Research Article

Provably Secure Crossdomain Multifactor Authentication Protocol for Wearable Health Monitoring Systems

Hui Zhang,1 Yuanyuan Qian,2 and Qi Jiang2

1School of Information Engineering, Yulin University, Yulin 719000, China
2School of Cyber Engineering, Xi’an 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

Copyright © 2020 Hui Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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...
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
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

![Figure 2: The authentication model for multidomain WHMSs.](image)

(a) issues the registration request to WCS through the secure channel

(b) When receiving the registration request, WCS selects an identity IDWD and calculates the shared key K<sub>WCS-WD</sub> = h(K<sub>WCS</sub>||ID<sub>WD</sub>||RT<sub>WD</sub>). Then, WCS stores {ID<sub>WD</sub>, K<sub>WCS-WD</sub>} in its database. Finally, the message <ID<sub>WD</sub>, K<sub>WCS-WD</sub>> is sent by WCS to WD via the secure channel

(c) WD stores the parameters {ID<sub>WD</sub>, K<sub>WCS-WD</sub>} in its memory

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

2.3. Mobile Device Registration Phase. The holder of MD (i.e., U') performs the following steps (see Figure 4):
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 $\sigma_i' = \text{Rep}(BIO_i, \tau_i)$ and $e_i' = h(h(ID_i, ||PW_i||\sigma_i) \mod l)$ 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 $T = C_x = f_i \oplus h(h(ID_i, ||\sigma_i'||PW_i))$. It continues to generate a random number $b \in Z_n^*$ and then computes $B = bP$, $C = bS = (C_x, C_y)$, $PID_{WD_j} = C_y \oplus ID_{WD_j}$, and $M_1 = h(ID_i || ID_{WD_j} || TC_{WD} || T_1 || C_y)$. $MD_i$ transmits a message $<\text{PID}_i, PID_{WD_j}, T_1, M_1, B>$ 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 $<\text{PID}_i, PID_{WD_j}, T_1, M_1, B>$ of $MD_i$, $MCS_m$ verifies $T_1$ according to the equation $|T_1 - T_1| \leq \Delta T$. If the timestamp is valid, it continues to calculate $C_x' = sB = (C_x', C_y')$, $ID'_{MD_i} = PID_{WD_j} \oplus C_x'$, and $ID_{WD_j}' = PID_{WD_j} \oplus C_y'$. $MCS_m$ obtains the corresponding $K_{MD_i}$ according to $ID'_{MD_i}$ and the table

---

**Table 1: Symbols.**

| Symbol          | Description                                                                 |
|-----------------|-----------------------------------------------------------------------------|
| $U_i$           | The doctor                                                                  |
| $WD_j$          | The wearable device of patients                                             |
| $MD_i$          | The mobile device of $U_i$                                                  |
| $ID_i, ID_{WD_j}$| The identifier of $U_i$ and $WD_j$                                           |
| $PW_i, BIO_i$   | The password and biometric template of $U_i                                |
| $Gen(\cdot), \text{Rep}(\cdot)$ | The generation and reproduction algorithm in a fuzzy extractor |
| $t$             | The registration timestamp                                                  |
| $T$             | The timestamp                                                               |
| $\Delta T$      | 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 $K_{MD}$ for this $MD$, and calculates temporal certificate $TC_i = h(ID_i || K_{MD} || RT_{MD})$. $MCS_m$ stores $\{ID_i, K_{MD}\}$ in its database. Then, $TC_i$ is sent to $MD_i$.

(c) $MD_i$ continues the calculation of $\text{Gen}(BIO_i) = (\sigma_i, \tau_i)$, where $\sigma_i$ is the biometric key and $\tau_i$ is the reproduction parameter. Then, $MD_i$ calculates the fuzzy verifiers $e_i = h(h(ID_i || PW_i || \sigma_i) \mod l)$ and $f_i = TC_i \oplus h(ID_i || \sigma_i || PW_i)$ and stores the parameters $\{\text{Gen}(\cdot), \text{Rep}(\cdot), \tau_i, h(\cdot), e_i, f_i, T\}$ in its memory.
Figure 4: Mobile device registration phase.

**MD$_{j}$**

- Select identity ID$_j$.
- Password PW$_j$.
- Imprint biometrics BIO$_j$.

![Diagram](image1)

Figure 5: Login and authentication phase.

**MD$_{j}$**

- Input ID$_j$, PW$_j$, imprint BIO$_j$.
- $\sigma_j' = \text{Rep(BIO}_j, r_j)$.
- $e_j' = h(h(ID_j || PW_j || \sigma_j') \mod l)$.
- Check if $e_j' = e_j$. If not, abort.
- Generate $T_j$.
- $TC_j = f \mod h(ID_j || \sigma_j' || PW_j)$.

**MCS$_m$**

- Check validity of $T_j$, if so, $C_j = (C_j', C_j)$.
- $ID_j' = \text{PID} @ C_j', ID_j' \rightarrow K_{MD_j}$.
- $ID_{MCS_m}' = \text{PID}_{MCS_m} @ C_j$.
- $TC_j' = h(ID_j || K_{MD_j} || RT_{MD_j})$.
- $M_j' = h(ID_j || ID_{MCS_m}' || TC_j' || T_j || C_j)$.
- Check if $M_j' = M_j$.
- If so, MCS$_m$ authenticates MD$_j$.
- Generate $T_j$, $ID_{MCS_m}' \rightarrow \text{WCS}_k$, $K_{CS_{m,k}}$.
- $M_j = (ID_j', ID_{MCS_m}', T_j || K_{CS_{m,k}})$.

**WCS$_k$**

- Check validy of $T_j$.
- $SK_{CS_{m,k}} = h(K_{CS_{m,k}} || T_j || T_j)$.
- $K_{CS_{m,k}}' = TK_{WCS_k} @ SK_{CS_{m,k}}$.
- $ID_{WCS_k}' || ID_{MCS_m} || ID_{WCS_k} || ID_{WCS_k} || T_j$.

- Check if $M_j' = M_j$.
- If so, MCS$_m$ authenticates WCS$_k$.
- Generate $T_j$, a temporary $K_{WCS_k}$.
- $Ticket_{WCS_k} = (ID_j', K_{WCS_k}, \text{lifetime})_{\text{WCS}_{k}, \text{WCS}_{k}}$.
- $M_j = h(K_{CS_{m,k}} || ID_j || ID_j || T_j)$.
- $ID_{WCS_k}' || ID_{CS_{m,k}} || ID_{WCS_k} || ID_{WCS_k} || Ticket_{WCS_k}$.
- $SK_{CS_{m,k}} = h(K_{CS_{m,k}} || T_j || T_j)$.
- $TK_{WCS_k} = K_{WCS_k} @ SK_{CS_{m,k}}$.

- Check validy of $T_j$.
- $K_{WCS_k}' = TK_{WCS_k} @ C$.
- $M_j' = h(Ticket_{WCS_k}' || ID_{WCS_k}' || T_j || C || Ticket_{WCS_k})$.

- Verify if $M_j' = M_j$.
- If so, MD$_j$ authenticates MCS$_m$. 

---

**MD$_{j}$**

- Generate secret key $K_{MD_j}$.
- $TC_j = h(ID_j || K_{MD_j} || RT_{MD_j})$.
- Store [ID$_j$, $K_{MD_j}$].
stored in its database and calculates $TC_i'' = h(ID_i)||K_{MD}||RT_{MD}$ and $M_1' = h(ID)||ID_{WD}'||TC ||T_1||C_i'$ 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 $K_{CS_{k_{na}}}$ according to ID$_{WD}$.$j$. Then, MCS$_m$ calculates $M_2 = h(K_{CS_{k_{na}}},||ID||ID_{WD}'||T_2')$ and $M_3 = \{ID_i',ID_{WD}'',ID_{MCS_{m}},T_2\}^{K_{CS_{na}}}$ and sends $M_2$, $M_3$, $T_2, ID_{MCS_{m}} > to WCS_k$

(b) WCS$_k$ receives the message $<M_2,M_3,T_2,ID_{MCS_{m}}>$ sent by MCS$_m$, and WCS$_k$ checks the validity of the timestamp $T_2$. If it is valid, WCS$_k$ gets the corresponding $K_{CS_{k_{na}}}$ according to ID$_{MCS_{m}}$, decrypts $M_3$ to obtain $ID_i',ID_{WD}'',ID_{MCS_{m}}$ with $K_{CS_{na}}$, and then checks the equation $ID_{MCS_{m}} = ID_{MCS_{m}}$. If it fails, the session is interrupted. Otherwise, it continues to calculate $M_2' = h(K_{CS_{k_{na}}}||ID_{WD}'||T_3)$ 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 $K_{WCS_{k_{-WD}}}$ according to ID$_{WD}$.$j$, generates the current timestamp $T_3$ and a temporary key $K_{WD}$.$j$, and calculates $SK_{CS_{k_{na}}} = h(K_{CS_{k_{na}}}||T_2||T_3)$, $TK_{WD} = K_{WD}||SK_{CS_{k_{na}}}$, and $M_4 = h(K_{CS_{k_{na}}}||K_{WD}||T_3||ID||ID_{WD}'||ID_{MCS_{m}}||ID_{WC_{k_{na}}}|(Ticket_{WD})$. It sends the message $<Ticket_{WD}, TK_{WD}, T_3, M_4 > to WCS_m$

(c) After receiving $<Ticket_{WD}, TK_{WD}, T_3, M_4 >$, MCS$_m$ checks the freshness of $T_3$. If the timestamp is valid, it continues to compute $SK_{MCS_{m}} = h(K_{CS_{k_{na}}}||T_3||T_3)$, $K_{WD}' = TK_{WD}||SK_{CS_{k_{na}}}$, and $M_4' = h(K_{CS_{k_{na}}}||K_{WD}||T_3||ID||ID_{WD}'||ID_{MCS_{m}}||ID_{WC_{k_{na}}}|(Ticket_{WD})$ 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 $M_5 = h(TC_i'||ID||ID_{WD}'||T_4||C||Ticket_{WD})$ and $TTK_{WD} = K_{WD}'||C$. It sends a message $<Ticket_{WD}, TTK_{WD}, T_4, M_5 > to MD_i$

(d) After $<Ticket_{WD}, TTK_{WD}, T_4, M_5 >$ is received, MD$_i$ checks the freshness of $T_4$ and calculates $K_{WD} = TTK_{WD}||C$ and $M_5' = h(TC_i'||ID||ID_{WD}'||T_4||C||Ticket_{WD})$. It checks if $M_5' = M_5$ is true. If true, established, MCS$_m$ is considered legal by MD$_i$.

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 $M_6 = h(K_{WD}'||T_5)$, and transmits a message $<Ticket_{WD}, T_5, M_6, B > to WD_j$

(b) After accepting $<Ticket_{WD}, T_5, M_6, B >$, WD$_j$ checks the freshness of the timestamp $T_5$. So it obtains $ID_{WD}, K_{WD}, lifetime$ by decrypting $Ticket_{WD}$ with key $K_{WCS_{k_{-WD}}}$ and verifies the validity of

Figure 6: Session key agreement phase.
Ticket\textsubscript{WD}. It continues to calculate $M'_7 = h(K_{WD}||T_9)$ and verifies if the equation $M'_7 = M_7$ is true. If it fails, the session is interrupted. Otherwise, WD\textsubscript{j} treats MD as legitimate. WD\textsubscript{j} generates the current $T_9$, selects a random number $d$, and computes $D = dP$, $M_7 = h(K_{WD}||T_3||T_5||ID_{WD}||ID_{WD}||dB)$, and $SK_{MD-WD} = h(ID_{ID}||ID_{WD}||T_6||T_8||K_{WD}||dB)$. Eventually, it sends $<T_6, M_7, D >$ to MD\textsubscript{i}.

(c) After receiving $<T_6, M_7, D >$, MD\textsubscript{i} calculates $M'_7 = h(K_{WD}''||T_3||T_6||ID_{WD}||ID_{WD}||dB)$ and then verifies if $M'_7 = M_7$ holds. If not, the session is interrupted. Conversely, WD\textsubscript{j} is considered legal by MD\textsubscript{i}. Finally, it calculates the session key $SK_{MD-WD} = h(ID_{ID}||ID_{WD}||T_6||T_8||K_{WD}||dB)$.

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\textsubscript{i}, password $PW_i^0$, and biometric $BIO_i^0$ on MD\textsubscript{i}. Then, MD\textsubscript{i} calculates $\sigma_i^0 = \text{Rep}(BIO_i^0, \tau_i^0)$ and $\epsilon_i^0 = h(h(ID_{ID}||PW_i^0||\sigma_i^0) \mod l)$ and checks if $\epsilon_i^0 = \epsilon_i$ 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\textsubscript{i} calculates the relevant parameters $\text{Gen}(BIO_i^n) = (\sigma_i^n, \tau_i^n), \epsilon_i^n = h(h(ID_{ID}||PW_i^n||\sigma_i^n) \mod l), \text{and} f_i^n = TC_i \oplus h(ID_{ID}||\sigma_i^n||PW_i^n)$. Finally, $U_i$ updates the original $\epsilon_i, f_i, \tau_i$ to $\epsilon_i^n, f_i^n, \tau_i^n$.

2.8. Dynamic Smart Device Addition Phase. New WD\textsubscript{j} and new MD\textsubscript{i} can be dynamically added at this phase.

(1) First, a new wearable device named WD\textsubscript{j}\textsubscript{new}. In essence, this process looks like the WD\textsubscript{j} initialization phase, so it just needs to register at WCS\textsubscript{k}.

(a) WD\textsubscript{j}\textsubscript{new} issues a registration request to WCS\textsubscript{k} through a secure channel.

(b) After the registration request is received, WCS\textsubscript{k} selects an identity ID\textsubscript{WD}\textsubscript{j}\textsubscript{new} for WD\textsubscript{j}\textsubscript{new} and calculates the share key $K_{WD_{new}} = h(K_{WD}||ID_{WD_{new}}||RT_{WD_{new}})$, in which RT\textsubscript{WD_{new}} represents the timestamp when registering WD\textsubscript{j}\textsubscript{new}. Then, WCS\textsubscript{k} stores $\{ID_{WD_{new}}, K_{WD_{new}}\}$ in its database. Finally, the message $<ID_{WD_{new}}, K_{WD_{new}}>$ is given to WD\textsubscript{j}\textsubscript{new} by WCS\textsubscript{k} over the secure channel.

(c) WD\textsubscript{j}\textsubscript{new} stores the parameters $\{ID_{WD_{new}}, K_{WD_{new}}\}$ in its memory.

(2) Secondly, add a new mobile device called MD\textsubscript{i}\textsubscript{new}.

(a) $U_i$ selects ID\textsubscript{MD}\textsubscript{i}\textsubscript{new} and $PW_i^n$ and enters $BIO_i^n$ on the mobile device MD\textsubscript{i}\textsubscript{new}. Then, ID\textsubscript{i}\textsubscript{new} is sent to MCS\textsubscript{m} by $U_i$ via a secure channel.

(b) After receiving the identity ID\textsubscript{MD}\textsubscript{i}\textsubscript{new} of $U_i$, MCS\textsubscript{m} generates a key $K_{MD_{new}}$ for this MD\textsubscript{i}\textsubscript{new} and calculates $RT_{MD_{new}} = h(ID_{ID}||K_{MD_{new}}||RT_{MD})$, in which RT\textsubscript{MD} represents the registration timestamp of MD\textsubscript{i}\textsubscript{new}. MCS\textsubscript{m} stores $\{ID_{MD_{new}}, K_{MD_{new}}\}$ in its database. Finally, $RT_{MD_{new}}$ is sent to MD\textsubscript{i}\textsubscript{new}.

(c) After receiving the message, MD\textsubscript{i}\textsubscript{new} calculates $\text{Gen}(BIO_{new}) = (\sigma_{i,1}^{new}, \tau_i^{new})$, where $\sigma_{i,1}^{new}$ is the biometric key and $\tau_i^{new}$ is the common recovery parameter.

(d) After the above process is completed, MD\textsubscript{i}\textsubscript{new} continues to calculate $\epsilon_i^{new} = h(h(ID_{ID}||PW_{new}||\sigma_i^{new}) \mod l)$ and $f_i^{new} = TC_i \oplus h(ID_{ID}||\sigma_i^{new}||PW_{new})$ and stores the parameters $\{\text{Gen}(\cdot), \text{Rep}(\cdot), \tau_i^{new}, h(\cdot), \epsilon_i^{new}, f_i^{new}, l\}$ 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 $D_{id} \times D_{pw}$ 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 $\Pi^i_U$ represent the $i$th 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 wrong message; when the output has no answer, we use $\bot$ 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 SID 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:

\[ \text{Execute}(\Pi_{\text{MCS}}^m, \Pi_{\text{MD}}^i, \Pi_{\text{WCS}}^k, \Pi_{\text{WD}}^l). \]

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.

\[ \text{Send}(\Pi_U^i, m). \]

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

\[ \text{Reveal}(\Pi^i_U). \]

When the instance $\Pi^i_U$ 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 $\bot$.

\[ \text{Corrupt}(\Pi^i_U). \]

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.

\[ \text{Test}(\Pi_U^i). \]

It can determine the security of the session key owned by the instance $\Pi^i_U$. 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 the queries. It is necessary for A to judge that the bit used by the simulator is 0 or 1 after the Test query. If A 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.$D$. The size of the dictionary space is $|D|$, and the advantage of the attacker to make this attack is defined as $\text{Adv}_{A}^{\text{ake}}(P) = 2\Pr[\text{Succ}] − 1$.

An authentication protocol is called semantically secure, if and only if for all polynomial time attackers, they have the advantage $\text{Adv}_{A}^{\text{ake}}(P)$ which is larger than $q_{\text{send}}/|D|$ that can be ignored, where $q_{\text{send}}$ 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. $\text{Adv}_{A}^{\text{ake}}(P)$ is the advantage for A to break the semantic security of the protocol $P$. A can execute at most $q_{\text{send}}$ send queries and $q_{\text{exe}}$ queries of different instances in the longest time $t$, so we have

\[ \text{Adv}_{A}^{\text{ake}}(P) \leq \frac{q_{\text{send}}}{|D|}. \]

Proof. We use a series of mixed experiments $E_{x_0}, E_{x_1}, E_{x_2}, \ldots, E_{x_7}$ 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 $\text{Adv}_{A}^{\text{ake}}(P)$ be the advantage of the attacker in $E_x$ and $\Delta_x$ denote the degree of distinction between $E_{x_i}$ and $E_{x_{i-1}}$.

\[ E_{x_0}. \]

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

\[ \text{Adv}_{A}^{\text{ake}}(P) = \text{Adv}_{A}^{\text{ake}}(E_{x_0}). \]

\[ E_{x_1}. \]

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 $(s, \text{value})$. 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 $E_x$ from $E_x$.  

$$
\Delta_0 = |\text{Adv}_1(A) - \text{Adv}_0(A)| \leq \text{neg}(\kappa). 
$$

$E_x$. 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  

$$
\Delta_1 = |\text{Adv}_2(A) - \text{Adv}_1(A)| \leq \text{neg}(\kappa). 
$$

$Ex$. Simulation of the active 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 $<M_1, M_2, M_4, M_5, M_6, M_7>$. $E_x$ and $E_x$ are indistinguishable from $A$ unless they provide valid information to end the game. Specifically, for the authentication message $M_1 = h(Id)||ID_{WD}||TC_1||C_x||C_y$, where $TC_i = h(Id)||K_{MD}||RT_{MD}$ or $TC_i = f_i \oplus h(Id)||\sigma_i^j||PW_i$ in the case that no corruption request is made, $\sigma_i^j$, $PW_i$ cannot be obtained or the key $K_{MD}$ is unknown to the attacker, and the valid information $M_1$ cannot be calculated, so the attacker has a negligible probability of success. So  

$$
\Delta_2 = |\text{Adv}_3(A) - \text{Adv}_2(A)| \leq \text{neg}(\kappa). 
$$

$E_x$. Simulation of the active session has been changed in the experiment. For a Send(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  

$$
\Delta_3 = |\text{Adv}_4(A) - \text{Adv}_3(A)| \leq \text{neg}(\kappa). 
$$

$Ex$. We continue to change the simulation of the active sessions during the experiment. If the attacker sends a Reveal (WCS$^5$) query to the WCS, it will get the session key $SK_{CS,d} = h(K_{CS,d}||T_2||T_3)$ between the WCS and the MCS and can also calculate the temporary key $K_{WD,t}$. However, in order to generate valid verification information $M_4$, $A$ needs to generate a valid Ticket$_{WD}$. It is able for $A$ to know the identity of Ticket$_{WD}$, 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 Ticket$_{WD}$, 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  

$$
\Delta_4 = |\text{Adv}_5(A) - \text{Adv}_4(A)| \leq \text{neg}(\kappa). 
$$

$Ex$. 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 $bP$ of the message $M_5$ is actually equivalent to the probability of solving the Elliptic 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  

$$
\Delta_5 = |\text{Adv}_6(A) - \text{Adv}_5(A)| \leq \text{neg}(\kappa). 
$$

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

$$
\Delta_6 = |\text{Adv}_7(A) - \text{Adv}_6(A)| \leq \text{neg}(\kappa). 
$$

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  

$$
\text{Adv}_{E,D}(A) \leq \frac{q_{send}}{|D|}. 
$$

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$_A$, it can get the information $\{\text{Gen}(\cdot), \text{Rep}(\cdot), \tau_i, h(\cdot), e_i, f_i, I\}$ stored in MD$_A$. First, the adversary $A$ wants to correctly guess the doctor’s password $PW_A$, and needs to guess the password and verify the security parameters $e_i = h(ID)||PW_i||\sigma_i^j \mod l$. 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 $|D_a|/l$ 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 \( f_i = T C_i \oplus h(\text{ID}_{i}||\text{PW}_{i}) \). However, \( T C_i = h(\text{ID}_{i}||K_{MD}||RT_{MD}) \), and it is protected by the key \( K_{MD} \), which is generated by \( M_{CSm} \) for \( \text{MD}_i \). 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 \(<\text{PID}_i, \text{PID}_\text{WD}, T_1, M_1, B \>, <M_2, M_3, T_2, \text{ID}_{MCSm} >, <\text{Ticket}_{WD}, \text{TK}_{WD}, T_3, M_4 \>, <\text{Ticket}_{WD}, \text{TTK}_{WD}, T_4, M_5, C \>, <\text{Ticket}_{WD}, T_5, M_6, B \>, \text{and} <T_6, M_7, D \> in the login phase, the authentication phase, and the session key negotiation phase. Then, A re-plays 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 \(<\text{PID}_i, \text{PID}_\text{WD}, T_1, M_1, B \> \) 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 \( B = aB, C = bC = (C_x, C_y) \), \( \text{PID}_i = \text{ID} \oplus C_x \), \( \text{PID}_\text{WD} = \text{ID}_{WD} \oplus C_y \), and \( M_1 = h(\text{ID}_{i}||\text{ID}_\text{WD}||TC_i||T_1||C_z) \). The message receiver will confirm whether the party is a legitimate one by verifying \( M_1 = M_1' \). Both of the messages \( TC_i = f_i \oplus h(\text{ID}_{i}||\sigma_{i}||\text{PW}_{i}) \quad \text{and} \quad TC_i = h(\text{ID}_{i}||K_{MD}||RT_{MD}) \) of \( M_4 \) are protected by a password or a key \( K_{MD} \), 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 \(<M_2, M_3, T_2, \text{ID}_{MCSm} > \) sent from \( M_{CSm} \) to \( \text{WCS}_k \), A intercepts the message as an intermediary and replaces it with its own messages. It wants to pass the verification of \( \text{WCS}_k \) and then needs to send the correct \(<M_2, M_1 \> \). To calculate \( M_2 = h(K_{CS_{mk}}||\text{ID}_\text{WD}||T_2 \) \) and \( M_2 = \{\text{ID}_1', \text{ID}_\text{WD}', \text{ID}_{MC_{Sm}}, T_2 \} K_{CS_{mk}} \), it needs the shared key \( K_{CS_{mk}} \) between \( \text{WCS}_k \) and \( M_{CSm} \), but it cannot get the key. Therefore, it cannot generate the message \(<M_2, M_3 \> \). Similarly, it does not correctly calculate \( \text{Ticket}_{WD}, \text{TK}_{WD}, \text{and} M_4 \) in the next message \(<\text{Ticket}_{WD}, \text{TK}_{WD}, T_3, M_4 > \), because they are both protected by the keys \( K_{WCS_{WD}} \), and \( K_{CS_{mk}} \). 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 \( \text{ID}_i, \text{PW}_i, \) and \( \text{BIO}_i' \) on \( \text{MD}_i \). Then, \( \text{MD}_i \) will calculate \( \sigma_i = \text{Rep}(\text{BIO}_i', \tau_i) \) and \( e_i = h(h(\text{ID}_i||\text{PW}_i||\sigma_i) \mod l) \). \( \text{MD}_i \) 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 \(<\text{PID}_i, \text{PID}_\text{WD}, T_1, M_1, B \>, <M_2, M_3, T_2, \text{ID}_{MCSm} >, <\text{Ticket}_{WD}, \text{TK}_{WD}, T_3, M_4 \>, <\text{Ticket}_{WD}, \text{TTK}_{WD}, T_4, M_5, C \>, <\text{Ticket}_{WD}, T_5, M_6, B \>, \text{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 \( U_i, \text{MD}_i, \) and \( \text{WCS}_k \). Therefore, our protocol achieves anonymity [32, 33]. On the other hand, we can also find that these messages are dynamic. The pseudonymity \( \text{PID}_i \) 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 \( TC_i = f_i \oplus h(\text{ID}_{i}||\sigma_{i}||\text{PW}_{i}) \) and \( M_1 = h(\text{ID}_{i}||\text{ID}_\text{WD}||TC_i||T_1||C_z) \). So MD_ can pass the authentication of \( MCS_{Sm} \) successfully via checking the correctness of \( M_1 \). Similarly, an adversary cannot calculate correct \( M'_1 = h(\text{TC}||\text{ID}_\text{WD}||T_4||C|| Ticket_{WD} \) without knowing \( TC_i '' \). Since only \( M_{CSm} \) knows the secret key \( s \), it can compute the valid \( TC_i '' \). Thus, MD_
In our scheme, the previous session key, our protocol accomplishes known key security. Thus, our protocol achieves mutual authentication between ID and UserID.

4.7. Known Key Security. It is assumed that the adversary has obtained the session key SKMD, and can authenticate kMD by verifying the correctness of TicketMD, since only the legal MD can decrypt KMD 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, Uk has long-term secrets PWk, BIOk, and ek = h(IDk||PWk||σk) mod l, and when the long-term secrets of Uk are leaked, the previous session key SKMD,WD = h(IDk||IDWD||Tk||KWD||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, documentors can freely update passwords or biometrics in our protocol, as shown in Section 2.7.

## 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: 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 | ✓                  | ✓                  | ✓          |

Table 3: Efficiency comparison.

| Schemes               | Our scheme | The scheme in [34] | The scheme in [35] |
|-----------------------|------------|--------------------|--------------------|
| MD(Ii)                | 8Tth + Tp  | 5Tth + 2Tp         | 5Tth + 3Tp         |
| MCSm(CS)              | 6Tth + 3Tp + Tp | 2Th + 3Tp         | 4Tth + Tp |
| WDj                   | 3Tth + Tj  | 2Tth + 2Tp         | 4Tth + Tp         |
| WCSk                  | 3Tth + 2Tp | —                  | —                  |

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...
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 cross-domain 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. 201JY016).

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.-L. 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 curve 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 curve 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.