT.
Table 1: Notations and abbreviations.

| Symbol               | Description                                                                 |
|----------------------|-----------------------------------------------------------------------------|
| GWN                  | Gateway node                                                                |
| ID<sub>GWN</sub>    | GWN's identity                                                              |
| U<sub>i</sub>        | i<sup>th</sup> user and i<sup>th</sup> sensing device                     |
| SD<sub>j</sub>       | U<sub>i</sub>'s and SD<sub>j</sub>'s identity, respectively               |
| r<sub>GWN</sub>      | A random nonce                                                              |
| PW<sub>i</sub>       | U<sub>i</sub>'s password and biometrics                                     |
| B<sub>i</sub>        | U<sub>i</sub>'s biometrics key and public reproduction parameter            |
| Gen(·) and Rep(·)   | Generation and reproduction algorithm of fuzzy extractor, respectively       |
| TS<sub>i</sub>       | Current timestamp                                                           |
| ΔTS                 | Maximum transmission delay                                                  |
| KEY<sub>GWN-U<sub>i</sub></sub> | Symmetric key between U<sub>i</sub> and GWN                              |
| SK                  | Session key between the user and sensing devices                            |
| S                   | Secret value utilized for secret sharing                                    |
| s<sub>i</sub>, f<sub>j</sub>, and k<sub>j</sub> | SD<sub>j</sub>'s secret parameters                                         |
| h(·)                | Hash function                                                               |
| ⊕ and ||          | Concatenation and bit-wise XOR operation                                     |

(1) To help sensing device SD<sub>j</sub> authenticate GWN. Finally, GWN broadcasts the message \{M<sub>4</sub>, M<sub>5</sub>, M<sub>6</sub>, TS<sub>3</sub>\} via a public channel.

(2) Step AKAP2: after receiving the broadcast message from GWN, each sensing device SD<sub>j</sub> verifies |TS<sub>i</sub> - TS<sub>j</sub>| ≤ ΔTS to check the freshness of the message firstly. If the inequality holds, SD<sub>j</sub> uses CRT to obtain r<sup>*</sup><sub>GWN</sub> = M<sub>4</sub> mod k<sub>j</sub> by its stored value k<sub>j</sub>. SD<sub>j</sub> then uses the group key r<sup>*</sup><sub>GWN</sub> to decrypt M<sub>5</sub> to attain the sensitive parameter ID<sub>j</sub>, ID<sub>GWN</sub>, r<sup>*</sup><sub>i</sub>, and r<sub>GWN</sub>\* KEY<sub>GWN-U<sub>i</sub></sub>. SD<sub>j</sub> further calculates M<sub>7</sub> = h(ID<sub>j</sub> || ID<sub>GWN</sub> || r<sup>*</sup><sub>i</sub> || KEY<sub>GWN-U<sub>i</sub></sub> \* r<sub>GWN</sub> \* TS<sub>2</sub>) with the condition M<sub>7</sub> = M<sub>6</sub> to verify GWN. If it is true, each SD<sub>j</sub> computes M<sub>8</sub> = Enc<sub>GWN</sub> (ID<sub>SD<sub>j</sub></sub>, s<sub>j</sub>, f<sub>j</sub>) to encrypt the legal share s<sub>j</sub> and f<sub>j</sub> and generates the current timestamp TS<sub>j</sub>. Then, each sensing device SD<sub>j</sub> sends the reply message \{M<sub>8</sub>, TS<sub>j</sub>\} to GWN securely.

(3) Step AKAP3: when receiving the message, GWN firstly verifies |TS<sub>j</sub> - TS<sub>i</sub>| ≤ ΔTS to check the freshness of the message. If it holds, GWN obtains share s<sub>j</sub> and f<sub>j</sub> by calculating Dec<sub>GWN</sub> (M<sub>8</sub>) = (ID<sub>SD<sub>j</sub></sub>, s<sub>j</sub>, f<sub>j</sub>). GWN further computes \[θ_1 = \sum_{i=1}^{l} λ_i s_i \] and \[θ_2 = \sum_{i=1}^{l} λ_i f_i \] and checks whether \[θ_1 = θ_2 \]. If it holds, GWN can reconstruct the secret successfully. Then, GWN computes M<sub>9</sub> = h(S || r<sub>GWN</sub>), M<sub>10</sub> = M<sub>9</sub> × y<sub>i</sub>, M<sub>11</sub> = h(M<sub>9</sub> \* M<sub>10</sub>), and M<sub>12</sub> = Enc<sub>GWN-U<sub>i</sub></sub> (r<sup>*</sup><sub>GWN</sub> \* r<sub>i</sub> \* M<sub>4</sub>). GWN then generates the current timestamps TS<sub>4</sub> and a new temporal identity TID<sub>4</sub>\* and calculates M<sub>13</sub> = h(ID<sub>i</sub> || KEY<sub>GWN-U<sub>i</sub></sub> || TS<sub>4</sub> \* TID<sub>4</sub>\* and M<sub>14</sub> = h(M<sub>12</sub> || M<sub>4</sub>). Finally, GWN broadcasts the message \{M<sub>10</sub>, M<sub>11</sub>\} to all the participants and sends the message \{M<sub>12</sub>, M<sub>13</sub>, M<sub>14</sub>, TS<sub>4</sub>\} to U<sub>i</sub>.

(4) Step AKAP4: when receiving the message \{M<sub>10</sub>, M<sub>11</sub>\} from GWN, each sensing device SD<sub>j</sub> calculates M<sub>6</sub> = M<sub>10</sub> mod k<sub>j</sub> and M<sub>15</sub> = h(M<sub>9</sub> \* M<sub>10</sub>). If M<sub>15</sub> = M<sub>11</sub>, each device SD<sub>j</sub> calculates SK as SK = h(ID<sub>j</sub> || ID<sub>GWN</sub> || r<sup>*</sup><sub>GWN</sub> \* M<sub>2</sub> || KEY<sub>GWN-U<sub>i</sub></sub>). GWN validates the shared session key by computing M<sub>16</sub> = h(SK || ID<sub>GWN</sub> || ID<sub>j</sub>) and sends it to U<sub>i</sub>.

(5) Step AKAP5: after receiving the message, U<sub>i</sub> firstly verifies |TS<sub>i</sub> - TS<sub>j</sub>| ≤ ΔTS to check the freshness of the message. If it holds, U<sub>i</sub> computes Dec<sub>GWN-U<sub>i</sub></sub> (M<sub>12</sub>) = (r<sup>*</sup><sub>GWN</sub> \* r<sub>i</sub> \* M<sub>9</sub>). Then, U<sub>i</sub> checks whether r<sup>*</sup><sub>GWN</sub> \* r<sub>i</sub> to validate the session consistency. If it holds, U<sub>i</sub> computes M<sub>17</sub> = h(M<sub>12</sub> || M<sub>9</sub>). If M<sub>17</sub> = M<sub>14</sub>, U<sub>i</sub> further calculates SK* as SK* = h(ID<sub>i</sub> || ID<sub>GWN</sub> || r<sup>*</sup><sub>GWN</sub> \* M<sub>9</sub> || KEY<sub>GWN-U<sub>i</sub></sub>) with sensing devices. U<sub>i</sub> calculates M<sub>18</sub> = h(SK* || ID<sub>GWN</sub> || ID<sub>j</sub>) after receiving the message from the sensing device and checks whether M<sub>18</sub> = M<sub>16</sub>. If it holds, U<sub>i</sub> needs to change TID<sub>4</sub>\* = h(ID<sub>i</sub> || KEY<sub>GWN-U<sub>i</sub></sub> || TS<sub>4</sub> \* TID<sub>4</sub>\* and M<sub>13</sub>.

3. Cryptanalysis of Vinoth et al.’s Scheme

For a multifactor authentication scheme, it is essential to create a concise and concrete adversarial model. In this section, we propose two attacks, a smart loss attack and a sensing device capture attack to show the vulnerabilities of the scheme. First of all, we refer to the adversary model proposed by Wang et al. [9] which is strict but reasonable. The assumptions below are about the adversary’s capabilities:

(1) There exist two kinds of communication channels: a secure channel and a public channel. The former is mainly used for registration, while the other is mainly used in login and authentication phases. The adversary \*\* can has full control of the public channel, i.e., \*\* can eavesdrop, intercept, modify, and redirect messages transmitted between communication participants [10, 11].
Table 2: Login and authenticated key agreement phase.

| User (U_i) | Gateway (GWN) | Sensing device (SD_j) |
|------------|--------------|-----------------------|
| L1 Insert SC_i to card reader. Input ID_i, PW_i, and B_i. Compute BK_i = Rep(B_i, r_i, RPW_i = h(ID_i, PW_i, BK_i), ′ a_i = D_i \oplus h(1D_i, BK_i), A_i = A_i \oplus a_i, and V_i = h(RPW_i, A_i, B_i, h(1D_i, BK_i)) \mod \omega Check whether V_i = V'_i; if so, generate a random nonce r_i and timestamps T_{S_i}, Calculate, |
| | M_2 = h(TID_i, | Check whether | |
| | M_2 \rightarrow \text{[open channel]} \langle M_{4i}, M_{7i}, M_{x}, T_{S_1} \rangle \checkmark | T_{S_1} = T_{S_i} \leq \Delta T \$ if so, compute \( r_{GWN}^* = r_i \); if so, compute M_{13} = h(M_{12}, M_{4i}, M_{7i}, T_{S_1}) \checkmark |
| | M_4 = h(DID_{GWN}, r_i, M_{4i}, T_{S_1}) \checkmark | Check whether | |
| | \text{[open channel]} \langle M_{4i}, M_{7i}, M_{x}, T_{S_2} \rangle \checkmark | T_{S_2} = T_{S_i} \leq \Delta T \$ if so, compute (ID_{ID}, S_i, f_i) = Dec_{GWN}(M_4, \theta_1 = \sum_{i \in I} \lambda_i f_i, and \theta_2 = \sum_{i \in I} \lambda_i f_i. Check whether \theta_1 = \theta_2; if so, return \theta_1 as S. Compute M_5 = h(S_{GWN}), M_{10} = M_{I_1}, and M_{11} = h(M_{I_1}). Check if M_{15} = M_{11}; if so, compute SK = h(ID_{ID}, TID_{GWN}, T_{S_1}, TID_{GWN}. |
| | V_3 Generate a temporal identity TID_{GWN} and timestamp T_{S_1}. Compute M_{12} = Enc_{KEY_{GWN}}(r_{GWN}^*, M_{12}), M_{13} = h(ID_{ID}, TID_{GWN}, T_{S_1}, TID_{GWN}, and M_{14} = h(M_{13}, r_{GWN}^*), | |
| | \text{[open channel]} \langle M_{10}, M_{14} \rangle \checkmark | Check whether | |
| | \text{[open channel]} \langle M_{12}, M_{13}, M_{14}, T_{S_1} \rangle \checkmark | T_{S_2} - T_{S_1} \leq \Delta T. If so, compute | |
(2) The adversary $\mathcal{A}$ can offline exhaust all the items in the Descartes space of identities and passwords which are of low entropy within polynomial time.

(3) When it comes to multifactor authentication, the scheme should be secure even if one or more factors are compromised, which is called truly multifactor security [12]. Therefore, it is reasonable to make an assumption that $\mathcal{A}$ may (i) obtain a victim’s password by performing shoulder surfing or phishing attacks, (ii) extract the secret parameters in the lost smart card by performing side-channel attack, or (iii) attain a victim’s biometric information using malicious devices. However, the above assumptions cannot be achieved at the same time; otherwise, it will be a trivial case.

(4) The adversary $\mathcal{A}$ could be the administrator of the server or a legitimate user in the system.

(5) The adversary $\mathcal{A}$ can determine victim’s identity.

It is worth noting that users can select his/her identity ID and password PW in many protocols. However, the user selected identities and passwords are usually of low entropy ($|D| \leq |P| \leq 10^6$) [13, 14]. Therefore, assumption (2) is realistic. Then, assumption (3) specifies truly three-factor security. And, assumption (4) can be used to capture the threats from the system when the server is corrupted or any legitimate users are malicious. Finally, assumption (5) describes the fact that most of the user identity are user’s e-mail addresses or phone numbers, which can be easily obtained.

The following analysis will take the five assumptions mentioned above into account.

### 3.1. Smart Card Loss Attack

We employ the user $U_i$ as the victim to show the process of this attack. According to assumption (3), it is reasonable for the adversary $\mathcal{A}$ to get $U_i$’s smart card SC (stolen or picked up) and corresponding biometrics $B_i'$. Besides, as a premeditated adversary, $\mathcal{A}$ has full control of the public channel, and she can collect a past transcript between $U_i$ and gateway node (GWN) (i.e., $\{\text{TID}_1, M_1, M_2, \text{TS}_1\}$). Then, $\mathcal{A}$ can guess $U_i$’s password and identity correctly as following steps:

1. **Step 1.** $\mathcal{A}$ computes $\text{BK}_i = \text{Rep}(B_i', \tau_i)$, where $\tau_i$ can be extracted from victim’s smart card.
2. **Step 2.** $\mathcal{A}$ chooses a pair $(\text{ID}_i^*, \text{PW}_i^*)$ from $D \times P$, where $D$ denotes the identity space and $P$ denotes the password space.
3. **Step 3.** $\mathcal{A}$ computes $\text{ID}_i^{**} = \mathcal{C}_{\mathcal{A}}(\text{ID}_i^* \oplus \text{BK}_i)$.
4. **Step 4.** $\mathcal{A}$ computes $\text{RPW}_i^* = h(\text{ID}_i^* \oplus \text{PW}_i^* \oplus \text{BK}_i)$.
5. **Step 5.** $\mathcal{A}$ computes $r_i^* = M_i \oplus A_i^* \oplus \text{RPW}_i^*$, noted that $\mathcal{A}$ can extract $A_i^*$ from victim’s smart card and collect $M_i$ from the past transcript.
6. **Step 6.** $\mathcal{A}$ computes $M_2^* = h(\text{TID}_i \oplus \text{ID}_i^{**} \oplus r_i^* \oplus \text{TS}_1)$ and verifies the correctness of $(\text{ID}_i^*, \text{PW}_i^*)$ pair by checking if $M_2^* = M_2$.

As mentioned before, users can choose his/her own ID and PW in most password-based authentication schemes (e.g., References [15–17]) aiming to achieve user-friendliness. And, Vinoth et al.’s scheme is no exception. It makes assumption (2) reasonable that users often select low entropy identities and passwords. Therefore, it is possible for $\mathcal{A}$ to exhaust all the (ID, PW) pairs offline within polynomial time. We can calculate the running time of the attack procedure as $O(3T_H \times |D| \times |P|)$, where $|D|$ represents the number of identities, $|P|$ represents the number of passwords, and $T_H$ represents the running time for Hash operation. Note that the operation time of bit-wise XOR operation in Step 3 can be ignored. Since $|D|$ and $|P|$ are very limited (e.g., $|D| \leq |P| \leq 10^6$) [13, 14], the attack mentioned above is significant and shows a challenge to user authentication protocols.

### 3.2. Sensing Device Capture Attack

According to Vinoth et al.’s threat model, the adversary $\mathcal{A}$ can compromise a sensing device (SD) and extract the parameters stored in it (i.e., $\{\text{ID}_{SD}, s_i, f_i, k_i, a_i\}$). We assume that SD$_j$ is captured by the adversary; then, $\mathcal{A}$ can successfully impersonate the user $U_j$ as follows:

1. **Step 1.** Computes $r_{GWN}^* = M_4 \mod k_j$, where $M_4$ is received from (GWN).
2. **Step 2.** Decrypts the received message $M_5$ by using the key $r_{GWN}^*$ and obtains the security parameters $(\text{ID}_{GWN} \oplus \text{KEY}_{GWN-U_i}, \text{ID}_{GWN} \oplus \text{ID}_1)$, and $r_1^*$ of the user $U_i$ who is sending the login request.
3. **Step 3.** Computes $\text{KEY}_{GWN-U_i} = \text{KEY}_{GWN-U_i} \oplus r_{GWN}^* \oplus \text{KEY}_{GWN-U_i}$.
4. **Step 4.** Computes $\text{TID}_i^{\text{new}} = h(\text{ID}_i \oplus \text{KEY}_{GWN-U_i} \oplus \text{TS}_1) \oplus M_{13}$, where $\text{TS}_4$ and $M_{13}$ are obtained from the public channel.
5. **Step 5.** Randomly chooses a new nonce $r_{\text{new}}^*$ and current timestamp $\text{TS}_1^{\text{new}}$.
6. **Step 6.** Computes $M_1^* = \text{KEY}_{GWN-U_i} \oplus r_{\text{new}}^*$ and $M_2^* = h(\text{TID}_i^{\text{new}} \oplus M_1^* \oplus \text{ID}_{GWN} \oplus r_{\text{new}}^* \oplus \text{TS}_1^{\text{new}})$.
7. **Step 7.** Sends the login request $(\text{TID}_i^{\text{new}}, M_1^*, M_2^*, \text{TS}_1^{\text{new}})$ to (GWN) and finishes the login phase.

After receiving the message, GWN first checks the freshness of the received message and computes $r_{\text{new}}^* = \text{KEY}_{GWN-U_i} \oplus M_1^*$, where KEY$_{GWN-U_i}$ is stored in the GWN’s database and retrieved according to corresponding TID. Then, GWN computes $M_3 = h(\text{TID}^{\text{new}} \oplus M_1^* \oplus \text{ID}_{GWN} \oplus r_{\text{new}}^* \oplus \text{TS}_1^{\text{new}})$ and verifies whether the calculated $M_3$ is equal to the received $M_2$. If it holds, GWN will authenticate the authenticity of $U_i$. Since the parameters are calculated correctly, the adversary $\mathcal{A}$ can pass the verification of the GWN. So far, the adversary has successfully impersonated user $U_j$.

### 3.3. No Forward Secrecy

When a scheme ensures that, even the long-term private keys (or secret) of communication participants are leaked, previously agreed session keys can
still be secure [18], then the scheme is called supporting forward secrecy. It is important for security critical systems to support forward secrecy, especially when there still exist many security and privacy problems in the IIoT environment.

If an attacker $\mathcal{A}$ has captured a sensing device $SD_j$, extracted the parameters $\{ID_{SD}, s_t, f_t, k_t\}$ from $SD_j$, and intercepted the messages $\{M_4, M_5, M_{10}\}$, the following method can be used to calculate the session key:

1. $\mathcal{A}$ computes $r_{GWN}^j = M_4 \mod k_j$, where $M_4$ is received from (GWN)
   
2. $\mathcal{A}$ decrypts the received message $M_5$ by using the key $r_{GWN}^j$ and obtains the security parameters $(r_{GWN}^j, ID_j, r_1^j, \text{and ID}_{GWN})$ of the user $U_i$
   
3. $\mathcal{A}$ computes $M_5^* = M_{10} \mod k_j$, where $M_{10}$ is received from (GWN)
   
4. $\mathcal{A}$ computes the session key $SK^* = h(ID_i \| ID_{GWN} \| r_{GWN}^j \| M_5^* \| \text{KEY}_{GWN-U_j})$

With the session key $SK^*$ computed, the entire session will be no secret to the adversary $\mathcal{A}$.

\section{Security Vulnerability Discussion}

In this section, we highlight again that when considering multifactor security, even if one or more authentication factors are obtained (not all) by the adversary, the scheme should not be broken. Based on this assumption, we proposed the smart card loss attack and the sensing device capture attack. Although Vinoth et al. have employed the fuzzy-verify technique proposed by Wang and Wang [12], the adversary can still obtain victim’s password in the way of offline guessing. This disappointing situation is caused that they do not employ the public-key cryptosystem and no public key material is used to construct the login message. To solve this problem, we suggest to use Diffie–Hellman key exchange scheme. Specifically, GWN computes $y = g^x \mod p$ and stores it into $U_i$’s smart card SC during the user registration phase, where $g$ is a generator of the group $G$, $p$ is a large prime number, and $x$ is GWN’s secret key. After that, when $U_i$ logs in, she chooses a random number $u$ and compute $y_1 = g^u \mod p$ and $C_1 = y_1 \mod p$ first; then, she constructs the login message $M_3 = h(TID_i \| ID_{GWN} \| r_1 \| C_1 \| Y_i \| T_{S_j})$. Since the adversary cannot calculate $Y_i$, the aforementioned smart card loss attack can be prevented.

Meanwhile, in the sensing device capture attack, an adversary $\mathcal{A}$ can impersonate user $U_i$ even without her password $PW_i$. Essentially, when authenticating the identity of user $U_i$, the GWN only checks whether the user $U_i$ who sends the login request holds the parameter $\text{KEY}_{\text{GWN-U}_j}$. Unfortunately, $\text{KEY}_{\text{GWN-U}_j}$ is encrypted by the group key $r_{GWN}^j$ in the message $M_5$, but obtaining the group key $r_{GWN}^j$ is easy for an adversary who has breached the SD and extracted $k_j$. After this, $\mathcal{A}$ could decrypt $M_5$ to get $\text{KEY}_{\text{GWN-U}_j}$, $ID_j$, $r_1^j$, and $\text{ID}_{GWN}$. With these parameters, $\mathcal{A}$ can bypass the system’s user authentication. One possible countermeasure to this problem is that GWN constructs the message $M_5 = \text{Enc}_{r_{GWN}^j}(ID_i, ID_{GWN}, \text{KEY}_{\text{GWN-U}_j} \| r_1^j)$ and $M_6 = h(ID_{GWN} \| ID_{GWN} \| \text{KEY}_{\text{GWN-U}_j} \| T_{S_j})$. As a result, when receiving the message, $\{M_4, M_5, M_6, T_{S_j}\}$, the adversary $\mathcal{A}$ who captures the sensing device can only obtain the parameter $\text{KEY}_{\text{GWN-U}_j} \| r_1^j$ by decrypting $M_5$. Therefore, $\mathcal{A}$ cannot impersonate $U_i$ since she cannot obtain $\text{KEY}_{\text{GWN-U}_j}$.

Note that, one may argue that when the victim user $U_i$ interacts with the GWN, she uses her temporary identity $TID_i$, and it seems impossible for an adversary to find victim’s message from the transcript. However, according to assumption (5), the adversary $\mathcal{A}$ can determine victim’s identity. Thus, a premeditated adversary may first compromise a sensing device and wait for the victim chosen by her to send the login request message. Then, $\mathcal{A}$ decrypts $M_5$ to get $ID$ and checks if this ID belongs to the victim. After that, $\mathcal{A}$ continues to monitor the channel until $U_i$’s session ends. Finally, $\mathcal{A}$ could calculate $TID_i^{\text{new}}$ as Step 4 of the sensing device capture attack.

In order to fix the defects of forward secrecy, we also rely on public key cryptography. Specifically, before computing $M_4$, $SD_j$ first chooses a random number $r_1$ and computes $Y_2 = g^{r_2} \mod p$. Then, $SD_j$ computes $M_8 = \text{Enc}_{GWN}(ID_{SD_j}, S_j, f_j, Y_2)$ and sends it to GWN. After that, GWN chooses a random number $x'$ and computes $C_2 = Y_2 S_j^{x'} \mod p$, $Y_{GWN-S_j} = g^{x'} \mod p$, $M_{10} = \text{Enc}_{C_2}(M_9)$, and $M_{11} = h(M_{10} \| M_9 \| Y_{GWN-S_j})$. In Section 3.4, we show that the adversary $\mathcal{A}$ can compute the session key $SK$, and this is caused by $\mathcal{A}$ to obtain $M_9$. However, $M_9$ is protected by the shared key $C_2$ now. As a result, $\mathcal{A}$ cannot calculate the previous session key. In order to be consistent with the previous modification, the session key is calculated as $SK = h(ID_{GWN} \| r_{GWN}^j \| M_5^* \| \text{KEY}_{\text{GWN-U}_j} \| r_1^j)$.

\section{4. Conclusion}

In this paper, we have revisited and analysed Vinoth et al.’s authentication scheme for IIoT environments. We demonstrated that their scheme suffers from the smart card loss attack and the sensing device capture attack although they claimed that their scheme has the ability to defend various known attacks. We have also briefly discussed the potential causes of these defects. It is hoped that the proposed attacks can help inspire new designs of secure and efficient multifactor authentication protocols for IIoT.

\section{Data Availability}

Data sharing is not applicable to this article as no new data was created or analysed in this study.

\section{Conflicts of Interest}

The authors declare that they have no conflicts of interest.

\section{References}

[1] Z. Meng, Z. Wu, C. Muvianto, and J. Gray, “A data-oriented M2M messaging mechanism for industrial IoT applications,”
IEEE Internet of Things Journal, vol. 4, no. 1, pp. 236–246, 2017.

[2] C. Yin, J. Xi, R. Sun, and J. Wang, “Location privacy protection based on differential privacy strategy for big data in industrial internet of things,” IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3628–3636, 2018.

[3] W. Z. Khan, M. H. U. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, “Industrial internet of things: recent advances, enabling technologies and open challenges,” Computers & Electrical Engineering, vol. 81, Article ID 106522, 2020.

[4] W. Ali, I. Ud Din, A. Almogren, M. Guizani, and M. Zuaier, “A lightweight privacy-aware IoT-based metering scheme for smart industrial ecosystems,” IEEE Transactions on Industrial Informatics, vol. 9, p. 1, 2020.

[5] J. Seto, Y. Wang, and X. Lin, “User-habit-oriented authentication model: toward secure, user-friendly authentication for mobile devices,” IEEE Transactions on Emerging Topics in Computing, vol. 3, no. 1, pp. 107–118, 2015.

[6] C. Wang, L. Xiao, J. Shen, and R. Huang, “Neighborhood trustworthiness-based vehicle-to-vehicle authentication scheme for vehicular ad hoc networks,” Concurrency and Computation: Practice & Experience, vol. 31, no. 21, Article ID e4643, 2019.

[7] J. Shen, T. Zhou, X. Liu, and Y.-C. Chang, “A novel Latin-square-based secret sharing for M2M communications,” IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3659–3668, 2018.

[8] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, “Secure multi-factor authenticated key agreement scheme for industrial IoT,” IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3801–3811, 2020.

[9] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, “Understanding node capture attacks in user authentication schemes for wireless sensor networks,” IEEE Transactions on Dependable and Secure Computing, p. 1, 2020.

[10] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K.-K. R. Choo, “Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks,” Computer Networks, vol. 129, pp. 429–443, 2017.

[11] D. He, S. Zeadally, L. Wu, and H. Wang, “Analysis of handover authentication protocols for mobile wireless networks using identity-based public key cryptography,” Computer Networks, vol. 128, pp. 154–163, 2017.

[12] D. Wang and P. Wang, “Two birds with one stone: two-factor authentication with security beyond conventional bound,” IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 4, pp. 708–722, 2018.

[13] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, “Targeted online password guessing: an underestimated threat,” in CCS’16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1242–1254, Vienna, Austria, October 2016.

[14] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, “Zipf’s law in passwords,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 11, pp. 2776–2791, 2017.

[15] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, “A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment,” Future Generation Computer Systems, vol. 78, pp. 1005–1019, 2018.

[16] F. Wu, X. Li, A. K. Sangaiah et al., “A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks,” Future Generation Computer Systems, vol. 82, pp. 727–737, 2018.