Hindawi
Security and Communication Networks
Volume 2020, Article ID 5047379, 11 pages
https://doi.org/10.1155/2020/5047379

Research Article

Cryptanalysis and Security Improvement of Two Authentication Schemes for Healthcare Systems Using Wireless Medical Sensor Networks

Jiaqing Mo,[1] **Zhongwang Hu**,[1] **and Yuhua Lin**[2]

[1]*School of Computer Science and Software, Zhaoqing University, Zhaoqing 526061, China*
[2]*Education Technology and Computer Center, Zhaoqing University, Zhaoqing 526061, China*

Correspondence should be addressed to Jiaqing Mo; mojiaqing@126.com

Received 18 October 2019; Accepted 18 January 2020; Published 19 February 2020

Guest Editor: Geethapriya Thamilarasu

Wireless medical sensor networks (WMSNs) play an important role in collecting healthcare data of the remote patient and transmitting them to the medical professional for proper diagnosis via wireless channel. To protect the patient's healthcare data which is private-related and sensitive, some authentication schemes for healthcare systems using WMSN have been proposed to ensure the secure communication between the medical sensors and the medical professional. Since cryptanalyzing the security defects of authenticated protocols is crucial to put forward solutions and propose truly robust protocols, we scrutinize two state-of-the-art authentication protocols using WMSN for healthcare systems. Firstly, we examine Ali et al.'s enhanced three-factor based authentication protocol and show that although it provides a formal proof and a security verification, it still fails to resist offline dictionary guessing attack, desynchronization attack, and privileged insider attack and contains a serious flaw in the password change phase. Secondly, we investigate Shuai et al.'s lightweight and three-factor based authentication protocol and point out that it cannot achieve high security level as they claimed; it is actually subject to offline dictionary guessing attack and privileged insider attack, and it also has a design flaw in the password change phase. In addition, we suggest several countermeasures to thwart these security weaknesses in these two schemes for WMSN and the similar kinds.

1. Introduction

Internet of Things (IoT), which enables a variety of things to connect each other via the Internet or wireless communication, by employing data-collecting devices such as sensors and radio frequency identification (RFID), has a wide range of applications [1, 2]. As an indispensable part of IoT, wireless sensor networks (WSNs) can collect data from specific objects and share them with human beings; thus, WSN is widely applied in many application scenarios, like healthcare service [3, 4], environment monitoring [5], and habitats [6]. Wireless medical sensor network (WMSN) is a popular application of WSN for healthcare systems, in which wearable sensors gather the patient's physiological information such as blood pressure, body temperature, and heart rate and send them to the medical professionals for diagnosis or further treatment [7]. It is obvious that WMSN not only monitors the patient in real-time but also saves his time and money and improves the efficiency of the medical professional. Generally, a typical WMSN mainly includes three entries: a gateway node, sensor nodes, and medical professional. The gateway node (GWN) has powerful computation and ammunition capabilities and plays the role of a communication bridge between the sensors and medical professionals. The sensor nodes, resource-restraint in computation and communication capabilities, are implanted or installed in the patient's body to gather the physiological information and transmit them to medical professionals in the distance with the help of GWN. However, the physiological information of the patient is sensitive, and they are transmitted over insecure wireless channel. If the attacker intercepts and modifies these physiological data, the doctor may make a wrong diagnosis.

Although some measures have been developed to protect the security of WSN at link layer and network layer in IEEE 802.15.4 by IETF [8, 9], it is still necessary to design a robust authentication mechanism in application layer to protect the sensitive sensed data from unauthorized access. That is to say, the identity legitimacy of the medical professional should be verified before accessing the sensor data. In addition, the sensor node to be accessed should be authenticated for the criticality and sensitivity of the sensed data from the patient. Particularly, a session key should be negotiated between the medical professional and the sensor node to secure the real-time access.

Over years, a series of authentication protocols have been proposed for WMSN to protect the transmitted data against unauthorized access from an attacker or a malicious user. We brief the previous schemes related to WMSN. Because of the limitations of wearable sensor's computation and communication capabilities, WMSN authentication schemes concern efficiency and adopt lightweight cryptography operations on the premise of ensuring security. In 2012, Kumar et al. [10] presented an efficient remote user authentication protocol named E-SAP for healthcare applications in WMSN environment and claimed that their scheme is secure against various known attacks. However, He et al. [11] showed that Kumar et al.'s scheme suffers from offline password guessing attack and privileged insider attack, as well as failure to provide user anonymity. In addition, they suggested a robust and efficient anonymous authentication protocol for patient monitoring using WMSN. Unfortunately, both Wu et al. [12] and Li et al. [13] indicated that the protocol in [11] is still vulnerable to some security weaknesses, such as denial of service attack, lack of wrong password detection mechanism, user impersonation attack, senor node capture attack, and offline password guessing attack. As a remedy, they also gave their enhanced protocol, respectively. However, Das et al. [14] observed that Li et al.'s scheme [13] is insecure to withstand sensor node capture attack, privileged insider attack, and lack of user anonymity. Further, they contributed an efficient and secure authentication protocol for WMSN. In the same year, Srinivas et al. [15] described that Wu et al.'s scheme [12] is subject to insider attack, user impersonation attack, and stolen smartcard attack. To thwart these security defects, they devised an efficient authentication scheme using lightweight operations for WMSN. But Wu et al. [16] pointed out that the scheme in [15] is unsuitable for practical deployment owing to security weaknesses like offline password guessing attack, and a lightweight two-factor authentication scheme for healthcare systems using WMSN was introduced to fix these drawbacks.

In 2016, Amin et al. [17] proposed a two-factor anonymous patient monitoring system using hash function in WMSN. The purpose of scheme in [17] is to design a robust and efficient user authentication protocol so as to provide secure data access in WMSN. However, Jiang et al. [18] claimed that scheme in [17] fails to resist stolen mobile device attack and desynchronization attack, as well as suffering from security issue of sensor key exposure. Afterwards, they devised an enhanced protocol. In addition, protocol in [17] was deemed vulnerable to user impersonation attack, offline password guessing attack, known session key temporary information, revelation of secret parameters, and identity guessing attack by Ali et al. [19], and then they proposed an enhanced three-factor authentication protocol to overcome these vulnerabilities. Although Jiang et al. [18] adopted fuzzy verifier technique and asserted that their protocol achieves admirable security properties, we find that their scheme is susceptible to privileged insider attack, denial of service attack, and known session special temporary information attack.

Since elliptical curve cryptography (ECC) can achieve the same symmetric cryptography algorithm (i.e., RSA) security level with faster computation and smaller key size, many authentication protocols have been developed for WMSN on ECC to enhance their security in recent years. In 2016, Hayajneh et al. [20] proposed an authentication protocol for remote patient monitoring with Rabin algorithm and used Tmote sky notes to prove its efficiency. In the same year, Liu and Chung [21] devised a remote user authentication scheme on bilinear pair to facilitate security and privacy protection in wireless healthcare sensor networks and asserted their scheme can resist various known attacks. But, Challa et al. [22] claimed that the protocol in [20] is susceptible to stolen smartcard attack, offline password guessing attack, privileged insider attack, user impersonation attack, and even inappropriate mutual authentication. To improve efficiency and security, they introduced a three-factor authentication protocol using lightweight ECC point multiplications with formal proof. In 2019, to ensure secure communication and privacy-preserving, Xie et al. [23] proposed an efficient and certificateless authentication scheme named CasCP with batch authentication in wireless body area networks. In the same year, Li et al. [2] considered that the protocol in [17] is vulnerable to denial of services (DoS) attack and cannot provide forward secrecy and proposed an ECC-based three-factor authentication protocol using fuzzy commitment and fuzzy verifier techniques to enhance the security of [17].

More recently, Ali et al. [19] analyzed protocol in [17] and showed that their protocol suffers from offline password guessing attack, user impersonation attack, and revelation of secret parameter, and a new three-factor protocol is introduced to resist various attacks. But, in this work, in contrast to their assertions, we examine Ali et al.'s protocol and point out that their scheme is still vulnerable to offline dictionary guessing attack, desynchronization attack, and privileged insider attack and has a flaw in the password change phase. In addition, Shuai et al. [24] in 2019 proposed a lightweight three-factor authentication scheme for patient monitoring using on-body wireless networks and employing one-time hash chain technique and pseudonym identity method to improve its security. The on-body wireless network is actually a WMSN, since the former is like the latter, in which the sensors installed on the patient collect physiological data and transmit them to the doctor or the health professional through GWN for further processing. However, in this paper, we prove that Shuai et al. [24] suffers from three security drawbacks, that is, offline dictionary guessing

attack, privileged insider attack, and flaw in the password change phase.

As two case studies, our analysis shows that a number of WMSN authentication protocols for healthcare systems and the similar kinds are not secure under some provable security models. Furthermore, our cryptanalysis of the two schemes highlights that it is important to pay attention to potential threats when proposing a new authentication protocol.

In brief, our main contributions are summarized as follows.

(1) First, we cryptanalyze Ali et al.'s protocol [19] and reveal that it cannot withstand offline dictionary guessing attack and desynchronization attack and contains a serious flaw in the password change phase.

(2) Second, we cryptanalyze Shuai et al.'s protocol [24] and show that their scheme is vulnerable to offline dictionary guessing attack as [19] and privileged insider attack. In addition, we point out a design flaw in the password change phase of their scheme.

(3) Third, we put forward some effective countermeasures to amend these two schemes and similar authentication protocols with the same defects.

The remainder of this work is organized as follows: In Section 2, we review Ali et al.'s protocol and show their security weaknesses. Shuai et al.'s protocol is reviewed and cryptanalyzed in Section 3. Section 4 puts forward several countermeasures to fix the discovered threats. Finally, conclusion is made in Section 5.

2. Cryptanalysis on Ali et al.'s Protocol

In this section, we briefly review and cryptanalyze Ali et al.'s protocol [19], which is a lightweight three-factor based authentication protocol for healthcare monitoring in WMSN environment. Their scheme consists of five phases: system setup, user registration, login, authentication, and password change. To facilitate description, we list notations in Table 1 and they will be used throughout this work.

Table 1: Notations.

Notation	Description
U_i	i^{th} user
RA	Registration agent
GWN	Gateway node
S_j	j^{th} sensor node
ID_i	The identity of U_i
SID_j	The identity of S_j
PW_i	The password of U_i
BIO_i	U_i's biometrics
$\oplus$	The bitwise XOR operation
$\|\|$	The concatenation operation
$\Longrightarrow$	The secure channel
$\longrightarrow$	The public channel
$h()$	A secure one-way hash function
$H()$	Bio-hash function
$E_k()/D_k()$	The symmetric encryption/decryption function with key k
T_i	The current timestamp, $i = 1, 2, \ldots$
MD	Mobile device of U_i
MID_i	Temporary identity of U_i
ID_g	The identity of GWN
NC_{k0}	Serial number in GWN side
NC_k	Serial number in S_j side

2.1. Review of Ali et al.'s Scheme

2.1.1. System Setup. Firstly, the administrator SA selects identity SID_j for each sensor node and computes $X_{\text{GS}} = h(\text{SID}_j \,||\, X_G)$, $K_j = h(X_{\text{GS}} \,||\, Y_G \,||\, X_G)$, where X_G and Y_G are secret keys of GWN. Afterwards, SA stores $\{X_{\text{GS}}, K_j\}$ in the memory of the sensor node S_j.

2.1.2. User Registration. If the user wants to access the sensor, he must register in the gateway node first.

(1) $U_i \Longrightarrow$ GWN: $\{\text{ID}_i, \text{RPW}_i, F_i\}$

$\text{RPW}_i = h(\text{ID}_i \,||\, \text{PW}_i \,||\, r_i)$, $F_i = H\ (\text{BIO}_i \,||\, r_i)$, where r_i is a random number of U_i.

(2) GWN $\Longrightarrow U_i$: smartcard $= \{A_i, C_i, D_i, \text{DID}_i, H(), h()\}$

$A_i = h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g) \oplus h(\text{RPW}_i \,||\, F_i)$, $C_i = R_g \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $D_i = h(\text{RPW}_i \,||\, R_g \,||\, F_i)$, and $\{\text{DID}_i, C_i\}$ is stored in GWN's database. DID_i is a dynamic identity chosen by GWN and R_g is a random number.

(3) U_i computes $R_n = r_i \oplus h(\text{ID}_i \,||\, \text{PW}_i \,||\, H(\text{BIO}_i))$ and stores it in the smartcard.

2.1.3. Login

(1) U_i inserts his smartcard, inputs ID_i, PW_i and imprints BIO_i, and then the smartcard computes $r_i = R_n \oplus h(\text{ID}_i \,||\, \text{PW}_i \,||\, H(\text{BIO}_i))$, $\text{RPW}_i = h(\text{ID}_i \,||\, \text{PW}_i \,||\, r_i)$, $F_i = H\ (\text{BIO}_i \,||\, r_i)$, $h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g) = A_i \oplus h(\text{RPW}_i \,||\, F_i)$, $R_g = C_i \oplus \text{h}(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, and $D_i' = h(\text{RPW}_i \,||\, R_g \,||\, F_i)$ and checks $D_i' ? = D_i$. If it fails, U_i aborts this login request.

(2) $U_i \longrightarrow$ GWN: $\{\text{DID}_i, M_1, M_2, M_3\}$.

$M_1 = M_i \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $M_2 = E_{h(M_i \| R_g)}(\text{ID}_i \| \text{SID}_j \| T_1 \| A_i)$, $M_3 = h(\text{ID}_i \,||\, \text{SID}_j \,||\, h(\text{RPW}_i \,||\, F_i))$, where M_i is a random nonce and T_1 is the current timestamp.

2.1.4. Authentication

(1) GWN $\longrightarrow S_j$: $\{M_1, M_4, M_5, M_6\}$

GWN computes $M_i = M_1 \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $R_g = C_i \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$ and decrypts M_2 to obtain $(\text{ID}_i \,||\, \text{SID}_j \,||\, T_1 \,||\, A_i)$ using $h(M_i \,||\, R_g)$. If T_1 is not fresh, GWN aborts this session; otherwise, GWN computes $h(\text{RPW}_i \,||\, F_i) = A_i \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $M_3' = h(\text{ID}_i \,||\, \text{SID}_j \,||\, h(\text{RPW}_i \,||\, F_i))$, and checks whether $M_3' ? = M_3$. If it is false, GWN terminates the

session. Otherwise, GWN computes $X_{GS} = h(SID_j \| X_G)$, $K_j = h(X_{GS} \| Y_G \| X_G)$, $M_4 = E_{h(X_{GS}\|K_j)}(ID_g \| ID_i \| M_i \| A_i \| T_3)$, $M_5 = N_i \oplus h(T_3 \| h(RPW_i \| F_i))$, and $M_6 = h(ID_i \| N_i \| T_3 \| ID_g)$.

(2) $S_j \longrightarrow$ GWN: $\{M_7, M_8, T_5\}$

S_j decrypts M_4 to get $(ID_g \| ID_i \| M_i \| A_i \| T_3)$ with key $h(X_{GS} \| K_j)$ and verifies the freshness of T_3. If not, S_j aborts the session. Otherwise, GWN computes $h(RPW_i \| F_i) = M_1 \oplus M_i \oplus A_i$, $N_i = h(T_3 \| h(RPW_i \| F_i)) \oplus M_5$, $M_6' = h(ID_i \| N_i \| T_3 \| ID_g)$ and checks whether $M_6' ?= M_6$. If not, GWN aborts the session. Otherwise, S_j computes $M_7 = V_i \oplus h(M_i \| N_i)$, $SK = h(h(RPW_i \| F_i) \| M_i \| N_i \| V_i)$, and $M_8 = h(SK \| ID_i \| ID_g \| T_5)$.

(3) GWN $\longrightarrow U_i$: $\{M_9, M_{10}\}$

GWN checks the freshness of T_5. GWN aborts the session if T_5 is not fresh. Otherwise, GWN computes $V_i = M_7 \oplus h(M_i \| N_i)$, $SK' = h(h(RPW_i \| F_i) \| M_i \| N_i \| V_i)$, and $M_8' = h(SK' \| ID_i \| ID_g \| T_5)$ and checks whether $M_8' ?= M_8$. If not, GWN aborts the session. Otherwise, GWN computes $C_i^n = R_g \oplus h(DID_i^n \| X_G \| ID_g)$, $M_9 = E_{h(RPW_i\|F_i)}(C_i^n \| N_i \| V_i \| DID_i^n)$, and $M_{10} = h(SK \| C_i^n \| DID_i^n)$ and updates the database with $\{DID_i^n, C_i^n\}$.

(4) U_i decrypts M_9 with $h(RPW_i \| F_i)$ to obtain $(C_i^n \| N_i \| V_i \| DID_i^n)$, computes $SK' = h(h(RPW_i \| F_i) \| M_i \| N_i \| V_i)$, and checks whether $M_{10} ?= h(SK' \| C_i^n \| DID_i^n)$. If yes, U_i replaces (DID_i, C_i) with (DID_i^n, C_i^n). Otherwise, U_i rejects the session.

2.1.5. Password Change. This phase is performed if U_i wants to change his password.

(1) U_i inserts smartcard and keys ID_i, PW_i, and imprints BIO_i, and then the smartcard computes $r_i = R_n \oplus h(ID_i \| PW_i \| H(BIO_i))$, $RPW_i = h(ID_i \| PW_i \| r_i)$, $F_i = H(BIO_i \| r_i)$, $h(DID_i \| X_G \| ID_g) = A_i \oplus h(RPW_i \| F_i)$, $R_g = C_i \oplus h(DID_i \| X_G \| ID_g)$, and $D_i' = h(RPW_i \| R_g \| F_i)$ and compares $D_i' ?= D_i$. If it fails, smartcard aborts the session. Otherwise, the procedure continues.

(2) U_i inputs his new password PW_i^{new}, and the smartcard computes $RPW_i^{new} = h(ID_i \| PW_i^{new} \| r_i)$, $A_i^{new} = A_i \oplus h(DID_i \| X_G \| ID_g) \oplus h(RPW_i^{new} \| F_i)$, and $D_i^{new} = h(RPW_i^{new} \| C_i \oplus h(DID_i \| X_G \| ID_g) \| F_i)$. Finally, smartcard replaces (A_i, D_i) with (A_i^{new}, D_i^{new}).

2.2. Cryptanalysis of Ali et al.'s Protocol. Although Ali et al.'s protocol [19] is equipped with a formal security proof to show that their scheme can withstand various known attacks, it still suffers from some security defects. In this subsection, we prove that their protocol cannot resist offline dictionary guessing attack, desynchronization attack, and privileged insider attack and has a flaw in the password change phase though they tried to fix the security drawbacks in Amin et al.'s scheme. Since it is crucial to depict the capabilities of the adversary in designing a robust authentication protocol in WSN environment, we summarize the adversary model as follows [19, 25–27].

(1) The attacker can intercept, delete, modify, and insert the messages exchanged between the related communication parties over public channel.

(2) The attacker cannot guess the secret key and random number since they are assumed sufficiently large.

(3) The attacker can offline enumerate the user-memorable identities and low-entropy passwords in polynomial time simultaneously.

(4) As far as privileged insider attack is taken into account, the privileged-insider in GWN being an attacker can learn the submitted information by the user during the registration phase of authentication protocol.

(5) When considering whether some multifactor authentication protocol can provide truly multifactor security (i.e., the n factors protocol is secure, even if n-1 factors are compromised), it is reasonable to suppose that (i) the attacker can somehow obtain the lost/stolen smartcard and retrieve the secret information by using side-channel attack [28, 29]. (ii) The attacker can collect the biometrics of the user through malicious device without awareness of victim.

2.2.1. Offline Dictionary Guessing Attack. It is widely regarded that the password-based authentication schemes are prone to password guessing attack [30–32], including online password guessing attack and offline password guessing attack, since the users tend to choose a password that is easy to remember. The online password guessing can be relatively detected by judging whether the time of logins exceeds the threshold. On the contrary, during this guessing attack, the attacker does not need to communicate with related communication parties, and thus the offline password guessing attack is not easily surmounted.

In Ali et al.'s scheme, they claimed their scheme not only can withstand password guessing attack, but also can withstand identity guessing attack. Unfortunately, we prove that their claim is not convincing as they claimed. According to the aforementioned adversary model, we assume that the user's lost/stolen smartcard is obtained by the attacker, and the user's biometrics is also collected by the attacker without awareness of owner, and the attackers can launch offline password guessing attack and offline identity guessing attack simultaneously in terms of item 3 in the adversary model, which we call offline dictionary guessing attack. The offline dictionary guessing attack is conducted to get the user's identity and password by the attacker with the following procedure.

Step 1: the attacker extracts the secret data $\{A_i, D_i, DID_i, H(), h(), R_n\}$ from the smartcard by using methods reported in [28].

Step 2: the attacker selects a candidate pair $(\mathrm{ID}_i^*, \mathrm{PW}_i^*)$ from D_{ID} and D_{PW}, where D_{ID} denotes the identity space and D_{PW} denotes the password space.

Step 3: the attacker computes $r_i^* = R_n \oplus h(\mathrm{ID}_i^* \,||\, \mathrm{PW}_i^* \,||\, H(\mathrm{BIO}_i))$, $\mathrm{RPW}_i = h(\mathrm{ID}_i^* \,||\, \mathrm{PW}_i^* \,||\, r_i^*)$, $F_i^* = H(\mathrm{BIO}_i \,||\, r_i^*)$, $h(\mathrm{DID}_i \,||\, X_G \,||\, \mathrm{ID}_g) = A_i \oplus h(\mathrm{RPW}_i^* || F_i^*)$, $R_g^* = C_i \oplus h(\mathrm{DID}_i \,||\, X_G \,||\, \mathrm{ID}_g)^*$.

Step 4: the attacker checks whether the extracted D_i equals the computed $h(\mathrm{RPW}_i^* \,||\, R_g^* \,||\, F_i^*)$.

Step 5: if it holds, the attacker has found a right pair $(\mathrm{ID}_i, \mathrm{PW}_i)$. Otherwise, the attacker repeats steps 2–4 until the right pair $(\mathrm{ID}_i, \mathrm{PW}_i)$ is found.

For ease of achieving user friendliness, Ali et al.'s scheme [19], like previous schemes [12, 17, 18], provides the password update phase, allowing the users to select their own ID and password and make changes. Generally, the user likes to choose an easy-to-remember identity and password, which are often low-entropy. Thus, this makes sense for the attacker to perform offline dictionary guessing attack by enumerating pairs $(\mathrm{ID}_i, \mathrm{PW}_i)$ in polynomial time. Let $|D_{\mathrm{ID}}|$ and $|D_{\mathrm{PW}}|$ represent the size of D_{ID} and D_{PW}, respectively. In addition, we set T_h and T_H as the execution time of hash function $h()$ and bio-hash function $H()$, respectively. The time complexity of the above attack procedure is $O(|D_{\mathrm{ID}}| * |D_{\mathrm{PW}}| * 4T_h * 2T_H)$. Since T_h and T_H are limited, it is clear that the time required by the attacker to carry out the above attack procedure is linear to $|D_{\mathrm{ID}}| * |D_{\mathrm{PW}}|$. As reported in [33, 34], both the identity space D_{ID} and the password space D_{PW} are rather limited in practice (e.g., $|D_{\mathrm{ID}}| \leq D_{\mathrm{PW}}| \leq 10^6$ [33, 34]), and thus, it is possible for the attacker to guess $(\mathrm{ID}_i, \mathrm{PW}_i)$ within polynomial time. Wang and Wang [35] even pointed out that the time spent on the above guessing attack can be greatly reduced to the level of seconds on an ordinary computer. Therefore, Ali et al.'s protocol [19] is vulnerable to offline dictionary guessing attack.

Based on the aforementioned attack, after the attacker has obtained the user's identity and password, he can impersonate the user to log onto GWN with the smartcard and the collected biometrics. In this regard, Ali et al.'s protocol suffers from user impersonation attack.

2.2.2. Desynchronization Attack. To achieve security features of user anonymity and user untraceability, Ali et al.'s protocol [19] makes use of synchronous update mechanism; that is, GWN updates the dynamic identity DID_i and C_i synchronously with U_i via message $\{M_9, M_{10}\}$. In this way, the attacker cannot trace a particular user by eavesdropping messages over the public channel. However, we point out that the attacker can breach this synchronous mechanism by blocking the last message $\{M_9, M_{10}\}$, leading to failure when the user logs onto GWN the next time. Such attack is illustrated as follows.

In Step 7 of the authentication phase after updating $\{\mathrm{DID}_i^n, C_i^n\}$ in the database, GWN sends message $\{M_9, M_{10}\}$ to U_i, where $M_9 = E_{h(\mathrm{RPW}_i \| F_i)}(C_i^n \| N_i \| V_i \| \mathrm{DID}_i^n)$, $M_{10} = h(\mathrm{SK} \,||\, C_i^n \,||\, \mathrm{DID}_i^n)$ and DID_i^n is a new dynamic identity. Upon receiving the message, U_i will generate a session key and replace $\{\mathrm{DID}_i, C_i\}$ with $\{\mathrm{DID}_i^n, C_i^n\}$. If the malicious attacker blocks this message at the end of authentication process, and the parameters $\{\mathrm{DID}_i, C_i\}$ in the user's smartcard remain unchanged while $\{\mathrm{DID}_i, C_i\}$ on the GWN side have been updated, it means the attacker has broken the dynamic identity synchronization mechanism between GWN and the user by means of blocking messages. As a result, the medical professional can no longer log onto GWN to access data from the sensor on the patient.

2.2.3. Privileged Insider Attack. According to item 4 of the adversary model, a privileged insider of GWN obtains the user's registration request information $\{\mathrm{ID}_i, \mathrm{RPW}_i, F_i\}$, as well as the secret data $\{A_i, C_i, D_i, \mathrm{DID}_i, H(), h()\}$ on the smartcard before GWN sent the smartcard to the user. With this information, he launches a privileged insider attack as follows.

Step 1: he eavesdrops the messages $\{M_1, M_4, M_5, M_6\}$ and $\{M_9, M_{10}\}$ from the public channel.

Step 2: then, he decrypts M_9 using decryption key $h(\mathrm{RPW}_i \,||\, F_i)$ to obtain N_i and V_i.

Step 3: further, he acquires M_i by computing $M_i = h(\mathrm{RPW}_i \,||\, F_i) \oplus M_1 \oplus A_i$.

Step 4: finally, with the known parameters $h(\mathrm{RPW}_i \,||\, F_i)$, M_i, N_i, V_i, the attacker can compute the session key $\mathrm{SK} = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, M_i \,||\, N_i \,||\, V_i)$.

Therefore, Ali et al.'s scheme suffers from privileged insider attack.

2.2.4. Flaw in Password Change Phase. In Ali et al.'s protocol, they provide a password change phase to allow users to freely change the password locally. However, our scrutiny reveals that their password change phase has a fatal flaw which will prevent the user from logging onto GWN. In their scheme, before changing the password, the user is asked to input his identity and old password and imprint his biometrics. If the identity legitimacy of the user is verified by the smartcard, the user is allowed to enter a new password to update the old one. Then, the smartcard computes $\mathrm{RPW}_i^{\mathrm{new}} = h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, r_i)$, $A_i^{\mathrm{new}} = A_i \oplus h(\mathrm{DID}_i \,||\, X_q \,||\, \mathrm{ID}_g) \oplus h(\mathrm{RPW}_i^{\mathrm{new}} \,||\, F_i)$, $D_i^{\mathrm{new}} = h(\mathrm{RPW}_i^{\mathrm{new}} \,||\, C_i \oplus h(\mathrm{DID}_i \,||\, X_G \,||\, \mathrm{ID}_g) \,||\, F_i))$. At last, the smartcard replaces $\{A_i, D_i\}$ with $\{A_i^{\mathrm{new}}, D_i^{\mathrm{new}}\}$. Note that R_n has not been updated with the new password $\mathrm{PW}_i^{\mathrm{new}}$. Thereafter, if the user wants to log onto GWN, he enters ID_i, $\mathrm{PW}_i^{\mathrm{new}}$ and imprints BIO_i, and the smartcard computes $r_i' = R_n \oplus h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, H(\mathrm{BIO}_i))$. It is evident that $r_i' \neq r_i$, since $\mathrm{PW}_i^{\mathrm{new}} \neq \mathrm{PW}_i$. Accordingly, because the calculation of D_i is related to r_i, the computed D_i is not equal to the stored D_i in the smartcard. For this reason, the legal user is always rejected from logging onto GWN once he changed his password. Thus, Ali et al.'s protocol suffers with a serious flaw in the password change phase.

3. Cryptanalysis on Shuai et al.'s Protocol

In this section, we review and cryptanalyze Shuai et al.'s protocol [24] proposed in 2019, which is an anonymous authentication scheme for remote patient monitoring. To achieve some desirable security attributes, their scheme employs pseudonym identity method to preserve user anonymity and adopts one-time hash chain technique to achieve forward secrecy. The serial number technique is also used to resist desynchronization attack. Furthermore, they conduct an informal security analysis to show that their scheme is secure against various attacks. However, in the following section, we find that their scheme is susceptible to offline dictionary guessing attack; that is, their protocol fails to provide truly a three-factor security. On the other hand, we show that their protocol is suspected to privileged insider attack.

3.1. Review of Shuai et al.'s Scheme. We will concisely review Shuai et al.'s scheme. Their protocol involves initialization phase, registration phase, login phase, authentication and key agreement phase, and password change phase.

3.1.1. Initialization Phase. The RA performs this phase offline. RA chooses two random numbers ID_g and K as the identity and master secret key to GWN, respectively. Next, RA chooses a collision-resistant cryptographic hash function $h()$ for all communication participants. Finally, RA chooses a unique identity SID_j for each wearable sensor node S_j and stores SID_j into S_j's memory.

3.1.2. Registration Phase. This phase consists of two points, that is, user registration phase and wearable sensor node registration phase.

(1) User registration

(*i*) U_i => RA: $\{ID_i, A_i\}$
The user U_i inputs his ID_i, PW_i, and imprints biometrics BIO_i to mobile device MD. Thereafter, MD computes $\text{Gen}(BIO_i) = (R_i, P_i)$, $A_i = h(PW_i \| R_i \| a_i)$, where Gen is a probabilistic generation procedure, R_i is a secret random key, P_i is an auxiliary string, and a_i is a random secret value generated by U_i.

(ii) RA => U_i: $\{MID_i, B_i, C_i, K_{GU}\}$
RA chooses three nonces b_i, r_1, r_2 and sets $K_{GU} = r_1$, $MID_i = MID_{i0} = r_2$, $MID_{i1} = \text{null}$. Afterwards, RA calculates $B_i = A_i \oplus h(ID_i \| K \| b_i)$, $C_i = h(h(ID_i \| K \| b_i) \| A_i)$, stores $\{ID_i, MID_{i0}, MID_{i1}, b_i, K_{GU}\}$ into the user information table, and copies this table to GWN.

(iii) U_i calculates $D_i = h(ID_i \| PW_i \| R_i) \oplus a_i$ and then stores $\{D_i, P_i\}$ into the MD's memory. Finally, MD contains secret data $\{MID_i, B_i, C_i, D_i, K_{GU}, P_i\}$.

(2) Wearable sensor node registration phase

(*i*) S_j => RA: $\{SID_j\}$.

(ii) RA => S_j: $\{K_{GS}, NC_K\}$.
RA chooses a random nonce K_{GS} and set $NC_K = NC_{K0} = 0$, and then RA stores $\{SID_j, K_{GS}, NC_{K0}\}$ into the sensor node information table and copies it to GWN.

(iii) On receipt of the message, S_j stores $\{K_{GS}, N_{CK}\}$ into its memory.

3.1.3. Login Phase. U_i keys his ID_i, PW_i, and imprints his biometrics BIO_i^* to MD, and MD computes $R_i^* = \text{Rep}(BIO_i^*, P_i)$, $a_i^* = D_i \oplus h(ID_j \| PW_i \| R_i^*)$, $A_i^* = h(PW_i \| R_i \| a_i^*)$, $h(ID_i \| K \| b_i) = B_i \oplus A_i^*$, $C_i^* = h(h(ID_i \| K \| b_i) \| A_i^*)$ and checks whether C_i^* equals the stored C_i. If it is false, MD aborts the session. Otherwise, MD chooses a random nonce R_1 and the current timestamp T_1, computes $MS_1 = (R_1 \| SID_j) \oplus h(MID_i \| h(ID_i \| K \| b_i) \| K_{GU})$, $V_1 = h(ID_i \| R_1 \| h(ID_i \| K \| b_i) \| MID_i \| K_{GU} \| T_1)$. Finally, U_i sends message $\{MID_i, MS_1, V_1, T_1\}$ to GWN.

3.1.4. Authentication and Key Agreement Phase

(1) On receiving the login request, GWN checks the freshness of timestamp T_1. If not, GWN rejects the request. Otherwise, the subsequent operations of GWN are divided into three cases.

Case 1: If $MID_i = MID_{i0}$, GWN extracts $\{ID_i, b_i, K_{GU}, MID_{i1}\}$ from the user information table in light of MID_i and then checks whether the one-time hash chain K_{GU} is updated.

(*i*) If $MID_{i1} = \text{NULL}$, it means that K_{GU} has been updated. GWN computes $(R_1^* \| SID_j) = MS_1 \oplus h(MID_{i0} \| h(ID_i \| K \| b_i) \| K_{GU})$, $V_i^* = h(ID_i \| R_1^* \| h(ID_i \| K \| b_i) \| MID_{i0} \| K_{GU} \| T_1)$, and checks whether $V_i^* = V_i$ holds. If not, GWN aborts the session. Otherwise, GWN chooses a new pseudonym identity MID_{i0}^* and sets $MID_{i1} = MID_{i0}$, $MID_{i0} = MID_{i0}^*$.

(ii) If $MID_{i1} \neq \text{NULL}$, it indicates K_{GU} is not updated in the last session. GWN computes $K_{GU}^* = h(K_{GU})$, $(R_1^* \| SID_j) = MS_i \oplus h(MID_{i0} \| h(ID_i \| K \| b_i) \| K_{GU}^*)$ and $V_i^* = h(ID_i \| R_1^* \| h(ID_i \| K \| b_i) \| MID_{i0} \| K_{GU} \| T_1)$ and checks whether V_i^* equals V_i. If not, GWN aborts this session. Otherwise, GWN generates a new random pseudonym identity MID_{i0}^* and sets $MID_{i1} = MID_{i0}$, $MID_{i0} = MID_{i0}^*$, $K_{GU} = K_{GU}^*$.
Case 2: If $MID_i = MID_{i1}$, GWN extracts $\{ID_i, b_i, K_{GU}\}$, computes $(R_1^* \| SID_j) = MS_1 \oplus h(MID_{i1} \| (h(ID_i \| K \| b_i) \| K_{GU})$, $V_i^* = h(ID_i \| R_1^* \| h(ID_i \| K \| b_i) \| MID_{i1} \| K_{GU} \| T_1)$, and verifies $V_i^* \stackrel{?}{=} V_i$. If not, GWN aborts this session. Otherwise, GWN selects a new random pseudonym identity MID_{i0}^* and sets $MID_{i0} = MID_{i0}^*$.
Case 3: If $MID_{i1} \neq MID_{i0}$ and $MID_i \neq MID_{i1}$, GWN aborts the session.

(2) GWN $\longrightarrow S_j$: $\{MS_2, V_2, NC_{k0}\}$

GWN chooses a random nonce R_2 and computes $MS_2 = (R_1 \| R_2 \| ID_i \| ID_g) \oplus h(K_{GS} \| SID_k \| NC_{k0})$, $V_2 = h(ID_i \| ID_g \| R_1 \| R_2 \| K_{GS} \| NC_{k0})$. Thereafter, GWN updates K_{GS} and NC_{k0} with $K_{GS} = h(K_{GS} \| SID_j)$ and $NC_{k0} = NC_{k0} + 1$, respectively.

(3) $S_j \longrightarrow$ GWN: $\{MS_3, V_3\}$

Upon receiving the message from GWN, S_j checks whether $1 \leq NC_{k0} - NC_k \leq N$ holds, where N is a threshold. If it is false, S_j aborts the session. Otherwise, after setting $K^*_{GS} = K_{GS}$, S_j computes $N-1$ times $K^*_{GS} = h(K^*_{GS} \| SID_j)$. If $N = 1$, S_j will not execute the above hash operation. Then, S_j computes $(R_1 \| R_2 \| ID_i \| ID_g) = MS_2 \oplus h(K^*_{GS} \| SID_j \| (NC_{k0} - 1))$, $V^*_2 = h(ID_i \| ID_g \| R_1 \| R_2 \| K^*_{GS} \| (NC_{k0} - 1))$, and verifies $V^*_2 ? = V_2$. If it is true, S_j sets $K_{GS} = h(K^*_{GS} \| SID_j)$ and $NC_k = NC_{k0}$. Then, S_j generates a random number R_3 and computes $SK = h(ID_i \| ID_g \| SID_k \| R_1 \| R_2 \| R_3)$, $MS_3 = R_3 \oplus h(K_{GS} \| SID_j \| NC_k)$, $V_3 = h(SID_j \| ID_i \| SK \| R_3 \| NC_k)$, and transmits $\{MS_3, V_3\}$ to GWN.

(4) GWN $\longrightarrow U_i$: $\{MS_4, V_4\}$

Upon receiving the message from S_j, GWN computes $R^*_3 = MS_3 \oplus h(K_{GS} \| SID_j \| NC_{k0})$, $SK = h(ID_i \| ID_g \| SID_j \| R_1 \| R_2 \| R^*_3)$, $V^*_3 = h(SID_j \| ID_i \| SK \| R^*_3 \| NC_{k0})$, and verifies $V^*_3 ? = V_3$. If it is false, GWN aborts the session. Otherwise, GWN computes $MS_4 = (R_2 \| R_3 \| ID_g \| MID_{i0}) \oplus h(R_1 \| h(ID_i \| K \| b_i) \| K_{GU} \| MID_{i1})$, $V_4 = h(ID_i \| SID_j \| SK \| R_2 \| MID_{i0})$, and sends $\{MS_4, V_4\}$ to U_i.

(5) $U_i \longrightarrow$ GWN: $\{V_5\}$

Upon receiving the message, U_i computes $(R_2 \| R_3 \| ID_g \| MID_{i0}) = MS_4 \oplus h(R_1 \| h(ID_i \| K \| b_i) \| K_{GU} \| MID_i)$, $SK = h(ID_i \| ID_g \| SID_j \| R_1 \| R_2 \| R_3)$, $V^*_4 = h(ID_i \| SID_j \| SK \| R_2 \| MID_{i0})$, and verifies $V^*_4 ? = V_4$. If it is false, U_i aborts the session. Otherwise, U_i computes $V_5 = h(ID_i \| ID_g \| SID_j \| MID_{i0} \| SK)$ and sets $K_{GU} = h(K_{GU})$ and $MID_i = MID_{i0}$. After that, U_i sends $\{V_5\}$ to GWN.

(6) GWN

Upon receiving $\{V_5\}$, GWN computes $V^*_5 = h(ID_i \| ID_g \| SID_j \| MID_{i0} \| SK)$ and verifies $V^*_5 ? = V_5$. If it is false, GWN aborts the session. Otherwise, GWN sets $K_{GU} = h(K_{GU})$ and $MID_{i1} = NULL$ and believes that U_i has shared a session key with S_j.

3.1.5. Password Change Phase. U_i inputs ID_i, PW_i and imprints BIO_i to mobile device MD. Then, MD computes $R^*_i = Rep(BIO_i, P_i)$, $a^*_i = D_i \oplus h(ID_i \| PW_i \| R^*_i)$, $A^*_i = h(PW_i \| R_i \| a^*_i)$, $h(ID_i \| K \| b_i) = B_i \oplus A^*_i$, $C^*_i = h(h(ID_i \| K \| b_i) \| A^*_i)$, and compares C^*_i with the stored C_i. If it is true, MD rejects the password change request. Otherwise, MD allows U_i to input a new password PW^{new}_i and computes $A^{new}_i = h(PW^{new}_i \| R_i \| a_i)$, $B^{new}_i = h(ID_i \| K \| b_i) \oplus A^{new}_i = B_i \oplus A_i \oplus A^{new}_i$, and $C^{new}_i = h(h(ID_i \| K \| b_i) \oplus A^{new}_i)$. Finally MD updates $\{B_i, C_i\}$ with $\{B^{new}_i, C^{new}_i\}$.

3.2. Cryptanalysis on Shuai et al.'s Scheme. Despite armed with three factors and formal security proof, Shuai et al.'s protocol [24] suffers from offline dictionary guessing attack and privileged insider attack and contains a serious design flaw in the password change phase.

3.2.1. Offline Dictionary Guessing Attack. Suppose the attacker has obtained the lost/stolen mobile device and extracted the secret data $\{MID_i, B_i, C_i, D_i, K_{GU}, P_i\}$ from it; meanwhile, he has collected biometrics BIO_i of the medical professional via a malicious terminal; the attacker can mount an offline dictionary guessing attack as follows.

Step 1: computes $R^*_i = Rep(BIO_i, P_i)$;

Step 2: chooses a pair (ID^*_i, PW^*_i) from the dictionary space DID and DPW, respectively.

Step 3: computes $a^*_i = D_i \oplus h(ID^*_i \| PW^*_i \| r^*_i)$, $A^*_i = h(PW^*_i \| r^*_i \| A^*_i)$, $h(ID^*_i \| K \| b_i)^* = B_i \oplus A^*_i$, $C^*_i = h(h(ID^*_i \| K \| b_i)^* \| A^*_i)$, where D_i and B_i are from the mobile device.

Step 4: verifies the correctness of (ID^*_i, PW^*_i) pair by checking whether the computed C^*_i equals the stored C_i. If it holds, the attacker has found the correct value of (ID^*_i, PW^*_i). Otherwise, the attacker repeats steps 2–4 until $C^*_i = C_i$.

It is clear that the time complexity of the above attack is $O(|D_{ID}| * |D_{PW}| * 3T_h)$, where T_h is the execution time of hash function. As analyzed in Section 2.2.1, such attack is quite efficient.

3.2.2. Privileged Insider Attack. Assume privileged insider of RA being an attacker, it is easy for him to know the registration information $\{ID_i, A_i\}$ during the user registration phase. Moreover, he also can learn $\{ID_i, MID_{i0}, MID_{i1}, b_i, K_{GU}\}$ from the user information table and the registration reply message $\{MID_{i1}, B_i, C_i, K_{GU}\}$ from the side of RA and mount a privileged insider attack. The similar attacks have been discussed in [14, 36–38]. Using these information, the attacker can reveal the session key with the following procedure.

Step 1: computes $h(ID_i \| K \| b_i)^* = A_i \oplus B_i$.

Step 2: intercepts the user's login request message $\{MID_i, MS_1, V_1, T_1\}$ and GWN's reply message $\{MS_4, V_4\}$ from the public channel.

Step 3: acquires R_1 and SID_j by computing $(R_1 \| SID_j)^* = MS_1 \oplus h(MID_i \| h(ID_i \| K \| b_i)^* \| K_{GU})$.

Step 4: acquires R_2, R_3, ID_g, and MID_{i0}, by computing $(R_2 \| R_3 \| ID_g, \| MID_{i0})^* = MS_4 \oplus h(R_1 \| h(ID_i \| K \| b_i)^* \| K_{GU} \| MID_i)$.

Step 5: computes the session key $SK = h(ID_i \| ID_g \| SID_j \| R_1 \| R_2 \| R_3)$.

With the session key, the attacker can decrypt all the messages between the user and the sensor. In this way, the patient's sensitive physiological information is exposed to the attacker. Therefore, Shuai et al.'s scheme fails to resist privileged insider attack.

3.2.3. Flaw in Password Change Phase. For ease of the password change phase, Shuai et al.'s scheme also provides the password change phase for U_i to change his password locally without contacting the RA. Unfortunately, similar to Ali et al.'s scheme, there is a serious security flaw in their password change phase which prevents the users who change their password from being able to log onto GWN again. Before allowing the user to change the password, the MD verifies his identity legitimacy based on the identity ID_i, password PW_i, and biometrics information BIO_i provided by the user. If the user is legitimate, MD allows U_i to input his new password $\mathrm{PW}_i^{\mathrm{new}}$. However, this password change phase only updates B_i and C_i stored on the mobile device according to the new password and does not update D_i with the new password, which is used to recover the secret random number a_i of U_i during the login phase. The user either writes the secret random number a_i on a paper or bears it in mind or updates D_i with the new password. Thus, if he intends to recover a_i by computing $a_i = D_i \oplus h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, R_i^*)$ when he logs onto GWN, he will fail because the previous PW_i is different from the new $\mathrm{PW}_i^{\mathrm{new}}$, and $h()$ is a collision-resistant function, which causes the computed value of $h(\mathrm{ID}_i \,||\, \mathrm{PW}_i \,||\, R_i^*)$ and $h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, R_i^*)$ not to be equal. As a result, the user who has changed his password will be rejected by MD when he intends to log onto GWN again. What is worse, the user can no longer change the password in the future, because MD also needs to verify the legitimacy of the user by recovering the user's secret random number a_i before changing his password.

4. Countermeasures

In order to address the security weaknesses in Ali et al.'s protocol and Shuai et al.'s protocol, we provide several possible countermeasures in this section.

4.1. Countermeasures to Offline Dictionary Guessing Attack. Our previous analysis shows that neither Ali et al.'s scheme nor Shuai et al.'s Scheme can provide truly three-factor security; that is, the attacker can launch an offline dictionary guessing attack to acquire the user's identity and password if he obtains the user's smartcard (or mobile device) and biometrics somehow. The root cause of this attack described above is that the password verifier $D_i = h(\mathrm{RFW}_i \,||\, R_g \,||\, F_i)$ of Ali et al.'s protocol and $C_i = h(h(\mathrm{ID}_i \,||\, K \,||\, b_i) \,||\, A_i)$ are stored in a smartcard (mobile device). Consequently, if the smartcard is obtained by the attacker, he will try to make a breach in the password verifier for offline dictionary guessing attack.

To thwart this security weakness without radical improvement while keeping usability, a feasible countermeasure is to utilize "fuzzy verifier" technique [25]. In the following, taking Ali et al.'s protocol as a case study to show how to integrate fuzzy verifier, we revise the password verifier D_i as $D_i = h(h(\mathrm{RPW}_i \,||\, R_g \,||\, F_i) \bmod n)$ during the user registration phase, where n represents the space size of $(\mathrm{ID}_i, \mathrm{PW}_i)$ pair. If the attacker has obtained the user's smartcard and biometrics, he picks up a pair $(\mathrm{ID}_i^*, \mathrm{PW}_i^*)$ from D_{ID} and D_{PW} to perform offline dictionary guessing attack as described in Section 2.2.1. However, it is hard for the attacker to find a correct pair $(\mathrm{ID}_i, \mathrm{PW}_i)$ since there are $(|D_{\mathrm{ID}}| * |D_{\mathrm{PW}}|)/n \approx 2^{32}$ candidates of $(\mathrm{ID}_i, \mathrm{PW}_i)$ pair (suppose $n = 2^8$, $|D_{\mathrm{ID}}| = |D_{\mathrm{PW}}| = 2^6$ [25, 33]). Someone may question if the attacker will just pick up an incorrect pair of $(\mathrm{ID}_i, \mathrm{PW}_i)$ but can satisfy $D_i = h(h(\mathrm{RPW}_i \,||\, R_g \,||\, F_i) \bmod n)$. The probability of such an event is $1/2^8$. Moreover, if the user is asked to enter the old/new password twice, and the hash function $h()$ responds as a random oracle, the probability will greatly reduce to $(1/2^8)^2 = 1/2^{16}$ [25, 33, 34]. Therefore, the fuzzy verifier that provides adequate candidate can effectively prevent the attacker from mounting offline dictionary guessing attack successfully. In addition, the effectiveness of fuzzy verifier technique has been discussed and verified in Section V-B of [24], and the interested readers can refer to it for more information.

4.2. Countermeasures to Desynchronization Attack. We have demonstrated that Ali et al.'s protocol is insecure against desynchronization attack in Section 2.2.2. Specifically, to provide user anonymity and untraceability, GWN chooses a new dynamic identity DID_i^n, computes the corresponding C_i^n, and stores $\{\mathrm{DID}_i^n, C_i^n\}$ in its database. Meanwhile, to keep consistency in the next login, the user needs to update $\{\mathrm{DID}_i^n, C_i^n\}$ in the smartcard simultaneously. However, Ali et al.'s protocol only considers the case where all messages in the ideal situation are successfully received by the receiver. If the attacker blocks the message $\{M_9, M_{10}\}$ from the GWN to the user to break the consistency in the authentication process, the authenticated parameters $\{\mathrm{DID}_i^n, C_i^n\}$ are made to be different between GWN and the user U_i, which means U_i could not log onto GWN ever since.

To cope with such an attack, an effective countermeasure is to avoid updating the user dynamic identity DID_i simultaneously on both sides of communication parties. That is, during the authentication phase, GWN chooses a new dynamic identity DID_i^n for U_i, but does not need to save it to the database. After decrypting M_9, U_i conceals DID_i^n with the new random number M_i and other information generated in each login, stores it in the smartcard, and restores DID_i^n on the next login. If message $\{M_9, M_{10}\}$ is blocked, on the one hand, the attacker cannot obtain the new DID_i^n because M_9 is encrypted; on the other hand, U_i does not update $D\mathrm{ID}_i$ in the smartcard since he has not received $\{M_9, M_{10}\}$. When U_i logs onto GWN next time, GWN can still recover M_i with the stored $D\mathrm{ID}_i$ instead of DID_i^n. In this way, although the attacker attempts to break the synchronization, he will not succeed because the dynamic identity information of the user has not been saved in GWN, and GWN will perform the subsequent procedure regardless of whether $\{M_9, M_{10}\}$ is blocked or not. Hence, the desynchronization attack is

thwarted effectively. It is worth noting that we only give the main idea of the measure, not a complete scheme, because the detailed solution requires a long paper. In addition, their user registration phase and the password change also need to be revised correspondingly, and we omitted them due to the space constraints.

4.3. Countermeasures to Privileged Insider Attack. Our aforementioned analysis shows that both of the two schemes suffer from privileged insider attack. The root cause is that to improve the computation efficiency, they use lightweight operations based on hash function and random numbers to generate the session key, which makes the leakage of a small amount of secret data easily lead to the leakage of other secret data. To thwart this attack, the public-key operations such as modular exponentiation or elliptic curve point multiplication should be adopted in their scheme [31]. We take the GWN and sensor side as the server side and keep the user as the client side; according to [31], modular exponentiation operation should be performed at least twice on the server side. Take Ali et al.'s scheme as an example and use elliptic curve point multiplication; without requiring radical improvement, the main idea of overcoming privileged insider attack during the login and authentication phase is sketched as follows.

Step 1: after generating the random nonce M_i in the login phase, U_i computes $W_1 = M_i P$ and sends the message containing W_1 to GWN. P is a generator in elliptic curve group over a finite field.

Step 2: because GWN does not need to participate in negotiating session key, GWN sends the message containing W_1 to S_j after the user's identity legitimacy verification is passed.

Step 3: if the legitimacy authentication of GWN is passed, the sensor S_j selects the random number V_i and calculates $W_2 = V_i P$ and computes the session key $\mathrm{SK} = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, W_1 \,||\, W_2 \,||\, V_i W_1) = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, M_i P \,||\, V_i P \,||\, V_i M_i P)$. Afterwards, S_j sends a message containing W_2 to U_i via GWN.

Step 4: if the legitimacy of GWN and S_j is ensured, U_i computes the session key $\mathrm{SK} = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, W_1 \,||\, W_2 \,||\, M_i W_2) = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, M_i P \,||\, V_i P \,||\, M_i V_i P)$.

If the attacker eavesdrops W_1 and W_2 from the public channel and intends to find M_i and V_i from W_1 and W_2, respectively, it is infeasible since he has to resolve elliptic curve discrete logarithm problem [2]; and if he intends to compute $M_i V_i P$ from W_1 and W_2, it is also impossible since he faces the hardness of elliptic curve computational Diffie-Hellman problem [2].

4.4. Countermeasures to Flaw in Password Change Phase. As we have analyzed before, both Ali et al.'s scheme and Shuai et al.'s scheme contain serious flaws in their password change phase which renders the user unable to log onto GWN again after changing his password. The reason is that none of their password change phase are designed to recover the secret random number for login. Thus, the countermeasures to fix these design flaws are obvious, and we describe them as follows.

(1) For Ali et al.'s protocol, $R_n^{\mathrm{new}} = r_i \oplus h(\mathrm{ID}_i^* \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, H(\mathrm{BIO}_i))$ should be added in step 2 of the password change phase, and R_n^{new} is also needed to replace the previous R_n in the smartcard.

(2) For Shuai et al.'s protocol, when performing step 2 of the password change phase, MD needs to additionally compute $D_i^{\mathrm{new}} = a_i \oplus h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, R_i)$ and replaces D_i with D_i^{new} in MD.

5. Conclusion

In the past few years, many three-factor authentication protocols have been proposed for WMSN and the similar environment. But, most of them are vulnerable to some inherent security defects more or less. In this paper, we briefly review and cryptanalyze the two quite recent and typical authentication protocols with key agreement presented by Ali et al. and Shuai et al., respectively. Firstly, we point out that although Ali et al. tried to overcome the security defects in the previous scheme and provide security proof with BAN logic and simulation under AVISPA, they are still vulnerable to offline dictionary guessing attack, desynchronization attack, and privileged insider attack and even contain a serious design flaw in the password change phase. Secondly, we demonstrate that Shuai et al.'s protocol is also insecure against offline dictionary guessing attack and privileged insider attack and has a design flaw in the password change phase. Thereafter, we put forward some possible countermeasures to eliminate these security weaknesses. Note that in this paper, the assumption that an attacker can simultaneously obtain both the secret information on the smartcard (mobile device) and the biometrics of the user is a trivial case, but it still cannot be ignored since security is one of the most important factors to consider in designing a protocol. Otherwise, if it is not based on this assumption, the attacker will require higher time complexity when carrying out offline ID and password dictionary attacks on the two protocols. Our efforts highlight that it is important to be aware of potential security risks in designing authentication protocols for WMSN and the similar kinds. This also indicates the necessity of our work.

Data Availability

(1) The reference data [19] used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s12652-018-1015-9]). (2) The reference data [24] used to support the findings of this study have been deposited in the [Hindawi] repository ([DOI: 10.1155/2019/8145087]).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Project no. 61672007) and Science and Technology Innovation Guidance Project 2017 (Project no. 201704030605).

References

[1] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, "On the security and privacy of internet of things architectures and systems," in *Proceedings of the 2015 International Workshop on Secure Internet of Things (SIoT)*, pp. 49–57, IEEE, Vienna, Austria, September 2015.

[2] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Systems Journal*, pp. 1–12, 2019.

[3] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "Iot healthcare analytics: the importance of anomaly detection," in *Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 994–997, IEEE, Crans-Montana, Switzerland, May 2016.

[4] S. H. Shah, A. Iqbal, and S. S. A. Shah, "Remote health monitoring through an integration of wireless sensor networks, mobile phones & cloud computing technologies," in *Proceedings of the 2013 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 401–405, IEEE, San Jose, CA, USA, October 2013.

[5] G. Mois, T. Sanislav, and S. C. Folea, "A cyber-physical system for environmental monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, pp. 1463–1471, 2016.

[6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, ACM, Atlanta, GA, USA, September 2002.

[7] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[8] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the internet of things," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147–159, 2011.

[9] J. Spins, "Security protocols for sensor networks," *Wireless Networks*, vol. 5, pp. 521–534, 2002.

[10] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-sap: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[11] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[12] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and anonymous two-factor authentication protocol for healthcare applications with wireless medical sensor networks," *Multimedia Systems*, vol. 23, no. 2, pp. 195–205, 2017.

[13] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655, 2016.

[14] A. K. Das, A. K. Sutrala, V. Odelu, and A. Goswami, "A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1899–1933, 2017.

[15] J. Srinivas, D. Mishra, and S. Mukhopadhyay, "A mutual authentication framework for wireless medical sensor networks," *Journal of Medical Systems*, vol. 41, no. 5, p. 80, 2017.

[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] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 80, pp. 483–495, 2018.

[18] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[19] R. Ali, A. K. Pal, S. Kumari, A. K. Sangaiah, X. Li, and F. Wu, "An enhanced three factor based authentication protocol using wireless medical sensor networks for healthcare monitoring," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–22, 2018.

[20] T. Hayajneh, B. Mohd, M. Imran, G. Almashaqbeh, and A. Vasilakos, "Secure authentication for remote patient monitoring with wireless medical sensor networks," *Sensors*, vol. 16, no. 4, p. 424, 2016.

[21] C.-H. Liu and Y.-F. Chung, "Secure user authentication scheme for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 59, pp. 250–261, 2017.

[22] S. Challa, A. K. Das, V. Odelu et al., "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 69, pp. 534–554, 2018.

[23] Y. Xie, S. Zhang, X. Li, Y. Li, and Y. Chai, "Cascp: efficient and secure certificateless authentication scheme for wireless body area networks with conditional privacy-preserving," *Security and Communication Networks*, vol. 2019, Article ID 5860286, 13 pages, 2019.

[24] M. Shuai, B. Liu, N. Yu, and L. Xiong, "Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks," *Security and Communication Networks*, vol. 2019, Article ID 8145087, 14 pages, 2019.

[25] 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, 2016.

[26] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[27] J. Mo, Z. Hu, H. Chen, and W. Shen, "An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 4520685, 12 pages, 2019.

[28] T. H. Kim, C. Kim, and I. Park, "Side channel analysis attacks using am demodulation on commercial smart cards with seed," *Journal of Systems and Software*, vol. 85, no. 12, pp. 2899–2908, 2012.

[29] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer Science & Business Media, Berlin, Germany, 2008.

[30] 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.

[31] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[32] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

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

[34] 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.

[35] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 1–15, 2014.

[36] T.-H. Chen, Y.-C. Chen, W.-K. Shih, and H.-W. Wei, "An efficient anonymous authentication protocol for mobile pay-tv," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1131–1137, 2011.

[37] J. Wei, W. Liu, and X. Hu, "Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 77, no. 3, pp. 2255–2269, 2014.

[38] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-to-peer Networking and Applications*, vol. 9, no. 1, pp. 223–244, 2016.