Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7579161, 16 pages
https://doi.org/10.1155/2018/7579161

Research Article

Trusted Authority Assisted Three-Factor Authentication and Key Agreement Protocol for the Implantable Medical System

Deming Mao,[1] **Ling Zhang**,[2] **Xiaoyu Li**,[1] **and Dejun Mu**[1]

[1] *School of Automation, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China*
[2] *Southwest Institute of Telecommunication, Chengdu 610041, Sichuan, China*

Correspondence should be addressed to Deming Mao; maodmnwpu@163.com

Received 5 April 2018; Accepted 27 June 2018; Published 29 July 2018

Academic Editor: Joseph Liu

The application of implantable medical devices (IMDs), which solves the problems of geographical distance limitation and real-time health monitoring that plague patients and doctors, has caused great repercussions in the medical community. Despite the great potential of wide application, it also brings some security and privacy issues, such as the leakage of health data and unauthorized access to IMDs. Although a number of authentication and key agreement (AKA) schemes have been developed, we find that some subtle attacks still remain to be addressed. Then we propose an improved AKA scheme which achieves strong security features including user anonymity and known key security. It is formally proved to be secure under the Real-or-Random model. Moreover, a comprehensive security analysis shows that our scheme can resist various attacks and satisfy the desired requirements. Finally, the performance analysis shows the superiority of our protocol which is suitable for the implantable medical system.

1. Introduction

With the improvement of wireless communication technologies, the implantable medical devices (IMDs), such as pacemakers, cranial nerve stimulators, and cochlear implants, have been widely used in the medical services field [1, 2]. All these micro devices implanted in patients' body can continuously monitor and collect data to reflect the patient's health. Through controller node (CN), implantable medical devices are able to transmit the data to the remote attending physician or the medical institution, which greatly simplifies the treatment process of patients and breaks the limitation of region. Generally speaking, the combination of these advanced technologies improves health care practices, urgent care, and preventive health [3].

A typical architecture of implantable medical system is shown in Figure 1. CN and IMDs firstly register to the trusted authority (TA) before they are deployed into the system. Then, IMDs collect data such as body temperature, heart beats, and blood pressure, which can be derived by CN via wireless communication technologies, such as Bluetooth or ZigBee [4]. After the collection process, the CN needs to be plugged into the Internet via an access point to be accessible by the attending physician or the medical institution. In the meantime, cloud servers may be used for storing collected health data to ease the storage burden on mobile devices [5, 6].

However, it is the application of wireless communication that makes the transmission of medical data face the potential security risks [7–9]. According to the Dolev-Yao threat model [10], the implantable medical system is facing a wide range of malicious attacks which may cause the leakage of health data and unauthorized access to IMDs. In response to the serious security threats, it is imperative to design a mutual authentication and key agreement (AKA) mechanism which can ensure the confidentiality of the transmitted sensor data and resist malicious attacks.

1.1. Related Work. With the wireless interface enabled, IMDs can be accessed by an authorized operator in physical proximity via the IMDs programmer. However, the wireless communication and networking capabilities of IMDs turn out to be the major sources of security vulnerabilities [11, 12]. For this purpose, access control for implantable medical

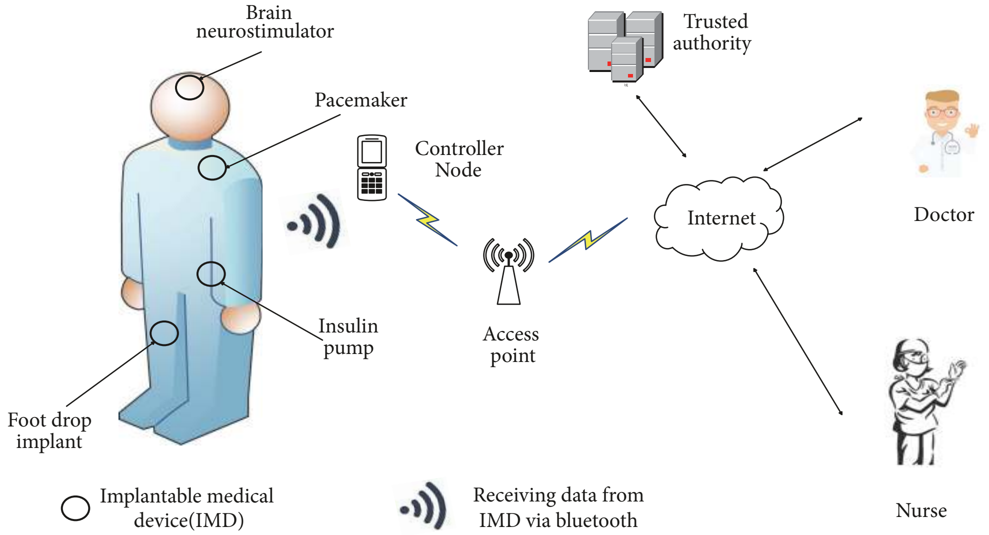


Figure 1: The network model of the implantable medical system.

system is highly desired and many schemes have also been put forward in this field.

Initially, considering the scarce energy reserves and limited communication capacity of IMDs, some schemes based on symmetric key cryptography [15–19] were proposed, they realized high encryption speed and efficiency at the same time but showed weaknesses of resisting against certain attacks, and the complexity of key management will introduce large memory and communication overhead which contradicts their original intentions. This means that the symmetric key cryptography based schemes are difficult to provide a complete security guarantee for implantable medical system.

Then, traditional public key cryptography (TPKC) based authentication schemes [20, 21] were implemented in IMDs. Unfortunately, the limited computing capability and battery capacity of the mobile device hinders the application of TPKC in implantable medical system. The concept of ECC (Elliptic curve cryptosystem) was then put forward [22] which provided the same security with a much smaller key size compare to the TPKC [23] so that many ECC-based protocols were proposed subsequently [13, 24]. In 2013, Liu et al. [25] put forward a scheme in which they used the bilinear pairing defined on the elliptic curve to design a new certificateless signature scheme, but later in 2014, Xiong [26] analyzed the Liu et al.'s authentication protocol and concluded that their scheme was prone to a kind of attack by a key replacement adversary [27]. In 2016, He et al. [28] also claimed that the Liu et al.'s scheme cannot resist the impersonation attack; meanwhile they put forward their own improved protocol. In 2018, Li et al. [29] analyzed the loopholes in each layer of the current implantable medical system and put forward a complete three-layer scheme.

As we know, each authentication factor has its own advantages and disadvantages. Passwords are prone to dictionary attacks while smart cards may be lost. A number of two-factor protocols [30–38] have been put forward. In these schemes, two kinds of factors, i.e., passwords and smart cards, are combined to achieve user authentication. In 2015, He et al. came up with a scheme [35] where the smart card is used to store some private parameters about healthcare applications using wireless medical sensor networks. Wei et al. proposed an anonymous authentication scheme [33] for wireless body area networks in 2017 as well as gave a formal security analysis of the protocol.

To further enhance the security strength of two-factor protocols, three-factor authentication (3FA) schemes which consolidate all three factors (i.e., passwords, smart cards, and biometrics) have attracted more and more attentions [14, 39–44]. In 2017, Wei applied the fuzzy extractor scheme into his newly proposed protocol [39] to handle the biometrics. Meanwhile Jiang et al. presented a scheme [41] where the biohashing is used to protect the biometrics. In 2016, Wu et al. proposed a 3FA scheme [43] aiming at summarizing the flaws that existed in previous typical protocols and came up with a more complete solution. In 2017, Li et al. [40] remedied flaws in Jiang et al.'s scheme [32] in which fuzzy commitment is used to protect biometrics. In 2017, Wazid et al. provided a 3FA scheme [14] for IMDs and claimed that their protocol could meet the known security, but we reveal that the protocol cannot achieve complete security.

1.2. Motivations and Contributions. With the popularity of the IMD, its safety and privacy protection have attracted great attention and a large number protocols in this field have emerged, but few of them can achieve the desired security guarantee. In such a situation, it is imperative to sum up the defects in previous protocols and propose new schemes to make the implantable medical system more secure and reliable. Among these protocols, we pick Wazid et al.'s scheme [14] as a typical case study to analyze some defects of the scheme. Then we propose a trusted authority assisted 3FA protocol which effectively solves the security vulnerabilities in the original protocol. Our contributions are summarized as follows:

(i) First, we find out three drawbacks of the most recent 3FA protocol of Wazid et al. To be specific, we find that the scheme cannot withstand offline password guessing attack, the CN impersonation attack, and the authentication phase of the protocol is problematic.

(ii) Second, we propose a trusted authority assisted 3FA protocol. Specifically, we introduce the fuzzy verifier [45] to effectively prevent offline password guessing attack during local login verification phase and adopt the widely used fuzzy vault [46] to protect the biometric template.

(iii) Third, we analyze the security of our protocol both formally and informally. Our protocol not only properly solves the shortcomings in the original scheme, but also achieves perfect forward security, user anonymity, know key security, and so forth. At the same time, our protocol can resist a variety of known attacks.

1.3. Organization of the Paper. The rest of the paper is organized as follows. In Section 2, we briefly review some preliminaries used in this paper, including ECC and the fuzzy vault. Section 3 depicts the details of Wazid et al.'s scheme. Then in Section 4, we present the vulnerabilities in their scheme. In Section 5, we propose an improved scheme. In Section 6, we have an elaborate analysis from both formal and informal point of view. The comparisons of efficiency and features are listed in Section 7. In the end, this paper is concluded in Section 8.

2. Preliminaries

2.1. Fuzzy Vault. The fuzzy vault is a constructor used to protect biometric templates BIO with various built-in algorithms. Its security relies on the secret key K and BIO. It works in key binding mode where the biometric and the key are monolithically bound within a binding mechanism. Compared with fuzzy extractor [47], the Euclidean distance measurement used in fuzzy vault has been widely accepted in most biostatistical applications [48]. Therefore, in view of the value in practice, we will adopt the fuzzy vault to protect biometric features in our improved scheme.

Specifically, the user selects a polynomial Pol which is used to encode secret key K and be evaluated on all elements in BIO. Then the biometric BIO which is imprinted by user can be converted into a set of L points which lie on the Pol according to $Gen(BIO, K, Pol) = L$. Then, taking L and CP which is a large set of "chaff points" as inputs of $Enc(\cdot)$, we can get the final vault V which equals $CP \cup L$, that is, $Enc(CP, L) = V$. Generally, we put the final vault V in the mobile device.

When the user wants to recover the secret key K, she/he can scan the biometric BIO^* on terminal firstly, then taking the vault V and BIO^* as the inputs of the algorithm $Dec(\cdot)$ which will output the Pol if and only if $|BIO - BIO^*| < \varepsilon$ where ε is the fuzziness parameter. The secret key K can be recovered with the input Pol by the algorithm $Rec(\cdot)$ finally.

2.2. Elliptic Curve Cryptosystem (ECC). Compared with the traditional RSA algorithm, ECC achieves the same security strength with much smaller key size, so ECC is more efficient than RSA. Elliptic curve equation is defined in such a form: nonsingular elliptic curve $E_p(a,b) : y^2 - x^3 + ax + b(\bmod p)$ over a prime finite field Z_p, where p is a large prime and $a, b \in Z_p^*$ satisfies $4a^3 + 27b^2 \neq 0(\bmod p)$.

Table 1: Notations.

Notations	Description
U_i, MD_i	i_{th} user and his/her mobile device
CN_j	j_{th} controller node
IMD_l	l_{th} implantable medical device
TA	Trusted authority
ID_i, PW_i, BIO_i	U_i's identity, password and biometric information
ID_{TA}, ID_{CN_j}	Identities of TA and controller node
$RID_i, RID_{CN_j}, RTS_{CN_j}$	Pseudo identities of U_i and CN_j, registration timestamp of CN_j
N	1024-bit secret number of TA
T_i	Current timestamp
ΔT	Maximum transmission delay associated with a message
t	Error tolerance threshold used in fuzzy extractor
$k \cdot P$	Elliptic curve point multiplication, $k \in Z_p^*,\ P \in E_p(a,b)$
$h(\cdot)$	Collision-resistant cryptographic hash function
$\|$	Concatenation operation
$\oplus$	Bitwise XOR operation

Besides, there are two difficult problems in ECC, namely, Elliptic Curve Discrete Logarithm Problem (ECDLP) and Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP). Specifically, the first one depicts that it is impossible to find an integer $x \in Z_p^*$ that satisfies the formula $Q = x \cdot P$ with two given points P and Q over $E_p(a,b)$. The other one describes that it is hard to calculate the value $xy \cdot P$ with the given points P, $x \cdot P$ and $y \cdot P$, $x, y \in Z_p^*$. These two hard problems guarantee the security of Elliptic Curve primitives, and an adversary still has a great deal of difficulty in getting the secret after obtaining the public values.

3. Review of Wazid et al.'s Scheme

In this section, we review the details of Wazid et al.'s scheme, which consists of eight phases, i.e., predeployment, postdeployment, registration, login, authentication and key agreement, password and biometric update, and dynamic control node addition, as well as dynamic IMD addition. The scheme is for the purpose of mutual authentication and key agreement establishment between the mobile device and IMDs. The notations used in this paper are listed in Table 1.

3.1. Predeployment Phase. Before deployment, a trusted authority TA needs to complete the registration for each CN_j

as well as IMD_l. TA first selects a secret 1024-bit number N for CN_j and IMD_l. Then TA picks the identity ID_{CN_j} for CN_j and calculates $RID_{TA} = h(ID_{TA} \parallel N)$, $RID_{CN_j} = h(ID_{CN_j} \parallel N)$, $TC_{CN_j} = h(ID_{TA} \parallel RTS_{CN_j} \parallel N)$. Meanwhile, TA constructs the univariate polynomial $P(RID_{CN_j}, y)$ according to the polynomial-based key distribution $P(x, y) = \sum_{i=0}^{n}\sum_{j=0}^{n} g_{i,j}x^{i}y^{j} \in GF(p)[x, y]$ proposed in [49] where the prime p is chosen as a large number and n is also large to preserve unconditional security and n-collusion resistant property against IMD capture attack. Finally, TA stores $\{RID_{TA}, RID_{CN_j}, TC_{CN_j}, P(RID_{CN_j}, y)\}$ in the memory of CN_j. Similar to the above calculations, TA generates a unique identity ID_{IMD_l} and calculates $RID_{IMD_l} = h(ID_{IMD_l} \parallel N)$, $P(RID_{IMD_l}, y)$ and then stores the information $\{RID_{IMD_l}, P(RID_{IMD_l}, y)\}$ in the memory of ID_{IMD_l}.

3.2. Postdeployment Phase. After the predeployment phase, CN_j and IMD_l establish a shared key using the information distributed during the predeployment phase. The details of the process are as follows. Firstly, IMD_l sends the message $\langle RID_{IMD_l}\rangle$ to CN_j. Once CN_j receives the message, CN_j responds with the message $\langle RID_{CN_j}\rangle$. Then they calculate the same shared secret key $SK_{IMD_l,CN_j} = P(RID_{IMD_l}, RID_{CN_j})$ and $SK_{CN_j,IMD_l} = P(RID_{IMD_l}, RID_{CN_j})$ on each own for future use.

3.3. Registration Phase. This phase has 4 steps.

Step 1. The user selects his/her identity ID_i at will and forwards it with registration request to TA in a secure channel.

Step 2. After accepting the request, TA computes the pseudo identity of U_i as $RID_i = h(ID_i \parallel N)$. Then TA continues to compute the value A_i as $A_i = h(RID_{TA} \parallel ID_i)$. TA sends the message $\langle RID_i, A_i, RID_{TA}\rangle$ to U_i.

Step 3. After receiving registration reply from TA, U_i further selects a private key $k \in Z_p^*$ and computes the corresponding public key $Q = k \cdot P$.

Step 4. U_i inputs his/her password PW_i and imprints fingerprint BIO_i in mobile device MD_i, then MD_i calculates $Gen(BIO_i) = (\sigma_i, \tau_i)$, $RID_i' = RID_i \oplus h(PW_i \parallel \sigma_i)$, $RPW_i = h(PW_i \parallel k)$, $D_i = k \oplus h(ID_i \parallel PW_i \parallel \sigma_i)$, $RID_{TA}' = RID_{TA} \oplus h(ID_i \parallel k \parallel \sigma_i)$, $A_i' = A_i \oplus h(k \parallel \sigma_i)$, $B_i = h(A_i \parallel RPW_i)$, and $C_i = h(ID_i \parallel RID_{TA} \parallel B_i \parallel \sigma_i)$. At last, MD_i keeps the data $\{RID_i', RID_{TA}', A_i', C_i, D_i, \tau_i, Gen(\cdot), Rep(\cdot), h(\cdot), t\}$ in its memory.

3.4. Login Phase. As depicted in Figure 2, to login to CN_j, U_i executes the following steps.

Step 1. U_i inputs his/her ID_i, PW_i and BIO_i', then MD_i retrieves the biometric key $\sigma_i' = Rep(BIO_i', \tau_i)$. Then MD_i computes $k' = D_i \oplus h(ID_i \parallel PW_i \parallel \sigma_i')$, $RPW_i' = h(PW_i \parallel k')$, $A_i^* = A_i' \oplus h(k' \parallel \sigma_i')$, $B_i^* = h(A_i^* \parallel RPW_i')$, $RID_{TA}^* = RID_{TA}' \oplus h(ID_i \parallel k' \parallel \sigma_i')$, $RID_i^* = RID_i' \oplus h(PW_i \parallel \sigma_i')$, and $C_i^* = h(ID_i \parallel RID_{TA}^* \parallel B_I^* \parallel \sigma_i')$. If C_i^* equals the stored C_i, it means that U_i's inputs are verified as correct; otherwise, the login phase will be terminated immediately.

Step 2. MD_i picks the current timestamp T_1 and a 160-bit random nonce r_i. Then MD_i computes $a_i = h(r_i \parallel T_1 \parallel RID_i^* \parallel RPW_i' \parallel \sigma_i')$, $b_i = h(RID_{TA}^* \parallel T_1)$, and $M_1 = a_i \cdot P$ as well as the signature $M_2 = a_i + k'b_i(\mathrm{mod}\,p)$. At last, MD_i forwards the message $\langle M_1, M_2, T_1\rangle$ to CN_j via a public channel.

3.5. Authentication and Key Agreement Phase. In this phase, U_i and CN_j need to authenticate each other as well as establish a session key between them for future safe communications; see Figure 2.

Step 1. After obtaining the message $\langle M_1, M_2, T_1\rangle$, CN_j first checks $|T_1 - T_1^*|? < \Delta T$, if two values are equal, CN_j calculates $b_i' = h(RID_{TA} \parallel T_1)$, and then checks $M_2 \cdot P? = M_1 + b_i' \cdot Q$. Similarly, if verification matches, it indicates that U_i is considered legitimate. Then CN_j chooses T_2 and a random number r_j and continues to compute $c_j = h(r_j \parallel T_2 \parallel RID_{CN_j} \parallel TC_{CN_j})$, $M_4 = c_j \cdot P$, $k_{ij} = c_j \cdot M_1 = (a_i c_j) \cdot P$, session key $SK_{ij} = h(k_{ij} \parallel RID_{TA} \parallel T_1 \parallel T_2)$, and $M_5 = h(SK_{ij} \parallel T_2)$. Finally, CN_j sends the message $\langle M_4, M_5, T_2\rangle$ to U_i through the public channel.

Step 2. After receiving the message from CN_j, U_i first judges $|T_2 - T_2^*|? < \Delta T$, then computes $k_{ij}^* = a_i \cdot M_4 = (a_i c_j) \cdot P$, $SK_{ij}^* = h(k_{ij}^* \parallel RID_{TA}^* \parallel T_1 \parallel T_2)$, and $M_6 = h(SK_{ij}^* \parallel T_2)$. If $M_6 = M_5$, it indicts that CN_j passes the verification. With that, U_i calculates $M_7 = h(SK_{ij}^* \parallel T_3)$ and forwards the message $\langle M_7, T_3\rangle$ to CN_j.

Step 3. CN_j checks $|T_3 - T_3^*|? < \Delta T$, then computes $M_8 = h(SK_{ij} \parallel T_3)$, and judges whether $M_8 = M_7$.

Finally, both CN_j and U_i complete the mutual authentication and agree on the same session key which will used for the secure communications in future.

3.6. Password and Biometric Update Phase. If U_i wants to change the password, he/she can execute ensuing procedure.

Step 1. Firstly, U_i inputs ID_i, PW_i^{old}, and BIO_i^{old}. MD_i computes $\sigma_i^{old} = Rep(BIO_i^{old}, \tau_i)$, $k = D_i \oplus h(ID_i \parallel PW_i^{old} \parallel \sigma_i^{old})$, $RPW_i^{old} = h(PW_i^{old} \parallel k)$, $A_i^{old} = A_i' \oplus h(k \parallel \sigma_i^{old})$, $B_i^{old} = h(A_i^{old} \parallel RPW_i^{old})$, and $RID_{TA} = RID_{TA}' \oplus h(ID_i \parallel k \parallel \sigma_i^{old})$ and checks if C_i^{old} equals $h(ID_i \parallel RID_{TA} \parallel B_i^{old} \parallel \sigma_i^{old})$. If it holds, MD_i asks U_i for the new password PW_i^{new}.

Step 2. After U_i inputs the PW_i^{new} and MD_i calculates $\sigma_i^{new} = Rep(BIO_i^{new}, \tau_i^{new})$, $RPW_i^{new} = h(PW_i^{new} \parallel k)$, $A_i^{new} = A_i^{old} \oplus h(k \parallel \sigma_i^{new})$, $B_i^{new} = h(A_i^{new} \parallel RPW_i^{new})$, $RID_{TA}'' = RID_{TA} \oplus h(ID_i \parallel k \parallel \sigma_i^{new})$, $C_i^{new} = h(ID_i \parallel RID_{TA}'' \parallel B_i^{new} \parallel \sigma_i^{new})$, and $D_i'' = k \oplus h(ID_i \parallel PW_i^{new} \parallel \sigma_i^{new})$. Finally, MD_i replaces RID_i',

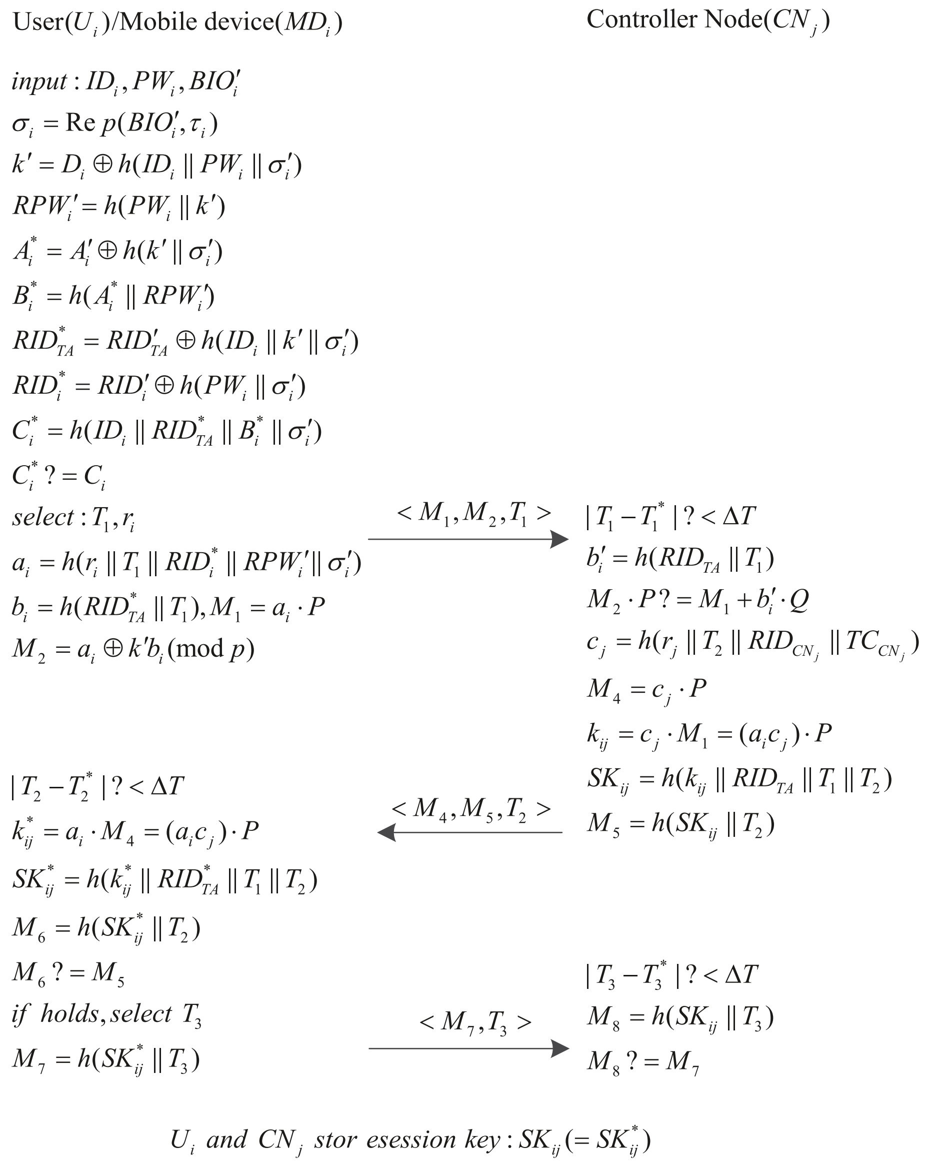

User(U_i)/Mobile device(MD_i)
Controller Node(CN_j)
$input: ID_i, PW_i, BIO'_i$
$\sigma_i = \mathrm{Re}\,p(BIO'_i, \tau_i)$
$k' = D_i \oplus h(ID_i \| PW_i \| \sigma'_i)$
$RPW'_i = h(PW_i \| k')$
$A^*_i = A'_i \oplus h(k' \| \sigma'_i)$
$B^*_i = h(A^*_i \| RPW'_i)$
$RID^*_{TA} = RID'_{TA} \oplus h(ID_i \| k' \| \sigma'_i)$
$RID^*_i = RID'_i \oplus h(PW_i \| \sigma'_i)$
$C^*_i = h(ID_i \| RID^*_{TA} \| B^*_i \| \sigma'_i)$
$C^*_i ?= C_i$
$select: T_1, r_i$
$< M_1, M_2, T_1 >$
$|T_1 - T^*_1| ?< \Delta T$
$a_i = h(r_i \| T_1 \| RID^*_i \| RPW'_i \| \sigma'_i)$
$b'_i = h(RID_{TA} \| T_1)$
$b_i = h(RID^*_{TA} \| T_1), M_1 = a_i \cdot P$
$M_2 \cdot P ?= M_1 + b'_i \cdot Q$
$M_2 = a_i \oplus k' b_i (\bmod p)$
$c_j = h(r_j \| T_2 \| RID_{CN_j} \| TC_{CN_j})$
$M_4 = c_j \cdot P$
$k_{ij} = c_j \cdot M_1 = (a_i c_j) \cdot P$
$SK_{ij} = h(k_{ij} \| RID_{TA} \| T_1 \| T_2)$
$|T_2 - T^*_2| ?< \Delta T$
$< M_4, M_5, T_2 >$
$M_5 = h(SK_{ij} \| T_2)$
$k^*_{ij} = a_i \cdot M_4 = (a_i c_j) \cdot P$
$SK^*_{ij} = h(k^*_{ij} \| RID^*_{TA} \| T_1 \| T_2)$
$M_6 = h(SK^*_{ij} \| T_2)$
$M_6 ?= M_5$
$|T_3 - T^*_3| ?< \Delta T$
$if\ holds, select\ T_3$
$< M_7, T_3 >$
$M_8 = h(SK_{ij} \| T_3)$
$M_7 = h(SK^*_{ij} \| T_3)$
$M_8 ?= M_7$
$U_i\ and\ CN_j\ stor\ esession\ key: SK_{ij} (= SK^*_{ij})$

Figure 2: Login and authentication phase of Wazid et al.'s scheme.

RID'_{TA}, A'_i, C_i, D_i, and τ_i with RID''_i, RID''_{TA}, A^{new}_i, C^{new}_i, D''_i, and τ^{new}_i, respectively.

3.7. Dynamic Controller Node Addition Phase. In this phase, a new controller node CN^{new}_j can be deployed as follows.

First, TA determines a new identity $ID^{new}_{CN_j}$ for CN^{new}_j and calculates $RID^{new}_{CN_j} = h(ID^{new}_{CN_j} \| N)$ and new polynomial $P(RID^{new}_{CN_j}, y)$ as well as $TC^{new}_{CN_j} = h(ID_{TA} \| RTS^{new}_{CN_j} \| N)$ in which the $RTS^{new}_{CN_j}$ is the newly generated registration timestamp. Finally, TA stores the parameters $\{RID^{new}_{CN_j}, TC^{new}_{CN_j}, RID_{TA}, P(RID^{new}_{CN_j}, y)\}$ into the memory of CN^{new}_j before it is deployed into the system.

3.8. Dynamic IMD Addition Phase. In this phase, we can deploy a new IMD (IMD^{new}_l). Specifically, TA computes $RID'_{IMD_l} = h(ID'_{IMD_l} \| N)$ and $P(RID'_{IMD_l}, y)$ and then stores $\{RID'^{new}_{IMD_l}, P(RID'^{new}_{IMD_l}, y)\}$ in the memory of IMD^{new}_l.

4. Weakness of the Wazid et al.'s Scheme

The widely accepted Dolev-Yao threat model (DY model) [10] demonstrates that the adversary A can fully control the public channel between communicators. That is, A is capable of eavesdropping, stealing, inserting, deleting, and modifying the messages in the open channel. Most recently, Wang et al. [45] have provided a complete summary of the adversary's capabilities and present twelve evaluation criteria for a secure protocol, i.e., no password verifier-table, no smart card loss attack, mutual authentication, and so forth. According to above evaluation criteria, we make a reasonable analysis of Wazid et al.'s scheme and find that the protocol

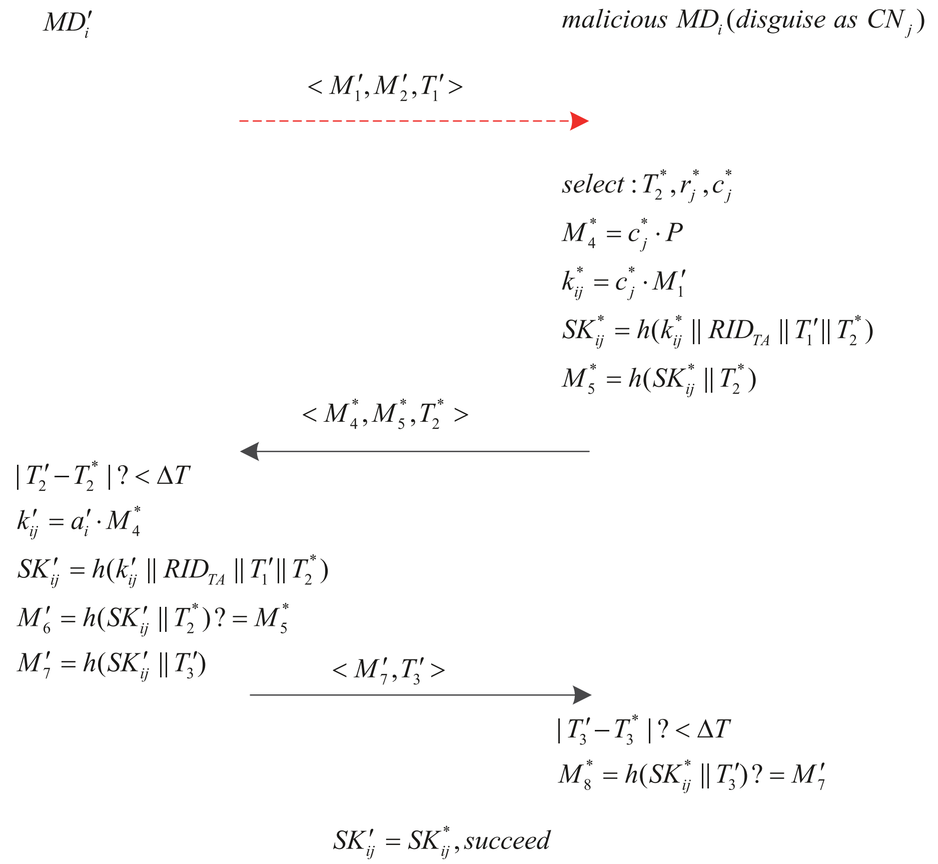

MD'_i
$malicious\ MD_i\,(disguise\ as\ CN_j)$
$< M'_1, M'_2, T'_1 >$
$select: T^*_2, r^*_j, c^*_j$
$M^*_4 = c^*_j \cdot P$
$k^*_{ij} = c^*_j \cdot M'_1$
$SK^*_{ij} = h(k^*_{ij} \| RID_{TA} \| T'_1 \| T^*_2)$
$M^*_5 = h(SK^*_{ij} \| T^*_2)$
$< M^*_4, M^*_5, T^*_2 >$
$| T'_2 - T^*_2 | ? < \Delta T$
$k'_{ij} = a'_i \cdot M^*_4$
$SK'_{ij} = h(k'_{ij} \| RID_{TA} \| T'_1 \| T^*_2)$
$M'_6 = h(SK'_{ij} \| T^*_2) ? = M^*_5$
$M'_7 = h(SK'_{ij} \| T'_3)$
$< M'_7, T'_3 >$
$| T'_3 - T^*_3 | ? < \Delta T$
$M^*_8 = h(SK^*_{ij} \| T'_3) ? = M'_7$
$SK'_{ij} = SK^*_{ij}, succeed$

Figure 3: The controller node impersonation attack in Wazid et al.'s scheme.

has the following three flaws, i.e., offline password guessing attack, controller node impersonation attack, and Incorrect authentication process. As a result, it cannot achieve mutual authentication; that is, the scheme fails to meet the security claimed by the authors.

4.1. Offline Password Guessing Attack. To achieve user friendliness, in registration phase, users are allowed to choose their own identities and passwords at will; the majority of users will choose easy-to-recall ID and PW; the combination of these low entropy ID and PW are likely to be vulnerable to offline guessing attack. A probabilistic polynomial time (PPT) adversary can offline enumerate all (ID, PW) pairs in Cartesian product $D_{id} * D_{pw}$, where D_{id} and D_{pw} represent ID space and PW space, respectively. In a 3FA protocol, we should ensure that even the MD_i and biometric have been corrupted, and the whole scheme can still resist this type attack to protect the security of user's secrets. Based on all above assumptions, the adversary can launch an offline password guessing attack through the following processes.

Step 1. We assume that the adversary A has acquired MD_i and biometric BIO_i of the user and then obtains the secret parameters $\{RID'_i, RID'_{TA}, A'_i, C_i, D_i, \tau_i, Gen(\cdot), Rep(\cdot), h(\cdot), t\}$ stored in the MD_i.

Step 2. The adversary A picks a (ID'_i, PW'_i) pair and calculates $\sigma_i = Rep(BIO_i, \tau_i)$, $k^* = D_i \oplus h(ID^*_i \parallel PW^*_i \parallel \sigma_i)$, $RPW^*_i = h(PW^*_i \parallel k^*)$, $A^*_i = A'_i \oplus h(k^* \parallel \sigma_i)$, $B^*_i = h(A^*_i \parallel RPW^*_i)$, $RID^*_{TA} = RID'_{TA} \oplus h(ID^*_i \parallel k^* \parallel \sigma_i)$, $RID^*_i = RID'_i \oplus h(PW^*_i \parallel \sigma_i)$, and $C^*_i = h(ID'_i \parallel RID'_{TA} \parallel B^*_i \parallel \sigma_i)$.

Step 3. Finally, A checks whether $C^*_i = C_i$, and if it holds, we can say that the (ID'_i, PW'_i) selected by the adversary is a legal one. Otherwise, A can choose another (ID_i, PW_i) pair to continue implementing above steps until success.

4.2. The Controller Node Impersonation Attack. In registration phase, TA picks a secret number N and calculates TA's pseudo identifier $RID_{TA} = h(ID_{TA} \parallel N)$ which is a fixed value. What is more, in predeployment phase, both MD_i and CN_j have obtained RID_{TA}; for a malicious MD_i, he/she can disguise himself/herself as CN_j to communicate with another MD'_i as shown in Figure 3.

Step 1. The malicious MD_i intercepts the first authentication message $\langle M'_1, M'_2, T'_1 \rangle$ sent by MD'_i which is ought to have been received by CN_j.

Step 2. Then MD_i can impersonate CN_j to communicate with MD'_i, MD_i selects time stamp T^*_2, random value r^*_j, and c_j^*, Then MD_i computes $M_4^* = c_j^* \cdot P$, $k_{ij}^* = c_j^* \cdot M_1'$, session key $SK_{ij}^* = h(k_{ij}^* \parallel RID_{TA} \parallel T_1' \parallel T_2^*)$, and $M_5^* = h(SK_{ij}^* \parallel T_2^*)$. Finally, MD_i forwards the constructed false message $\langle M_4^*, M_5^*, T_2^* \rangle$ to MD_i'.

Step 3. After receiving the message from MD_i, MD_i' will check $|T_2' - T_2^*|? < \Delta T$ and then calculate $k_{ij}' = a_i' \cdot M_4^*$, session key $SK_{ij}' = h(k_{ij}' \parallel RID_{TA} \parallel T_1' \parallel T_2^*)$ and $M_6' = h(SK_{ij}' \parallel T_2^*)$, and obviously M_6' equals M_5^* which means that MD_i passes the verification of MD_i'. Then MD_i' computes $M_7' = h(SK_{ij}' \parallel T_3')$ and sends the message $\langle M_7', T_3' \rangle$ to MD_i.

Step 4. Once MD_i receives the message, MD_i checks $|T_3' - T_3^*|? < \Delta T$ and computes $M_8^* = h(SK_{ij}^* \parallel T_3')$, then he/she will successfully verify that M_8^* equals the received message M_7'.

At this point, MD_i and MD_i' have completed mutual authentication and negotiated the same session key ($SK_{ij}' = SK_{ij}^*$) used in future sessions. In real life, this situation is manifested as the adversary (MD_i, e.g., a doctor) successfully disguises as another patient and sends false health information to his/her attending doctor, which is easy to cause medical accident as well as being extremely harmful to the patient.

4.3. Incorrect Authentication Process. In authentication phase, U_i computes $M_1 = a_i \cdot P$ and $M_2 = a_i + k' b_i (\text{mod} p)$ and then sends the message $\langle M_1, M_2, T_1 \rangle$ to CN_j. Normally, after CN_j receiving the message, she/he computes $b_i' = h(RID_{TA} \parallel T_1)$ and then judges the legality of M_2 via checking $M_2 \cdot P? = M_1 + b_i' \cdot Q$. But it is not hard to notice that the message $\langle M_1, M_2, T_1 \rangle$ does not contain the public key Q. Without knowledge of Q, CN_j cannot complete the judgement of signature, so that CN_j fails to authenticate U_i.

5. The Proposed Scheme

To correct these shortcomings in Section 4, we remedy the protocol of Wazid et al. from the following aspects. (1) In the predeployment phase, TA chooses a random value $x \in Z_p$ as the private key and computes the corresponding public key $Q_{TA} = x \cdot P$. (2) We add the fuzzy verifier to prevent the offline password guessing attack in login phase. (3) We adopt the more widely used fuzzy vault to protect biometric templates instead of fuzzy extractor.

There are also eight phases in our proposed scheme: predeployment, postdeployment, registration, login, authentication and key agreement, password and biometric update, and dynamic control node addition as well as dynamic IMD addition.

5.1. Predeployment Phase. TA first selects a secret 1024-bit number N and chooses the finite cyclic additional group G generated by a point P with a large prime order n over a finite field Z_p on an elliptic curve. Then TA selects the private key $x \in Z_p$ only known to itself, whose corresponding public key is $Q_{TA} = x \cdot P$ which is made public.

TA computes the value $TC_{CN_j} = h(ID_{CN_j} \parallel RTS_{CN_j} \parallel N)$ and stores $\{TC_{CN_j}, ID_{CN_j}\}$ in the memory of TA as well as CN_j and then adds the univariate polynomial $P(TC_{CN_j}, y)$ to the memory of CN_j.

The computing processes in predeployment phase of the IMD_l is the same as that of Wazid et al.'s scheme, so the details are omitted.

5.2. Postdeployment Phase. The specific process of this phase is as follows.

Firstly, IMD_l sends the message $\langle RID_{IMD_l} \rangle$ to CN_j; once CN_j receives the message, CN_j responds with the message $\langle TC_{CN_j} \rangle$. At the same time, they calculate the same shared secret key $SK_{IMD_l, CN_j} = P(RID_{IMD_l}, TC_{CN_j})$ and $SK_{CN_j, IMD_l} = P(RID_{IMD_l}, TC_{CN_j})$ on each own for future use.

5.3. User Registration Phase. In this phase, U_i registers with TA by executing ensuing procedure as shown in Figure 4.

Step 1. U_i inputs the selected ID_i and password PW_i and imprints the biometric BIO_i into the MD_i. MD_i chooses the private key $k \in Z_p$ and computes the corresponding public key $Q_u = k \cdot P$, as well as keeping the both secret. Finally, U_i submits the ID_i and Q_u to TA via the secure channel.

Step 2. After receiving the registration request from U_i, TA calculates $RID_i = h(ID_i \parallel x \parallel N)$ and stores specific $\{ID_i, Q_u\}$ of U_i in the memory. Then TA forwards the value RID_i to U_i.

Step 3. Upon receiving the message, MD_i chooses a random number K and calculates fuzzy vault parameters $Gen(Pol, BIO_i, K) = L$ and $Enc(CP, L) = V$ as well as $RPW_i = h(PW_i \parallel k)$ and $D_i = k \oplus h(ID_i \parallel PW_i \parallel K)$. Then, MD_i computes the verification value $T_i = h(h(ID_i \parallel RPW_i \parallel K) \text{mod} l)$ where $2^8 \le l \le 2^{16}$ is a medium integer which represents the capacity of the pool of the $\langle ID_i, PW_i \rangle$ pair against the offline password guessing attack in the Wazid et al.'s scheme. After the calculation of $RID_i' = RID_i \oplus h(PW_i \parallel K \parallel k)$, MD_i stores the parameters $\{T_i, D_i, RID_i', V, Q_u, l, h(\cdot), Dec(\cdot), Rec(\cdot), Gen(\cdot), Enc(\cdot)\}$.

5.4. Login Phase. As showed in Figure 5, in this phase, U_i inputs ID_i, PW_i, and the biometric BIO_i' on the MD_i. Then U_i regains the fuzzy vault parameter K' by computing the value $Dec(BIO_i', V) = Pol'$ and $Rec(Pol') = K'$. With K', MD_i continues to calculate $k = D_i \oplus h(ID_i \parallel PW_i \parallel K')$ and $RPW_i' = h(PW_i \parallel k)$ and checks $T_i? = h(h(ID_i \parallel RPW_i' \parallel K') \text{mod} l)$. If two values are not equal, MD_i refuses the login request; otherwise, MD_i believes that ID_i, PW_i, and BIO_i' are legitimate and continues to compute $RID_i = RID_i' \oplus h(PW_i \parallel K \parallel k)$. Then, MD_i generates the current timestamp T_1 and random numbers a_i and c_i. With these numbers, MD_i continues to calculate $b_i = h(RID_i \parallel T_1 \parallel ID_i \parallel ID_{CN_j})$, $M_1 = a_i \cdot P$, $M_2 = a_i + k b_i (\text{mod} p)$, $M_3 = a_i \cdot Q_{TA}$, $UID_i = (ID_i \parallel ID_{CN_j}) \oplus h(M_3)$, and $PKS_i = c_i \oplus h(RID_i \parallel M_3 \parallel T_1)$.

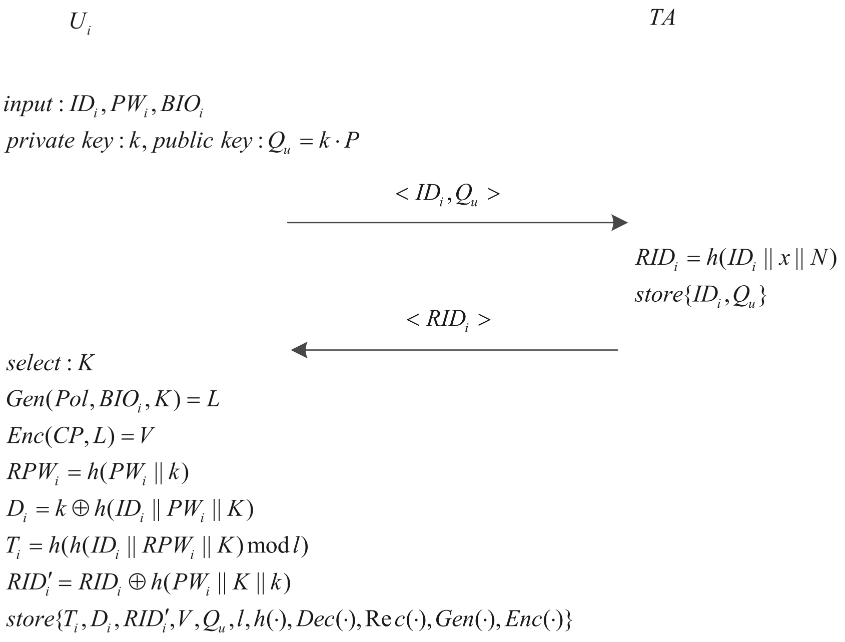

U_i
TA
$input: ID_i, PW_i, BIO_i$
$private\ key: k, public\ key: Q_u = k \cdot P$
$< ID_i, Q_u >$
$RID_i = h(ID_i \parallel x \parallel N)$
$store\{ID_i, Q_u\}$
$< RID_i >$
$select: K$
$Gen(Pol, BIO_i, K) = L$
$Enc(CP, L) = V$
$RPW_i = h(PW_i \parallel k)$
$D_i = k \oplus h(ID_i \parallel PW_i \parallel K)$
$T_i = h(h(ID_i \parallel RPW_i \parallel K) \bmod l)$
$RID_i' = RID_i \oplus h(PW_i \parallel K \parallel k)$
$store\{T_i, D_i, RID_i', V, Q_u, l, h(\cdot), Dec(\cdot), \mathrm{Re}c(\cdot), Gen(\cdot), Enc(\cdot)\}$

Figure 4: User registration phase of our scheme.

Finally, MD_i sends the message $\{M_1, M_2, PKS_i, T_1, UID_i\}$ to TA via a public channel.

5.5. Authentication and Key Agreement Phase. By executing following procedures, mutual authentication is established among U_i, TA, and CN_j, and a secure session key is negotiated between U_i and CN_j.

Step 1. After receiving the login request $\{M_1, M_2, PKS_i, T_1, UID_i\}$, TA first judges if $|T_1 - T_2| \le \Delta T$ holds, where T_2 is the current timestamp and ΔT is the maximum transmission delay. If it is invalid, TA terminates the session; otherwise, TA computes the value $ID_i^* \parallel ID_{CN_j}^* = UID_i \oplus h(x \cdot M_1)$ and retrieves Q_u^* (i.e., the public key of U_i) corresponding to ID_i^*. Then TA computes $RID_i^* = h(ID_i^* \parallel x \parallel N)$ and $b_i^* = h(RID_i^* \parallel T_1 \parallel ID_i^* \parallel ID_{CN_j}^*)$ and checks the validation of the signature by checking if the equation $M_2 \cdot P = M_1 + b_i^* \cdot Q_u^*$ holds. Specifically, the equality means that TA certifies U_i's legitimacy; otherwise, TA terminates the session. Then, TA continues to calculate $c_i = PKS_i \oplus h(RID_i \parallel x \cdot M_1 \parallel T_1)$, $M_4 = h(ID_i \parallel TC_{CN_j} \parallel c_i)$, $DID_{TA} = ID_i \oplus h(UID_i \parallel TC_{CN_j} \parallel T_2)$, and $PKS_{TA} = c_i \oplus h(TC_{CN_j} \parallel ID_i \parallel T_2)$. Finally, TA sends the message $\{M_1, DID_{TA}, PKS_{TA}, UID_i, T_2, M_4\}$ to CN_j via the public channel.

Step 2. After receiving the message from TA, CN_j first checks the validation of the condition $|T_2 - T_3| \le \Delta T$ where T_3 is the current timestamp. If it does not hold, the session is terminated here; otherwise, CN_j regains the value of ID_i and c_i by computing $ID_i^* = DID_{TA} \oplus h(UID_i \parallel TC_{CN_j} \parallel T_2)$ as well as $c_i^* = PKS_{TA} \oplus h(TC_{CN_j} \parallel ID_i^* \parallel T_2)$. Then, CN_j checks if M_4 equals the result of the computation of $h(ID_i^* \parallel TC_{CN_j} \parallel c_i^*)$. If it does not hold, CN_j terminates the session; otherwise, it means that CN_j verifies TA's legality. Then CN_j selects a random number c_j and goes on with the computation of $M_5 = c_j \cdot P$, $M_6 = c_j \cdot M_1$, the session key $SK_{ji} = h(ID_i \parallel ID_{CN_j} \parallel M_6)$, and $M_7 = h(SK_{ji} \parallel c_i \parallel T_3)$. Finally, the massage $\{M_5, T_3, M_7\}$ will be sent to U_i for authentication.

Step 3. When receiving the massage $\{M_5, T_3, M_7\}$ from CN_j, U_i will first check the validation of condition $|T_3 - T_4|? \le \Delta T$; if it holds, U_i continues to calculate the session key $SK_{ij} = h(ID_i \parallel ID_{CN_j} \parallel a_i \cdot M_5)$ and judge if the value M_7 equals $h(SK_{ij} \parallel c_i \parallel T_3)$. The final verification shows that the mutual authentication among the U_i, TA, and CN_j is accomplished and the session key $SK_{ij} = h(ID_i \parallel ID_{CN_j} \parallel a_i \cdot M_5) = h(ID_i \parallel ID_{CN_j} \parallel a_j \cdot M_1) = SK_{ji}$ is established for future sessions.

5.6. Password and Biometric Update Phase. In this phase, we allow U_i to update the password at will by the following process, which is executed locally without involving TA for security reasons.

Step 1. First, U_i inputs her/his ID_i, PW_i^{old}, and BIO_i' on the terminal. Then MD_i calculates fuzzy vault parameters $Dec(BIO_i', V) = Pol'$ and $Rec(Pol') = K'$ and regains the private key $k' = D_i \oplus h(ID_i \parallel PW_i^{old} \parallel K')$ and $RPW_i' = h(PW_i^{old} \parallel k')$. MD_i checks whether T_i equals $h(h(ID_i \parallel RPW_i' \parallel K') \bmod l)$ or not. If it does not hold, MD_i rejects the request; otherwise, MD_i claims for the new PW_i^{new}.

Step 2. When U_i inputs the new password PW_i^{new}, MD_i computes $RPW_i^{new} = h(PW_i^{new} \parallel k)$, $D_i^{new} = k \oplus h(ID_i \parallel PW_i^{new} \parallel K)$, $T_i^{new} = h(h(ID_i \parallel RPW_i^{new} \parallel K) \bmod l)$, and $RID_i'^{new} = RID_i' \oplus h(PW_i^{new} \parallel K \parallel k) \oplus h(PW_i^{new} \parallel K \parallel k)$.

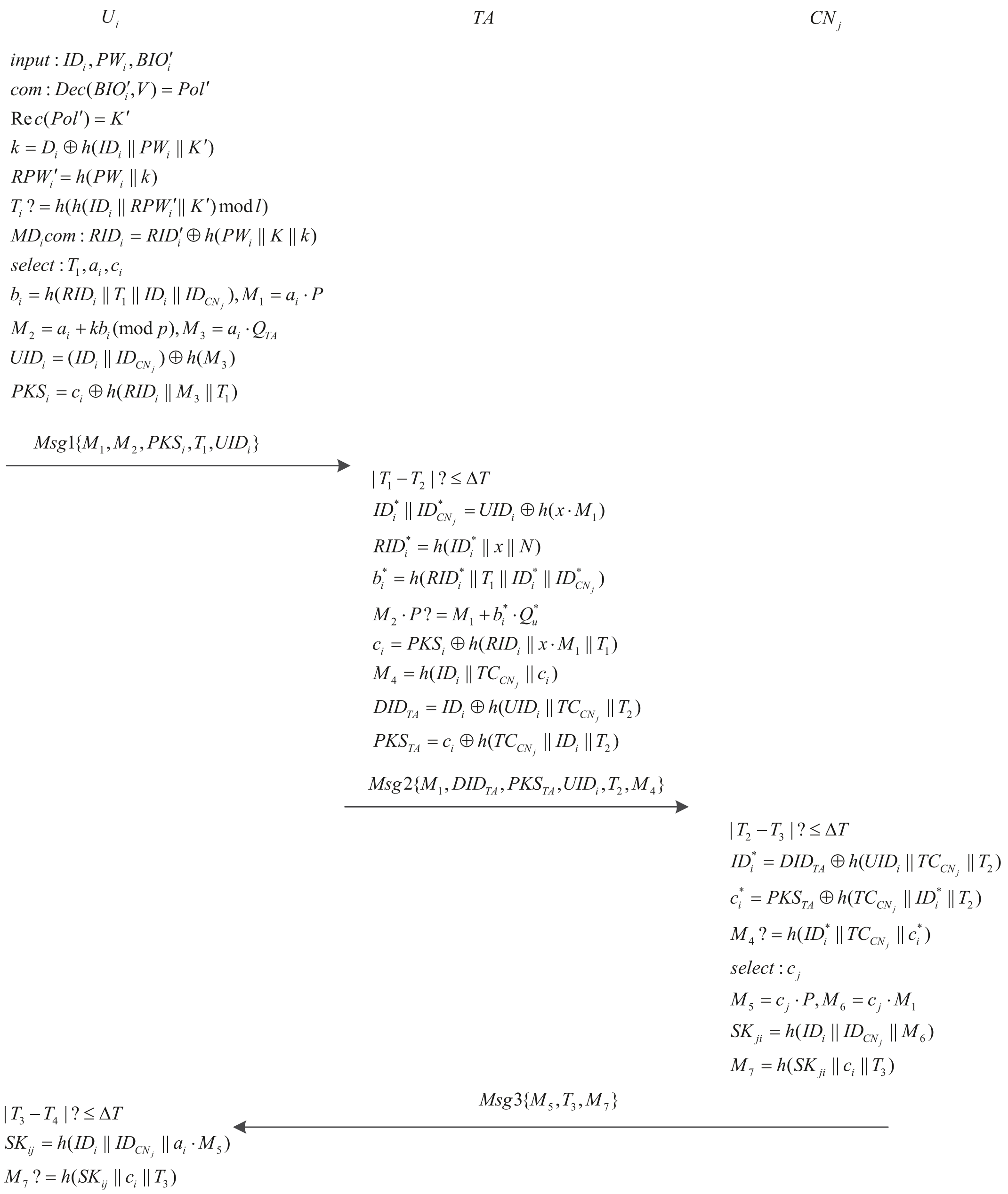

U_i
TA
CN_j
$input : ID_i, PW_i, BIO'_i$
$com : Dec(BIO'_i, V) = Pol'$
$\mathrm{Re}\,c(Pol') = K'$
$k = D_i \oplus h(ID_i \parallel PW_i \parallel K')$
$RPW'_i = h(PW_i \parallel k)$
$T_i ? = h(h(ID_i \parallel RPW'_i \parallel K') \bmod l)$
$MD_i com : RID_i = RID'_i \oplus h(PW_i \parallel K \parallel k)$
$select : T_1, a_i, c_i$
$b_i = h(RID_i \parallel T_1 \parallel ID_i \parallel ID_{CN_j}), M_1 = a_i \cdot P$
$M_2 = a_i + kb_i (\mathrm{mod}\ p), M_3 = a_i \cdot Q_{TA}$
$UID_i = (ID_i \parallel ID_{CN_j}) \oplus h(M_3)$
$PKS_i = c_i \oplus h(RID_i \parallel M_3 \parallel T_1)$
$Msg1\{M_1, M_2, PKS_i, T_1, UID_i\}$
$| T_1 - T_2 | ? \le \Delta T$
$ID_i^* \parallel ID_{CN_j}^* = UID_i \oplus h(x \cdot M_1)$
$RID_i^* = h(ID_i^* \parallel x \parallel N)$
$b_i^* = h(RID_i^* \parallel T_1 \parallel ID_i^* \parallel ID_{CN_j}^*)$
$M_2 \cdot P ? = M_1 + b_i^* \cdot Q_u^*$
$c_i = PKS_i \oplus h(RID_i \parallel x \cdot M_1 \parallel T_1)$
$M_4 = h(ID_i \parallel TC_{CN_j} \parallel c_i)$
$DID_{TA} = ID_i \oplus h(UID_i \parallel TC_{CN_j} \parallel T_2)$
$PKS_{TA} = c_i \oplus h(TC_{CN_j} \parallel ID_i \parallel T_2)$
$Msg2\{M_1, DID_{TA}, PKS_{TA}, UID_i, T_2, M_4\}$
$| T_2 - T_3 | ? \le \Delta T$
$ID_i^* = DID_{TA} \oplus h(UID_i \parallel TC_{CN_j} \parallel T_2)$
$c_i^* = PKS_{TA} \oplus h(TC_{CN_j} \parallel ID_i^* \parallel T_2)$
$M_4 ? = h(ID_i^* \parallel TC_{CN_j} \parallel c_i^*)$
$select : c_j$
$M_5 = c_j \cdot P, M_6 = c_j \cdot M_1$
$SK_{ji} = h(ID_i \parallel ID_{CN_j} \parallel M_6)$
$M_7 = h(SK_{ji} \parallel c_i \parallel T_3)$
$Msg3\{M_5, T_3, M_7\}$
$| T_3 - T_4 | ? \le \Delta T$
$SK_{ij} = h(ID_i \parallel ID_{CN_j} \parallel a_i \cdot M_5)$
$M_7 ? = h(SK_{ij} \parallel c_i \parallel T_3)$

Figure 5: Login and authentication phase of our scheme.

Step 3. After the computation, MD_i updates the value of D_i^{new}, T_i^{new}, and $RID_i'^{new}$ in the list. Above processes simulate the situation that user only wants to update the password and maintains original biometric where $BIO_i^{new} = BIO_i$. The password and biometric update phase are summarized in Figure 6.

5.7. Dynamic Controller Node Addition Phase. In this phase, we can deploy a new control node as follows.

Step 1. TA first picks a new identity for CN_j^{new}, called $ID_{CN_j}^{new}$, then TA repeats the calculation $TC_{CN_j}^{new} = h(ID_{CN_j}^{new} \parallel RTS_{CN_j}^{new} \parallel N)$ of CN_j^{new} in the predeployment phase where $RTS_{CN_j}^{new}$ is newly generated registration timestamp. Next, TA calculates the univariate polynomial $P(TC_{CN_j}^{new}, y)$.

Step 2. Finally, TA stores the parameters $\{TC_{CN_j}^{new}, ID_{CN_j}^{new}\}$ into its memory and stores the credentials $\{TC_{CN_j}^{new}, ID_{CN_j}^{new}, P(TC_{CN_j}^{new}, y)\}$ into the memory of CN_j^{new} prior to its deployment.

5.8. Dynamic IMD Addition Phase. Depending on the real situation, the patient needs to check the state of the implantable device in time to ensure that accurate health data

User(U_i)	Mobile device(MD_i)
$input : ID_i, PW_i^{old}, BIO_i'$	$MD_i com : Dec(BIO_i', V) = Pol'$ $\mathrm{Re}\,c(Pol') = K'$ $k' = D_i \oplus h(ID_i \parallel PW_i^{old} \parallel K')$ $RPW_i' = h(PW_i^{old} \parallel k')$ $T_i = h(h(ID_i \parallel RPW_i' \parallel K') \bmod l)$
$input : PW_i^{new}$	$RPW_i^{new} = h(PW_i^{new} \parallel k)$ $D_i^{new} = k \oplus h(ID_i \parallel PW_i^{new} \parallel K)$ $T_i^{new} = h(h(ID_i \parallel RPW_i^{new} \parallel K) \bmod l)$ $RID_i'^{new} = RID_i' \oplus h(PW_i^{new} \parallel K \parallel k) \oplus h(PW_i^{new} \parallel K \parallel k)$ replace D_i, T_i and RID_i' with D_i^{new}, T_i^{new} and $RID_i'^{new}$

Figure 6: Password and biometric update phase of our scheme.

is conveyed, so we often need to replace an old IMD or add a new IMD. In the case that we use a new IMD_l^{new} to replace the existing one, please refer to Wazid et al.'s scheme for the details.

6. Security Analysis

We analyze the security of our proposed scheme in this section; it fully proves that our scheme can solve the shortcomings of Wazid et al.'s scheme and resist all kinds of known attacks. The security features such as user anonymity and forward secrecy are guaranteed in our protocol.

6.1. Security Model. Our scheme involves three interacting entities, such as U_i with $\{PW_i, BIO_i, MD_i, k\}$, CN_j with TC_{CN_j}, and TA which keeps his/her private key x. Each participant can activate multiple protocol instances and run multiple session instances in parallel. The U_i^{th} is defined as the ith instance of U_i, and the same rules apply to CN_j^{th} and TA^a. All of these instances can be seen as oracles which have three states below.

(i) Accept state: when the oracle has received the last valid message of the protocol, we can say the oracle accepts the message.

(ii) Reject state: when the oracle has received any incorrect message, the oracle will reject the received message.

(iii) $\perp$ state: when the oracle outputs no answer of the queries, we say that the oracle is in an unresponsive state which is defined as $\perp$ state.

We give the security model of our scheme, which combines the security models of [33, 45].

Definition 1 (partnering). If the instances of U_i^{th} and CN_j^{th} satisfy the following three conditions meanwhile, we determine that they are partnered to each other. (1) One of the instances is the target object of session for the other instances in the protocol, that is, the partner identification of U_i^{th} is CN_j^{th} and vice versa. (2) Both instances accept the messages mutually and negotiate the same secure session key. (3) Both instances share the same session identifier.

Definition 2 (freshness). An instance called fresh must meet the following conditions. (1) Before the instance U_i^{th} accepts the protocol run and generates the session key, neither the participants U_i^{th} nor the partners of the instance U_i^{th} are completely corrupted. (2) Neither U_i^{th} nor his/her partner instances are queried of Reveal(U_i^{th}/CN_j^{th}) by the adversary or disclose the session key.

Definition 3 (correctness). When U_i^{th} and CN_j^{th} are partnered as well as accepted, they will agree on the same session key.

Definition 4 (adversary capabilities). Interaction between the adversary A and participants in the protocol is implemented via oracle queries to simulate the abilities of attackers in reality. All oracle queries are listed as follows.

(i) Execute($U_i^{th}, CN_j^{th}, TA^a$): this oracle simulates the passive attacks (such as eavesdropping, tracking) where the adversary can get all response messages $\langle Msg1, Msg2, Msg3\rangle$ exchanged during the honest execution of authentication process.

(ii) Send($U_i^{th}/CN_j^{th}/TA^a, m$): this oracle models the active attacks where the adversary can forward a modified message m to $U_i^{th}/CN_j^{th}/TA^a$. Then he/she will get the response generated from $U_i^{th}/CN_j^{th}/TA^a$ who executes the procedure of honest protocol after receiving m. Additionally, the query Send(U_i^{th}, start) initials the protocol.

(iii) Test(U_i^{th}/CN_j^{th}): this query does not model the actual attack capabilities of adversary A but rather measures the semantic security of the session key SK. For a participant instance U_i^{th}/CN_j^{th}, if the instance does not generate the session key, an undefined symbol $\perp$ will be returned. Otherwise, a uniform coin is thrown, if the result is 1, the true session key of the instance U_i^{th}/CN_j^{th} is returned; otherwise, a random number of the same length as the session key is returned. The adversary needs to guess the result of the toss to see whether he/she gets a real session key or a random number. Notice that the Test(U_i^{th}/CN_j^{th}) oracle query can only be used for fresh instance and up to once.

(iv) Reveal(U_i^{th}/CN_j^{th}): this oracle simulates the reveal of session key SK to adversary if U_i^{th}/CN_j^{th} really holds SK and has not been queried by a Test(U_i^{th}/CN_j^{th}) before. Otherwise the $\perp$ will be returned.

(v) Corrupt(U_i^{th}, a): this oracle query is used to model the corruption ability of the adversary; we assume A can get any one factor of U_i^{th} but not all.

If $a = 1$, it responses A with the password PW_i of U_i^{th}.

If $a = 2$, it responses A with all the security parameters stored in the MD_i of U_i^{th}.

If $a = 3$, it responses A with the biometric BIO_i of U_i^{th}.

If $a = 4$, it responses A with the private key k of U_i^{th}.

(vi) Corrupt(CN_j^{th}/TA^a): the adversary can get the long-term secret values of CN_j^{th}/TA^a, such as $TC_{CN_j}$of CN_j^{th} or the private key x of TA^a.

Definition 5 (random oracle). We determine the cryptographic one-way hash function H which can be accessed by all participants including A as a random oracle.

A 3FA protocol should guarantee the semantic security which is defined from Test-query. In the process run of the protocol P, A can ask the Test-query just once while other queries; i.e., Execute-query, Reveal-query, or Send-query can be asked multiple times in polynomial time. Besides, A can only make Test-query on a fresh instance. The adversary's operation is to guess the result of the coin toss in the Test-query, then we treat the event in which the adversary correctly guesses the result as a successful attack, credited as Succ(A). Only after the participants have completed the strict mutual authentication can a common session key be negotiated. The advantage of an adversary$A$ breaking the session key security of protocol P is defined as $\text{Adv}_{P,D}^{ake}(A) = 2\text{Pr}[\text{Succ}(A)] - 1$ where D denotes the password space whose distribution follows a Zipf's law [50].

Theorem 6 (semantic security). *Given a 3FA protocol P, if the advantage* $\text{Adv}_{P,D}^{ake}(A)$ *of an arbitrary PPT adversary breaking the session key security of the protocol is at most a negligible amount n(l) larger than* $C' \cdot q_{send}^{s'}$, *then we believe that the P satisfies the semantic security, where the* q_{send} *denotes the number of active attacks by the PPT adversary and n(l) represents a negligible function for the security parameter l.*

$$\text{Adv}_{P,D}^{ake}(A) \le C' \cdot q_{send}^{s'} + n(l) \tag{1}$$

As shown above, $C' = 0.062239$ and $s' = 0.155478$ represent the Zipf parameters put forward by Wang et al. [50].

6.2. Security Proof. Assuming that DDH holds in a cyclic group, the public key encryption algorithm used in the protocol is CCA secure, and the signature algorithm is unforgeable for adaptively chosen messages. Here we prove Theorem 6 by simulating several mixing games. The mixing games start with a real attack game, and then we gradually modify the simulation rules in each game until the adversary's attack advantage to distinguish the correct session key from a random key of the same length becomes zero and then the game ends. For two adjacent mixing games, we will calculate the upper bound of the attacker's advantage gap and finally calculate the upper bound of adversary's attack on this 3FA protocol. We use Δ_i to indicate the difference between mixing games G_i and G_{i+1} and use $Adv_i(A)$ to denote the advantage of A in hybrid games G_i.

(i) G_0: this experiment is the start game which simulates the real attack mode of the adversary we demonstrate in Section 6. So, we can get

$$Adv_{P,D}^{ake}(A) = Adv_0(A) \tag{2}$$

(ii) G_1: in this game, we simulate all random oracles H in the protocol by maintaining a hash query list l_{hash}. Besides, we also simulate a private hash oracle H' by holding another list l'_{hash} which records the Hash-query directly implemented by the adversary. Obviously, the game is indistinguishable from a real one, so we have

$$\Delta_1 = \left|Adv_1(A) - Adv_1(A)\right| \le n(l) \tag{3}$$

(iii) G_2: we exclude some impossible collisions in the G_2, i.e., the collisions of messages $\langle Msg1, Msg2, Msg3\rangle$ in sessions and the collisions in the outputs of Hash-query. According to the birthday paradox, we have

$$\Delta_2 = \left|Adv_2(A) - Adv_1(A)\right| \le n(l) \tag{4}$$

(iv) G_3: we will revise the session simulation rules for the passive attacks that the adversary asks through the Execute-query. We suppose that U_i constructs the $Msg1$ using another (ID_i^*, PW_i^*) pair chosen from Cartesian product $D_{id} * D_{pw}$ instead of the real one. That is, parameters $k^* = D_i \oplus h(ID_i^* \parallel PW_i^* \parallel K)$, $RID_i^* = RID_i' \oplus h(PW_i^* \parallel K \parallel k)$, and $b_i = h(RID_i \parallel T_1 \parallel ID_i^* \parallel ID_{CN_j})$ are calculated and so that the signature can be calculated as $M_2 = a_i + k^* b_i^* (\text{mod} p)$. Upon receiving the message $Msg1$, TA continues to simulate session with the false identity. If TA is lucky enough to guess the real (ID_i, PW_i), the game is terminated. The real (ID_i, PW_i) and the pseudo (ID_i^*, PW_i^*) can be seen as two challenge messages for the encryption algorithm, so the difference between the games G_3 and G_2 is at most the advantage of A breaking the encryption algorithm's CPA security of the signature. And the CPA security of the signature can be reduced to the DDH hypothesis. So, we can conclude

$$\Delta_3 = \left|Adv_3(A) - Adv_2(A)\right| \leq n(l) \tag{5}$$

(v) G_4: in this game, we continue to revise the simulation session rules in passive attacks. We use the private hashing function H' to compute the session key SK_{ij} without the Diffie-Hellman parameters a_i and c_j, that is, $SK_{ij} = H'(ID_i \parallel ID_{CN_j})$. Since we have excluded the collisions in the previous game, only A computes the valid Diffie-Hellman parameters $a_i c_j \cdot P$ and sends the query $(ID_i, ID_{CN_j}, a_i c_j \cdot P)$ to H and can A distinguish the difference between G_4 and the previous one. But the capability of A is limited by the hardness of DDH security where given g^a, g^b, g^{ab} and g^a, g^b, g^c, A cannot tell g^{ab} from g^c. Based on the intractability of the DDH problem, we have

$$\Delta_4 = \left|Adv_4(A) - Adv_3(A)\right| \leq n(l) \tag{6}$$

(vi) G_5: in this game, we start to revise the simulation session rules by active attacks. We take the Send$(TA, (Msg1))$ as the example, and if U_i is not corrupted and A correctly constructs the signature, then we say that A wins the game and terminate the simulation. Based on the unforgeability security of the signature, then we have

$$\Delta_5 = \left|Adv_5(A) - Adv_4(A)\right| \leq n(l) \tag{7}$$

(vii) G_6: we continue to revise the simulation session rules in active sessions. We acknowledge that A wins the game when A has successfully fabricated the message $Msg\{M_1, DID_{TA}, PKS_{TA}, UID_{TA}, M_4\}$ and sent it to TA. We use the private hash function H' to simulate the active sessions. The authenticator M_4 is calculated as $M_4 = H'(ID_i \parallel TC_{CN_j} \parallel c_i)$ where the c_i is randomly selected from a cyclic group. When the c_i corresponds to a fake PW_i^*, the distribution of c_i is indistinguishable from the uniform distribution on a cyclic group. Then we have

$$\Delta_6 = \left|Adv_6(A) - Adv_5(A)\right| \leq n(l) \tag{8}$$

(viii) G_7: we change the simulation rules in active sessions for the last time in this game. If A correctly forge the message $Msg3\{M_5, M_7, T_3\}$, then we say A wins the game and terminate the game. The authenticator M_7 contains the random number c_i which is unknown to A. We have eliminated this situation in previous game. So, we have

$$\Delta_7 = \left|Adv_7(A) - Adv_6(A)\right| \leq n(l) \tag{9}$$

The only way to succeed in this game is to obtain the parameters in MD_i and guess U_i's real password. A is unable to get any information of PW_i from simulation, according to the Zipf law, we get

$$Adv_8(A) \leq n(l) \leq C' \cdot q_{send}^{s'} \tag{10}$$

Therefore, Theorem 6 is proved.

6.3. Other Discussions. In this aspect, we demonstrate that our protocol can resist various known attacks as well as achieve security characteristics such as user anonymity, forward security, and key security.

6.3.1. Privileged Insider Attack. In the registration phase of our protocol, U_i sends the message consisting of the identity ID_i and corresponding public key Q_u without any knowledge of the password PW_i, so that TA has no approach to derive PW_i. Obviously, our scheme can withstand the privileged insider attack.

6.3.2. Stolen-Verifier Attack. In this attack mode, an attacker can steal the verification parameters stored by TA to cheat U_i, while we just put ID_i and Q_u in the verification table which contains no knowledge about password PW_i. Therefore, our scheme is immune to the stolen-verifier attack.

6.3.3. Offline Password Guessing Attack with Stolen Mobile Device. For this situation, we usually suppose that the A has gained the security parameters $\{T_i, D_i, RID_i', Q_u, V, l\}$ stored in the MD_i and the biometric BIO_i simultaneously; A can eavesdrop authentication messages $\langle Msg1, Msg2, Msg3\rangle$ transmitted via the public channel.

A picks a candidate $\langle ID_i^*, PW_i^*\rangle$ pair in the Cartesian product $D_{id} * D_{pw}$ and computes $Dec(BIO_i, V) = Pol$, $Rec(Pol) = K$, $k^* = D_i \oplus h(ID_i^* \parallel PW_i^* \parallel K)$, and $RPW_i^* = h(PW_i^* \parallel k)$ as well as the verification value $T_i^* = h(h(ID_i^* \parallel RPW_i^* \parallel K) \bmod l)$. In general, A can determine the chosen $\langle ID_i^*, PW_i^*\rangle$ pair's validation by checking if T_i^* equals the stored value T_i. If it holds, it means that A has guessed the correct $\langle ID_i^*, PW_i^*\rangle$ of U_i successfully; otherwise, he/she can pick another $\langle ID_i^*, PW_i^*\rangle$ pair continuing to attack. However, we introduce the fuzzy-verifier $T_i = h(h(ID_i \parallel RPW_i \parallel K) \bmod l)$ which is effective in leaving adequate candidates for A to identify and thus making it impossible for a PPT adversary to successfully guess the password.

Hence, the offline password guessing attack can not damage U_i's security.

6.3.4. Undetectable Online Password Guessing Attack. In the proposed scheme, once A tries initialing the protocol, he/she needs to make sure that the chosen password PW_i^* is valid to construct the verification signature $M_2 = a_i + kb_i (\bmod p)$ which will pass authentication of TA. Otherwise, the wrong PW_i^* will be observed easily by TA. So, our scheme can withstand the undetectable online password guessing attack.

6.3.5. Modification Attack. In our protocol, even A intercepts the messages transmitted in the channel, it is still impossible for A to construct $Msg1\{M_1, M_2, PKS_i, T_1, UID_i\}$, $Msg2\{M_1, DID_{TA}, PKS_{TA}, UID_i, T_2, M_4\}$, and $Msg3\{M_5, T_3, M_7\}$ which are protected by the secret value, private key or hash functions to pass the message verification. For example, in $Msg1$ A is unable to calculate the value $M_2 = a_i + kb_i(\mathrm{mod}\,p)$, since $b_i = h(RID_i \parallel T_1 \parallel ID_i \parallel ID_{CN_j})$ where $RID_i = RID_i' \oplus h(PW_i \parallel K \parallel k) = h(ID_i \parallel x \parallel N)$ consists of secret values only known to U_i or TA such as PW_i, private key k, and x, so that A's login request will be rejected by TA. Similarly, A cannot construct the valid verification parameters M_4 without knowledge of TC_{CN_j} or M_7 due to the hardness of ECCDH problem introduced in Section 2.2. Thus, all modified messages will be detected and rejected by receiver simultaneously.

In conclusion, modification attack is impossible in our scheme.

6.3.6. User Impersonation Attack. We suppose that A plans to impersonate as a legitimate user U_i to interact with TA. The key step is to construct a valid value M_2 to pass the verification of TA. However, A is unable to calculate $M_2 = a_i + kb_i(\mathrm{mod}\,p)$ without b_i. To get $b_i = h(RID_i \parallel T_1 \parallel ID_i \parallel ID_{CN_j})$, he/she needs to know the most of long-term values. Therefore, our proposed scheme is immune to the user impersonation attack.

6.3.7. Control Node Impersonation Attack. We have analyzed that the malicious MD_i may successfully impersonate CN_j to cheat another MD_i^* in Wazid et al.'s scheme. On the one hand, both MD_i and CN_j hold the same parameter RID_{TA} which composes the correct verification value $M_5 = h(SK_{ij} \parallel T_2)$ and $SK_{ij} = h(k_{ij} \parallel RID_{TA} \parallel T_1 \parallel T_2)$. On the other hand, in Wazid et al.'s scheme, the essential parameter c_j is not verified when it is sent to MD_i. But in our scheme, this attack mode cannot be implemented, and the malicious MD_i is unable to fabricate M_7 without knowing c_i of MD_i^*, so we solve the potential pitfall in Wazid et al.'s scheme.

From another point of view, an adversary A cannot construct the verification value M_7 due to the hardness of ECCDH, so A fails to impersonate a CN_j. In a word, the control node impersonation attack has no threat to our protocol.

6.3.8. TA Impersonation Attack. For A, it is computationally infeasible to get the value $M_4 = h(ID_i \parallel TC_{CN_j} \parallel c_i)$ which is protected by hash function and critical parameters TC_{CN_j} as well as nonce c_i. The c_i can be derived from two functions as $c_i = PKS_{TA} \oplus h(TC_{CN_j} \parallel ID_i \parallel T_2) = PKS_i \oplus h(RID_i \parallel M_5 \parallel T_1)$, but even A has intercepted the parameters PKS_{TA}, UID_i, and PKS_i; he/she still cannot calculate c_i without RID_{CN_j}, RID_i, or M_3, and then M_4 cannot be computed. In short, our scheme is immune to the TA impersonation attack.

6.3.9. Denial-of-Service (DoS) Attack. Before U_i's login request is sent to TA, the password PW_i^*, identity ID_i^*, and biometric BIO_i^* input in the terminal by U_i will be determined locally by verifying the value of T_i^*. According to the protocol, only when $T_i^* = T_i$, the process will continue. Hence, our protocol can withstand such an attack.

6.3.10. Replay Attack. When an adversary A wants to send the intercepted messages $\langle Msg1, Msg2, Msg3\rangle$ to receiver again, it will fail to pass the protection of timestamp $\langle T_1, T_2, T_3, T_4\rangle$. All these intercepted messages will be seen overdue. So, our scheme can withstand this attack effectively.

6.3.11. Mutual Authentication. Mutual authentication means that before the doctor gets health information from CN_j, U_i, TA, and CN_j have confirmed the legitimacy of the other two parties. In our protocol, TA holds the public key Q_u to verify the signature M_2, and then U_i is authenticated. In the same way, we take the verification values M_4 and M_7 which consist of some parameters only known to them just like private key or nonce to accomplish mutual authentication. That is, when they affirm that each other is legal, a secure session key is negotiated between U_i and CN_j.

6.3.12. Known Key Security. Our entire protocol's purpose is to ensure the safety of subsequent medical information delivery after mutual authentication is completed. The session key $SK_{ij} = h(ID_i \parallel ID_{CN_j} \parallel a_i c_j \cdot P)$ which depends on random numbers a_i and c_j can be different and independent in every key agreement phase. Even some session keys are disclosed, in the next session, the SK_{ij} will maintain secure. Hence, our protocol guarantees the security of the session key.

6.3.13. Perfect forward Secrecy. At the final step of authentication phase, U_i and CN_j negotiate a session key $SK_{ij} = h(ID_i \parallel ID_{CN_j} \parallel a_i c_j \cdot P) = h(ID_i \parallel ID_{CN_j} \parallel a_i \cdot M_5) = h(ID_i \parallel ID_{CN_j} \parallel c_j \cdot M_1)$. To calculate the session key with $M_1 = a_i \cdot P$, A has to solve the ECCDH problem as we showed before. It follows that even long-term keys of U_i and CN_j are disclosed, the session key still maintains secure. Hence, the proposed protocol achieves perfect forward secrecy.

6.3.14. User Anonymity. In the proposed protocol, we conceal the identity ID_i in the $b_i = h(RID_i \parallel T_1 \parallel ID_i \parallel ID_{CN_j})$, $UID_i = (ID_i \parallel ID_{CN_j}) \oplus h(M_3)$, and $DID_{TA} = ID_i \oplus h(UID_i \parallel TC_{CN_j} \parallel T_2)$. It shows that ID_i is protected by private key x in $RID_i = h(ID_i \parallel x \parallel N)$, nonce a_i in $M_3 = a_i \cdot Q_{TA}$. That means in addition to the U_i, TA, and CN_j, no one knows the ID_i. So, our scheme achieves user anonymity.

6.3.15. User Untraceability. In the proposed protocol, messages $Msg1\{M_1, M_2, PKS_i, T_1, UID_i\}$, $Msg2\{M_1, DID_{TA}, PKS_{TA}, UID_i, T_2, M_4\}$, and $Msg3\{M_5, T_3, M_7\}$ transmitted among U_i, TA, and CN_j are dynamic and different from before ones because the sender randomly selects a number to compose messages. For instance, in $Msg1$, the introductions of a_i and c_j make the parameters different for each login phase to prevent A from using static values to track user. In short, it is impossible for A to track U_i in our scheme.

6.3.16. Biometric Template Privacy. Our scheme can effectively maintain the privacy of biometric BIO_i. On the one hand, user does not offer CN_j the biometric template, and there is no knowledge about U_i's biometric template in the

Table 2: Comparison of security features.

	Wang et al.'s scheme [13]	Wazid et al.'s scheme [14]	Our scheme
Mutual authentication	×	√	√
Known key security	√	√	√
Perfect forward secrecy	√	√	√
User anonymity	√	√	√
Biometric template privacy	√	√	√
Resisting modification attack	√	√	√
Resisting user impersonation attack	×	√	√
Resisting server(CN) impersonation attack	√	×	√
Resisting man-in-the-middle attack	√	√	√
Resisting stolen-verifier attack	√	√	√
Resisting privileged insider attack	√	√	√
Resisting replay attack	√	√	√
Resisting modification attack	√	√	√
Resisting password guessing attack	√	×	√
Resisting secure key agreement	×	×	√

Table 3: Comparison of computation cost.

scheme	Wang et al.'s [13]	Wazid et al.'s [14]	Ours
$U_i(MD_i)$	$T_{bp} + 3T_{em} + 5T_H + T_{se}$	$3T_{em} + 12T_H + T_{as} + T_{me}$	$3T_{em} + 10T_H + T_{as} + T_{me}$
TA	-	-	$4T_{em} + 7T_H + T_{as}$
CN_j	$T_{bp} + 2T_{em} + 5T_H + T_{se}$	$4T_{em} + 5T_H + T_{as}$	$2T_{em} + 5T_H$
Overall	$2T_{bp} + 5T_{em} + 10T_H + 2T_{se}$	$7T_{em} + 17T_H + 2T_{as} + T_{me}$	$9T_{em} + 22T_H + 2T_{as} + T_{me}$

memory of CN_j. On the other hand, we firstly use fuzzy vault to convert the form of biometric template to V. Even A obtains the V form MD_i, he/she still cannot recover the biometric template because the algorithms of fuzzy vault are one-way operations. Moreover, the biometric template itself is difficult to lose or falsify. In short, our protocol guarantees the privacy of biometric template.

7. Features and Efficiency Comparison

This section shows the comparisons of our scheme and other two related works (Wang et al. [13], Wazid et al. [14]) in efficiency and the advantages/disadvantages showed in Tables 3 and 2, respectively. Specifically, we analyze the computation cost from the point of time complexity to compare the efficiency. What needs to be explained is that we only focus on the login and authentication phases and ignore the bit-XOR operation due to its low computation consumption. Besides, we use the symbols of T_{em}, T_H, T_{bp}, T_{se}, T_{me}, and T_{as} to represent the time cost of elliptic curve point multiplication, hash function, bilinear pairing, symmetric key encryption/decryption, modular exponentiation, and asymmetric key encryption/decryption, respectively.

From Tables 2 and 3, it could be seen that although the calculation cost of our scheme is a little higher than the other two solutions, we have greatly satisfied various security standards in terms of security, which is superior to Wang et al.'s protocol [13] in resisting impersonation attack and achieving mutual authentication. And our scheme makes up for the flaws we analyzed in Wazid et al.'s protocol [14]. In general, our protocol is more suitable for use in implantable medical system, within the acceptable computational energy consumption of the devices.

8. Conclusion

We take the most recent scheme of Wazid et al. as a typical example to show the subtlety of the design of 3FA for the implantable medical system. We have found that the scheme cannot resist three types of drawbacks, i.e., password guessing attack, controller node impersonation attack, and the incorrect authentication process. Then we have presented a trusted authority assisted 3FA protocol for the implantable medical system. Specifically, we have made the following amendments. TA is introduced in the authentication phase of the newly proposed solution. We have also replaced fuzzy extractor with the more widely applied fuzzy vault to the biometrics. The new protocol is provably secure under DDH assumption; the efficiency comparison and features analysis indicate that while a little efficiency is sacrificed, our protocol satisfies all the required security features. Overall, our new protocol is suitable for use in the implantable medical system.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All the authors have contributed equally to this work.

Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61672433) and Basic Research Project from Science and Innovation Council of Shenzhen (nos. 201703063000511 and 201703063000517).

References

[1] R. Altawy and A. M. Youssef, "Security tradeoffs in cyber physical systems: a case study survey on implantable medical devices," *IEEE Access*, vol. 4, pp. 959–979, 2016.

[2] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "SoK: security and privacy in implantable medical devices and body area networks," in *Proceedings of the 35th IEEE Symposium on Security and Privacy (SP '14)*, pp. 524–539, San Jose, Calif , USA, May 2014.

[3] P. K. Sahoo, "Efficient security mechanisms for mhealth applications using wireless body sensor networks," *Sensors*, vol. 12, no. 9, pp. 12606–12633, 2012.

[4] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social Attribute Aware Incentive Mechanism for Device-to-Device Video Distribution," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1908–1920, 2017.

[5] J. Xiong, Y. Zhang, L. Lin et al., "ms-PoSW: A multi-server aided proof of shared ownership scheme for secure deduplication in cloud," *Concurrency & Computation Practice & Experience*, no. 5, Article ID e4252, 2017.

[6] S. Kumari, X. Li, F. Wu, A. K. Das, K.-K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.

[7] G. Zheng, R. Shankaran, M. A. Orgun, L. Qiao, and K. Saleem, "Ideas and Challenges for Securing Wireless Implantable Medical Devices: A Review," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 562–576, 2017.

[8] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1-1, 2017.

[9] D. Wu, F. Zhang, H. Wang, and R. Wang, "Security-oriented opportunistic data forwarding in Mobile Social Networks," *Future Generation Computer Systems*, vol. 87, pp. 803–815, 2018.

[10] D. Dolev and A. C.-C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[11] L. Wu, X. Du, M. Guizani, and A. Mohamed, "Access Control Schemes for Implantable Medical Devices: A Survey," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1272–1283, 2017.

[12] E. Marin, D. Singelée, F. D. Garcia, T. Chothia, R. Willems, and B. Preneel, "On the (in)security of the latest generation implantable cardiac defibrillators and how to secure them," in *Proceedings of the 32nd Annual Computer Security Applications Conference, ACSAC 2016*, pp. 226–236, Los Angeles, Calif, USA, December 2016.

[13] C. Wang and Y. Zhang, "New authentication scheme for wireless body area networks using the bilinear pairing," *Journal of Medical Systems*, vol. 39, no. 11, article 136, 2015.

[14] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, 2018.

[15] A. Perrig, "The tesla broadcast authentication protocol," *Rsa Cryptobytes*, vol. 20, no. 2, p. 2002, 2005.

[16] C.-C. Lee, C.-W. Hsu, Y.-M. Lai, and A. Vasilakos, "An enhanced mobile-healthcare emergency system based on extended chaotic maps," *Journal of Medical Systems*, vol. 37, no. 5, article 9973, 2013.

[17] F. Wu, X. Li, L. Xu et al., "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server," *Computers & Electrical Engineering*, vol. 63, pp. 168–181, 2017.

[18] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Computer Networks*, 2017.

[19] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, vol. 63, pp. 56–75, 2016.

[20] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[21] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 26, no. 1, pp. 96–99, 1983.

[22] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO'85*, H. C. Williams, Ed., vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, Springer, 1986.

[23] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[24] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Generation Computer Systems*, 2016.

[25] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 332–342, 2014.

[26] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 12, pp. 2327–2339, 2014.

[27] B. Hu, D. Wong, Z. Zhang, and X. Deng, "Key replacement attack against a generic construction of certificateless signature," in *Information Security and Privacy*, vol. 4058 of *Lecture Notes in Computer Science*, pp. 235–246, Springer, Berlin, Germany, 2006.

[28] D. He, S. Zeadally, N. Kumar, and J. H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Systems Journal*, vol. 11, no. 4, pp. 1–12, 2016.

[29] X. Li, M. H. Ibrahim, S. Kumari, and R. Kumar, "Secure and efficient anonymous authentication scheme for three-tier mobile healthcare systems with wearable sensors," *Telecommunication Systems*, vol. 67, no. 3, pp. 1–26, 2018.

[30] C.-T. Li, C.-C. Lee, C.-Y. Weng, and S.-J. Chen, "A Secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-Healthcare systems," *Journal of Medical Systems*, vol. 40, no. 11, article 233, 2016.

[31] Q. Jiang, J. Ma, and C. Yang, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers Electrical Engineering*, 2017.

[32] 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.

[33] F. Wei, P. Vijayakumar, J. Shen, R. Zhang, and L. Li, "A provably secure password-based anonymous authentication scheme for wireless body area networks," *Computers and Electrical Engineering*, vol. 65, pp. 322–331, 2018.

[34] 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.

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

[36] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Networks*, vol. 27, pp. 159–194, 2015.

[37] T. Chen, C. Lee, M. Hwang et al., "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, 2013.

[38] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-I. Fan, "An extended multi-server-based user authentication and key agreement scheme with user anonymity," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 1, pp. 119–131, 2013.

[39] F.-S. Wei, Q. Jiang, R.-J. Zhang, and C.-G. Ma, "A privacy-preserving multi-factor authenticated key exchange protocol with provable security for cloud computing," *Journal of Information Science and Engineering*, vol. 33, no. 4, pp. 907–921, 2017.

[40] X. Li, J. Niu, S. Kumari et al., "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network & Computer Applications*, vol. 103, no. 1, pp. 194–204, 2018.

[41] Q. Jiang, Z. Chen, B. Li et al., "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence & Humanized Computing*, pp. 1–13, 2017.

[42] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[43] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and provably secure three-factor user authentication scheme for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 1–20, 2016.

[44] C. Lee, C. Chen, P. Wu, and T. Chen, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices," *IET Computers & Digital Techniques*, vol. 7, no. 1, pp. 48–55, 2013.

[45] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: an overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2015.

[46] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography. An International Journal*, vol. 38, no. 2, pp. 237–257, 2006.

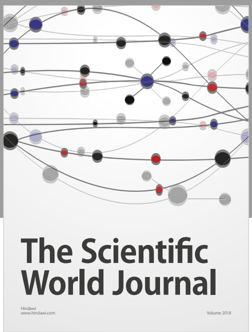
[47] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2008.

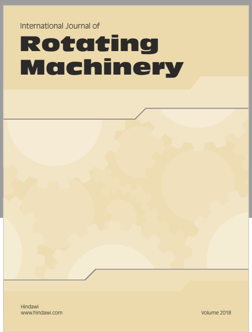
[48] J. Yu, G. Wang, Y. Mu, and W. Gao, "An efficient generic framework for three-factor authentication with provably secure instantiation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2302–2313, 2014.

[49] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," in *Advances in Cryptology —CRYPTO' 92*, vol. 740 of *Lecture Notes in Computer Science*, pp. 471–486, 1993.

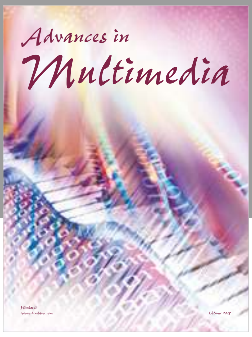
[50] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipfs law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

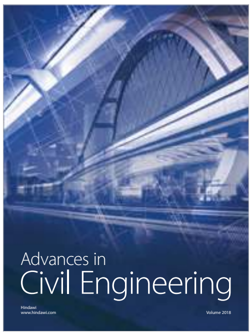
Journal of
Engineering

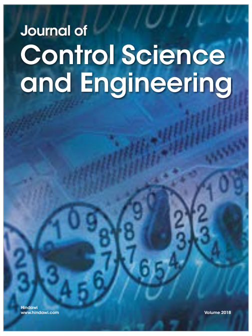

The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

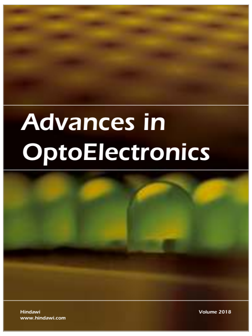

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

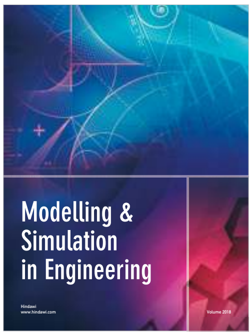
Journal of
Robotics

Journal of
Electrical and Computer Engineering


Advances in
OptoElectronics

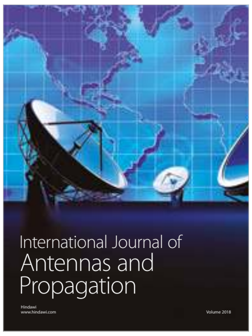
VLSI Design

International Journal of
Navigation and Observation

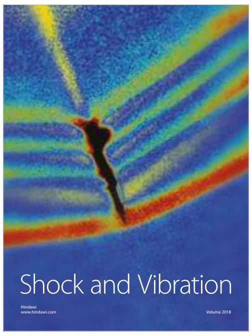

Modelling &
Simulation
in Engineering

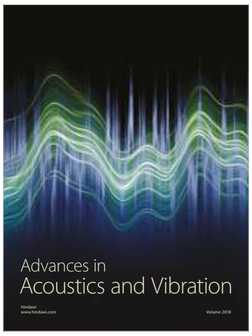
International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration