Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8818704, 13 pages
https://doi.org/10.1155/2020/8818704

Research Article

Provably Secure Crossdomain Multifactor Authentication Protocol for Wearable Health Monitoring Systems

Hui Zhang,[1] **Yuanyuan Qian**,[2] **and Qi Jiang**[2]

[1]*School of Information Engineering, Yulin University, Yulin 719000, China*
[2]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Hui Zhang; zhanghui@yulinu.edu.cn

Received 25 July 2020; Revised 25 August 2020; Accepted 3 September 2020; Published 24 September 2020

Academic Editor: Weizhi Meng

Wearable health monitoring systems (WHMSs) have become the most effective and practical solutions to provide users with low-cost, noninvasive, long-term continuous health monitoring. Authentication is one of the key means to ensure physiological information security and privacy. Although numerous authentication protocols have been proposed, few of them cater to crossdomain WHMSs. In this paper, we present an efficient and provably secure crossdomain multifactor authentication protocol for WHMSs. First, we propose a ticket-based authentication model for multidomain WHMSs. Specifically, a mobile device of one domain can request a ticket from the cloud server of another domain with which wearable devices are registered and remotely access the wearable devices with the ticket. Secondly, we propose a crossdomain three-factor authentication scheme based on the above model. Only a doctor who can present all three factors can request a legitimate ticket and use it to access the wearable devices. Finally, a comprehensive security analysis of the proposed scheme is carried out. In particular, we give a provable security analysis in the random oracle model. The comparisons of security and efficiency with the related schemes demonstrate that the proposed scheme is secure and practical.

1. Introduction

The advance in technologies such as sensing devices and wireless communication has propelled the wide application of Internet of things in the medical field [1–3]. One of the typical applications is wearable health monitoring systems (WHMSs), which is an effective and practical solution to provide users with ubiquitous, low-cost, noninvasive, long-term continuous health monitoring.

In the classic WHMS model [4], there are three types of participants in a single security domain, i.e., wearable device (WD), cloud server (CS), and mobile device (MD). Typically, various WDs, such as smart bracelets and smart shoes worn on users, can send the collected data to CS via the MD held by the users through Bluetooth, Wi-Fi, or other wireless networks [5]. The CS, as a trusted entity, is mainly in charge of device registration and private information storage. A MD (such as a smartphone) connected to the Internet can access the WDs with the aid of CS.

To achieve ubiquity, it is impractical to deploy a single-domain WHMS which includes all entities. In this paper, we mainly focus on multidomain WHMSs (see Figure 1). Without loss of generality, we suppose that there are two different domains, i.e., D1 and D2. The patient in domain D1 has a variety of WDs for collecting physiological data, while in another domain D2, the doctor monitors the patient through the MD and analyzes the patient's health data for medical treatment.

Although WHMSs bring great convenience to people, they also pose many security and privacy issues, such as sensitive personal information leakage and unauthorized access to device information [6]. Therefore, as one of the key means to fulfill data security and privacy protection [7], the authentication protocol is the focus of this paper.

To this end, numerous authentication protocols have been proposed in [8–10]. Most of them mainly concern a single domain where the wearable device collecting data and the mobile device accessing data held by the user are registered

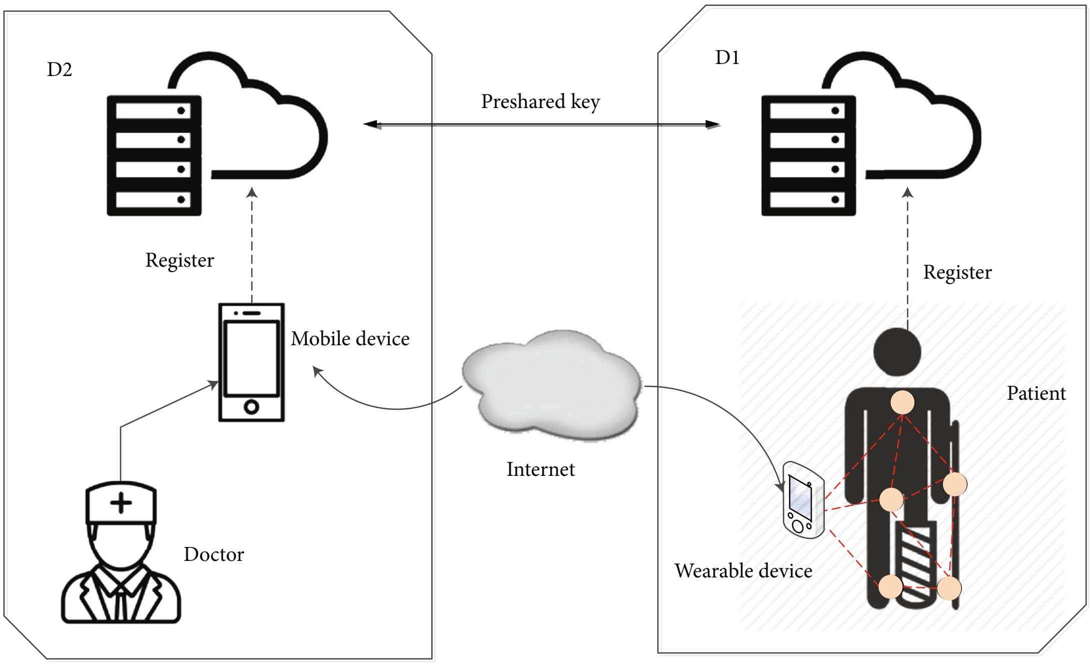


Figure 1: System model of crossdomain cloud-assisted WHMSs.

with the same server. However, in this paper, the two may be from two different domains. That is, few of them fit for multidomain WHMSs. Therefore, it is urgent to propose a multidomain authentication protocol for WHMSs.

1.1. Related Work. In order to resist malicious attacks on communication between wearable devices and smart devices, a number of authentication and key agreement (AKA) protocols for WHMSs have been put forward.

Kumar et al. [11] presented a two-factor authentication protocol based on a password and smart card (i.e., E-SAP), in which only symmetric key primitives are involved to achieve mutual authentication and key establishment. Li et al. [12] revealed that many previous schemes could not hide the user's identity information during the login session phase. Therefore, in order to protect the privacy of user identity, the dynamic identity-based AKA scheme was proposed. Amin et al. [13] designed a two-way AKA protocol for a medical monitoring system to realize the anonymity of medical staff. However, Jiang et al. [14] analyzed Amin et al.'s scheme [13] and pointed out that it could not prevent mobile device stealing attacks and sensor key exposure. Once a smart device is stolen or lost, it may lead to sensitive data leakage in the device. In order to mitigate the above situation, the biometric is introduced as the third authentication factor, resulting in a large number of three-factor authentication protocols [15–18].

In recent years, the rapid development of cloud technology has made it possible to transfer computation and storage burdens of wearable devices to cloud servers, which greatly reduces the computation cost of deploying WHMSs. To this end, cloud-assisted AKA protocols are proposed.

In 2016, the yoking proof-based AKA protocol was proposed in [19], which is applied to the deployment of wearable devices with the aid of cloud servers. Specifically, local authentication is performed between the mobile device and two wearable devices, while remote authentication is performed by a cloud server. In the same year, a new asymmetric three-party authentication scheme for mutual authentication between wearable devices and mobile devices was proposed in [20]. But in [21], it is pointed out that one of the hypotheses in [19] is impractical; that is, a long-term key shared between the mobile devices and the wearable device is required before the protocol starts. In addition, in terms of security, the scheme in [19] is not resilient to desynchronization attacks. Moreover, it is also revealed in [20] that an out-of-band channel is needed in the authentication phase of the scheme in [21], while in general, it is assumed that a secure channel is only needed in the registration phase.

In 2017, Wu et al. [20] provided a cloud server-assisted AKA scheme for the wearable computing, which realizes mutual authentication and anonymity for the wearable device. In their scheme, the cloud server can be considered a trusted entity. In 2018, Srinivas et al. [22] proposed a novel cloud server-centric authentication scheme for medical surveillance systems, in which the cloud server acts as a relay in the authentication procedure between the users and wearable sensor nodes. Most recently, a cloud-centric three-factor AKA protocol was proposed in [23], which unifies three biometric encryption methods.

In a multidomain scenario, smart devices located in one security domain want to access wearable devices in another domain. In this direction, a multigateway authentication scheme is proposed for a wireless sensor network in [24]. However, the scheme is prone to lost smart card attack since it does not involve public key cryptographic primitives.

1.2. Our Contributions. For the security and privacy of personal private data in multidomain WHMSs [25], we

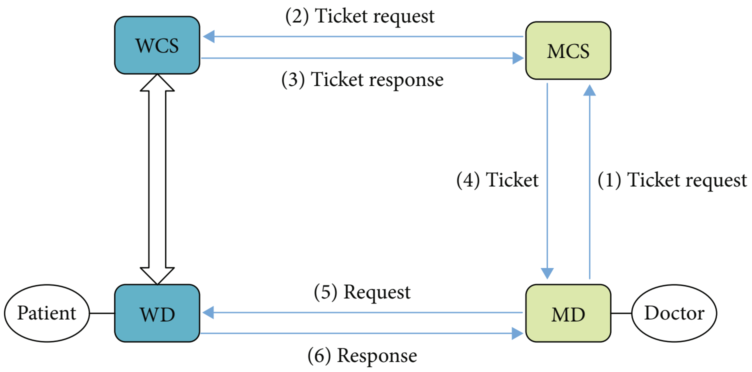


FIGURE 2: The authentication model for multidomain WHMSs.

design a crossdomain multifactor authentication protocol. Our contributions are summarized as follows.

Firstly, we propose a ticket-based authentication model for multidomain WHMSs. Specifically, a MD of a doctor and a WD are registered with MCS and WCS, respectively. The two CSs have established a trust relationship. The MD can request a ticket from MCS and remotely access the WD.

Secondly, we propose a crossdomain three-factor authentication scheme based on the above model. Only a doctor who can present all three factors can request a legal ticket which can be used to access the wearable devices. Moreover, both Elliptical Curve Cryptography (ECC) and fuzzy verifier [26] are introduced to avoid lost smart card attacks, and the Elliptic Curve Diffie-Hellman (ECDH) is employed to fulfill the strong confidentiality of the protocol.

Finally, we present the security and performance analysis of the proposed scheme. The provable security analysis under the random oracle model is given. By comparing its security and efficiency with the related schemes, the security and practicability of the scheme are demonstrated.

1.3. Organization of This Paper. The paper is organized as follows. In Section 2, we propose a crossdomain three-factor AKA scheme for WHMSs. The provable security analysis and informal security analysis are presented in Sections 3 and 4, respectively. Section 5 provides security analysis and efficiency comparison. The conclusion is given in Section 6.

2. The Proposed Protocol

In this paper, we are committed to a crossdomain scenario. Specifically, security domain D1 contains several WDs of a patient and the cloud server WCS, and security domain D2 contains the MD of a doctor and the cloud server MCS. The MD used by the doctor needs to access the WD that collects the patient's physiological data in the case of remote diagnosis [27].

We provide an authentication model for multidomain WHMSs (see Figure 2), which achieves mutual authentication and key agreement between WD and MD from two different domains [28]. The details are as follows. First, the MD sends an access request to the MCS to which it belongs. The MCS sends a ticket request to the WCS, and then, WCS responds to the MCS with the ticket, which contains the secret information associated with the WD. After obtaining the ticket forwarded to the MD through the MCS, the MD can use it to initiate an access request to the WD, and WD will send a response message after the authentication from WD. Finally, the WD and the MD achieves mutual authentication and also negotiates the session key for the future communication.

We present a crossdomain three-factor authentication protocol which includes 8 stages, i.e., (1) initialization phase, (2) wearable device registration phase, (3) mobile device registration phase, (4) login phase, (5) authentication phase, (6) session key agreement phase, (7) password and biometric update phase, and (8) dynamic smart device addition phase. The symbols and their descriptions in the scheme are shown in Table 1.

2.1. Initialization Phase. At this stage, MCS_m and WCS_k pre-share the key $K_{\mathrm{CS}_{m,k}}$. Each $\{\mathrm{MCS}_m \mathrm{WCS}_k\}$ pair has a shared key and can be identified based on each other's identity. A finite cyclic group G generated by a point P of a large prime n on the elliptic curve is selected by MCS_m. It selects s as a private key, calculates the public key $S = sP$, and publishes it. WCS_k stores its $\mathrm{ID}_{\mathrm{WCS}_k}$ and the private key K_{WCS_k} in the database.

2.2. Wearable Device Registration Phase. The holder of WD_j performs the following steps (see Figure 3):

(a) WD_j issues the registration request to WCS_k through the secure channel

(b) When receiving the registration request, WCS_k selects an identity $\mathrm{ID}_{\mathrm{WD}_j}$ for WD_j and calculates the shared key $K_{\mathrm{WCS}_k-\mathrm{WD}_j} = h(K_{\mathrm{WCS}_k} \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{RT}_{\mathrm{WD}_j})$. Then, WCS_k stores $\{\mathrm{ID}_{\mathrm{WD}_j}, K_{\mathrm{WCS}_k-\mathrm{WD}_j}\}$ in its database. Finally, the message $<\mathrm{ID}_{\mathrm{WD}_j}, K_{\mathrm{WCS}_k-\mathrm{WD}_j}>$ is sent by WCS_k to WD_j via the secure channel

(c) WD_j stores the parameters $\{\mathrm{ID}_{\mathrm{WD}_j}, K_{\mathrm{WCS}_k-\mathrm{WD}_j}\}$ in its memory

2.3. Mobile Device Registration Phase. The holder of MD_i (i.e., U_i) performs the following steps (see Figure 4):

Table 1: Symbols.

Symbol	Description
U_i	The doctor
WD_j	The wearable device of patients
MD_i	The mobile device of U_i
ID_i, $\mathrm{ID}_{\mathrm{WD}_j}$	The identifier of U_i and WD_j
PW_i, BIO_i	The password and biometric template of U_i
$\mathrm{Gen}(\cdot)$, $\mathrm{Rep}(\cdot)$	The generation and reproduction algorithm in a fuzzy extractor
t	The fault tolerance threshold used by $\mathrm{Rep}(\cdot)$
RT	The registration timestamp
T	The timestamp
ΔT	The time threshold
$h(\cdot)$	The hash function
$\oplus$	The exclusive or
$\|$	The concatenation
A	The adversary

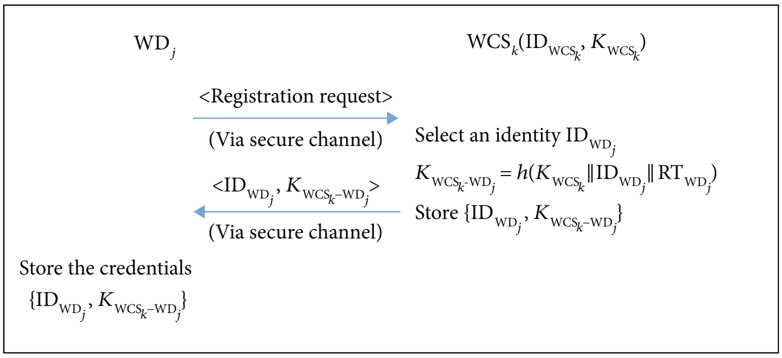


Figure 3: Wearable device registration phase.

(a) U_i selects ID_i and PW_i and enters BIO_i (e.g., fingerprint) on the mobile device MD_i. Then, U_i sends them to MCS_m with the identity ID_i through a secure channel

(b) Once the identity ID_i of U_i is received, MCS_m generates a key K_{MD_i} for this MD_i and calculates temporal certificate $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$. MCS_m stores $\{\mathrm{ID}_i, K_{\mathrm{MD}_i}\}$ in its database. Then, TC_i is sent to MD_i

(c) MD_i continues the calculation of $\mathrm{Gen}(\mathrm{BIO}_i) = (\sigma_i, \tau_i)$, where σ_i is the biometric key and τ_i is the reproduction parameter. Then, MD_i calculates the fuzzy verifiers $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i) \bmod l)$ and $f_i = \mathrm{TC}_i \oplus h(\mathrm{ID}_i \mid \| \sigma_i \| \mathrm{PW}_i)$ and stores the parameters $\{\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), \tau_i, h(\cdot), e_i, f_i, l\}$ in its memory

2.4. Login Phase. As shown in Figure 5, U_i enters ID_i, PW_i, and BIO_i' (e.g., fingerprint). Then, MD_i calculates $\sigma_i' = \mathrm{Rep}(\mathrm{BIO}_i', \tau_i)$ and $e_i' = h(h(\mathrm{ID}_i \| PW_i \| \sigma_i) \bmod l)$ and checks if $e_i' = e_i$ holds. If not, MD_i interrupts the request. Otherwise, it selects the current timestamp T_1 and calculates T $C_i' = f_i \oplus h(\mathrm{ID}_i \| \sigma_i' \| \mathrm{PW}_i)$. It continues to generate a random number $b \in Z_n^*$ and then computes $B = bP$, $C = bS = (C_x, C_y)$, $\mathrm{PID}_{\mathrm{WD}_j} = C_y \oplus \mathrm{ID}_{\mathrm{WD}_j}$, $\mathrm{PID}_i = \mathrm{ID}_i \oplus C_x$, and $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$. MD_i transmits a message $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$ to MCS_m.

2.5. Authentication Phase. At this stage, the mutual authentication between the participants is realized, as shown in Figure 5.

(a) After receiving the message $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$ of MD_i, MCS_m verifies T_1 according to the equation $|T_1' - T_1| \le \Delta T$. If the timestamp is valid, it continues to calculate $C' = sB = (C_x', C_y')$, $\mathrm{ID}_i' = \mathrm{PID}_i \oplus C_x'$, and $\mathrm{ID}_{\mathrm{WD}_j}' = \mathrm{PID}_{\mathrm{WD}_j} \oplus C_y'$. MCS_m obtains the corresponding K_{MD_i} according to ID_i' and the table

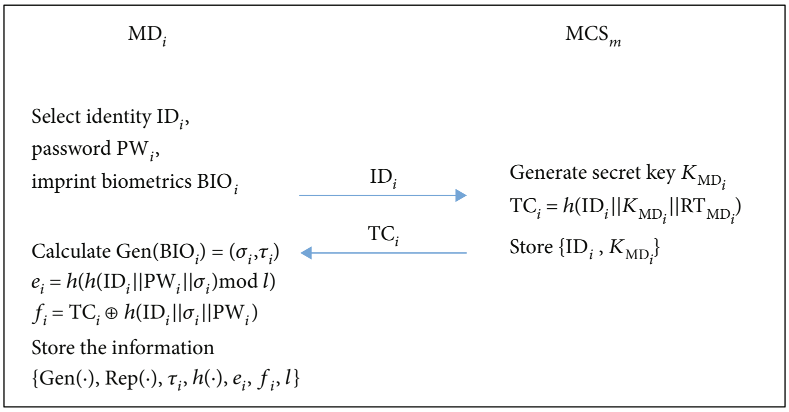
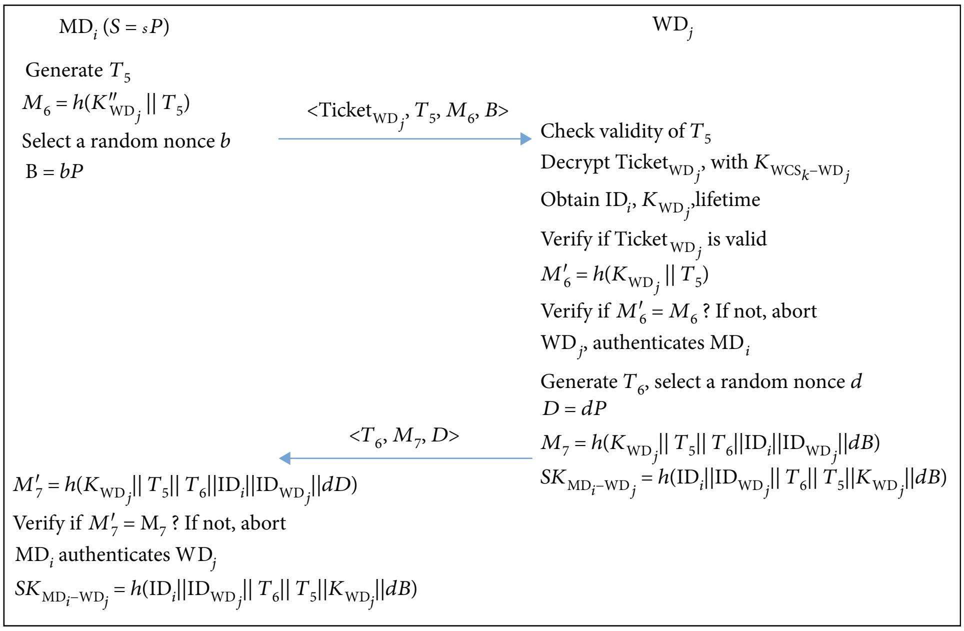


FIGURE 4: Mobile device registration phase.

FIGURE 5: Login and authentication phase.



MD_i $(S = sP)$

WD_j

Generate T_5

$M_6 = h(K''_{\mathrm{WD}_j} \| T_5)$

Select a random nonce b

$B = bP$

$<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$

Check validity of T_5

Decrypt $\mathrm{Ticket}_{\mathrm{WD}_j}$, with $K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}$

Obtain ID_i, K_{WD_j},lifetime

Verify if $\mathrm{Ticket}_{\mathrm{WD}_j}$ is valid

$M_6' = h(K_{\mathrm{WD}_j} \| T_5)$

Verify if $M_6' = M_6$? If not, abort

WD_j, authenticates MD_i

Generate T_6, select a random nonce d

$D = dP$

$<T_6, M_7, D>$

$M_7 = h(K_{\mathrm{WD}_j} \| T_5 \| T_6 \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| dB)$

$SK_{\mathrm{MD}_i\text{-}\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| dB)$

$M_7' = h(K_{\mathrm{WD}_j} \| T_5 \| T_6 \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| dD)$

Verify if $M_7' = M_7$? If not, abort

MD_i authenticates WD_j

$SK_{\mathrm{MD}_i\text{-}\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| dB)$

FIGURE 6: Session key agreement phase.

stored in its database and calculates $\mathrm{TC}_i'' = h(\mathrm{ID}_i' \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$ and $M_1' = h(\mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| \mathrm{TC}_i'' \| T_1 \| C_x')$ and checks if the equation $M_1 = M_1'$ holds. If so, MD_i is considered legal by MCS_m. It continues to generate the current timestamp T_2 and determines which WCS_k to be requested as well as the corresponding share key $K_{\mathrm{CS}_{m,k}}$ according to $\mathrm{ID}_{\mathrm{WD}_j}'$. Then, MCS_m calculates $M_2 = h(K_{\mathrm{CS}_{m,k}} \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_2)$ and $M_3 = \{\mathrm{ID}_i', \mathrm{ID}_{\mathrm{WD}_j}', \mathrm{ID}_{\mathrm{MCS}_m}, T_2\}_{K_{\mathrm{CS}_{m,k}}}$ and sends $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$ to WCS_k

(b) WCS_k receives the message $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$ sent by MCS_m, and WCS_k checks the validity of the timestamp T_2. If it is valid, WCS_k gets the corresponding $K_{\mathrm{CS}_{m,k}}$ according to $\mathrm{ID}_{\mathrm{MCS}_m}$, decrypts M_3 to obtain ID_i', $\mathrm{ID}_{\mathrm{WD}_j}'$, $\mathrm{ID}_{\mathrm{MCS}_m}'$ with $K_{\mathrm{CS}_{m,k}}$, and then checks the equation $\mathrm{ID}_{\mathrm{MCS}_m}' = \mathrm{ID}_{\mathrm{MCS}_m}$. If it fails, the session is interrupted. Otherwise, it continues to calculate $M_2' = h(K_{\mathrm{CS}_{m,k}} \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_2)$ and verifies if $M_2' = M_2$ is true. If true, MCS_m is considered legal by WCS_k. WCS_k obtains the responding key $K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}$ according to $\mathrm{ID}_{\mathrm{WD}_j}'$, generates the current timestamp T_3 and a temporary key K_{WD_j}, and calculates $\mathrm{Ticket}_{\mathrm{WD}_j} = \{\mathrm{ID}_i', K_{\mathrm{WD}_j}, \mathrm{lifetime}\}_{K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}}$, $\mathrm{SK}_{\mathrm{CS}_{m,k}} = h(K_{\mathrm{CS}_{m,k}} \| T_2 \| T_3)$, $\mathrm{TK}_{\mathrm{WD}_j} = K_{\mathrm{WD}_j} \oplus \mathrm{SK}_{\mathrm{CS}_{m,k}}$, and $M_4 = h(K_{\mathrm{CS}_{m,k}} \| K_{\mathrm{WD}_j} \| T_3 \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| \mathrm{ID}_{\mathrm{MCS}_m} \| \mathrm{ID}_{\mathrm{WCS}_k} \| \mathrm{Ticket}_{\mathrm{WD}_j})$. It sends the message $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$ to MCS_m

(c) After receiving $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, MCS_m checks the freshness of T_3. If the timestamp is valid, it continues to compute $\mathrm{SK}_{\mathrm{CS}_{m,k}} = h(K_{\mathrm{CS}_{m,k}} \| T_2 \| T_3)$, $K_{\mathrm{WD}_j}' = \mathrm{TK}_{\mathrm{WD}_j} \oplus \mathrm{SK}_{\mathrm{CS}_{m,k}}$, and $M_4' = h(K_{\mathrm{CS}_{m,k}} \| K_{\mathrm{WD}_j}' \| T_3 \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| \mathrm{ID}_{\mathrm{MCS}_m} \| \mathrm{ID}_{\mathrm{WCS}_k} \| \mathrm{Ticket}_{\mathrm{WD}_j})$ and verifies if $M_4' = M_4$ holds. If true, WCS_k is considered legal by MCS_m. MCS_m generates the current timestamp T_4 and calculates $M_5 = h(\mathrm{TC}_i'' \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_4 \| C \| \mathrm{Ticket}_{\mathrm{WD}_j})$ and $\mathrm{TTK}_{\mathrm{WD}_j} = K_{\mathrm{WD}_j}' \oplus C$. It sends a message $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TTK}_{\mathrm{WD}_j}, T_4, M_5>$ to MD_i

(d) After $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TTK}_{\mathrm{WD}_j}, T_4, M_5>$ is received, MD_i checks the freshness of T_4 and calculates $K_{\mathrm{WD}_j}'' = \mathrm{TTK}_{\mathrm{WD}_j} \oplus C$ and $M_5' = h(\mathrm{TC}_i' \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_4 \| C \| \mathrm{Ticket}_{\mathrm{WD}_j})$. It checks if $M_5 = M_5'$ is true. If established, MCS_m is considered legal by MD_i

2.6. Session Key Agreement Phase.

2.6. Session Key Agreement Phase. At this stage, a session key is established between MD_i and WD_j, as shown in Figure 6.

(a) MD_i generates a timestamp T_5, selects a random number b and computes $B = bP$ and $M_6 = h(K_{\mathrm{WD}_j}'' \| T_5)$, and transmits a message $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$ to WD_j

(b) After accepting $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$, WD_j checks the freshness of the timestamp T_5. So it obtains ID_i, K_{WD_j}, lifetime by decrypting $\mathrm{Ticket}_{\mathrm{WD}_j}$ with key $K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}$ and verifies the validity of

$\mathrm{Ticket}_{\mathrm{WD}_j}$. It continues to calculate $M_6' = h(K_{\mathrm{WD}_j} \| T_5)$ and verifies if the equation $M_6' = M_6$ is true. If it fails, the session is interrupted. Otherwise, WD_j treats MD_i as legitimate. WD_j generates the current T_6, selects a random number d, and computes $D = dP$, $M_7 = h(K_{\mathrm{WD}_j} \| T_5 \| T_6 \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| dB)$, and $\mathrm{SK}_{\mathrm{MD}_i\text{-}\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| dB)$. Eventually, it sends $<T_6, M_7, D>$ to MD_i

(c) After receiving $<T_6, M_7, D>$, MD_i calculates $M_7' = h(K_{\mathrm{WD}_j}'' \| T_5 \| T_6 \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| bD)$ and then verifies if $M_7' = M_7$ holds. If not, the session is interrupted. Conversely, WD_j is considered legal by MD_i. Finally, it calculates the session key $\mathrm{SK}_{\mathrm{MD}_i\text{-}\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| bD)$

2.7. Password and Biometric Update Phase. At this stage, the old password and biometric are updated with new ones. The details are as follows.

(a) Firstly, U_i inputs identity ID_i, password $\mathrm{PW}_i^{\,0}$, and biometric BIO_i^0 on MD_i. Then, MD_i calculates $\sigma_i^0 = \mathrm{Rep}\,(\mathrm{BIO}_i^0, \tau_i)$ and $e_i^{\,0} = h(h(\mathrm{ID}_i \| \mathrm{PW}_i^{\,0} \| \sigma_i^{\,0}) \bmod l)$ and checks if $e_i^0 = e_i$ is true. If so, the previously entered information is considered valid and continues to enter the new password and biometrics that the doctor wants to update in the next step; otherwise, the session is terminated

(b) U_i enters a new password PW_i^n and/or BIO_i^n. Then, MD_i calculates the relevant parameters $\mathrm{Gen}(\mathrm{BIO}_i^{\,n}) = (\sigma_i^{\,n}, \tau_i^{\,n})$, $e_i^{\,n} = h(h(\mathrm{ID}_i \| \mathrm{PW}_i^{\,n} \| \sigma_i^{\,n}) \bmod l)$, and $f_i^{\,n} = \mathrm{TC}_i \oplus h(\mathrm{ID}_i \| \| \sigma_i^{\,n} \| \mathrm{PW}_i^{\,n})$. Finally, U_i updates the original e_i, f_j, τ_i to e_i^n, f_j^n, τ_i^n

2.8. Dynamic Smart Device Addition Phase. New WD_j and new MD_i can be dynamically added at this phase.

(1) First, add a new wearable device named $\mathrm{WD}_j^{\mathrm{new}}$. In essence, this process looks like the WD_j initialization phase, so it just needs to register at WCS_k:

(a) $\mathrm{WD}_j^{\mathrm{new}}$ issues a registration request to WCS_k through a secure channel

(b) After the registration request is received, WCS_k selects an identity $\mathrm{ID}_{\mathrm{WD}_j}^{\mathrm{new}}$ for $\mathrm{WD}_j^{\mathrm{new}}$ and calculates the share key $K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}^{\mathrm{new}} = h(K_{\mathrm{WCS}_k} \| \mathrm{ID}_{\mathrm{WD}_j}^{\mathrm{new}} \| \mathrm{RT}_{\mathrm{WD}_j}^{\mathrm{new}})$, in which $\mathrm{RT}_{\mathrm{WD}_j}^{\mathrm{new}}$ represents the timestamp when registering $\mathrm{WD}_j^{\mathrm{new}}$. Then, WCS_k stores $\{\mathrm{ID}_{\mathrm{WD}_j}^{\mathrm{new}}, K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}^{\mathrm{new}}, \mathrm{RT}_{\mathrm{WD}_j}^{\mathrm{new}}\}$ in its database. Finally, the message $<\mathrm{ID}_{\mathrm{WD}_j}^{\mathrm{new}}, K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}^{\mathrm{new}}>$ is given to $\mathrm{WD}_j^{\mathrm{new}}$ by WCS_k over the secure channel

(c) $\mathrm{WD}_j^{\mathrm{new}}$ stores the parameters $\{\mathrm{ID}_{\mathrm{WD}_j}^{\mathrm{new}}, K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}^{\mathrm{new}}\}$ into their memory

(2) Secondly, add a new mobile device called $\mathrm{MD}_i^{\mathrm{new}}$:

(a) U_i selects $\mathrm{ID}_i^{\mathrm{new}}$ and $\mathrm{PW}_i^{\mathrm{new}}$ and enters $\mathrm{BIO}_i^{\mathrm{new}}$ on the mobile device $\mathrm{MD}_i^{\mathrm{new}}$. Then, $\mathrm{ID}_i^{\mathrm{new}}$ is sent to MCS_m by U_i via a secure channel

(b) After receiving the identity $\mathrm{ID}_i^{\mathrm{new}}$ of U_i, MCS_m generates a key $K_{\mathrm{MD}_i}^{\mathrm{new}}$ for this $\mathrm{MD}_i^{\mathrm{new}}$ and calculates $\mathrm{TC}_i^{\mathrm{new}} = h(\mathrm{ID}_i^{\mathrm{new}} \| K_{\mathrm{MD}_i}^{\mathrm{new}} \| \mathrm{RT}_{\mathrm{MD}_i}^{\mathrm{new}})$, in which $\mathrm{RT}_{\mathrm{MD}_i}^{\mathrm{new}}$ represents the registration timestamp of $\mathrm{MD}_i^{\mathrm{new}}$. MCS_m stores $\{\mathrm{ID}_i^{\mathrm{new}}, K_{\mathrm{MD}_i}^{\mathrm{new}}\}$ in its database. Then, $\mathrm{TC}_i^{\mathrm{new}}$ is sent to $\mathrm{MD}_i^{\mathrm{new}}$

(c) After receiving the message, $\mathrm{MD}_i^{\mathrm{new}}$ calculates $\mathrm{Gen}(\mathrm{BIO}_i^{\mathrm{new}}) = (\sigma_i^{\mathrm{new}}, \tau_i^{\mathrm{new}})$, where σ_i^{new} is the biometric key and τ_i^{new} is the common recovery parameter

(d) After the above process is completed, $\mathrm{MD}_i^{\mathrm{new}}$ continues to calculate $e_i^{\mathrm{new}} = h(h(\mathrm{ID}_i^{\mathrm{new}} \| \mathrm{PW}_i^{\mathrm{new}} \| \sigma_i^{\mathrm{new}}) \bmod l)$ and $f_i^{\mathrm{new}} = \mathrm{TC}_i^{\mathrm{new}} \oplus h(\mathrm{ID}_i^{\mathrm{new}} \| \sigma_i^{\mathrm{new}} \| \mathrm{PW}_i^{\mathrm{new}})$ and stores the parameters $\{\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), \tau_i^{\mathrm{new}}, h(\cdot), e_i^{\mathrm{new}}, f_i^{\mathrm{new}}, l\}$ in its memory

3. Provable Security Analysis

3.1. Adversary Model. We give the security model in this paper. It is assumed that the cryptographic primitives used are secure. That is, A is not capable of guessing the result of the hash functions, the random numbers, and the preshared keys of both parties used in the protocol.

Hypothesis 1. Communication channels are mainly divided into a private channel (i.e., a secure channel) and a public channel (i.e., an unsecure channel). For the public channel, we use the classic Dolev-Yao model [29], where an adversary can eavesdrop, intercept, delete, or modify any messages sent through the open channel. However, for a secure channel generally used in the registration phase, the adversary cannot obtain any information through this channel.

Hypothesis 2. According to [26], with the improvement of the attacker's ability, the privacy information in a smart card can be obtained by power analysis attacks or by exploiting software vulnerabilities. Therefore, we assume in this paper that an adversary can resolve the confidential information after obtaining the smart card.

Hypothesis 3. As the adversary model proposed in [26], the adversary A can offline exhaust all elements of the Cartesian product $D_{\mathrm{id}} \times D_{\mathrm{pw}}$ during the polynomial time, where D_{pw} and D_{id} denotes the password space and the identity space, respectively.

Hypothesis 4. As the security model of the three-factor AKA protocol proposed in [30], any two of three authentication factors can be obtained by A. However, it does not have the ability to obtain all authentication factors at the same time. The three cases are as follows:

(a) A can get the doctor's passwords and MDs

(b) A can get passwords and biometrics

(c) A can get MDs and biometrics

Hypothesis 5. The adversary A can get a session key established in the previous session.

3.2. Security Model. We explain the security model used by the security proof in this section. There are four main participants in this paper: WD, WCS, MD, and MCS.

Generally, the adversary of the authentication protocol is a probabilistic polynomial time adversary, who can control the transmission channel, passively eavesdropping or actively modifying or delaying messages [31].

Participants. Let Π_U^i represent the ith session instance of the participant U, also known as the oracle.

Status. There are generally three states: accept, reject, and invalid. It is in the "accept" state when the oracle receives the correct message. It is in the "reject" state when the oracle receives the error message; when the output has no answer, we use $\perp$ to indicate the invalid result.

Partnering. Instances of two participants can become partners of each other if and only if (1) both instances are in the "accept" state and have the same session key, (2) both instances share the same identity, (3) the ID of the former is the partner ID of the latter and vice versa, and (4) no other instance accepts the same session ID as both instances.

Freshness. An instance is said to be "fresh" if and only if (1) the attacker did not send a Reveal query to this instance or its partner instance and (2) the attacker did not corrupt the instance before the instance is in the accept state.

Adversary. The ability of the adversary can be simulated by the following queries to oracles:

Execute$(\Pi_{MCS}^m, \Pi_{MD}^i, \Pi_{WCS}^k, \Pi_{WD}^j)$. This query simulates passive eavesdropping attacks of A. For this query, the public-transmitted content of authentication instances executed between all participants will be obtained by A.

Send(Π_U^i, m). This oracle query simulates an active attack, and A sends the modified message to the instance Π_U^i on behalf of another party. After the instance Π_U^i receives the message, A will get a response message generated by the participant Π_U^i. Π_U^i can be a wearable device, a mobile device, and a cloud server in both domains.

Reveal(Π_U^i). When the instance Π_U^i obtains a session key, the attacker has the ability to get the key. When an instance does not have a session key, the attacker gets an invalid flag $\perp$.

Corrupt(Π_U^i). Through this query, A can get secret credentials of corrupted participants, such as passwords, biometrics, and mobile devices. This query can simulate the forward security of the session key.

Test(Π_U^i). It can determine the security of the session key owned by the instance Π_U^i. After the simulator receives this query, it will perform a flip coin operation. When the result is 1, the attacker returns a real session key; when the result is 0, the attacker returns a random key string with the same length as the key. In this case, A must distinguish whether the returned value is a real session key or a random value, and the probability is 1/2.

We define the semantic security of the session key. A can only perform the Test query to fresh instances, and there are no restrictions on other queries. It is necessary for A to judge that the bit used by the simulator is 0 or 1 after the Test query. If it can guess the correct result, the attacker is considered to have succeeded in the semantic security of the protocol P and defined this successful event as Succ. The size of the dictionary space is $|D|$, and the advantage of the attacker to make this attack is defined as $\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A) = 2\Pr[\mathrm{Succ}] - 1$. An authentication protocol is called semantically secure, if and only if for all probability polynomial time attackers, they have the advantage $\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A)$ which is larger than $kq_{\mathrm{send}}/|D|$ that can be ignored, where q_{send} is the number of active attacks by A.

3.3. Security Proof

Theorem 1. *Suppose that P is the proposed authentication protocol, E_p is an elliptic curve group, and A is a PPT adversary. $Adv_{P,D}^{ake}(A)$ is the advantage for A to break the semantic security of the protocol P. A can execute at most q_{send} send queries and q_{exe} queries of different instances in the longest time t, so we have*

$$Adv_{P,D}^{ake}(A) \leq \frac{q_{send}}{|D|}. \tag{1}$$

Proof. We use a series of mixed experiments $\mathrm{Ex}_0, \mathrm{Ex}_1, \mathrm{Ex}_2, \cdots, \mathrm{Ex}_7$ to prove that the protocol is AKA secure. These experimental games start from a real attack scenario. Through continually changing some simulation rules in the experiments, we have the final experiment in which the attacker has little advantage in distinguishing between a session key and a random key of the same length. Let $\mathrm{Adv}_i(A)$ be the advantage of the attacker in Ex_i and Δ_i denote the degree of distinction between Ex_i and Ex_{i+1}.

Ex_0. This is a scheme under the random oracle model. According to the definition of the advantage of the previous attacker, we have

$$\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A) = \mathrm{Adv}_0(A). \tag{2}$$

Ex_1. In the hybrid experiment, we maintain a hash table H list to simulate all random oracles. When s is a string and wants to query $H(s)$, the oracle first searches the H list for the corresponding record $\{s, \mathrm{value}\}$. If found, the value corresponding to the record is returned. Conversely, the oracle produces a random bit string $b \in \{0,1\}^l$, returns the value to the interrogator, and stores the record $\{s, b\}$ in the hash table. Since the random oracle is perfectly simulated in polynomial time, the attacker cannot distinguish Ex_0 from Ex_1.

$$\varDelta_0 = |\mathrm{Adv}_1(A) - \mathrm{Adv}_0(A)| \le \mathrm{neg}l(\kappa). \tag{3}$$

Ex$_2$. In the previous experiment, we have known that the oracle is perfectly simulated in polynomial time, so we exclude relatively unlikely hash collisions. When a collision occurs in the passive session or oracle simulation, then we will end the simulation of the entire game and believe that the attacker has won the game. Based on a birthday paradox, we have

$$\varDelta_1 = |\mathrm{Adv}_2(A) - \mathrm{Adv}_1(A)| \le \mathrm{neg}l(\kappa). \tag{4}$$

Ex$_3$. Simulation of the passive session has been changed in the experiment, considering the probability that the attacker would not make any random oracle query but can forge the authentication information $<M_1, M_2, M_4, M_5, M_6, M_7>$. Ex_2 and Ex_3 are indistinguishable from A unless they provide valid information to end the game. Specifically, for the authentication message $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$, where $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$ or $\mathrm{TC}_i' = f_i \oplus h(\mathrm{ID}_i \| \sigma_i' \| \mathrm{PW}_i)$ in the case that no corruption request is made, σ_i', PW_i cannot be obtained or the key K_{MD_i} is unknown to the attacker, and the valid information M_1 cannot be calculated, so the attacker has a negligible probability of success. So

$$\varDelta_2 = |\mathrm{Adv}_3(A) - \mathrm{Adv}_2(A)| \le \mathrm{neg}l(\kappa). \tag{5}$$

Ex$_4$. Simulation of the active session has been changed in the experiment. For a $\mathrm{Send}(\mathrm{MCS}^m, (B, M_1))$ query, if A does not corrupt the MD, while M_1 is the valid verification message generated by A, then we only need to let A achieve the final victory of the game and stop the simulation game. If such events occur, the attacker can get the random number b when knowing B, P and generate the random number C, in which $B = bP$, $b \in Z_n^*$, and $C = bS = (C_x, C_y)$ and the message M_1 contains C_x. The probability of successful construction of the message M_1 described above is equal to the probability of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) in ECC. The ECDLP is a difficult problem in cryptography, so the probability of an attacker's success is negligible. In short, we have

$$\varDelta_3 = |\mathrm{Adv}_4(A) - \mathrm{Adv}_3(A)| \le \mathrm{neg}l(\kappa). \tag{6}$$

Ex$_5$. We continue to change the simulation of the active sessions during the experiment. If the attacker sends a Reveal (WCS^k) query to the WCS, it will get the session key $\mathrm{SK}_{\mathrm{CS}_{m,k}} = h(K_{\mathrm{CS}_{m,k}} \| T_2 \| T_3)$ between the WCS and the MCS and can also calculate the temporary key K_{WD_j}. However, in order to generate valid verification information M_4, A needs to generate a valid $\mathrm{Ticket}_{\mathrm{WD}_j}$. It is able for A to know the identity of $\mathrm{Ticket}_{\mathrm{WD}_j}$ and specify the lifetime according to the general rules, but A cannot get the key shared by WCS and WD in advance. If A can guess and get a valid $\mathrm{Ticket}_{\mathrm{WD}_j}$, we terminate the simulation of the game and declare that the attacker has already won the game. The probability of such an event is negligible, so there will be

$$\varDelta_4 = |\mathrm{Adv}_5(A) - \mathrm{Adv}_4(A)| \le \mathrm{neg}l(\kappa). \tag{7}$$

Ex$_6$. We change the simulation rules of the activity sessions again in the experiment. Specifically, for message M_5, assume that A previously obtained the value of S and B by eavesdropping, where $B = bP$, the random number $b \in Z_n^*$, but the probability of successfully forging bsP of the message M_5 is actually equivalent to the probability of solving the Elliptical Curve Computational Diffie-Hellman Problem (ECCDHP). It is well known that ECCDHP is a difficult problem in cryptography, so the success probability of an attacker is negligible, so there are

$$\varDelta_5 = |\mathrm{Adv}_6(A) - \mathrm{Adv}_5(A)| \le \mathrm{neg}l(\kappa). \tag{8}$$

Ex$_7$. Finally, we change the simulation of the activity sessions in the experiment. During the session key agreement phase, an attacker may have previously obtained $\mathrm{Ticket}_{\mathrm{WD}_j}$ by eavesdropping. If A fakes the message $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$ and sends it to WD_j, then we just need to let A win and terminate the simulation. However, it should be noted that K_{WD_j} is an unknown security parameter, so the probability that A can effectively generate this information is negligible. Based on the above, we have

$$\varDelta_6 = |\mathrm{Adv}_7(A) - \mathrm{Adv}_6(A)| \le \mathrm{neg}l(\kappa). \tag{9}$$

In the final experiment, there is no real password-related information in the session using the Execute query from A, so there is no advantage, and the active attack through the Send query is only

$$\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A) \le \frac{q_{\mathrm{send}}}{|D|}. \tag{10}$$

4. Informal Security Analysis

This section shows that our scheme can achieve many security attributes.

4.1. Preventing Stolen Mobile Device Attack. If A has got a stolen or lost MD_i, it can get the information $\{\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), \tau_i, h(\cdot), e_i, f_i, l\}$ stored in MD_i. First, the adversary A wants to correctly guess the doctor's password PW_i and needs to guess the password and verify the security parameters $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i') \bmod l)$. According to the assumptions about the ability of the adversary given in this paper, A can get both identity ID_i and biometric BIO_i, but e_i is a fuzzy verifier $(2^4 < l < 2^8)$, so there are $|D_{\mathrm{id}}|/l$ possible password alternatives. To get the only correct password, A has to identify it online, and this can be prevented by implementing a number-limiting strategy. On the other hand, A may also try to get a unique correct password by $f_i = \mathrm{TC}_i \oplus h(\mathrm{ID}_i \| \sigma_i \| \mathrm{PW}_i)$. However, $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$, and it is protected by the key K_{MD_i}, which is generated by MCS_m for MD_i. So, this method cannot be implemented. Therefore, it is found that the above two possible attack methods are not feasible; that is, our protocol can prevent such attack.

4.2. Preventing Replay Attack. Suppose that A has eavesdropped all the information $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$, $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TTK}_{\mathrm{WD}_j}, T_4, M_5, C>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$, and $<T_6, M_7, D>$ in the login phase, the authentication phase, and the session key negotiation phase. Then, A replays them on the public channel, but it is intuitive to see that all of the messages we transmit contain the timestamp, which is the time when the message is sent. We use timestamps and random nonce in the protocol to guarantee the freshness of the transmitted information. If there is an adversary attempting to repeatedly send these messages, the existence of this situation will be found by verifying the validity of the timestamp. In addition, it is not feasible for an adversary to bypass the message recipient's verification of the timestamp because all messages contain a key-protected hash value. Therefore, our protocol can prevent replay attacks.

4.3. Preventing Man-in-the-Middle Attack. It is assumed that A is able to intercept the sent messages in the login phase, authentication phase, and key agreement phase and replace those messages with its own messages to perform the attack as a middleman.

Specifically, if A wants to modify the message $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$ and the key to the parameter M_1, B is to generate a random number $b \in Z_n^*$, A can randomly select $b \in Z_n^*$ and calculate $B = bP$, $C = bS = (C_x, C_y)$, $\mathrm{PID}_i = \mathrm{ID}_i \oplus C_x$, $\mathrm{PID}_{\mathrm{WD}_j} = \mathrm{ID}_{\mathrm{WD}_j} \oplus C_y$, and $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$. The message receiver will confirm whether the party is a legitimate one by verifying $M_1 = M_1'$. Both of the messages $\mathrm{TC}_i = f_i \oplus h(\mathrm{ID}_i \| \sigma_i \| \mathrm{PW}_i)$ and $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$ of M_1 are protected by a password or a key K_{MD_i}, so A cannot calculate TC_i. It can be seen that A cannot replace the real message M_1 with his fake message and gain the trust of the receiver as an intermediary. For the message $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$ sent from MCS_m to WCS_k, A intercepts the message as an intermediary and replaces it with its own messages. It wants to pass the verification of WCS_k and then needs to send the correct $<M_2, M_3>$. To calculate $M_2 = h(K_{\mathrm{CS}_{m,k}} \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_2)$ and $M_3 = \{\mathrm{ID}_i', \mathrm{ID}_{\mathrm{WD}_j}', \mathrm{ID}_{\mathrm{MCS}_m}, T_2\}_{K_{\mathrm{CS}_{m,k}}}$, it needs the shared key $K_{\mathrm{CS}_{m,k}}$ between WCS_k and MCS_m, but it cannot get the key. Therefore, it cannot generate the message $<M_2, M_3>$. Similarly, it does not correctly calculate $\mathrm{Ticket}_{\mathrm{WD}_j}$, $\mathrm{TK}_{\mathrm{WD}_j}$, and M_4 in the next message $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, because they are both protected by the keys $K_{\mathrm{WCS}_k\text{-}\mathrm{WD}_j}$ and $K_{\mathrm{CS}_{m,k}}$. In the same way, A cannot generate other valid messages. Although the message is modified and sent to the intended recipient, it cannot be verified by the recipient. In short, our protocol can achieve mutual authentication among all participants. Therefore, the protocol can defend against man-in-the-middle attacks.

4.4. Efficient Unauthorized Login Detection. During protocol execution, unauthorized access should be detected in the login phase, and the session is terminated when the request is rejected. This not only saves unnecessary communication costs and calculation costs but also enables update operations such as password update. In the actual scenario, if the doctor enters an incorrect password, a detection mechanism in our protocol can verify the validity of the information provided by the doctor and provide timely feedback. The protocol is specifically implemented in this way, and we use a fuzzy extractor to verify the validity of the doctor's biometrics. In the login phase of the protocol, U_i enters ID_i, PW_i, and $\mathrm{BIO}_i{}'$ on MD_i. Then, MD_i will calculate $\sigma_i' = \mathrm{Rep}(\mathrm{BIO}_i', \tau_i)$ and $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i') \bmod l)$. MD_i verifies if $e_i' = e_i$ holds. If not, the login request is rejected.

Therefore, our protocol can detect unauthorized login by user doctor's error input or intentional attack by the attacker during the login phase.

4.5. Anonymity and Untraceability. We assume that A intercepts all information $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$, $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TTK}_{\mathrm{WD}_j}, T_4, M_5, C>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$, and $<T_6, M_7, D>$ transmitted on the public channel during the login phase, the authentication phase, and the session key negotiation phase.

It can be seen from all messages that they contain timestamps or nonces and are protected by their own keys or shared keys, thus ensuring confidentiality. Only when A knows these secret parameters can A obtain the identity information related to U_i, MD_i, and WD_j. Therefore, our protocol achieves anonymity [32, 33]. On the other hand, we can also find that these messages are dynamic. The pseudoidentity PID_i of users is different in each session, and $b \in Z_n^*$ is randomly selected. Therefore, the message fields in each session are different, and the adversary cannot obtain useful information through different sessions, so untraceability is realized.

4.6. Mutual Authentication. In our protocol, only the legal patient processing the correct password and biometrics and the corresponding wearable device can compute $\mathrm{TC}_i' = f_i \oplus h(\mathrm{ID}_i \| \sigma_i' \| \mathrm{PW}_i)$ and $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$. So MD_i can pass the authentication of MCS_m successfully via checking the correctness of M_1. Similarly, an adversary cannot calculate correct $M_5' = h(\mathrm{TC}_i' \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_4 \| C \| \mathrm{Ticket}_{\mathrm{WD}_j})$ without knowing TC''. Since only MCS_m knows the secret key s, it can compute the valid TC''. Thus, MD_i

Table 2: Comparison of security attributes.

Schemes	The scheme in [34]	The scheme in [35]	Our scheme
Preventing stolen mobile device attack	☒	☒	✓
Preventing replay attack	✓	✓	✓
Preventing man-in-the-middle attack	✓	✓	✓
Efficient unauthorized login detection	✓	✓	✓
Anonymity and untraceability	✓	✓	✓
Mutual authentication	☒	✓	✓
Known key security	✓	✓	✓
Perfect forward secrecy	✓	✓	✓
Extensibility	✓	✓	✓
Efficient password and biometric update	✓	✓	✓

can authenticate MCS_m by verifying the correctness of M_5. Thus, our protocol achieves mutual authentication between MD_i and MCS_m.

In the communication between MCS_m and WCS_k, WCS_k authenticates MCS_m via checking the correctness of $M_2' = h(K_{\mathrm{CS}_{m,k}} \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_2)$, since only the legal MCS_m stores the valid share key $K_{\mathrm{CS}_{m,k}}$. Similarly, MCS_m authenticates WCS_k via checking the correctness of $M_4' = h(K_{\mathrm{CS}_{m,k}} \| K_{\mathrm{WD}_j}' \| T_3 \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| \mathrm{ID}_{\mathrm{MCS}_m} \| \mathrm{ID}_{\mathrm{WCS}_k} \| \mathrm{Ticket}_{\mathrm{WD}_j})$ because only the valid WCS_k processing the valid share key $K_{\mathrm{CS}_{m,k}}$ can decrypt M_3 to obtain ID_i', $\mathrm{ID}_{\mathrm{WD}_j}'$, and $\mathrm{ID}_{\mathrm{MCS}_m}'$. Thus, MCS_m and WCS_k accomplish mutual authentication.

4.7. Known Key Security. It is assumed that the adversary A has obtained the session key $\mathrm{SK}_{\mathrm{MD}_i-\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| bdP)$ shared by MD_i and WD_j. However, because our protocol uses timestamps and each session includes a randomly chosen temporary key K_{WD_j} to guarantee that the session key of the current session is totally different from the previous session key, our protocol accomplishes known key security.

4.8. Perfect Forward Secrecy. In our scheme, U_i has long-term secrets PW_i, BIO_i, and $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i') \bmod l)$, and when the long-term secrets of U_i are leaked, the previous session key $\mathrm{SK}_{\mathrm{MD}_i-\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| bdP)$ will not be leaked. Because b and d are randomly selected, it is difficult to calculate bdP by bP and dP according to ECCDHP.

4.9. Extensibility. The protocol includes a mobile device or wearable device dynamic addition phase, so it can provide extensibility. Through this phase, we are able to dynamically add mobile devices or wearable devices, which only need to interact with the cloud servers of the security domain to which they belong. The cloud server maintains a table. Therefore, the protocol can provide the security features of extensibility.

4.10. Efficient Password and Biometric Update. Because of the efficient detection mechanism of unauthorized logins, doctors can freely update passwords or biometrics in our protocol, as shown in Section 2.7.

Table 3: Efficiency comparison.

Schemes	Our scheme	The scheme in [34]	The scheme in [35]
$\mathrm{MD}_i(U_i)$	$8T_h + T_p$	$5T_h + 2T_p$	$5T_h + 3T_p$
$\mathrm{MCS}_m(\mathrm{CS})$	$6T_h + 3T_p + T_s$	$2T_h + 3T_p$	$4T_h + T_p$
WD_j	$3T_h + T_s$	$2T_h + 2T_p$	$4T_h + T_p$
WCS_k	$3T_h + 2T_s$	—	—

5. Security and Efficiency Comparison

5.1. Security Comparison. The security comparison of our scheme with [34, 35] is shown in Table 2.

Table 2 shows that the schemes in [34, 35] fail to meet all the security features listed in the table, such as inability to defend against MD stolen attacks. Our scheme can satisfy a number of security features, which has been proven in previous security analysis.

5.2. Efficiency Comparison. For efficiency, we mainly pay attention to the login, authentication, and session key agreement phases. The following symbols are used to define various calculations as well as their specific time consumption.

T_s: the time complexity of symmetric encryption and decryption (0.0214385 ms) [35].

T_p: the time complexity of point multiplication operation of an elliptic curve (0.427576 ms) [35].

T_h: the time complexity of computing hash functions (0.0000328 ms) [35].

The efficiency comparison of our scheme with [34, 35] is shown in Table 3.

Our scheme has two cloud servers, and each domain has one cloud server. Different from our scheme in the number of participants, there is only one cloud server in schemes [34, 35]. Since the cloud server has stronger computing power and more resource [36], we only pay attention to the calculation of time consumption of mobile devices and

Table 4: Time-cost comparison (ms).

Schemes	Our scheme	The scheme in [34]	The scheme in [35]
$\mathrm{MD}_i(U_i)$	0.4278384	0.8553160	1.2828920
$\mathrm{MCS}_m(\mathrm{CS})$	1.3043633	1.2827936	0.4277072
WD_j	0.0215369	0.8552176	0.4277072
WCS_k	0.0429754	—	—

wearable devices. As shown in Table 4, our scheme has obvious performance advantages.

Therefore, our scheme has better performance and meets a variety of common security demands, which is suitable for use in a wearable environment.

6. Conclusion

In practical WHMSs, single-domain authentication schemes can no longer meet the growing number of users and devices and crossdomain authentication schemes are urgently needed. In this paper, we proposed a ticket-based authentication model for multidomain WHMSs. Specifically, a mobile device of one domain can request a ticket from the cloud server of another domain with which wearable devices are registered and remotely access the wearable devices with the ticket. Then, we proposed a crossdomain three-factor authentication scheme based on the above model. Only a doctor who can present all three factors can request a legal ticket which can be used to access the wearable devices. Both the elliptical curve and fuzzy verifier are introduced to avoid lost smart card attack and to strengthen the confidentiality of the protocol. Finally, we presented the security and performance analysis of the proposed scheme. We carried out provable security analysis in a random oracle model and compared its security and efficiency with those of related schemes. The result shows the security and practicability of the proposed scheme.

Data Availability

The article contains data supporting the results of this study.

Conflicts of Interest

The authors claim that there is no conflict of interest.

Authors' Contributions

All authors made equal contribution to the work.

Acknowledgments

This study was supported by the Research and Development Program for Science and Technology Department of Shaanxi Province (Program No. 2020NY-163), Research and Development Program for Science and Technology of Yulin (Program No. 2019-77-2), Young Elite Scientist Sponsorship Program of the Yulin Association for Science and Technology (Program No. 20190127), National Natural Science Foundation of China (Program No. 61672413), and Scientific Research Program Funded by the Education Department of Shaanxi Province (Program No. 20JY016).

References

[1] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.

[2] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving fusion of IoT and big data for e-health," *Future Generation Computer Systems*, vol. 86, pp. 1437–1455, 2018.

[3] Y. Yang, X. Liu, and R. H. Deng, "Lightweight break-glass access control system for healthcare Internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3610–3617, 2018.

[4] 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.

[5] Q. Jiang, Z. Chen, J. Ma, X. Ma, J. Shen, and D. Wu, "Optimized fuzzy commitment based key agreement protocol for wireless body area network," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[6] 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.

[7] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[8] C.-T. Li, C. C. Lee, and C. Y. Weng, "An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1133–1143, 2013.

[9] 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.

[10] T.-Y. Chen, C. C. Lee, M. S. Hwang, and J. K. Jan, "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.

[11] 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.

[12] 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.

[13] R. Amin, S. K. 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.

[14] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, "User centric three-factor authentication protocol for cloud-assisted

wearable devices," *International Journal of Communication Systems*, vol. 32, no. 6, 2019.

[15] A. K. Das, "A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks," *Wireless Personal Communications*, vol. 82, no. 3, pp. 1377–1404, 2015.

[16] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, pp. 42–62, 2016.

[17] 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.

[18] A.-K. Das, M. Wazid, N. Kumar, M. K. Khan, K. K. R. Choo, and Y. H. Park, "Design of secure and lightweight authentication protocol for wearable devices environment," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1310–1322, 2018.

[19] W. Liu, H. Liu, Y. Wan, H. Kong, and H. Ning, "The yoking-proof-based authentication protocol for cloud-assisted wearable devices," *Personal and Ubiquitous Computing*, vol. 20, no. 3, pp. 469–479, 2016.

[20] F. Wu, X. Li, L. Xu, S. Kumari, M. Karuppiah, and J. Shen, "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server," *Computers & Electrical Engineering*, vol. 63, pp. 168–181, 2017.

[21] S. Liu, S. Hu, J. Weng, S. Zhu, and Z. Chen, "A novel asymmetric three-party based authentication scheme in wearable devices environment," *Journal of Network and Computer Applications*, vol. 60, pp. 144–154, 2016.

[22] S. Jangirala, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 942–956, 2020.

[23] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, p. 1, 2020.

[24] F. Wu, L. Xu, S. Kumari et al., "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *Journal of Network and Computer Applications*, vol. 89, pp. 72–85, 2017.

[25] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.

[26] 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.

[27] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2739–2750, 2019.

[28] Y. Yang, X. Zheng, X. Liu, S. Zhong, and V. Chang, "Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system," *Future Generation Computer Systems*, vol. 84, pp. 160–176, 2018.

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

[30] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[31] 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.

[32] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Generation Computer Systems*, vol. 84, pp. 239–251, 2018.

[33] C.-T. Li, C.-C. Lee, and C.-Y. Weng, "A secure cloud-assisted wireless body area network in mobile emergency medical care system," *Journal of Medical Systems*, vol. 40, no. 5, p. 117, 2016.

[34] H.-L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim, and H. W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[35] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, 59 pages, 2017.

[36] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Systems Journal*, vol. 14, no. 1, pp. 310–320, 2020.