Research Article

Revisiting a Multifactor Authentication Scheme in Industrial IoT

Ding Wang ^{1,2}, **Shuhong Hong** ^{1,2} and **Qingxuan Wang** ^{1,2}

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

Academic Editor: AnMin Fu

Copyright © 2021 Ding Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, as one of the key applications of Internet of Things, Industry IoT (IIoT) has recently received significant attention and has facilitated our life. In IIoT environments, an amount of data generally requires to be transmitted between the user and sensing devices in an open channel. In order to ensure safe transmission of these data, it is necessary for the user and sensing devices to authenticate each other and establish a secure channel between them. Recently, a multifactor authenticated key agreement scheme for IIoT was proposed, which aims to tackle this problem and provide solutions for user multiple sensing devices' access. This work claims that the proposed scheme is secure against various attacks and has less communication and computational costs than other existing related schemes. Unfortunately, we find that this scheme cannot resist smart card attack and sensing device capture attack. Furthermore, we show that this scheme fails to provide forward secrecy, which is essential for a secure multifactor authentication scheme.

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

FIGURE 1: Network model for IIoT.

designed by Vinoth et al., we do not think it can provide truly multifactor security.

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 SD_j is registered by GWN in offline and is distributed a unique identity ID_{SD_j} . In order to calculate the secret, GWN chooses a secret value S and two vectors $Vector_1$ and $Vector_2$. Assume that $S = Vector_1 \cdot x_0$ and $S^2 = Vector_2 \cdot x_0$. GWN then computes $s_j = Vector_1 \cdot x_j$ and $f_j = Vector_2 \cdot x_j$ and picks pair-wise relative positive numbers $k_1, \dots, k_n$ for each sensing device SD_j . GWN computes $Mul = \prod_{j=1}^n k_j$ and $Mul_j = (Mul/k_j)$. Then, GWN generates a random nonce $Nonce_j$, which satisfies $Mul_j \times Nonce_j \equiv 1 \pmod{k_j}$. GWN calculates γ as $\gamma = \sum_{j=1}^n Var_j = \sum_{j=1}^n Mul_j \times Nonce_j$ and stores it. GWN sends $\langle ID_{SD_j}, s_j, f_j, k_j \rangle$ 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 $(BK_i, \tau_i) = Gen(B_i)$. It notes that the algorithm is built into the fuzzy extractor. U_i generates a 128-bit random nonce a and computes TPW_i as $TPW_i = h(ID_i \| PW_i \| BK_i) \oplus a$. Finally, U_i sends a message $\{ID_i, TPW_i\}$ to GWN.

- (2) Step URP2: after receiving the message $\{ID_i, TPW_i\}$, GWN generates a 1024-bit random secret key KEY_{GWN} and further calculates $KEY_{GWN-U_i} = h(ID_i \| KEY_{GWN})$. Then, GWN calculates A_i as $A_i = KEY_{GWN-U_i} \oplus TPW_i$ and C_i as $C_i = ID_{GWN} \oplus TPW_i$. In addition, for each user U_i , GWN generates a 128-bit identity 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 $RPW_i = h(ID_i \| PW_i \| BK_i)$ and $A'_i = A_i \oplus TPW_i \oplus RPW_i$. U_i then computes $D_i = a \oplus h(ID_i \| BK_i)$ and $C'_i = C_i \oplus TPW_i \oplus h(ID_i \| BK_i)$. U_i further calculates $V_i = h(RPW_i \| A'_i \| a \| h(ID_i \| BK_i)) \pmod{\omega}$. Finally, U_i needs to store $\{TID_i, A'_i, C'_i, D_i, V_i, Gen(\cdot), \tau_i, \omega, Rep(\cdot), h(\cdot)\}$ 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 $\{TID_i, M_1, M_2, TS_1\}$, GWN firstly verifies whether $|TS_1 - TS'_1| \leq \Delta TS$ to check the freshness of login request. If it is true, GWN obtains ID_i and KEY_{GWN-U_i} corresponding to TID_i by retrieving the database. GWN calculates $r_i^* = M_1 \oplus KEY_{GWN-U_i}$, $M_3 = h(TID_i \| M_1 \| ID_{GWN} \| r_i^* \| TS_1)$. 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 $r_{GWN} (r_{GWN} \leq \min\{k_j\})$. GWN calculates M_4 as $M_4 = r_{GWN} \times \gamma$ to securely send r_{GWN} to each sensing device. Then, GWN calculates $M_5 = Enc_{r_{GWN}}(ID_i, ID_{GWN}, r_i^*, r_{GWN} \oplus KEY_{GWN-U_i})$ to encrypt the parameters. After that, GWN computes $M_6 = h(ID_i \| ID_{GWN} \| r_i^* \| M_4 \| TS_2 \text{Vert} \| KEY_{GWN-U_i})$

TABLE 1: Notations and abbreviations.

Symbol	Description
GWN	Gateway node
ID_{GWN}	GWN's identity
U_i and SD_j	i^{th} user and j^{th} sensing device
ID_i and ID_{SD_j}	U_i 's and SD_j 's identity, respectively
r_{GWN}	A random nonce
PW_i and B_i	U_i 's password and biometrics
BK_i and τ_i	U_i 's biometrics key and public reproduction parameter
$Gen(\cdot)$ and $Rep(\cdot)$	Generation and reproduction algorithm of fuzzy extractor, respectively
TS_i	Current timestamp
ΔTS	Maximum transmission delay
KEY_{GWN-U_i}	Symmetric key between U_i and GWN
SK	Session key between the user and sensing devices
S	Secret value utilized for secret sharing
s_j, f_j , and k_j	SD_j 's secret parameters
$h(\cdot)$	Hash function
$\oplus$ and $\parallel$	Concatenation and bit-wise XOR operation

to help sensing device SD_j authenticate GWN. Finally, GWN broadcasts the message $\{M_4, M_5, M_6, TS_2\}$ via a public channel.

- (2) Step AKAP2: after receiving the broadcast message from GWN, each sensing device SD_j verifies $|TS_2 - TS'_2| \leq \Delta TS$ to check the freshness of the message firstly. If the inequality holds, SD_j uses CRT to obtain $r_{GWN}^* = M_4 \bmod k_j$ by its stored value k_j . SD_j then uses the group key r_{GWN}^* to decrypt M_5 to attain the sensitive parameter ID_i, ID_{GWN}, r_i^* , and $r_{GWN} \oplus KEY_{GWN-U_i}$. SD_j further calculates $M_7 = h(ID_i \parallel ID_{GWN} \parallel r_i^* \parallel M_4 \parallel KEY_{GWN-U_i} \oplus r_{GWN} \oplus r_{GWN}^* \parallel TS_2)$ with the condition $M_7 = M_6$ to verify GWN. If it is true, each SD_j computes $M_8 = Enc_{r_{GWN}^*}(ID_{SD_j}, s_j f_j)$ to encrypt the legal share s_j and f_j and generates the current timestamp TS_3 . Then, each sensing device SD_j sends the reply message $\{M_8, TS_3\}$ to GWN securely.
- (3) Step AKAP3: when receiving the message, GWN firstly verifies $|TS_3 - TS'_3| \leq \Delta TS$ to check the freshness of the message. If it holds, GWN obtains share s_j and f_j by calculating $Dec_{r_{GWN}^*}(M_8) = (ID_{SD_j}, s_j, f_j)$. GWN further computes $\theta_1 = \sum_{t=1}^l \lambda_t s_t$ and $\theta_2 = \sum_{t=1}^l \lambda_t f_t$ and checks whether $\theta_1^2 = \theta_2$. If it holds, GWN can reconstruct the secret successfully. Then, GWN computes $M_9 = h(S \parallel r_{GWN})$, $M_{10} = M_9 \times \gamma$, $M_{11} = h(M_9 \parallel M_{10})$, and $M_{12} = Enc_{KEY_{GWN-U_i}}(r_{GWN}, r_i^*, M_9)$. GWN then generates the current timestamps TS_4 and a new temporal identity TID_i^{new} and calculates $M_{13} = h(ID_i \parallel KEY_{GWN-U_i} \parallel TS_4) \oplus TID_i^{new}$ and $M_{14} = h(M_{12} \parallel M_9 \parallel r_i^*)$. Finally, GWN broadcasts the message $\{M_{10}, M_{11}\}$ to all the participants and sends the message $\{M_{12}, M_{13}, M_{14}, TS_4\}$ to U_i .
- (4) Step AKAP4: when receiving the message $\{M_{10}, M_{11}\}$ from GWN, each sensing device SD_j

calculates $M_9^* = M_{10} \bmod k_j$ and $M_{15} = h(M_9^* \parallel M_{10})$. If $M_{15} = M_{11}$, each device SD_j calculates SK as $SK = h(ID_i \parallel ID_{GWN} \parallel r_{GWN}^* \parallel r_i^* \parallel M_9^* \parallel KEY_{GWN-U_i})$. GWN validates the shared session key by computing $M_{16} = h(SK \parallel ID_{GWN} \parallel ID_i)$ and sends it to U_i .

- (5) Step AKAP5: after receiving the message, U_i firstly verifies $|TS_4 - TS'_4| \leq \Delta TS$ to check the freshness of the message. If it holds, U_i computes $Dec_{KEY_{GWN-U_i}}(M_{12}) = (r_{GWN}^*, r_i^*, M_9^*)$. Then, U_i checks whether $r_i^* = r_i$ to validate the session consistency. If it holds, U_i computes $M_{17} = h(M_{12} \parallel M_9^*)$. If $M_{17} = M_{14}$, U_i further calculates SK^* as $SK^* = h(ID_i \parallel ID_{GWN} \parallel r_{GWN}^* \parallel r_i^* \parallel M_9^* \parallel KEY_{GWN-U_i})$ with sensing devices. U_i calculates $M_{18} = h(SK^* \parallel ID_{GWN} \parallel ID_i)$ after receiving the message from the sensing device and checks whether $M_{18} = M_{16}$. If it holds, U_i needs to change $TID_i^{new} = h(ID_i \parallel KEY_{GWN-U_i} \parallel TS_4) \oplus M_{13}$.

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 $\mathcal{A}$ has full control of the public channel, i.e., $\mathcal{A}$ 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 $_j$)
	<p>Insert SC_i to card reader. Input ID_i, PW_i, and B_i^*.</p> <p>Compute $BK_i^* = \text{Rep}(B_i^*, \tau_i)$, $RPW_i^* = h(ID_i \ PW_i \ BK_i^*)$, $a^* = D_i \oplus h(ID_i \ BK_i^*)$, $A_i^* = A_i' \oplus a^*$, and $V_i^* = h(RPW_i^* \ A_i^* \ a^* \ h(ID_i \ BK_i^*)) \bmod \omega$</p> <p>Check whether $V_i = V_i^*$, if so, generate a random nonce r_i and timestamps TS_1. Calculate, $ID_{GWN}^* = C_i \oplus h(ID_i \ BK_i^*)$, $M_1 = A_i \oplus RPW_i^* \oplus r_i$, $M_2 = h(TID_i \ M_1 \ ID_{GWN}^* \ r_i \ TS_1)$ $\rightarrow [\text{open channel}] \langle M_4, M_5, M_6, TS_2 \rangle$</p> <p>Check whether $TS_4 - TS_1 \leq \Delta TS$ if so, compute $(r_{GWN}^*, r_i^*, M_9^*) = \text{Dec}_{\text{KEY}_{GWN-U_i}}(M_{12})$; check whether $r_i^* = r_i$; if so, compute $M_{17} = h(M_{12} \ M_9^* \ r_i)$. Check whether $M_{17} = M_{14}$; if so, compute $SK^* = h(ID_i \ ID_{GWN} \ r_{GWN}^* \ r_i \ M_9^* \ \text{KEY}_{GWN-U_i})$ and $M_{18} = h(SK^* \ ID_{GWN} \ ID_i)$. Check whether $M_{16} = M_{18}$; if so, the session key is established successfully. Compute $TID_i^{\text{new}} = h(ID_i \ \text{KEY}_{GWN-U_i} \ TS_4) \oplus M_{13}$. Update TID_i with TID_i^{new}</p>	<p>Check whether $TS_1 - TS_3 \leq \Delta TS$. If so, extract ID_i and KEY_{GWN-U_i}. Compute $r_i^* = M_1 \oplus \text{KEY}_{GWN-U_i}$ and $M_3 = h(TID_i \ M_1 \ ID_{GWN} \ r_i^* \ TS_1)$. Check whether $M_3 = M_2$; if so, generate a random nonce r_{GWN} and timestamps TS_2. Compute $M_4 = r_{GWN} \times \gamma$, $M_5 = \text{Enc}_{r_{GWN}}(ID_i, ID_{GWN}^*, r_i^*, r_{GWN} \oplus \text{KEY}_{GWN-U_i})$, and $M_6 = h(ID_i \ ID_{GWN} \ r_i^* \ M_4 \ \text{KEY}_{GWN-U_i} \ TS_2)$, $\rightarrow [\text{open channel}] \langle M_4, M_5, M_6, TS_2 \rangle$</p> <p>Check whether $TS_3 - TS_3' \leq \Delta TS$; if so, compute $(ID_{SD_j}, s_j, f_j) = \text{Dec}_{r_{GWN}}(M_8)$, $\theta_1 = \sum_{i=1}^I \lambda_i s_i$, and $\theta_2 = \sum_{i=1}^I \lambda_i f_i$. Check whether $\theta_2^1 = \theta_2$; if so, return θ_1 as S. Compute $M_9 = h(S \ r_{GWN})$, $M_{10} = M_9 \times \gamma$, and $M_{11} = h(M_9 \ M_{10})$.</p> <p>Generate a temporal identity TID_i^{new} and timestamp TS_4. Compute $M_{12} = \text{Enc}_{\text{KEY}_{GWN-U_i}}(r_{GWN}, r_i^*, M_9)$, $M_{13} = h(ID_i \ \text{KEY}_{GWN-U_i} \ TS_4 \oplus TID_i^{\text{new}}$, and $M_{14} = h(M_{12} \ M_9 \ r_i^*)$. $\rightarrow [\text{open channel}] \langle M_{10}, M_{11} \rangle \xleftarrow{[\text{open channel}]} \langle M_{12}, M_{13}, M_{14}, TS_4 \rangle$</p>	<p>Check whether $TS_2 - TS_2' \leq \Delta TS$. If so, compute $r_{GWN}^* = M_4 \bmod k_j$, $(ID_i, ID_{GWN}, r_i^*, r_{GWN} \oplus \text{KEY}_{GWN-U_i}) = \text{Dec}_{r_{GWN}}(M_5)$, and $M_7 = h(ID_i \ ID_{GWN} \ r_i^* \ M_4 \ \text{KEY}_{GWN-U_i} \oplus r_{GWN} \oplus r_{GWN}^* \ TS_2)$ Check whether $M_7 = M_6$; if so, generate a timestamp TS_3. Compute $M_8 = \text{Enc}_{r_{GWN}}(ID_{SD_j}, s_j, f_j)$, $\xleftarrow{[\text{open channel}]} \langle M_8, TS_3 \rangle$</p> <p>Compute $M_9^* = M_{10}$ and $M_{15} = h(M_9^* \ M_{10})$. Check if $M_{15} = M_{11}$; if so, compute $SK = h(ID_i \ ID_{GWN} \ r_{GWN}^* \ r_i^* \ M_9^* \ \text{KEY}_{GWN-U_i})$ and $M_{16} = h(SK \ ID_{GWN} \ ID_i) \xleftarrow{[\text{open channel}]} \langle M_{16} \rangle$</p>

- (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 ($|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 10^6$) [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., $\{TID_i, M_1, M_2, TS_1\}$). Then, $\mathcal{A}$ can guess U_i 's password and identity correctly as following steps:

Step 1. $\mathcal{A}$ computes $BK_i = \text{Rep}(B_i^*, \tau_i)$, where τ_i can be extracted from victim's smart card

Step 2. $\mathcal{A}$ chooses a pair (ID_i^*, PW_i^*) from $\mathcal{D}_{id} \times \mathcal{D}_{pw}$, where $\mathcal{D}_{id}$ denotes the identity space and $\mathcal{D}_{pw}$ denotes the password space

Step 3. $\mathcal{A}$ computes $ID_{GWN}^* = C_i' \oplus h(ID_i^* \| BK_i)$

Step 4. $\mathcal{A}$ computes $RPW_i^* = h(ID_i^* \| PW_i^* \| BK_i)$

Step 5. $\mathcal{A}$ computes $r_i^* = M_1 \oplus A_i' \oplus RPW_i^*$, noted that $\mathcal{A}$ can extract A_i' from victim's smart card and collect M_1 from the past transcript

Step 6. $\mathcal{A}$ computes $M_2^* = h(TID_i \| M_i \| ID_{GWN}^* \| r_i^* \| TS_1)$ and verifies the correctness of (ID_i^*, PW_i^*) pair by checking if $M_2^* = M_2$

Step 7. $\mathcal{A}$ executes the steps 2 ~ 6 repeatedly until finding the correct values

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 $\mathcal{O}(3T_H \times |\mathcal{D}_{id}| \times |\mathcal{D}_{pw}|)$, where $|\mathcal{D}_{id}|$ represents the number of identities, $|\mathcal{D}_{pw}|$ 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 $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ are very limited (e.g., $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 10^6$) [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., $\{ID_{SD}, s_i, f_i, k_j, a\}$). We assume that SD_j is captured by the adversary; then, $\mathcal{A}$ can successfully impersonate the user U_i as follows:

Step 1. Computes $r_{GWN}^* = M_4 \bmod k_j$, where M_4 is received from (GWN)

Step 2. Decrypts the received message M_5 by using the key r_{GWN}^* and obtains the security parameters ($r_{GWN} \oplus \text{KEY}_{GWN-U_i}$, ID_{GWN} , ID_i , and r_i^*) of the user U_i who is sending the login request

Step 3. Computes KEY_{GWN-U_i} as $\text{KEY}_{GWN-U_i} = r_{GWN}^* \oplus r_{GWN} \oplus \text{KEY}_{GWN-U_i}$

Step 4. Computes $TID_i^{\text{new}} = h(ID_i \| \text{KEY}_{GWN-U_i} \| TS_4) \oplus M_{13}$, where TS_4 and M_{13} are obtained from the public channel

Step 5. Randomly chooses a new nonce r_i^{new} and current timestamp TS_1^{new}

Step 6. Computes M_1^* as $M_1^* = \text{KEY}_{GWN-U_i} \oplus r_i^{\text{new}}$ and M_2^* as $M_2^* = h(TID_i^{\text{new}} \| M_1^* \| ID_{GWN} \| r_i^{\text{new}} \| TS_1^{\text{new}})$

Step 7. Sends the login request $\{TID_i^{\text{new}}, M_1^*, M_2^*, TS_1^{\text{new}}\}$ to (GWN) and finishes the login phase

After receiving the message, GWN first checks the freshness of the received message and computes $r_i^{\text{new}} = \text{KEY}_{GWN-U_i} \oplus M_1^*$, where KEY_{GWN-U_i} is stored in the GWN's database and retrieved according to corresponding TID_i . Then, GWN computes $M_3 = h(TID_i^{\text{new}} \| M_1^* \| ID_{GWN} \| r_i^{\text{new}} \| TS_1^{\text{new}})$ 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_i .

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 $\{ID_{SD_j}, s_i, f_i, k_j\}$ from SD_j , and intercepted the messages $\{M_4, M_5, M_{10}\}$, the following method can be used to calculate the session key:

Step 1. $\mathcal{A}$ computes $r_{GWN}^* = M_4 \bmod k_j$, where M_4 is received from (GWN)

Step 2. $\mathcal{A}$ decrypts the received message M_5 by using the key r_{GWN}^* and obtains the security parameters ($r_{GWN} \oplus KEY_{GWN-U_i}$, ID_i, r_i^* , and ID_{GWN}) of the user U_i

Step 3. $\mathcal{A}$ computes $M_9^* = M_{10} \bmod k_j$, where M_{10} is received from (GWN)

Step 4. $\mathcal{A}$ computes the session key $SK^* = h(ID_i \| ID_{GWN} \| r_{GWN}^* \| M_9^* \| KEY_{GWN-U_i})$

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

3.4. 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 $y = g^x \bmod p$ 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 should choose a random number u and compute $Y_1 = g^u \bmod p$ and $C_1 = y^u \bmod p$ first; then, she constructs the login message $M_2 = h(TID_i \| M_1 \| ID_{GWN}^* \| r_i \| C_1 \| Y_1 \| TS_1)$. Since the adversary cannot calculate Y_1 , 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 KEY_{GWN-U_i} . Unfortunately, KEY_{GWN-U_i} is encrypted by the group key r_{GWN}^* in the message M_5 , but obtaining the group key r_{GWN}^* is easy for an adversary who has breached the SD_j and extracted k_j . After this, $\mathcal{A}$ could decrypt M_5 to get KEY_{GWN-U_i} , ID_i, r_i^* , and 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 $M_5 = Enc_{r_{GWN}}(ID_i, ID_{GWN}, KEY_{GWN-U_i} \oplus r_i^*)$ and

$M_6 = h(ID_i \| ID_{GWN} \| M_4 \| KEY_{GWN-U_i} \| TS_2)$. As a result, when receiving the message, $\{M_4, M_5, M_6, TS_2\}$, the adversary $\mathcal{A}$ who captures the sensing device can only obtain the parameter $KEY_{GWN-U_i} \oplus r_i^*$ by decrypting M_5 . Therefore, $\mathcal{A}$ cannot impersonate U_i since she cannot obtain KEY_{GWN-U_i} .

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 impressible 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_i 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^{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_8 , SD_j first chooses a random number r_s and computes $Y_2 = g^{r_s} \bmod p$. Then, SD_j computes $M_8 = Enc_{r_{GWN}}(ID_{SD_j}, S_j, f_j, Y_2)$ and sends it to GWN. After that, GWN chooses a random number x' and computes $C_2 = Y_2 Z^{x'} \bmod p$, $Y_{GWN-S_j} = g^{x'} \bmod p$, $M_{10} = Enc_{C_2}(M_9)$, 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 $SK = h(ID_i \| ID_{GWN} \| r_{GWN}^* \| M_9^* \| KEY_{GWN-U_i} \oplus r_i^*)$.

4. Conclusion

In this paper, we have revisited and analysed Vinoth et al.'s authentication scheme for IIoT environments. We demonstrate 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.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

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. Zuair, “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.
 - [17] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, “An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.
 - [18] C. Wang and G. Xu, “Cryptanalysis of three password-based remote user authentication schemes with non-tamper-resistant smart card,” *Security and Communication Networks*, vol. 2017, Article ID 1619741, 14 pages, 2017.