Hindawi
Security and Communication Networks
Volume 2022, Article ID 6735003, 18 pages
https://doi.org/10.1155/2022/6735003

Research Article

A Blockchain-Based User Authentication Scheme with Access Control for Telehealth Systems

Shuyun Shi,[1,2] Min Luo,[1,3] Yihong Wen,[4] Lianhai Wang,[3] and Debiao He[1,5]

[1]*School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China*
[4]*54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China*
[5]*Shanghai Key Laboratory of Privacy-Preserving Computation, MatrixElements Technologies, Shanghai 201204, China*

Correspondence should be addressed to Min Luo; mluo@whu.edu.cn

Received 20 October 2021; Accepted 7 February 2022; Published 30 March 2022

Academic Editor: Mamoun Alazab

With the development of telecommunication systems and customized monitoring devices, telehealth has been widely used to improve medical quality and reduce overall health costs. However, the convenience of connection between the providers and patients through a public channel also leads to significant security and privacy concerns. Though there have been many authentication schemes designed for secure communications in telecare systems, most of them suffer from malicious attacks or have heavy computation and communication costs. Thus, in this article, we proposed a blockchain-based user authentication scheme integrating with access control and physical unclonable function (PUF). Permissioned blockchain and PUF are used to support secure data sharing across the healthcare service providers and identify the devices, respectively. Security analysis shows that our protocol satisfies the security requirements for telehealth services and is provably secure in the random oracle model. The performance evaluation demonstrates that it has less computation and communication costs compared with three of the latest schemes.

1. Introduction

Telecare medical information systems (TMIS) support remote medical services by providing online patient diagnosis to reduce healthcare service costs and improve patient health outcomes. The COVID-19 pandemic has promoted the use of telehealth to deliver, which can interrupt the transmission of the disease and facilitate public health mitigation by reducing outings.

As shown in Figure 1, patients at home are equipped with wearable monitoring devices. These devices can continually collect and transmit health data (e.g., blood pressure, blood sugar, heart rate, and more) to mobile devices. Health data will be transmitted to healthcare service providers (e.g., hospitals and health authorities) over the open internet. Then, care staff (e.g., doctors and nurses) can monitor the patient's condition remotely and make timely treatment decisions for better outcomes.

Vulnerabilities in the wireless networks offer some easy entry points for malicious adversaries, yet these networks are important for connections between patients at home and remote healthcare organizations. Many countries have laws that are designed to protect the patient's privacy, such as the Health Insurance Portability and Accounting Act (HIPAA) in the United States. Security in telecare services, i.e., how to ensure patient data security and privacy during transmission through the public channel, becomes a significant concern [1, 2].

User authentication is a necessary first step to ensure that only authorized users have access to protected data. Password-based user authentication scheme as the most convenient mechanism is widely employed, however, it is vulnerable to various attacks and could be a threat to data

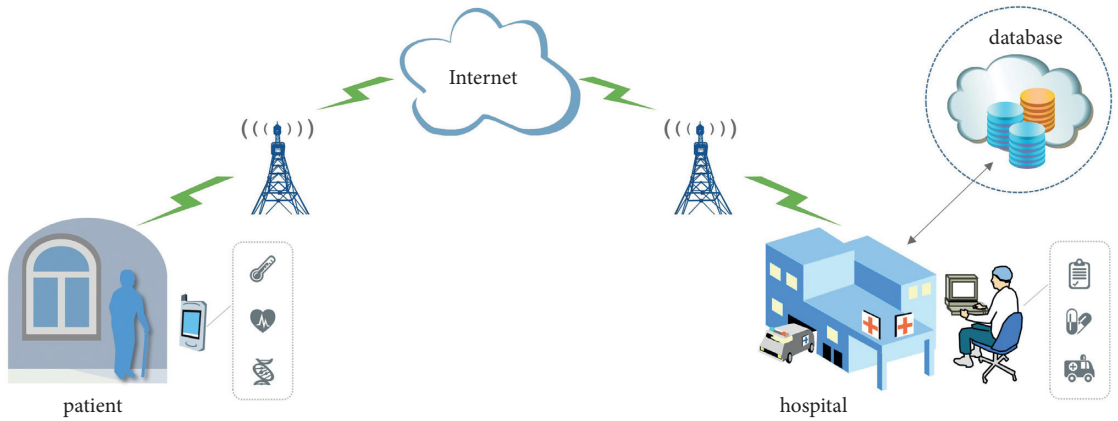

Figure 1: General architecture for telehealth system.

security. Multifactor authentication is a recommended approach in which any user is granted access to certain data after validating two or more pieces of evidence the user has.

In the meantime, people's mobility between different healthcare institutions raises some problems: user authentication in multiple servers and secure data sharing across different servicers. To address the above issues, multiserver authentication is used for access to multiple servers with one credential [3] in existing schemes. However, most schemes [4–6] have unsatisfactory performance or may suffer serious security problems in the telehealth services environment.

Furthermore, interaction with patient records from personal devices may be fraught with peril since it is difficult for healthcare servers to verify whether those devices meet security configuration requirements. Attackers could impersonate legal users for free treatment or profit once they guess the correct password. Otherwise, they could impersonate the healthcare providers to offer false treatment, which will cause a critical medical accident.

Therefore, it is necessary to design a new authentication scheme to ensure security and data sharing and resist various attacks in the telecare services environment.

1.1. Our Contribution. In this paper, we propose a multiserver authentication scheme with the integration of user authentication and access control, which determines the access control to achieve selective data disclosure, enhance data privacy, and improve the scheme's efficiency.

Blockchain has been widely applied in different areas because of its properties of immutability and decentralization [7–9]. In our scheme, some registration information will be recorded in the blockchain served as a trusted bulletin board to achieve secure data sharing across different healthcare provider servers.

PUF has been a promising cryptographic primitive, and it has been demonstrated in the security domain as well [10]. It serves as one of the authentication factors to identify the devices in the IoT systems, with the characteristics of being unclonable, unpredictable, and computable. We will introduce this technology to resist various attacks (e.g., impersonation attacks and physical attacks) for telehealth services.

The major contributions of this paper are summarized as follows:

(i) Firstly, by leveraging blockchain technology and PUFs, we design an efficient user authentication protocol with access control for the telehealth system.

(ii) Secondly, a comprehensive security analysis is given to show that the proposed protocol is provably secure and can satisfy the security requirements in the telehealth system.

(iii) Finally, we implement a prototype by smart contract based on Hyperledger Fabric. We analyze the performance to show that our scheme has less computation and communication costs than previously proposed protocols.

1.2. Organization of Our Paper. The paper is organized as follows: in section 2, we discuss some related literature. In section 3, we introduce the preliminaries used in this paper. In sections 4 and 5, we present the details of the system framework and proposed authentication protocol for telecare service. In sections 6 and 7, we give provable security in the random oracle model and evaluate the performance of the proposed scheme. Finally, section 8 concludes this paper.

2. Related Work

To secure remote healthcare services, some authentication schemes have been proposed. Debiao et al. [11] proposed an improved scheme to overcome the weakness of Wu et al.'s scheme [12] for TMIS. Their scheme is more efficient and applies to low-power mobile devices. However, the scheme cannot resist the offline password attack. Kumari et al. [13] proposed an improved user authentication scheme for applications in TMIS. Chen et al. [14] proposed a medical data exchange protocol based on the cloud environment for medical advice. Patient inspection information can be protected by asymmetric encryption. However, the scheme cannot provide user anonymity and has heavy computation costs.

Chiou et al. [15] resolved these security problems and provided a complete system implementation. Mohit et al. [4] reviewed Chiou et al.'s [15] protocol and found that it is susceptible to user anonymity and some attacks. They designed a lightweight authentication scheme in the cloud environment for TMIS. However, Kumar et al. [5] found that Mohit et al.'s [4] scheme is vulnerable to various attacks and cannot provide user anonymity and session key protection. They proposed an improved protocol for single-server architecture in TMIS but could not satisfy the perfect forward secrecy or multiserver environment.

Multiserver authentication scheme was first proposed by [3] using a neural network. Because of the complexity of the neural network, lots of schemes [16–19] were proposed to improve the performance and enhance the security. Multiserver authentication scheme without online RC [20, 21] is suitable for various applications, which reduces the cost of trusted RC establishment and authentication communication.

Recently, blockchain has become a research hotspot in telemedicine to ensure healthcare data security and to support data sharing. Liu et al. [22] proposed privacy-preserving mutual authentication in TMIS, which provides user anonymity and malicious users traceability if necessary. Yazdinejad et al. [23] designed a blockchain-based authentication scheme to reduce reauthentication across different hospitals, which can increase throughput and reduce time overhead for resource-limited devices. Li et al. [24] proposed a group authentication mechanism for authorized group members to access sensitive health records in the remote medical monitoring scene. Cheng et al. [25] designed a multiple identity authentication for a secure medical data sharing model based on blockchain to avoid over-reliance on a third party. Lin et al. [26] designed a transitively closed undirected graph authentication scheme for dynamic blockchain-based identity management systems, which is efficient for signers to update their certificates without signing again. Wang et al. [27] proposed a decentralized, secure, and lightweight certificateless signature (CLS) protocol by transforming the logic of KGC into smart contract code, which can resist key generation center compromised attacks and distributed denial of service (DDoS) attacks. Xiong et al. [28] presented ECDSA batch verification protocol in the blockchain-enabled Internet of Medical Things (IoMT) to enhance authentication efficiency and support identification algorithms for false signatures.

Nevertheless, all of the above schemes have no consideration for integration authentication with access control to improve the system efficiency. The idea of the integration of user authentication and access control was first proposed by Harn and Lin [29] to avoid potential security problems between these two protection mechanisms. Later, some improved protocols [30–32] were designed to enhance security and implement a protection scheme in distributed systems. Recently, Lin et al. [33] presented a remote mutual authentication scheme with fine-grained access control based on blockchain for industry 4.0 deployment. Xiong et al. [34] proposed an integrated scheme for a mobile cloud environment (MCC) without the trusted party to store the access control list. However, the computational overhead is expensive for limited mobile devices in telemedicine services.

Additionally, many PUF-based authentication schemes have been proposed for IoT systems, wireless sensor networks (WSNs), and so on. Since smart cards/devices are not tamper-resistant, some two-factor authentication schemes are susceptible to various physical attacks. To address the issues, PUFs are used as one of the authentication factors presented in some literature [35–38] to enhance the properties of lightweight authentication solutions.

In this paper, we will construct a blockchain-based user authentication scheme for better efficiency and security, in which access control integration can enhance the authentication efficiency, and a physical unclonable function is applied to identify the devices against various attacks in the telehealth environment.

3. Preliminaries

3.1. Bilinear Pairings. Let p, q be large prime numbers. Let G_1 be a cyclic additive group and G_2 be a multiplicative group with the same order q. e: $G_1 \times G_1 \longrightarrow G_2$ denotes a bilinear map, where a generator P of G_1 and a generator $g = e(P, P)$ of G_2, when the subsequent conditions meet.

(i) *Bilinear*: $e(aP, bQ) = e(P, Q)^{ab}$ for all $a, b \in Z_q^*$ and $P, Q \in G_1$.

(ii) *Nondegeneracy*: there exists an element $P \in G_1$ such that $e(P, P) \neq 1_{G_2}$.

(iii) *Computability*: given any two elements $P, Q \in G_1$, it is efficient to compute $e(P, Q)$.

The following mathematical problems are difficult, i.e., there is no polynomial algorithm to solve will be used in our proposed scheme.

(i) *Discrete Logarithm (DL) Problem*: given an element $Z \in G_1 (z \in G_2)$, find α such that $Z = \alpha P (z = g^\alpha)$.

(ii) *Computational Diffie-Hellman (CDH) Problem*: given two elements $xP, yP \in G_1 (g^x, g^y \in G_2)$, where $x, y \in Z_q^*$ are unknown, calculate $xyP \in G_1 (g^{xy} \in G_2)$.

(iii) *Modified Bilinear Inverse Diffie-Hellman with k Value (k-mBIDH) Problem*: given k elements $a_1, a_2, \ldots, a_k \in Z_q^*$ and k+2 elements $\widehat{s}P, \widehat{t}P$, $(1/\widehat{s} + a_1)P$, $(1/\widehat{s} + a_2)P, \ldots, (1/\widehat{s} + a_k)P \in G_1$, where $\widehat{s}, \widehat{t} \in Z_q^*$ is unknown, calculate $e(P, P)^{\widehat{t}/\widehat{s}+a}$ for any $a \notin \{a_1, a_2, \ldots, a_k\}$.

3.2. Chinese Remainder Theorem (CRT). Given N coprime integers $m_1, m_2, \ldots, m_k$, where $\gcd(m_i, m_j) = 1$ for $i, j = 1, 2, \ldots k$ and $i \neq j$. For integer a_i, there exists the following:

$$\begin{cases} X \equiv a_1 \bmod m_1 \\ X \equiv a_2 \bmod m_2 \\ \vdots \\ X \equiv a_k \bmod m_k \end{cases} . \quad (1)$$

The common solution X can be computed as follows:

$$X \equiv \sum_{i=1}^{k} \frac{M}{m_i} \cdot e_i \cdot a_i, \quad (2)$$

where $M = \prod_{i=1}^{k} m_i$ and $M/m_i \cdot e_i \equiv 1 \bmod m_i$ for $i = 1, 2, \ldots, k$.

3.3. Physical Unclonable Function. The physical unclonable function is a one-way mapping from a challenge space $\mathbb{C}$ to a response space $\mathbb{R}$ based on the unclonable characteristic of the underlying physical device. A set of challenge-response pairs (CRPs) is unique for each device, which can be used to identify the device. A PUF circuit must meet the properties below:

(i) *Unpredictability*: given any challenge $\mathscr{C}$, the probability to evaluate the response $\mathscr{R}$ of PUF is negligibly small without PUF instance.

(ii) *Computability*: given any challenge $\mathscr{C}$, it is easy to evaluate the response $\mathscr{R}$ for any PUF instance.

(iii) *Uniqueness*: for any two PUF_1 and PUF_2, given the same challenge $\mathscr{C}$, the probability to evaluate the same response $PUF_1(\mathscr{C}) = PUF_2(\mathscr{C})$ is negligibly small.

(iv) *One-way*: given any $\mathscr{R} \in \mathbb{R}$, there exists no polynomial algorithm InversePUF: $\mathscr{R} \longrightarrow \mathscr{C}$ for any challenge $\mathscr{C} \in \mathbb{C}$.

3.4. Fuzzy Extractor. The PUF circuit is susceptible to interference to generate the noisy response with low entropy, which may fail to authenticate the device. Fuzzy extractor has been developed to recover a reliable high-entropy response from a noisy response to enhance the security of authentication. Fuzzy extractor consists of the following two algorithms:

(i) $Gen(\cdot)$: a probabilistic key generation algorithm $(\kappa, h\ d) = Gen(R)$ to generate the key κ and helper data $h\ d$.

(ii) $\mathrm{Rec}(\cdot)$: a deterministic reconstruction algorithm $(\kappa) = Rec(R, \mathrm{hd})$ to recover the key κ from a noisy response R' and helper data $h\ d$, which the Hamming distance between the original response R and the noisy response R' is at most d.

3.5. Blockchain and Smart Contract. Blockchain technology is an immutable and distributed data ledger consisting of an append-only sequence of blocks chained by the cryptographic hash function. Based on permission authorized to network nodes, blockchain platforms are divided into three types: private blockchain, consortium blockchain, and public blockchain. In the proposed scheme, Hyperledger Fabric is chosen to support the flexible transaction and Turing-complete smart contracts.

The smart contract is an autoexecuted program deployed in the blockchain network, which can achieve complex functions, and it can be invoked by authorized nodes sending a legal transaction. The transaction is contained in the block after verification by the nodes. In our proposed scheme, we design the smart contract for registration information sharing between a trusted register center and multiple healthcare servers. The register center and servers can upload, query, and update the information by sending a transaction signed by its private key to invoke the smart contract. Users can query the information to ensure that their information can be authenticated by the whole network.

4. System Framework and Security Model

4.1. Network Model. Figure 2 describes the network model that consists of four types of entities in our proposed scheme, namely the trusted registration center (RC), the permissioned blockchain network (BC), the servers (hospital and medical institutions), and the mobile users (patients and healthcare servicers).

(i) *RC*: it is a trusted third party and is responsible for system initialization, user/server registration, and registration information recorded in the blockchain network. It generates a master private key and issues a private key of the user/server according to their identities. Besides, it also randomly generates a challenge and sends it to the user's devices. The registration record and challenge-response pair will be uploaded in the permissioned blockchain.

(ii) *BC*: it acts as a trusted recorder for registration, sharing, and updating by the smart contract. The trusted RC and servers join as network nodes to maintain the permissioned blockchain network together.

(iii) *Healthcare Servers*: the servers provide data storage for patient health records from wearable devices and support remote healthcare services between the healthcare provider and patients at home.

(iv) *Mobile Users*: there are two types of mobile users: patients and healthcare providers. We assume that all the mobile devices are equipped with a PUF and fuzzy extractor. The output of the PUF is used as one of the authentication factors. Moreover, a fuzzy extractor has been employed to recover the noisy PUF. The mobile devices of patients can upload biomedical data collected by wearable devices to remote servers for personalized medical services after mutual authentication. The mobile devices of healthcare providers can get patient conditions and offer clinical diagnostics after authentication.

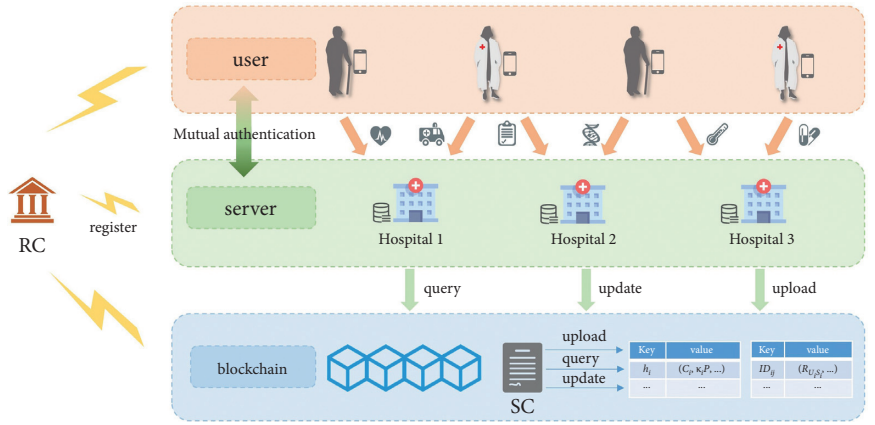

Figure 2: System model for remote healthcare service.

4.2. Network Assumptions. Patients can enjoy remote medical service by off-chain subscription service that hospitals/medical institutions provide and generate the access control list representing patients' service time. Healthcare providers should also get permission (e.g., read, write, delete) to operate the data to achieve diverse data sharing across different institutions.

Each subscription service maps to a specific value according to the mapping rules AC_{S_j} predefined and will be stored in the blockchain by each server, which can be retrieved by any server to check the validity. The access control list is represented by $ACL = \langle S, R \rangle$, where $S = \{S_1, S_2, \ldots, S_n\}$, $R = \{R_{S_1}, R_{S_2}, \ldots, R_{S_n}\}$, where $R_{U_i S_j}$ represents the access permission the user applies for in S_i, as shown in Tables 1 and 2. The server predefines the mapping rules AC_{S_j} of access permission, as shown in Table 3, and sends it to the RC in the registration phase.

Following the research efforts [39], after mapping, RC calculates the common solution M of each user using CRT as follows:

$$\begin{cases} M \equiv \mathrm{Num}_{R_1} \bmod N_{S_1} \\ M \equiv \mathrm{Num}_{R_2} \bmod N_{S_2} \\ \vdots \\ M \equiv \mathrm{Num}_{R_k} \bmod N_{S_n} \end{cases}. \tag{3}$$

In the authentication phase, the specific access permission can be calculated by the common solution.

$$\mathrm{Num}_{R_i} \equiv M \bmod N_{S_i}, \tag{4}$$

then the server can check whether the user has been authorized or expired by equation (4).

4.3. Security Requirements. To ensure the security of the remote healthcare service, the proposed scheme should satisfy the following requirements [21, 34–36, 40]:

Single registration: for convenience to users, the proposed scheme should provide single registration. Users only register with the trusted RC once before they can communicate with healthcare servers.

Table 1: The access control list of patients.

S_i	S_1	S_2	S_3	...	S_n
$R_{U_i S_j}$	One month	One month	Three months	...	Seven months

Table 2: The access control list of healthcare providers.

S_i	S_1	S_2	S_3	...	S_n
$R_{U_i S_j}$	Read	Read write	Delete	...	Read Write Delete

No online registration center: the proposed scheme should achieve mutual authentication without RC to relieve the communication overhead and resist the single point of failure.

Mutual authentication: to ensure that legal users and servers could access healthcare data, the proposed scheme should provide mutual authentication between U_i and S_j to verify the legality of each other.

User anonymity: to preserve the privacy of users, the proposed scheme should protect user identity. Any adversary cannot extract the real identity from the intercepted message.

Untraceability: for the better protection of user privacy, no potential adversary can trace the user's activities and behavior patterns by analyzing the intercepted message.

Session key agreement: the proposed scheme should generate the session key between the users and servers to encrypt the message in future communications.

Perfect forward secrecy: the proposed scheme should provide perfect forward secrecy to ensure the security of messages in the previous sessions. No potential attackers can generate the session key of previous sessions even if they obtain the long-term private key of two participants.

Two-factor security: the proposed scheme should provide two-factor security, i.e., password and mobile device embedded with PUF.

Table 3: The mapping rules of access permission.

Permission	One month	Three months	Five months	Seven months	Only read	Read write	Read write delete
Numbers	3	4	5	7	11	13	17

Access control for data privacy: the healthcare service providers can provide personalized remote healthcare services that users subscribe to for a specific service time. In the meantime, healthcare providers can be authorized to access the data with different permissions for preserving data privacy.

Resistance of various attacks: to resist known attacks existing in the service system, the proposed scheme should resist known attacks, including impersonation attacks, physical attacks, replay attacks, man-in-the-middle attacks, etc.

5. The Proposed Scheme

We describe the proposed scheme in this section. The proposed scheme consists of seven phases: blockchain initialization, system setup phase, server registration phase, user registration phase, mutual authentication phase, password update phase, and access rights update phase.

5.1. Blockchain Initialization. RC establishes a consortium blockchain (e.g., Hyperledger Fabric) among RC and servers as network nodes to maintain the blockchain. The servers must register and enroll the identities as legitimate members to engage in the consensus process.

In the meantime, the smart contract (SC) will be deployed in the blockchain network and can be invoked by transaction. Algorithms 1 and 2 show that the smart contract supports the upload and query of challenge-response pair (CRP) and access permission. In the subscription phase, the servers will upload the subscription service time (for patients) or service permission (for healthcare providers) into the blockchain, which can be validated when RC computes users' common solution. In the registration phase, RC will randomly generate the challenge and then receive the response produced by user's fuzzy extractor via a secure channel. The challenge-response pair will be stored in the ledger to share with servers in the authentication process.

5.2. System Setup Phase. In this phase, the trusted RC generates system parameters and selects the master private key.

(1) RC selects additive group G_1 and multiplicative group G_2 with the same prime order q and a bilinear pairing e: $G_1 \times G_1 \longrightarrow G_2$. RC also chooses a generator P of G_1 and then computes $g = e(P, P)$ of G_2.

(2) RC randomly chooses $s \in Z_q^*$ as the master private key and calculates its public key $P_{\text{pub}} = sP$.

(3) RC selects seven secure hash functions h_0: $\{0,1\}^* \longrightarrow \{0,1\}^{l_0}$, h_1: $\{0,1\}^* \times G_1 \longrightarrow Z_q^*$, h_1': $\{0,1\}^* \longrightarrow Z_q^*$, h_2: $\{0,1\}^* \times \{0,1\}^* \times G_1 \times \{0,1\}^* \times G_1 \times \{0,1\}^* \longrightarrow \{0,1\}^*$, h_3: $G_2 \longrightarrow \{0,1\}^*$, h_4: $\{0,1\}^* \times G_1 \times \{0,1\}^* \longrightarrow Z_q^*$, h_5: $\{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* G_1 \times G_1 \times G_1 \times \{0,1\}^* \longrightarrow \{0,1\}^*$, h_6: $G_1 \times G_1 \times G_1 \times G_1 \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \longrightarrow Z_q^*$

(4) RC publishes system parameters params $= \{G_1, G_2, P, e, q, g, P_{\text{pub}}, h_0, h_1, h_1', h_2, h_3, h_4, h_5, h_6\}$ and keeps $\{s\}$ securely.

5.3. Server Registration Phase. As shown in Figure 3, the server S_j registers with the RC to obtain its long-term private key through a secure channel. The steps below will be executed between S_j and RC.

(1) The server S_j selects its ID_{S_j} and predefines the access control mapping rules AC_{S_j}. Then, S_j transmits them to RC.

(2) RC computes $\mathscr{D}_{S_j} = 1/s + h_1'(ID_{S_j}) \cdot P$, selects a coprime integer N_{S_j}, and sends $\{\mathscr{D}_{S_j}, N_{S_j}\}$ to S_j. RC stores $\{S_j, N_{S_j}, AC_{S_j}\}$ securely.

(3) After receiving, S_j keeps $\{\mathscr{D}_{S_j}, N_{S_j}\}$ secretly.

5.4. User Registration Phase. There are two types of mobile users in this scheme, including patients (remote service subscribers) and healthcare providers (remote service providers). The following steps will be executed by mobile users to register with RC, as shown in Figure 4.

(1) The mobile user U_i selects an identity ID_i and a password PW_i and generates a nonce n_i. After subscription service, the access control list $ACL_i = \langle S_j, R_{U_i} \rangle$ and $P_i = h_0(ID_i \| PW_i \| n_i)$ will be sent to RC through a secure channel.

(2) After receiving the request, RC computes the common solution M using CRT according to the access control list of user $R_{U_i} = \{R_{U_iS_1}, R_{U_iS_2}, \ldots, R_{U_iS_n}\}$, mapping rules $\{AC_{S_1}, AC_{S_2}, \ldots, AC_{S_n}\}$, and $N = \{N_{S_1}, N_{S_2}, \ldots, N_{S_n}\}$ by equation (3). After that, RC generates a random number $r_i \in Z_q^*$, $R_i = r_iP$, and calculates $\mathscr{D}_i = (r_i + sh_i) \bmod q$ using $h_i = h_1(ID_i \| M \| R_i)$ and its master private key. RC also generates a random challenge C_i and sends it to U_i.

(3) After receiving the challenge C_i, U_i extracts the outputs of the PUF $\gamma_i = PUF(C_i)$ and then obtains the secret key κ_i and helper data $h\ d$ from $FE.\text{Gen}(\gamma_i)$ using a fuzzy extractor. After that, U_i sends $\{\kappa_i, h\ d\}$ to RC for authentication in future.

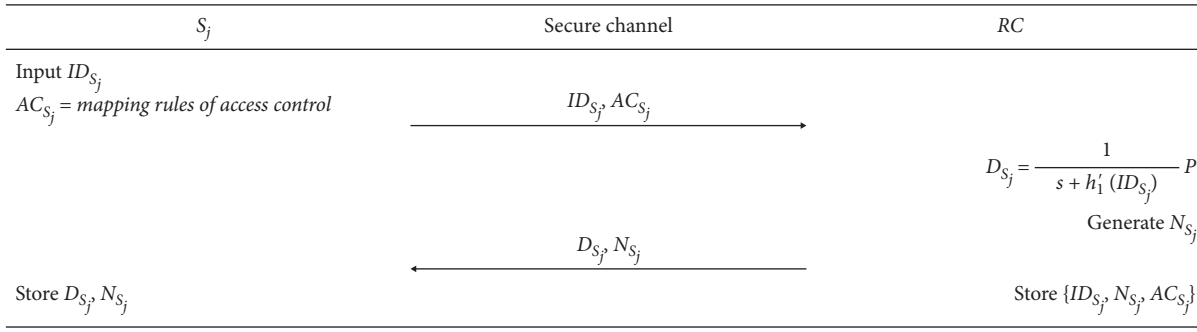

Figure 3: Server registration.

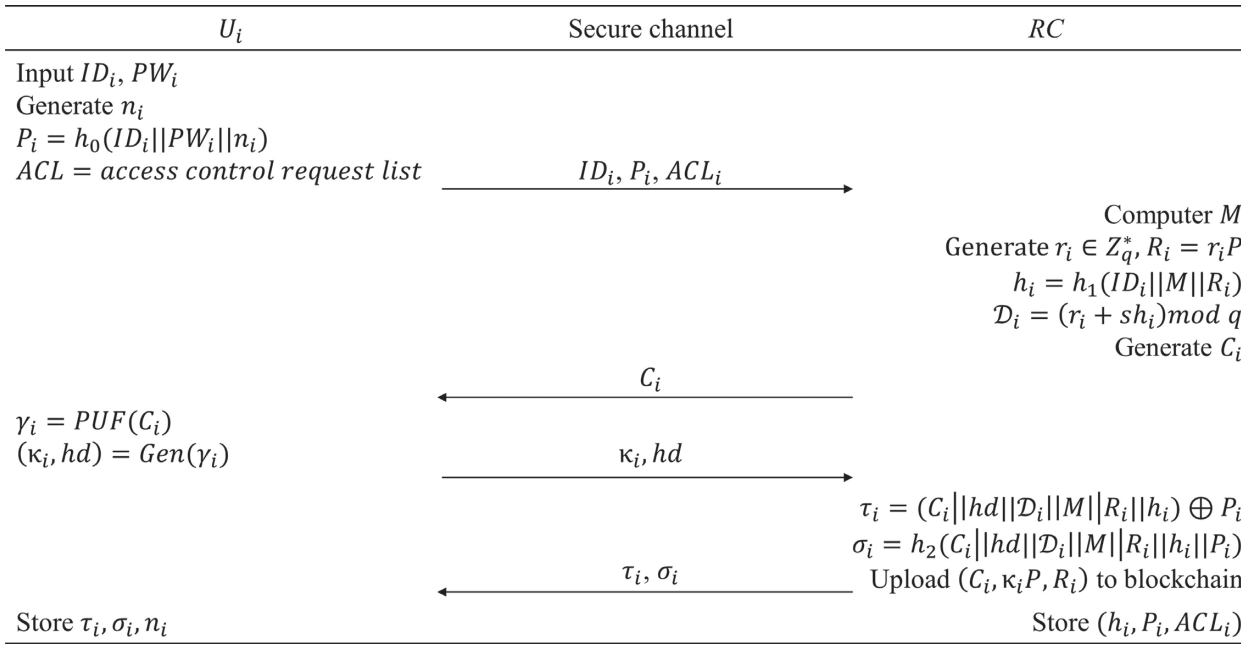

Figure 4: Mobile user registration.

(4) RC computes $\tau_i = (C_i \| \mathrm{hd} \| \mathscr{D}_i \| M \| R_i \| h_i) \oplus P_i$, $\sigma_i = h_2(C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i \| P_i)$, and $\kappa_i P$ and then uploads the "challenge-response" pair $\{C_i, \kappa_i P, R_i\}$ in the blockchain to share with the servers (healthcare providers), where t_s denotes the service start time as shown in Algorithm 2. After that, RC sends $\{\tau_i, \sigma_i\}$ to U_i and stores the user registration table $\{ID_i, P_i, ACL_i\}$ in the blockchain.

(5) U_i stores $\{\tau_i, \sigma_i, n_i\}$ in the secure memory.

5.5. Authentication Phase. As depicted in Figure 5, the user U_i authenticates with the server S_j, and then a common session key SK is generated for secure communications.

(1) U_i inputs his identity ID_i and password PW_i. His/Her mobile device computes $P_i = h_0(ID_i \| PW_i \| n_i), C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i = \tau_i \oplus P_i, \sigma_i' = h_2(C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i \| P_i)$ and then checks whether the equation $\sigma_i' = \sigma_i$ holds. If not, it terminates the request. Otherwise, the device extracts the PUF output $\gamma_i' = PUF(C_i)$ and $\kappa_i = \mathrm{Rec}(\gamma_i', \mathrm{hd})$ using a fuzzy extractor. After that, it chooses a random number $x \in Z_q^*$ and calculates $X' = g^x, X = xP, U_1 = x(P_{\mathrm{pub}} + h_1'(ID_{S_j})P), N = h_3(X') \oplus (ID_i \| M \| h_i \| X), w = h_4(N \| U_1 \| \kappa_i \| T_{us}) \| U_2 = \mathscr{D}_i + xw + \kappa_i$, where T_{us} denotes the current timestamp. U_i sends $\{N, U_1, U_2, w, T_{us}\}$ to S_j via a public channel.

(2) On receiving the message, S_j first checks whether the timestamp is fresh. If not, S_j rejects the session. Otherwise, S_j computes $X' = e(U_1, \mathscr{D}_{S_j})$ by its secret key and $ID_i \| M \| h_i \| X = N \oplus h_3(X')$. S_j invokes the smart contract to get $\{C_i, \kappa_i P, R_i, t_s, \mathrm{status}\}$ by input parameters h_i. After that, S_j checks whether the equation $U_2 P = R_i + h_i P_{\mathrm{pub}} + wX + \kappa_i P$ holds. If not, S_j fails to authenticate U_i. Otherwise, S_j calculates $R_{U_i S_j} \equiv M \mathrm{mod} N_{S_j}$ to check whether the access control permission has expired by the equation $R_{U_i S_j} > |t_e - t_s|$, where t_e denotes current time. If expired/not authorized, S_j terminates the request. Otherwise, S_j selects a random number $y \in Z_q^*$, computes $Y = yP, K = yX, V_j = h_5(ID_i \| ID_{S_j} \| M \| R_{U_i S_j} \| X \| Y \| K \| T_{su})$, and sets the session key

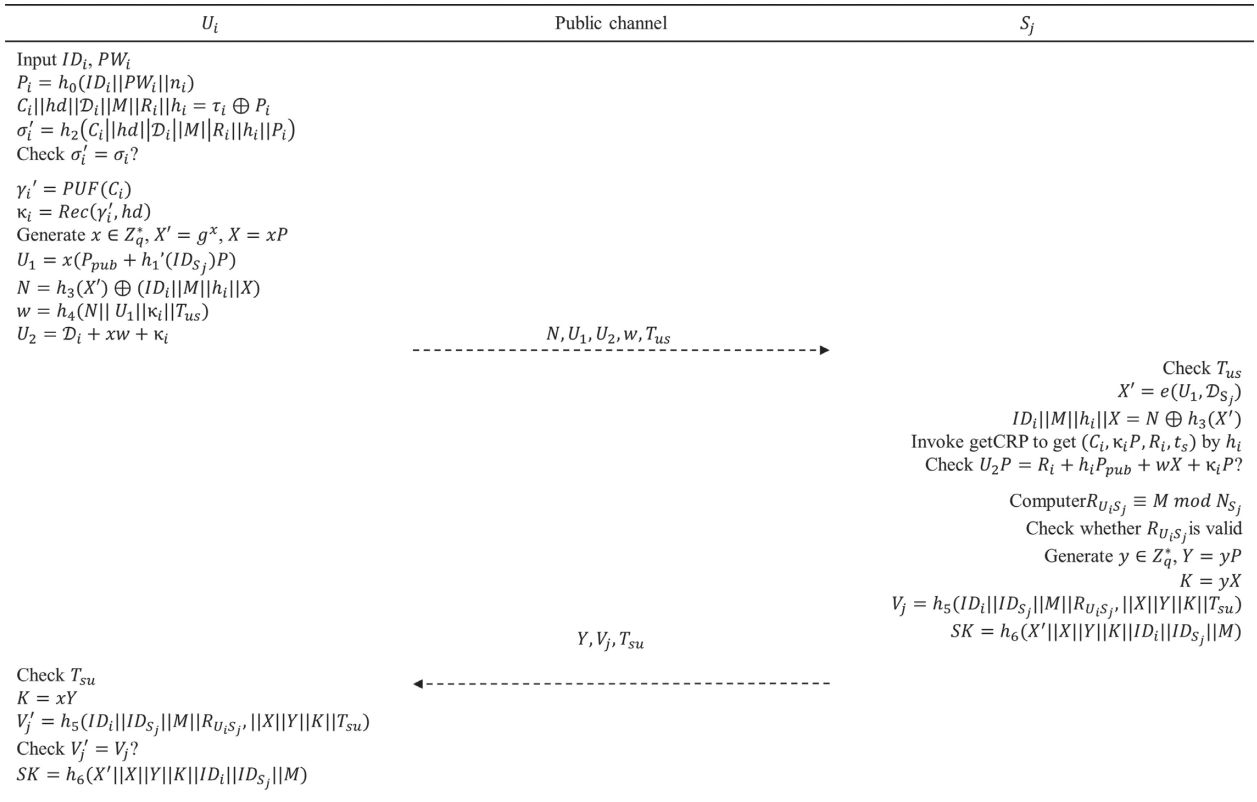

Figure 5: Mutual authentication.

$SK = h_6(X'\|X\|Y\|K\|ID_i\|ID_{S_j}\|M)$, where T_{su} is the current timestamp. Then, S_j sends $\{Y, V_j, T_{su}\}$ to U_i.

(3) After receiving the message, U_i first checks whether T_{su} is fresh. If not, U_i terminates the session. Otherwise, U_i computes $K = xY, V_j' = h_5(ID_i\|I\ D_{S_j}\|M\ \|R_{U_iS_j}\|X\|Y\|K\|T_{su})$ to check whether V_j' and V_j are equal. If not, U_i aborts the session. Otherwise, U_i calculates the session key $SK = h_6(X'\|X\|Y\ \|K\|ID_i\ \|ID_{S_j}\|M)$.

Note that the server can obtain X' and validate the user identity from the two following equations:

$$
\begin{aligned}
e\left(U_1, \mathscr{D}_{S_j}\right) &= e\left(x\left(P_{\text{pub}} + h_1'\left(ID_{S_j}\right)P\right), \frac{1}{s + h_1'\left(ID_{S_j}\right)}P\right) \\
&= e\left(x\left(sP + h_1'\left(ID_{S_j}\right)P\right), \frac{1}{s + h_1'\left(ID_{S_j}\right)}P\right) \\
&= e(P,P)^{x\left(s+h_1'\left(ID_{S_j}\right)\right)\cdot\frac{1}{s + h_1'\left(ID_{S_j}\right)}} = e(P,P)^x \\
&= g^x = X' \\
U_2P &= \left(\mathscr{D}_i + xw + \kappa_i\right)P \\
&= \mathscr{D}_iP + xwP + \kappa_iP \\
&= \left(r_i + sh_i\right)P + wX + \kappa_iP = R_i + h_iP_{\text{pub}} + wX + \kappa_iP.
\end{aligned}
\tag{5}
$$

5.6. Password Update Phase. If U_i wants to update the password, the following steps are executed between U_i and his/her mobile device:

(1) U_i inputs ID_i, old PW_i, and new PW_i^* into the mobile device.

(2) The mobile device computes $P_i = h_0(ID_i \| PW_i \| n_i)$, $C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i = \tau_i \oplus P_i, \sigma_i' = h_2(C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i \| P_i)$ and checks whether $\sigma_i' = \sigma_i$ holds. If not, the mobile device rejects the request. Otherwise, the mobile device calculates $P_i^* = h_0(ID_i \| PW_i^* \| n_i)$, $\tau_i^* = (C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i) \oplus P_i^*, \sigma_i^* = h_2(C_i \| h\ d \| \mathscr{D}_i \| M \| R_i \| h_i \| P_i^*)$. Then, the mobile device replaces $\{\tau_i, \sigma_i\}$ with $\{\tau_i^*, \sigma_i^*\}$.

5.7. Access Rights Update Phase. U_i must update the access control list and send a new ACL_i^* to the RC after the renewal of subscription. The steps will be carried out between U_i and RC.

(1) U_i generates a new access control list ACL_i^* and sends $\{ID_i, P_i, ACL_i^*\}$ to RC via a secure channel.

(2) After receiving the message, RC obtains the old access control list ACL_i from the user registration table by the index ID_i and updates ACL_i. Then, RC computes the new common solution M' using CRT. After that, RC regenerates $r_i' \in Z_q^*, h_i' = h_1(ID_i \| M \prime \| R_i'), \mathscr{D}_i' = (r_i' + sh_i') \bmod q, \tau_i' = (C_i \| h\ d \| \mathscr{D}_i' \| M' \| R_i' \| h_i') \oplus P_i, \sigma_i' = h_2(C_i \| h\ d \| \mathscr{D}_i' \| M' \| R_i' \| h_i' \| P)_i$ and sends $\{\tau_i', \sigma_i'\}$ to U_i. Finally, RC updates the user registration table $\{ID_i, P_i, ACL_i^*\}$.

6. Security Analysis

We will demonstrate that the proposed scheme is provably secure under the random oracle model and satisfy the security requirements mentioned in Section 4.3.

6.1. Security Model. We define the formal security model for the proposed scheme through a game played between an adversary $\mathscr{A}$ and a challenger $\mathscr{C}$. Let Π_P^i denote the i^{th} instance of the participant P, where $P \in \{U_i, S_j\}$ denotes two participants: the mobile user and the server who are involved in the execution of the proposed scheme. In the game, $\mathscr{A}$ can initiate a series of queries to $\mathscr{C}$, which can answer them as follows:

(i) $h_i(m)$: when $\mathscr{A}$ sends a message m, $\mathscr{C}$ first checks whether m is in the hash-list L_{h_i}. If so, return the value. Otherwise, $\mathscr{C}$ generates a random number $r_i \in Z_q^*$ and stores (m_i, r_i) in the hash-list L_{h_i}. After that, return r_i to $\mathscr{A}$.

(ii) *ExtractUserID* (ID_{U_i}): when $\mathscr{A}$ sends a query with the user's U_i identity ID_{U_i}, $\mathscr{C}$ generates a private key and stores (ID_{U_i}, d_i) in the user-list L_{U_i}.

(iii) *ExtractServerID* (ID_{S_j}): when $\mathscr{A}$ sends a query with the server's S_j identity ID_{S_j}, $\mathscr{C}$ generates a private key and stores (ID_{S_j}, d_j) in the server-list L_{S_j}.

(iv) *Send* (Π_P^i, m_i): when $\mathscr{A}$ sends a query with a message m_i, $\mathscr{C}$ responds with the result by executing the proposed scheme. $\mathscr{A}$ can start the proposed scheme by sending (Π_P^i, START).

(v) *Reveal* (Π_P^i): when $\mathscr{A}$ sends this query, $\mathscr{C}$ returns the session key of the i^{th} instance.

(vi) *Corrupt* (Π_P): when $\mathscr{A}$ sends this query with the participant $P \in \{U_i, S_j\}$ identity, $\mathscr{C}$ returns the corresponding private key.

(vii) *Test* (Π_P^i): $\mathscr{A}$ can send this query only once. Upon receiving the query, $\mathscr{C}$ flips a coin $c \in \{0, 1\}$. If $c = 1$, $\mathscr{C}$ returns the session key of i^{th} instance. Otherwise, $\mathscr{C}$ selects a random number and returns it.

After executing the aforementioned queries, $\mathscr{A}$ output the guess c' in the test query phase. If $c' = c$, we say $\mathscr{A}$ breaks the semantic security of the proposed scheme. The advantage that $\mathscr{A}$ violates the authenticated key agreement (AKA) of this scheme Γ is denoted by $A\ dv_\Gamma^{AKA} = |2Pr[c' = c] - 1|$.

Definition 1. (AKA-security): we say an authentication scheme Γ is AKA-security if $A\ dv_\Gamma^{AKA} = |2Pr[c' = c] - 1|$ is negligible for any polynomial adversary $\mathscr{A}$.

Let E_{U-S} and E_{S-U} denote the events $\mathscr{A}$ that can violate $U - to - S$ authentication by forging a login message and $S - to - U$ authentication by forging a response message, respectively. The advantage that $\mathscr{A}$ violates the mutual authentication of this scheme Γ is denoted by $A\ dv_\Gamma^{MA} = |\Pr[E_{U-S}] + \Pr[E_{S-U}]|$.

Definition 2. (MA-security): we say an authentication scheme Γ is MA-security if $A\ dv_\Gamma^{MA} = \Pr[E_{U-S}] + \Pr[E_{S-U}]$ is negligible for any polynomial adversary $\mathscr{A}$.

6.2. Provable Security. We will prove that our proposed scheme is MA-security and AKA-security in the security model above.

Lemma 1. *No polynomial adversary can forge a legal login message if the DL problem is difficult.*

Proof. suppose the adversary $\mathscr{A}$ can output a legal login message with non-negligible probability ε, then the challenger $\mathscr{C}$ can solve the DL problem with non-negligible advantage.

Given a DL instance $(P, \theta = \tau P)$, the task of $\mathscr{C}$ is to compute τ, where $\tau \in Z_q^*$ is unknown to $\mathscr{C}$. $\mathscr{C}$ generates a random number $\hat{\tau} \in Z_q^*$, sets $P_{\text{pub}} \leftarrow \theta, \hat{P}_{\text{pub}} = \hat{\tau} P$, and sends the system parameters params $= \{G_1, G_2, P, e, q, P_{\text{pub}}, \hat{P}_{\text{pub}}, h_0, h_1, h_1', h_2, h_3, h_4, h_5, h_6\}$ to $\mathscr{A}$. $\mathscr{C}$ will maintain seven hash-lists L_{h_i} $(i = 0, 1, 2, 3, 4, 5, 6)$, a mobile user list L_{U_i}, and a server list L_{S_j}. $\mathscr{C}$ randomly chooses a user's identity $ID_{U_i^*}$ as the challenge identity and answers $\mathscr{A}$'s queries as follows:

(i) $h_i(m)$: $\mathscr{C}$ maintains the lists L_{h_i}, which are initially empty. $\mathscr{C}$, firstly, checks whether (m_i, r_i) exists in L_{h_i}. If it does, $\mathscr{C}$ returns r_i to $\mathscr{A}$. Otherwise, $\mathscr{C}$ generates a random number r_i, stores (m_i, r_i) in L_{h_i}, and returns r_i to $\mathscr{A}$.

(ii) *ExtractUserID* (ID_{U_i}): $\mathscr{C}$ maintains the mobile user list L_{U_i}, which is initially empty. $\mathscr{C}$, firstly, checks whether $(ID_{U_i}, R_i, d_i, \kappa_i)$ exists in L_{U_i}. If it does, $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$. Otherwise, $\mathscr{C}$ executes the following steps:

(a) If $ID_{U_i} \neq ID_{U_i^*}$, $\mathscr{C}$ randomly selects $r_{U_i}, h_{1_{U_i}} \in Z_q^*$, and a random number κ_i as the output of fuzzy extractor. $\mathscr{C}$ computes $R_{U_i} = r_{U_i}P - h_{U_i}P_{pub}$. Otherwise, $\mathscr{C}$ sets $d_i \leftarrow r_{U_i}$ and stores $(ID_{U_i}, R_i, d_i, \kappa_i)$ and $(ID_{U_i}, R_{U_i}, h_{1_{U_i}})$ in L_{U_i} and L_{h_1}, respectively, and returns ID_{U_i} to $\mathscr{A}$.

(b) If $ID_{U_i} = ID_{U_i^*}$, $\mathscr{C}$ generates two random numbers $r_{U_i}^*, h_{1_{U_i}}^* \in Z_q^*$, computes $R_{U_i}^* = r_{U_i}^* P$, sets $d_{U_i}^* \leftarrow \perp$, selects a random number κ^* as the output of fuzzy extractor, stores $(ID_{U_i^*}, R_{U_i}^*, \perp, \kappa^*)$ and $(ID_{U_i^*}, R_{U_i}^*, h_{1_{U_i}}^*)$ in L_{U_i} and L_{h_1}, respectively, and returns $ID_{U_i^*}$ to $\mathscr{A}$.

(iii) *ExtractServerID* (ID_{S_j}): $\mathscr{C}$ maintains the server list L_{S_j}, which is initially empty. $\mathscr{C}$, firstly, checks whether (ID_{S_j}, d_j) is in L_{S_j}. If it does, $\mathscr{C}$ returns ID_{S_j} to $\mathscr{A}$. Otherwise, $\mathscr{C}$ selects $e_j \in Z_q^*$, computes $d_j = 1/\widehat{\tau} + e_j P$, stores (ID_{S_j}, d_j) and (ID_{S_j}, e_j) in L_{S_j} and L_{h_1}, respectively, and then returns ID_{S_j} to $\mathscr{A}$.

(iv) *Send* (Π_P^i, m_i): $\mathscr{A}$ can send the following queries:

(a) Send (U_i^t, START): when $\mathscr{A}$ sends this query, $\mathscr{C}$, firstly, checks if $ID_{U_i} = ID_{U_i^*}$ holds. If it does, $\mathscr{C}$ aborts the game. Otherwise, $\mathscr{C}$ checks whether ID_{U_i} exists in L_{U_i}. If not, $\mathscr{C}$ executes the ExtractUserID (ID_{U_i}) query. After that, with the private key d_i, $\mathscr{C}$ generates a random number $x \in Z_q^*$ and calculates $X' = g^x, X = xP, U_1, U_2, N, \omega$ as described. $\mathscr{C}$ stores $(ID_{U_i}, ID_{S_j}, t, x, X, M, R_{U_iS_j})$ in $L_{U_iS_j}$ and returns (U_1, U_2, N, ω) to $\mathscr{A}$.

(b) Send $(S_j^l, N, U_1, U_2, \omega, T_{us})$: when $\mathscr{A}$ sends this query, $\mathscr{C}$ checks whether ID_{S_j} exists in L_{S_j}. If not, $\mathscr{C}$ executes the ExtractServerID (ID_{S_j}). After that, with the private key d_j, $\mathscr{C}$ computes $X' = e(U_1, d_j)$ and extracts $ID_{U_i} \| M \| h_i \| X = N \oplus h_3(X')$. $\mathscr{C}$ obtains h_i, κ_i from L_{U_i}, verifies if $U_2P = R_i + h_iP_{\text{pub}} + \omega X + \kappa_i P$ holds. If not, $\mathscr{C}$ rejects the message. Otherwise, $\mathscr{C}$ generates a random number $y \in Z_q^*$, computes Y, V_j, T_{su} as described, and returns it to $\mathscr{A}$. If $ID_{U_i} = ID_{U_i^*}$, then $\mathscr{A}$ successfully forges a legal login message.

(c) Send (U_i^t, Y, V_j, T_{su}): upon receiving this message, $\mathscr{C}$ checks if $V_j' = V_j$ holds with $(ID_{U_i}, ID_{S_j}, t, x, X, M, R_{U_iS_j})$ in $L_{U_iS_j}$. If not, $\mathscr{C}$ rejects the message. Otherwise, $\mathscr{C}$ authenticates $\mathscr{A}$.

(v) *Reveal* (Π_P^i): upon receiving this message, $\mathscr{C}$ returns the session key SK if SK is accepted. Otherwise, $\mathscr{C}$ returns '$\perp$' to $\mathscr{A}$.

(vi) *Corrupt* (Π_P): upon receiving the identity ID_{U_i} (or ID_{S_j}), $\mathscr{C}$ looks up L_{U_i} (or L_{S_j}) and returns d_i (or d_j) to $\mathscr{A}$.

(vii) *Test* (Π_P^i): $\mathscr{C}$ generates a random number with the same length of session key and returns it to $\mathscr{A}$.

Suppose $\mathscr{A}$ forges a legal login message with $ID_{U_i} = ID_{U_i^*}$. By applying forking lemma, $\mathscr{A}$ can generate another legal login message U_2 with the same input of the simulation and different hash oracle queries. Then, we can get the following equations:

$$U_2P = R_i^* + h_i^* P_{\text{pub}} + \omega X + \kappa_i^* P, \tag{6}$$

$$U_2'P = R_i^* + h_i^* P_{\text{pub}} + \omega X + \kappa_i^* P. \tag{7}$$

Based on equations (7) and (8), we have the following:

$$\begin{aligned}(U_2 - U_2')P &= \left(R_i^* + h_i^* P_{\text{pub}} + \omega X + \kappa_i^* P\right) \\ &\quad - \left(R_i^* + h_i^* P_{\text{pub}} + \omega X + \kappa_i^* P\right) \\ &= (h_i^* - h_i^*)P_{\text{pub}} \\ &= (h_i^* - h_i^*)\tau P.\end{aligned} \tag{8}$$

By equations (8), $(U_2 - U_2')(h_i^* - h_i^*)^{-1} \bmod q$ is the answer of DL problem. The advantage that $\mathscr{C}$ solves the DL problem is given below. Firstly, some events are defined as follows:

(i) E_1: the simulation does not abort in any Send query.

(ii) E_2: $\mathscr{A}$ successfully forges a legal message.

(iii) E_3: $ID_{U_i} = ID_{U_i^*}$.

Let q_{h_1} and q_s denote the number of h_1 queries and Send queries. Then, we have the following equations:

$$\begin{aligned}\Pr[E_1] &= \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \\ \Pr[E_2 | E_1] &\geqslant \varepsilon \\ \Pr[E_3 | E_2 \wedge E_1] &\geqslant \frac{1}{q_{h_1}}.\end{aligned} \tag{9}$$

By equation (9), the advantage that $\mathscr{C}$ solves the DL problem is as follows:

$$\begin{aligned}\epsilon &= \Pr[E_3 \wedge E_2 \wedge E_1] \\ &= \Pr[E_3 | E_2 \wedge E_1] \cdot \Pr[E_2 || E_1] \cdot \Pr[E_1] \geqslant \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \frac{1}{q_{h_1}} \cdot \varepsilon.\end{aligned} \tag{10}$$

Thus, $\mathscr{C}$ can solve the DL problem with non-negligible probability ϵ by playing the game with $\mathscr{A}$. It contradicts with the difficulty of DL problem. We conclude that no

polynomial adversary can forge a legal login message with non-negligible probability. □

Lemma 2. *No polynomial adversary can forge a legal response message if the k-mBIDH Problem is difficult.*

Proof. suppose the adversary $\mathscr{A}$ can forge a legal response message with non-negligible probability ε, then the challenger $\mathscr{C}$ can solve the k-mBIDH problem with non-negligible advantage. Given a k-mBIDH instance, $P, P_{\text{pub}} = sP, \iota P, \{e_1, e_2, \ldots, e_k\} \in Z_q^*, 1/s + e_1 P, 1/s + e_2 P, \ldots, 1/s + e_k P \in G_1$, the task of $\mathscr{C}$ is to compute $e(P,P)^{\iota/s+e^*}$, where s, ι are unknown to $\mathscr{C}$ and $e^* \notin \{e_1, e_2, \ldots, e_k\}$. $\mathscr{C}$ sends the system parameters params $= \{G_1, G_2, q, e, g, P, P_{pub}, h_0, h_1, h_1', h_2, h_3, h_4, h_5, h_6\}$ to $\mathscr{A}$. $\mathscr{C}$ will maintain seven hash-lists L_{h_i} ($i = 0, 1, 2, 3, 4, 5, 6$), a mobile user list L_{U_i}, and a server list L_{S_j}. $\mathscr{C}$ chooses a random identity $ID_{S_j^*}$ as the challenge identity.

h_i ($i = 0, 1, 2, 3, 4, 6$), Reveal, Corrupt, and Test query are the same as those in the simulation of Lemma 1. h_5, Extract, and Send query are executed as follows:

(i) *ExtractServerID* (ID_{S_j}): $\mathscr{C}$ maintains the server list L_{S_j}, which is initially empty. $\mathscr{C}$, firstly, checks whether (ID_{S_j}, e_j) is in L_{S_j}. If it is, $\mathscr{C}$ returns ID_{S_j} to $\mathscr{A}$. Otherwise, $\mathscr{C}$ executes the following steps:

(a) If $ID_{S_j} \neq ID_{S^*}$, $\mathscr{C}$ selects $e_j \in \{e_1, e_2, \ldots, e_k\}$ and $d_j = (1/s + e_j)P$. $\mathscr{C}$ stores (ID_{S_j}, d_j) and (ID_{S_j}, e_j) in L_{S_j} and $L_{h_1'}$, respectively, and returns ID_{S_j} to $\mathscr{A}$.

(b) If $ID_{S_j} = ID_{S_j^*}$, $\mathscr{C}$ sets $h_1'(ID_{S_j}) = e^*$, stores $(ID_{S_j}, \perp)$ and (ID_{S_j}, e^*) in L_{S_j} and $L_{h_1'}$, respectively, and returns ID_{S_j} to $\mathscr{A}$.

(ii) *ExtractUserID* (ID_{U_i}): $\mathscr{C}$ maintains the mobile user list L_{U_i}, which is initially empty. $\mathscr{C}$, firstly, checks whether $(ID_{U_i}, R_{U_i}, d_i, \kappa_i)$ exists in L_{U_i}. If it does, $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$. Otherwise, $\mathscr{C}$ randomly selects $r_{U_i}, h_{1_{U_i}} \in Z_q^*$ and computes $R_{U_i} = r_{U_i} P - h_{1_{U_i}} P_{\text{pub}}$. $\mathscr{C}$ sets $d_i \leftarrow r_{U_i}$ and generates a random number κ_i as the output of fuzzy extractor. $\mathscr{C}$ stores $(ID_{U_i}, R_{U_i}, d_i, \kappa_i)$ and $(ID_{U_i}, R_{U_i}, h_{1_{U_i}})$ in L_{U_i} and L_{h_1}, respectively, and returns ID_{U_i} to $\mathscr{A}$.

(iii) *Send* (Π_P^i, m_i): $\mathscr{A}$ can send the following queries:

(a) Send (U_i^t, START): If U_i's partner is S_j^*, $\mathscr{C}$ sets $U_1 = \iota P$ and calculates U_2, N, ω as described. Otherwise, $\mathscr{C}$ looks up $(ID_{U_i}, R_{U_i}, h_{1_{U_i}})$ from L_{h_1} and generates a legal login message as described.

(b) Send $(S_j^l, N, U_1, U_2, \omega, T_{us})$: When $\mathscr{A}$ sends this query, $\mathscr{C}$ checks whether $ID_{S_j} = ID_{S_j^*}$ holds. If it does, $\mathscr{C}$ aborts the game. Otherwise, $\mathscr{C}$ behaves the operations as described.

(c) Send (U_i^t, Y, V_j, T_{su}): Upon receiving this message, $\mathscr{C}$ checks if $V' = V_j$ holds.

If not, $\mathscr{C}$ rejects the message. Otherwise, $\mathscr{C}$ authenticates $\mathscr{A}$. If $ID_{S_j} = ID_{S_j^*}$, it means that $\mathscr{A}$ successfully forges a legal response message.

Suppose $\mathscr{A}$ forges a legal response message with $ID_{S_j} = ID_{S_j^*}$. In the game, $\mathscr{A}$ must send $V_j = h_5(ID_i, ID_{S_j}, M, R_{U_i S_j}, X, Y, K, T_{su})$ query after recovering $ID_{U_i} \| M \| h_{1_{U_i}} \| X = N \oplus h_3(X')$. Thus, $\mathscr{A}$ must have executed the h_3 query with the message X'. We have the following:

$$\begin{aligned} X' &= e\left(U_1, d_j^*\right) \\ &= e\left(\iota P, \frac{1}{s + e^*} P\right) \\ &= e(P,P)^{\iota/s+e^*} \end{aligned} \tag{11}$$

The advantage that $\mathscr{C}$ solves the k-mBIDH Problem is given below. Some events are defined as follows:

(i) E_1: the simulation does not abort.

(ii) E_2: $\mathscr{A}$ successfully forges a legal response message.

(iii) E_3: $ID_{S_j} = ID_{S_j^*}$.

(iv) E_4: $\mathscr{C}$ selects a correct tuple from L_{h_3}.

Let q_{h_1}, q_{h_3}, and q_s denote the number of h_1, h_3 query and Send query. Then, we have the following equations:

$$\begin{aligned} &\Pr[E_1] = \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \\ &\Pr[E_2 | E_1] \geqslant \varepsilon \\ &\Pr[E_3 | E_2 \wedge E_1] \geqslant \frac{1}{q_{h_1}} \\ &\Pr[E_4 | E_3 \wedge E_2 \wedge E_1] \geqslant \frac{1}{q_{h_3}}. \end{aligned} \tag{12}$$

By equation (12), the advantage that $\mathscr{C}$ solves the k-mBIDH Problem is as follows:

$$\begin{aligned} \epsilon &= \Pr[E_4 \wedge E_3 \wedge E_2 \wedge E_1] \\ &= \Pr[E_4 | E_3 \wedge E_2 \wedge E_1] \cdot \Pr[E_3 | E_2 \wedge E_1] \cdot \Pr[E_2 | E_1] \cdot \Pr[E_1] \\ &\geqslant \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \frac{1}{q_{h_1}} \frac{1}{q_{h_3}} \cdot \varepsilon. \end{aligned} \tag{13}$$

Thus, $\mathscr{C}$ can solve the k-mBIDH problem with non-negligible probability ϵ by playing the game with $\mathscr{A}$. It contradicts with the hardness of k-mBIDH problem. We conclude that no polynomial adversary can forge a legal response message with non-negligible probability. □

Theorem 1. *The proposed scheme is MA-security if the DL problem and k-mBIDH problem are hard.*

Proof. Lemma 1 and 2 demonstrate that no polynomial adversary can forge a legal login message or response message if the DL problem and k-mBIDH problem are hard. In other words, the mobile user U_i and the server S_j can

authenticate each other. Therefore, the proposed scheme is MA-security. □

Theorem 2. *The proposed scheme is AKA-security if CDH problem is hard.*

Proof. suppose $\mathscr{A}$ can guess b correctly with non-negligible probability ε in the Test query, then $\mathscr{C}$ can solve the CDH problem with non-negligible probability. Some events are defined as follows:

(i) E_b: $\mathscr{A}$ guesses the value of b correctly.

(ii) E_{TU}: $\mathscr{A}$ makes the Test query to the user.

(iii) E_{TS}: $\mathscr{A}$ makes the Test query to the server.

(iv) E_{US}: $\mathscr{A}$ forges a legal login message between the user and server.

Since the probability that $\mathscr{A}$ can guess the value of b correctly is at least 1/2, then we have $\Pr[E_b] \geqslant \varepsilon/2$. We can get the equations as follows:

$$\begin{aligned}\frac{\varepsilon}{2} \leqslant \Pr[E_b] &= \Pr[E_b \wedge E_{TU}] \\ &+ \Pr[E_b \wedge E_{TS} \wedge E_{US}] \\ &+ \Pr[E_b \wedge E_{TS} \wedge E_{US}] \\ &\leqslant \Pr[E_b \wedge E_{TU}] + \Pr[E_{US}] \\ &+ \Pr[E_b \wedge E_{TS} \wedge E_{US}].\end{aligned} \tag{14}$$

Then,

$$\Pr[E_b \wedge E_{TU}] + \Pr[E_b \wedge E_{TS} \wedge E_{US}] \geqslant \frac{\varepsilon}{2} - \Pr[E_{US}]. \tag{15}$$

The event $E_{TS} \wedge E_{US}$ and E_{TU} are equal. We can get the following:

$$2Pr[E_b \wedge E_{TU}] \geqslant \frac{\varepsilon}{2} - \Pr[E_{US}]. \tag{16}$$

Then, the advantage that $\mathscr{C}$ solves the CDH problem is as follows:

$$\epsilon = \Pr[E_b \wedge E_{TU}] \geqslant \frac{\varepsilon}{4} - \frac{\Pr[E_{US}]}{2}. \tag{17}$$

The event $E_b \wedge E_{TU}$ means that $\mathscr{A}$ impersonates the user and has $K = xY$, which is the solution of CDH problem. According to Lemma 1, $\Pr[E_{US}]$ is negligible. Then, we can get $\Pr[E_b \wedge E_{TU}]$, which is non-negligible because ε is non-negligible. It means that $\mathscr{C}$ can solve the CDH problem with non-negligible probability ϵ by playing the game with $\mathscr{A}$. It contradicts the difficulty of CDH problem. We conclude that no polynomial adversary can guess b correctly, and the proposed scheme is AKA-security. □

6.3. Security Requirement Analysis. We also show that the proposed scheme satisfies the security requirements described in Section 4.3.

Single registration: According to the description of the proposed scheme, the trusted RC generates the registration information for users. Then, users can be authenticated by other healthcare service providers.

No online registration center: According to the specification of the proposed scheme, RC is not involved in the mutual authentication.

Mutual authentication: According to the proof of Lemma 1 and Lemma 2, we know that no polynomial adversary can forge a legal login message and response message. Thus, the user U_i and the server S_j can authenticate each other by the received message.

User anonymity: Based on the description of the proposed scheme, the user identity ID_{U_i} is hidden in the login message $\{N, U_1, U_2, \omega, T_{us}\}$, where $N = h_3(X') \oplus (ID_{U_i} \| M \| h_i \| X)$. An adversary can extract ID_{U_i} only if he/she can get $X' = e(U_1, \mathscr{D}_{S_j})$ after solving the k-mBIDH problem. Besides, the registration recorded in the blockchain is hidden by the hash function $h_i = h_1(ID_i \| M \| R_i)$.

Untraceability: In the proposed scheme, random x is generated in each new session to compute a new login message $\{N, U_1, U_2, \omega, T_{us}\}$, where $X' = g^x, X = xP$, $U_1 = x(PK + h_1(ID_{S_j})P), N = h_3(X') \oplus (ID_{U_i} \| M \| h_i \| X), \omega = h_4(N, U_1, \kappa_i, T_{us}), U_2 = \mathscr{D}_{U_i} + x\omega + \kappa_i$. Because of the randomness of x, the adversary cannot find any relation among these login messages of different sessions and thus cannot trace the users' behavior.

Session key agreement: Based on the description of the proposed scheme, the user and the server will generate a session key $SK = h_6(X' \| X \| Y \| K_{ij} \| ID_i \| ID_{S_j})$ for future secure communications. We know that no adversary can compute xyP from xP and yP since the CDH problem is hard.

Perfect forward secrecy: Assume that there is an adversary who gets the long-term private key of the user and the server and intercepts the exchange message of previous sessions. Then, the adversary cannot compute $K_{ij} = xyP$ in $SK = h_6(X' \| X \| Y \| K_{ij} \| ID_i \| ID_{S_j})$ even if he gets X and Y from U_1, N, Y, V_j since the CDH problem is hard. Therefore, the adversary cannot generate previous session keys even if he knows both private keys of the user and server.

Two-factor security with PUF: Assume that an adversary steals the mobile device and extracts the data $n_i, P_i, \tau_i, \sigma_i$ by a side channel attack. He can guess the password PW_i, however, he cannot verify the correctness of the password without knowing the identity ID_{U_i}. He cannot get the response output of PUF without C_i.

Moreover, the adversary cannot be authenticated in his mobile device because of the uncloneability of PUF even if he knows the password. Thus, our proposed scheme can satisfy the two-factor security.

Access control for data privacy: The proposed scheme provides access control in the authentication process. The common solution M can be used to determine the access permission NumR_i. The server can determine the access permission of the user efficiently without any other access control list to make data disclosure minimum. Besides, the common solution is associated with the RC's private key s, which means no adversary can forge the access permission message.

Resistance of various attacks: To resist known attacks existing in the service system, the proposed scheme should resist known attacks.

(i) *Insider attack*: An insider in the system only knows user identity ID_i but cannot verify the password or private key without n_i, P_i. Thus, the proposed scheme can resist an insider attack.

(ii) *Stolen card attack*: An adversary gets the user's smart card and extracts the data $n_i, P_i, \tau_i, \sigma_i$. However, he cannot verify the password without the user's identity. Therefore, the proposed scheme can withstand the stolen card attack.

(iii) *Offline password guessing attack*: An adversary gets the user's smart card and extracts the data $n_i, P_i, \tau_i, \sigma_i$. However, he cannot verify the password without the user's identity. Therefore, the proposed scheme can resist the offline password guessing attack.

(iv) *Replay attack*: According to the description of the proposed scheme, we use the timestamp to check the freshness. Besides, the user and the server will generate new random numbers $x, y \in Z_q^*$ in each session. Both of them can find the replay of a message by the validity of the received message.

(v) *User impersonation attack*: Based on the proof of Lemma 1, we show that no adversary can forge a legal login message without the user's private key. The server can check the validity of the login message.

(vi) *Server impersonation attack*: Based on the proof of Lemma 2, we show that no adversary can forge a legal response message without server's private key. The user can check the validity of the response message V_j since the server will extract X', X using its private key.

(vii) *Man-in-the-middle attack*: According to Theorem 1, we conclude that our proposed scheme provides mutual authentication, which means no adversary can forge a legal message without knowing the private keys.

(viii) *Physical attack*: No adversary can recreate the same PUF to authenticate the device since PUFs are almost impossible to clone. Besides, the response R and helper data hd are kept a secret, and only (C_i, g^κ) is stored in the blockchain, which can resist the modeling attack.

6.4. Security Comparisons. We now compare our proposed scheme with other prior related schemes, namely those of Xiong et al. [34], Jia et al. [21], and Son et al. [6]. As shown in Table 4, [6, 21, 34] cannot provide two-factor security, in which PUF serves as one of the authentication factors. [21] cannot support access control in the authentication process. All of them cannot resist physical attacks and others. Therefore, our scheme can satisfy all the listed security requirements and have better security.

7. Performance Analysis

In this section, we discuss the computational and communication costs of the proposed scheme and other related schemes [6, 21, 34, 37].

We will directly use the parameters of Jia et al. [21] scheme for the comparison in Table 5. We do not consider the execution time of registration phases, since they are executed only once. We also focus on the communication costs in the registration phase. The size of the element in G_1, G_2, and Z_q is 1024, 1024, and 160 bits, respectively, denoted by $|G_1|, |G_2|, |Z_q|$. Suppose the output length of the hash function $|H|$ and user's identity $|I\ D|$ is 256 bits. The length of the timestamp $|T|$ and common solution is 32 bits.

The summary of computation and communication costs is shown in Tables 6, 7 and Figures 6, 7. The proposed scheme has a lower computation cost compared to that in the scheme [6, 34] on the mobile users' side and [6, 21] on the server side. Besides, the proposed protocol has lower communication costs compared to those in [21]. Though the scheme [6, 34] has lower communication costs than ours, our scheme can be more efficient and meet more desirable security requirements.

We implement a software prototype based on Hyperledger Fabric 2.0 on a single machine running an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz, 8 GB RAM, and Ubuntu 18.04 for 64 bits operations system. We deploy a test network that consists of three peer organizations and a single orderer organization with three ordering nodes in a single channel to simulate the healthcare provider servicers in certain areas. The block size is 100 transactions in a batch and the block timeout is 2s to wait before creating a batch. The raft consensus algorithm is used for agreement on the consistence of transactions across the network. The smart contracts written in Go are deployed over the network, which is shown in Algorithm 1 and 2. All transactions of uploading, query, and update are driven via Hyperledger Fabric gateway to evaluate the throughput (TPS) and latency as shown in Figures 8 and 9. The performance demonstrates that the maximum throughput is 310 TPS for uploading the transactions and 600 TPS for query and update. The average latency for all transactions increases as increased in the total transactions. There are lots of factors that will affect the performance of the framework, such as the choices of ledger database, endorsement policy, network configuration parameters, and so on.

```
function uploadSubs(h_i, ID_{S_j}, R_{U_iS_j}) {
  ID_ij = h_i || ID_{S_j}
  if userExists (ID_ij) == false
    expiretime = time.Now().Month() + R_{U_iS_j}
    service = Service \{ID_ij, R_{U_iS_j}, expiretime\}
    return putState(ID_ij, service)
  else
    return Errorf("the user has already exists")
function querySubs(h_i, ID_{S_j}) {
  ID_ij = h_i || ID_{S_j}
  err, result = getState (ID_ij)
  if (err == null)
    return result
  else
    return err
function updateSubs(h_i, R_{U_iS_j}) {
  ID_ij = h_i || ID_{S_j}
  if userExists (ID_ij) == true
    expiretime = time.Now().Month() + R_{U_iS_j}
    service = Service \{ID_ij, R_{U_iS_j}, expiretime\}
    return putState(ID_ij, R_{U_iS_j})
  else
    return Errorf("the user does not exist")
```

Algorithm 1: Subscription service contract.

```
function uploadCRP(h_i, C_i, κ_iP, R_i, t_s) {
  if userExists (h_i) == false
    status = 'valid'
    t_s = time.Now()
    user = User \{h_i, C_i, κ_iP, R_i, t_s, status\}
    return putState(h_i, user)
  else
    return Errorf("the user has already exists")
function queryCRP (h_i) {
  err, result = getState (h_i)
  if (err == null && result.getStatus() == 'valid')
    return result
  else if (result.getStatus() != "valid")
    return Errorf("the CRP of user has been expired")
  return err
function updateCRP(h_i, C_i, κ_iP, R_i) {
  if userExists (h_i) == true
    status = 'valid'
    t_s = time.Now()
    user = User \{h_i, C_i, κ_iP, R_i, t_s, status\}
    return putState(h_i, user)
  else
    return Errorf("the user does not exist")
```

Algorithm 2: Challenge-response pair contract.

Table 4: Security comparison.

Security requirements	[34]	[21]	[6]	Ours
Single registration	✓	✓	✓	✓
No online registration center	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓
User anonymity	✓	✓	✓	✓
Untraceability	✓	✓	✓	✓
Session key agreement	✓	✓	✓	✓
Perfect forward secrecy	✓	✓	✓	✓
Two-factor security with PUF	×	×	×	✓
Access control	✓	×	✓	✓
Resistance of known attacks	×	×	×	✓

Table 5: Running time of basic operation.

	Description	Alibaba cloud	Google nexus
T_{G_b}	Bilinear pairing	5.275	48.66
T_{G_m}	Scalar multiplication	1.97	19.919
T_a	Point multiplication	0.012	0.118
T_h	Hash function	0.009	0.089
T_e	Modular exponentiation	0.339	3.328

Table 6: Computation costs comparison (ms).

Scheme	Computation costs(User)	Computation costs(Server)
[34]	$2\,T_{G_m} + T_{G_a} + T_e + T_{G_b} + 8\,T_{G_h} \approx 92.656$	$2\,T_{G_m} + T_{G_a} + T_{G_b} + 5\,T_h \approx 9.272$
[21]	$4\,T_{G_m} + T_e + 5\,T_h \approx 83.449$	$T_b + 5\,T_{G_m} + 3\,T_{G_a} + 5\,T_h \approx 15.206$
[6]	$5\,T_{G_m} + 8\,T_h \approx 100.307$	$2\,T_{G_b} + 5\,T_{G_m} + 5\,T_h \approx 20.445$
Ours	$4\,T_{G_m} + T_e + 7\,T_h + T_a \approx 83.745$	$T_{G_h} + 4\,T_{G_m} + 3\,T_h + 3\,T_a \approx 13.218$

Table 7: Communication costs comparison.

Scheme	Communication costs	Length/bits
[34]	$2\,\|G\| + 3\,\|H\|$	2816
[21]	$4\,\|G\| + 2\,\|Z_q\| + 2\,\|H\| + \|I\ D\|$	4736
[6]	$3\,\|G\| + 2\,\|H\| + 2\,\|T\|$	3648
Ours	$3\,\|G\| + \|Z_q\| + 3\,\|H\| + \|I\ D\| + 2\,\|T\|$	4320

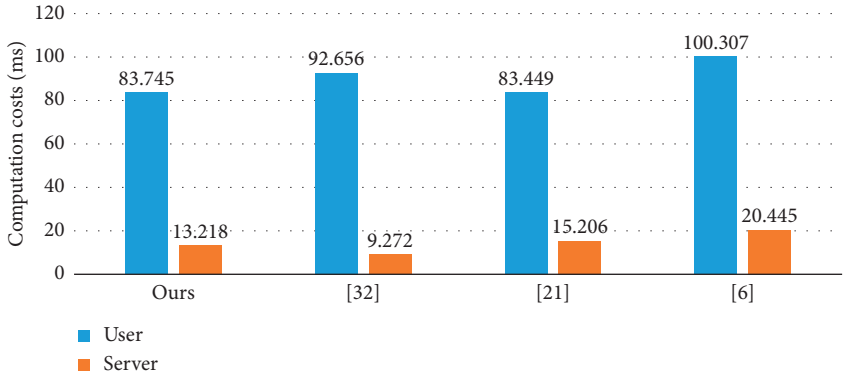

Figure 6: Computation costs comparison.

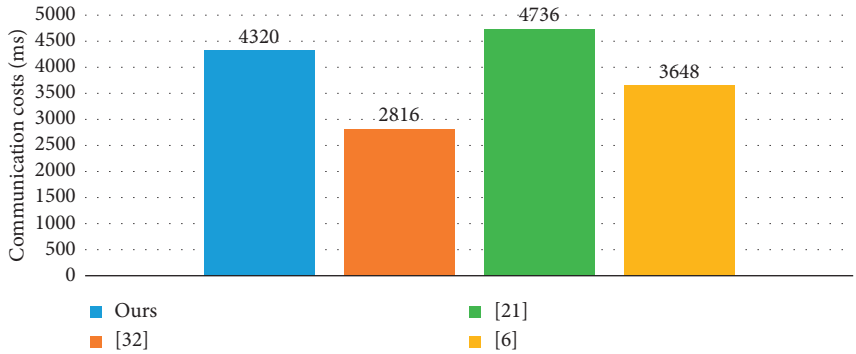

Figure 7: Communication costs comparison.

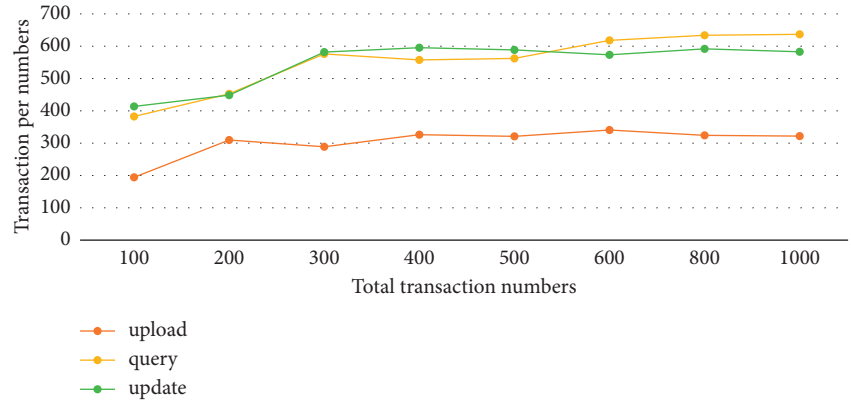

Figure 8: Transactions per second in the blockchain.

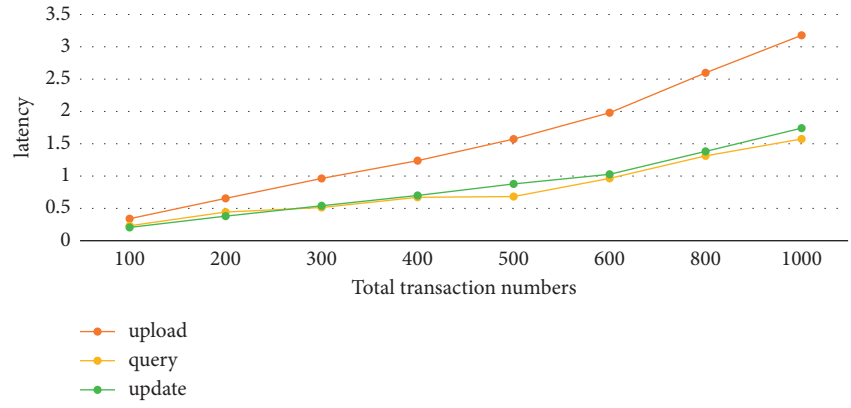

Figure 9: Latency for all transactions in the blockchain.

8. Conclusion

In this paper, we proposed a user authentication scheme with access control based on CRT for multiserver architectures, which not only secures sensitive health data transmission but enhances data privacy. The highlight of our protocol is that the decent integration of blockchain and PUF can achieve secure data sharing across medical institutions and identify each device for further security in the telehealth environment. Security analysis shows that our proposed scheme meets all of the desirable requirements above. Moreover, the performance analysis demonstrated that this protocol has lower communication and computation costs.

Future research will focus on more flexible access control updates and the improvement of computational and communication efficiency for resource-limited devices. We will also explore a privacy-preserving data sharing mechanism based on the blockchain and the improvement of blockchain performance. In addition, there is no doubt that a centralized registration center is convenient and available but our protocol is susceptible to some common security flaws, such as single point of failure, distributed denial of service (DDoS) attacks, and register center compromised attacks.

Decentralized blockchain technology to mitigate central registration center problems is also a significant direction.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The work was supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the Shandong Provincial Key Research and Development Program (Nos. 2021CXGC010107and 2020CXGC010107), the National Natural Science Foundation of China (Nos. 62172307, U21A20466, 61972294, and 61932016), the Blockchain Core Technology Strategic Research Program of Ministry of Education of China (No. 2020KJ010301), the Special Project on Science and Technology Program of Hubei Province (No. 2020AEA013), the Natural Science Foundation of Hubei Province (No. 2020CFA052), the Wuhan Municipal Science and Technology Project (No. 2020010601012187), and the Guangxi Key Laboratory of Trusted Software (No. kx202001).

References

[1] Q. Jiang, J. Ma, G. Li, and L. Yang, "An efficient ticket based authentication protocol with unlinkability for wireless access networks," *Wireless Personal Communications*, vol. 77, no. 2, pp. 1489–1506, 2014.

[2] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2021.

[3] L. H. Li-Hua Li, L.-C. Luon-Chang Lin, and M.-S. Min-Shiang Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[4] P. Mohit, R. Amin, A. Karati, G. P. Biswas, and M. K. Khan, "A standard mutual authentication protocol for cloud computing based health care system," *Journal of Medical Systems*, vol. 41, no. 4, p. 50, 2017.

[5] V. Kumar, S. Jangirala, and M. Ahmad, "An efficient mutual authentication framework for healthcare system in cloud computing," *Journal of Medical Systems*, vol. 42, no. 8, pp. 1–25, 2018.

[6] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, "Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain," *IEEE Access*, vol. 8, Article ID 192177, 2020.

[7] M. A. Khan and K. Salah, "Iot security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[8] M. Andoni, V. Robu, D. Flynn et al., "Blockchain technology in the energy sector: a systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[9] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: a survey," *Computers & Security*, vol. 97, Article ID 101966, 2020.

[10] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[11] H. Debiao, C. Jianhua, and Z. Rui, "A more secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1989–1995, 2012.

[12] Z.-Y. Wu, Y.-C. Lee, F. Lai, H.-C. Lee, and Y. Chung, "A secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1529–1535, 2012.

[13] S. Kumari, M. K. Khan, and R. Kumar, "Cryptanalysis and improvement of 'a privacy enhanced scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 37, no. 4, pp. 1–11, 2013.

[14] C. L. Chen, T. T. Yang, and T. F. Shih, "A secure medical data exchange protocol based on cloud environment," *Journal of Medical Systems*, vol. 38, no. 9, pp. 112–12, 2014.

[15] S.-Y. Chiou, Z. Ying, and J. Liu, "Improvement of a privacy authentication scheme based on cloud for medical environment," *Journal of Medical Systems*, vol. 40, no. 4, p. 101, 2016.

[16] I.-C. Lin, M.-S. Hwang, and L.-H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[17] W.-S. Wen-Shenq Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.

[18] J.-L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, vol. 27, no. 3-4, pp. 115–121, 2008.

[19] W.-J. Tsaur, J.-H. Li, and W.-B. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *Journal of Systems and Software*, vol. 85, no. 4, pp. 876–882, 2012.

[20] J.-L. Tsai and N.-W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE systems journal*, vol. 9, no. 3, pp. 805–815, 2015.

[21] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2019.

[22] X. Liu, W. Ma, and H. Cao, "Mbpa: a medibchain-based privacy-preserving mutual authentication in tmis for mobile medical cloud architecture," *IEEE Access*, vol. 7, Article ID 149282, 2019.

[23] A. Yazdinejad, G. Srivastava, R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, and M. Aledhari, "Decentralized authentication of distributed patients in hospital networks using blockchain," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2146–2156, 2020.

[24] C.-T. Li, D.-H. Shih, C.-C. Wang, C.-L. Chen, and C.-C. Lee, "A blockchain based data aggregation and group authentication scheme for electronic medical system," *IEEE Access*, vol. 8, Article ID 173904, 2020.

[25] X. Cheng, F. Chen, D. Xie, H. Sun, and C. Huang, "Design of a secure medical data sharing scheme based on blockchain," *Journal of Medical Systems*, vol. 44, no. 2, pp. 1–11, 2020.

[26] C. Lin, D. He, X. Huang, M. Khurram Khan, and K.-K. R. Choo, "A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems," *IEEE Access*, vol. 6, Article ID 28203, 2018.

[27] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for iiot devices," *IEEE Transactions on Industrial Informatics*, 2021.

[28] H. Xiong, C. Jin, M. Alazab et al., "On the design of blockchain-based ecdsa with fault-tolerant batch verication protocol for blockchain-enabled iomt," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[29] L. Harn and H.-Y. Lin, "Integration of user authentication and access control," *IEE Proceedings - Computers and Digital Techniques*, vol. 139, no. 2, pp. 139–143, 1992.

[30] N.-Y. Nam-Yih Lee, "Integrating access control with user authentication using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 943–948, 2000.

[31] Y.-C. Chen and L.-Y. Yeh, "An efficient authentication and access control scheme using smart cards," in *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, vol. 2, pp. 78–82, IEEE, 2005.

[32] C. Yang, Z. Jiang, and J. Yang, "Novel access control scheme with user authentication using smart cards," in *2010 Third International Joint Conference on Computational Science and Optimization*, vol. 2, pp. 387–389, IEEE, 2010.

[33] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "Bsein: a blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 2018.

[34] L. Xiong, F. Li, M. He, Z. Liu, and T. Peng, "An efficient privacy-aware authentication scheme with hierarchical access control for mobile cloud computing services," *IEEE Transactions on Cloud Computing*, 2020.

[35] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[36] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

[37] H. Boyapally, P. Mathew, S. Patranabis et al., "Safe is the new smart: puf-based authentication for load modification-resistant smart meters," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, 2020.

[38] M. Fakroon, F. Gebali, and M. Mamun, "Multifactor authentication scheme using physically unclonable functions," *Internet of Things*, vol. 13, Article ID 100343, 2021.

[39] C. C. Chang and J.-Y. Kuo, "An efficient multi-server password authenticated key agreement scheme using smart cards with access control," in *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, vol. 2, pp. 257–260, IEEE, 2005.

[40] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.