Hindawi
Mobile Information Systems
Volume 2020, Article ID 3984048, 12 pages
https://doi.org/10.1155/2020/3984048

Research Article

Traceable Attribute-Based Secure Data Sharing with Hidden Policies in Mobile Health Networks

Xueyan Liu, Yukun Luo, and Xiaotao Yang

College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, China

Correspondence should be addressed to Xueyan Liu; liuxy@nwnu.edu.cn

Received 8 February 2020; Revised 26 June 2020; Accepted 15 July 2020; Published 3 August 2020

Academic Editor: Juraj Machaj

The growing need to store, share, and manage medical and health records has resulted in electronic medical health sharing system (mHealth), which provides intelligent medical treatment for people. Attribute-based encryption (ABE) is regarded as a new cryptology to enhance fine-grained access control over encrypted sharing data in mHealth. However, some existing attribute-based mHealth systems not only violate the one-to-many application characteristics of attribute-based encryption mechanism but also destroy the anonymity of user. In this study, an efficient scheme is proposed to tackle the above defaults and offer two-way anonymity of data owner and data user by introducing a pseudoidentity. The computation of hidden access policy is reduced by removing the bilinear pairing, whereas the interaction between cloud storage and data user is avoided to save bandwidth during trapdoor generation. We also consider the temporal factor of the uploaded information by introducing access validity. Security and performance analyses show that the proposed scheme is efficient without reducing security.

1. Introduction

Given the rapid progress of cloud computing and mobile communication technology with ubiquitous mobile intelligent devices, the electronic medical health sharing system (mHealth) has been developed, which can provide intelligent healthcare services without temporal and spatial restrictions; specifically, mHealth allows patients to record body indicators and upload records, physicians to diagnose patients' illness remotely, and researchers to explore medical records [1]. The application of mHealth reshapes healthcare services model [2]. Figure 1 shows a typical architecture of mHealth sharing system, wherein implanted and wearable sensor devices collect various physiological indicators of patients and then deliver the gathered information to a personal server, such as mobile device. Patients may upload these data to a cloud server (CS) to save a personal storage space and allow doctors, family, other patients, and researchers to access such information. CS provides storage and retrieval services, wherein all kinds of users can apply for access to cloud data according to their own requirements. These services are also fast and efficient.

Although mHealth provides convenience in people's lives, promotes better quality of life, and exhibits good application prospect, it also raises a series of security issues [3]. After a patient uploads his/her electronic health records (EHRs) to cloud using personal service provider, other users may access such data in the cloud through various devices, laptops, personal computer, and mobile phones. EHRs contain physiological data (heartbeat, blood pressure, medications, and dosages) and sensitive information of patients (patient name, medical history, ID number, and phone number) and hospitals (hospital name and attending doctor). If EHRs are directly uploaded to the cloud for sharing, then the information of patients and hospitals will inevitably be leaked to the cloud server and various users, which may cause hidden danger to patients' health, threaten users' life and health, and affect hospitals. One of the solutions to these security issues is to encrypt EHRs before uploading them [4]. However, new problems may arise as follows: Firstly, who and how to obtain access? Secondly, patients and users operate EHRs through mobile devices, but the storage capacity, computing power, and overall capability of mobile devices are limited. Thirdly, patients do not

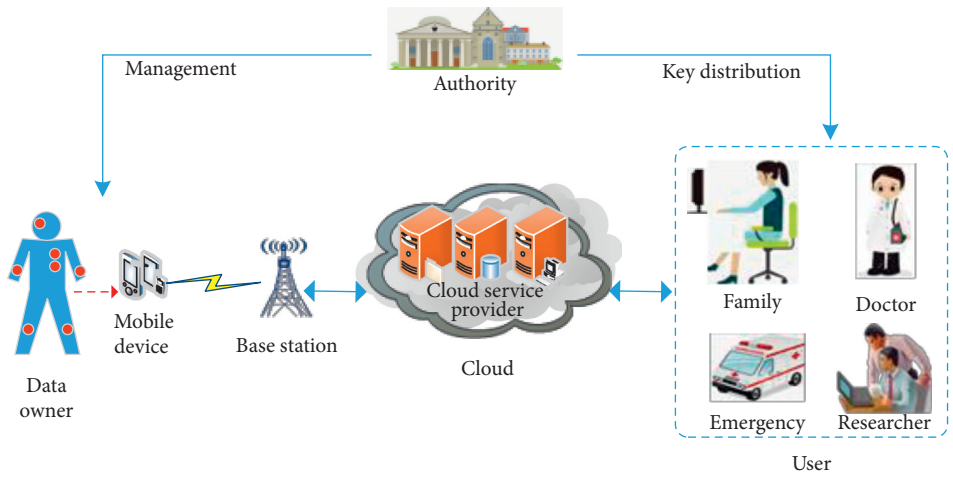

Figure 1: The architecture of the mHealth sharing system.

want others to know their real identity, and users do not want to reveal what EHRs they have accessed.

In sum, the mHealth system needs to solve the abovementioned problems through the following steps:

(1) Ensure data confidentiality: ensure CS and illegal users cannot obtain any information about EHRs.

(2) Proper access control: a patient needs to grant access permissions for different potential users to achieve flexible access control on EHRs in the mHealth system by encrypting once. In this way, unauthorised users cannot access shared EHRs.

(3) Lightweight cryptography: given the limitations of intelligent and mobile devices, an algorithm with little computation and communication costs should be provided.

(4) Two-way anonymity: ensure anonymity for both patients and data users.

To reduce local storage load and achieve resource sharing, increasing numbers of personal and medical institutions upload EHRs to CS. However, CS is not completely trusted and patients do not want to public their EHRs. Hence, EHRs must be encrypted by patients before uploading to CS, which can avoid information leakage. For example, encrypting EHRs of infected patients and then uploading them to the cloud can protect the privacy of patients and hospitals. However, encrypted EHRs can no longer be provided to other data users. Hence, user access authorisation has become a research focus. In general, public key cryptography (such identity-based encryption and certificateless encryption) solves the authorisation problem by sending the key to potential users in advance [5, 6]. However, predicting the exact identity of users is impossible, whereas data owners cannot provide authorisation service each time data users send authorisation requests. Hence, traditional public key cryptography is not suitable for online healthcare system. Attribute-based encryption (ABE) is the most attractive and popular in one-to-many application. Data owners usually use ABE technique to solve the problem of multiple users' flexible authorisation without the need to know the identity of potential users' in advance. In ABE, only the users whose attribute set can satisfy the access policy can obtain access to ensure the anonymity of data users [7–9].

However, in mHealth system, EHRs are important and sensitive, but in which the access policy includes sensitive information. For example, a patient with heart disease uploads his own EHR encrypted by the defined access policy {cardiovascular department∧{chief doctor∨nurse}}, which will easily categorise such EHR as heart disease. Therefore, the access policy should also be hidden. Existing research on hidden access policy provided answers to maintain the confidentiality of access policy [10, 11].

In the general scheme of attribute cryptography, the length of ciphertext and the computation of encryption and decryption are related to the number of attributes, which increases linearly and hence limits the application of this technique. Therefore, the use of fixed or small length ciphertext is a popular solution [12, 13], whereas outsourcing decryption is a good alternative [14].

This solution can help patients and data owners who do not want to disclose their identity preserve their anonymity whilst sharing their own EHRs. Specifically, an infected patient who wants to upload personal encrypted records to provide information to scientific research, but due to some social factors, he/she does not want to let other users know his/her real identity [15]. In this case, the anonymity is of great significance.

In addition to the abovementioned problems, time is also an important factor to be considered in the system. The delay of medical data transmission and access may cause serious consequences, including patient casualties.

1.1. Related Work. Sahai and Waters presented attribute-based encryption firstly [16]. Compared with traditional public key encryption, ciphertext can only be decrypted by one user, while the ABE ciphertext can be provided to multiple users. The encryptor encrypts a message based on an access policy, and only the user whose attributes set satisfies the requirement of encryptor can obtain the message. This mechanism establishes the one-to-many relationship between data owner and data users and enables the

fine-grained access control. Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) are the two categories of ABE technique, whose division is proposed in literature [17].

ABE is an excellent approach to ensure the secure access control of encrypted data and is widely used in many fields, such as cloud computing [18] and searchable encryption [19]. Most of the existing researches focus on the expressiveness of access policies [20]. However, in most ABE schemes, the policy is uploaded with the related ciphertext, which is public to CS and all users including illegal users [21–24]. Hence, any user who obtains the ciphertext can know what the content is about, which will disclose sensitive information about the shared data. Meanwhile, the access policy must be properly handled before sharing with ciphertex, that is, hidden. A series of research results on hidden access policy have been published [25–28]. Frikken et al. presented a protocol that protected both sensitive credentials and sensitive policies [25]. Lai et al. proposed another construction of CP-ABE scheme, which is a partially hidden access policy [26]. In response to the question of confidentiality, Hahn et al. proposed an attribute-based secure data sharing with hidden policies, which can be used in resource constrained environment [27]. These results provide better confidentiality of shared data and the anonymity of data user.

In general ABE schemes, the length of ciphertext, computation of encryption, and decryption are related with the number of attributes of data user, which restricts the use of this technique. To resolve this defect, there are two main solutions. One is to reduce the length of ciphertext as much as possible or adopt fixed length ciphertext. To address this problem, Emura et al. introduced the concept of constant ciphetext length in 2009 [29]. After that, many similar schemes were proposed for constant ciphertext length [23, 30]. The other is to introduce outsourcing decryption to reduce the computation load [31–33]. Li et al. gave a solution to implement attribute-based access control system by introducing secure outsourcing techniques into ABE [31]. In order to decrease computation, Zuo et al. proposed the CCA security model for ABE with outsourced decryption and then presented a concrete CCA-secure ABE scheme with outsourced decryption [32]. In these schemes, data users only perform a small amount of computation by outsourcing a large amount of computing to the cloud service provider.

Some studies with other characteristics have also been proposed, such as decentralized multiauthority scheme [34, 35], traceable scheme [36], and leakage-resilience scheme [37] and reduce online computation load scheme [38]. These studies provide applications in different focus areas.

1.2. Contribution. Given the continuous development of modern mobile communication and sensor technology, mHealth becomes a hot topic in the academe and healthcare industry. In view of the problems existing in the current mHealth system and the problems discussed in [27], this work proposes an improved attribute-based secure data sharing scheme for mHealth with hidden policies and traceability. Specifically, this study aims to (1) solve the problem of identity disclosure by introducing a concept of public pseudo-identity, wherein the real identity is only known by the centre authority (CA); (2) save bandwidth, wherein the interaction between CS and data user is avoided during token generation; and (3) meet the application needs of mobile medical system, wherein the temporal factor is introduced by setting the validity period of shared information by the data owner.

1.3. Organization. The rest of paper is organized as follows. We introduce the cryptographic primitives and describe the access policy and mHealth system model in Section 2. In Section 3, we review the scheme in [27] and give a detailed discussion. Section 4 gives an improved scheme, followed by security and performance analysis. Finally, Section 5 shows a conclusion of this paper.

2. Preliminaries

2.1. Bilinear Map. Let G_0 and G_1 be two groups with prime order p. e, a bilinear map, e: $G_0 \times G_0 \longrightarrow G_1$, satisfies the following properties:

(1) *Bilinearty.* $e(u^a, v^b) = e(uv)^{ab}$, where $u, v \in G_0$ and $a, b \in Z_p^*$

(2) *Nondegenerate.* $e(g, g) \neq 1$, where g is the generator of G_0

(3) *Computable.* There is an efficient algorithm to compute the bilinear map e

2.2. Security Assumption

2.2.1. k – BDHE Problem (Bilinear Diffie–Hellman Exponent) [39]. Let G_0 be a bilinear group with prime order p and g is the generator of G_0. The k – BDHE problem in G_0 states that given a vector of $2K + 1$ elements $(h, g, g^a, g^{a^2}, \ldots, g^{a^K}, g^{a^{K+2}}, \ldots, g^{a^{2K}}) \in G_0^{2K+1}$, it is computationally intractable to compute the value $e(g, h)^{\alpha^{K+1}}$.

Define the set $Y_{g,\alpha,K}$ as $Y_{g,\alpha,K} = \langle g^a, g^{a^2}, \ldots, g^{a^K}, g^{a^{K+2}}, \ldots, g^{a^{2K}} \rangle$.

Definition 1 (Decisional$k$ – BDHE). The decisional k – BDHE assumption is said to be held in G_1, if there is no probabilistic polynomial time adversary with nonnegligible advantage to distinguish

$$\begin{aligned} &\left\langle h, g, Y_{g,\alpha,K}, e(g, h)^{\alpha^{K+1}} \right\rangle, \\ &\left\langle h, g, Y_{g,\alpha,K}, e(g, h)^{R} \right\rangle, \end{aligned} \tag{1}$$

where $\alpha, R \in Z_p^*$ and $g, h \in G_0$.

2.2.2. l – SDH Assumption (Strong Diffie–Hellman) [40]. Given a $(l + 1)$–tuple $(g, g^a, g^{a^2}, \ldots, g^{a^l})$ as input, output $(c, g^{(1/(\alpha+c))}) \in Z_p^* \times G_0$. An algorithm $\mathbb{C}$ has advantage ε in solving l – SDH in G_0 if the following holds:

$$Pr\left[\mathbb{C}\left(g, g^a, g^{a^2}, \ldots, g^{a^l}\right) = \left(c, g^{(1/(\alpha+c))}\right)\right] \geq \varepsilon, \tag{2}$$

where random $\alpha \in Z_p^*$.

Definition 2. The l – SDH assumption is (t, ε)–secure if no t–time algorithm has advantage at least ε in solving the l – SDH problem in G_0.

2.3. Access Policy. The attribute universe is $U = \{A_1, A_2, \ldots, A_k\}$, each $A_i \in \{A_i^+, A_i^-, A_i^*\}$, where $A_i^+ = i$ denotes that the user has A_i, $A_i^- = k + i$ denotes that the user has no A_i, $A_i^* = 2k + i$ denotes a wildcard specifying do not care, $i = 1, 2, \ldots, k$. Let $L_{\text{ua}} = \{L_{\text{ua}}[1], L_{\text{ua}}[2], \ldots, L_{\text{ua}}[k]\}$ be an attribute set of user, where $L_{\text{ua}}[i] \in \{A_i^+, A_i^-\}$.

Let $W = \{W[1], W[2], \ldots, W[k]\}$ be an AND-gate access policy, where $W[i] \in \{A_i^+, A_i^-, A_i^*\}$. Denote $L_{\text{ua}}| = W$ that the attribute set L_{ua} of user satisfies W. Then,

$$L_{\text{ua}}\big| = W \Longleftrightarrow W \sqsubset L_{\text{ua}} \sqcup \{A_1^*, A_2^*, \ldots, A_i^*\}. \tag{3}$$

2.4. A System Model. In mHealth systems, individual intelligent sensors monitor certain physiological signal and send to the mobile device. Then, the mobile devices upload the received data to the cloud. Users with requirements can initiate requests to obtain retrieval authorisation. In Figure 2, there exist four types of entities in the improved scheme as follows:

(1) Centre authority (CA): it is a trusted entity that generates the system master key and public parameters and issues user's private key on his attributes

(2) Data owner: this is a patient who encrypts his data and generates the encrypted keyword index and then uploads them to cloud server

(3) Data user: patient, physician, nurse, researcher, etc. can be such entities who obtain his private key from CA; he/she generates the token of keyword and gets search authorized to decrypt a ciphertext only if his/her attribute set satisfies the corresponding access policy

(4) Cloud sever (CS): this is a storage centre that stores electric medical and health records and carries searching and some other work, such as partially decryption.

Table 1 gives the notations used in the paper.

3. The Scheme in Reference [27]

In this section, we review the scheme in Reference [27] and give a detailed discussion.

3.1. Review of the Scheme

3.1.1. Setup. Let G_0, G_1 be two bilinear groups of prime order p with generator $g \in G_0$, e: $G_0 \times G_0 \longrightarrow G_1$, and random $\alpha, \beta, \gamma \in Z_p$. H: $Z_p^* \longrightarrow G_0$is a hash function. CA computes $g_j = g^{\alpha^j}$, $j = 1, 2, \ldots, K, K+2, \ldots, 2K$, $K = 3k$. Then, it computes $v = g^\gamma$ and $h = g^\beta$. The public and master keys are PK $= (g, g_1, \ldots, g_K, g_{K+2}, \ldots, g_{2K}, v, h)$, MSK $= (\alpha, \beta, \gamma)$, and an identity table $T = \varnothing$ is initialized.

3.1.2. KeyGen. Assume that each data user U_a with identity id_{ua} and an attribute set $L_{\text{ua}} = \{L_{\text{ua}}[1], L_{\text{ua}}[2], \ldots, L_{\text{ua}}[k]\}$, if U_a has attribute A_i, $L_{\text{ua}}[i] = i$, else $L_{\text{ua}}[i] = k + i$. CA randomly chooses $c, \mu_a \in Z_p^*$, $\{r_1, r_2, \ldots, r_k\} \in Z_p$ and computes $r = \sum_{i=1}^k r_i$, $D = g^{((r\gamma)/(\mu_a + c))}$, $D' = c$, $D'' = g^r$, and $D_a = g^{\mu_a}$. For each $i \in [1, k]$, the following are computed:

$$\begin{aligned} D_i &= H(j)^\beta, \\ D_{i1} &= \begin{cases} g^{\gamma\left(c\alpha^j + \left(r_j/(\mu_a + c)\right)\right)}, & j \in [1, k], \\ g^{\gamma\left(c\alpha^j + \left(r_{j-k}/(\mu_a + c)\right)\right)}, & j \in [k+1, 2k], \end{cases} \\ D_{i2} &= g^{\gamma\left(c\alpha^j + \left(r_{j-2k}/(\mu_a + c)\right)\right)}, \quad j \in [2k+1, 3k]. \end{aligned} \tag{4}$$

The private key of user U_a is

$$\text{SK}_{\text{ua}} = \left(D, \{D_{i1}, D_{i2}, D_i\}, D', D'', D_a\right). \tag{5}$$

At last, CA puts a tuple $(c, \text{id}_{\text{ua}})$ into table T and uploads tuple $(\text{id}_{\text{ua}}, \{l_i = g_j^{D'}\}_{i=1}^k)$ to the CS.

3.1.3. Encrypt. The data owner U_o specifies an access policy W, where each attribute is either positive/negative or wildcard. U_o chooses a random $t \in Z_p$ and computes Key $= e(g_K, g_1)^{kt}$, $C = \{M\}_{\text{Key}}$, $C_0 = g^t$, and $C_1 = (v\prod_{j \in W} g_{K+1-j})^t$.

Next, U_o chooses a random $b \in Z_p^*$, computer $C_2 = g^b$, $s_i = e(h^b, H(j))$, $H(s_i)$ for $i \in [1, k]$, then the W is obfuscated as $\overline{W}$. Then, the cipher is

$$\text{CT} = (\overline{W}, C, C_0, C_1, C_2, \text{id}_{\text{ua}}). \tag{6}$$

3.1.4. GenToken. The data user U_a with a set of attribute L_{ua} wants to access the shared data of owner U_b, gets C_2 from CS, and computes $s_i = e(C_2, D_i) = e(g^b, H(j)^\beta)$, $I_i = H(s_i)$. As a result, the attribute set L_{ua} is transformed into $\overline{L}_{\text{ua}}$. The token is

$$\text{TK}_{\text{ua}} = \left(\{I_i\}_{i=1}^k\right). \tag{7}$$

3.1.5. PDecrypt. The CS checks whether the attribute set $\overline{L}_{\text{ua}}$ satisfies the access policy $\overline{W}$. If satisfies, CSP searches $(\text{id}_{\text{ua}}, \{l_i = g_j^{D'}\}_{i=1}^k)$ to partially decrypt CT as follows:

$$\begin{aligned} A_i &= e(l_i, C_1) = e\left(g_j^{D'}, C_1\right) \\ &= e\left(g, v\prod_{s \in W} g_{K+1-s}\right)^{\alpha^j t D'} \\ &= e\left(g, g^{\left(\gamma + \sum_{s \in W} \alpha^{K+1-s}\right)}\right)^{\alpha^j t D'} \\ &= e(g, g)^{\left(\alpha^j t D' \gamma + t D' \sum_{s \in W} \alpha^{K+1-s+j}\right)}, \end{aligned} \tag{8}$$

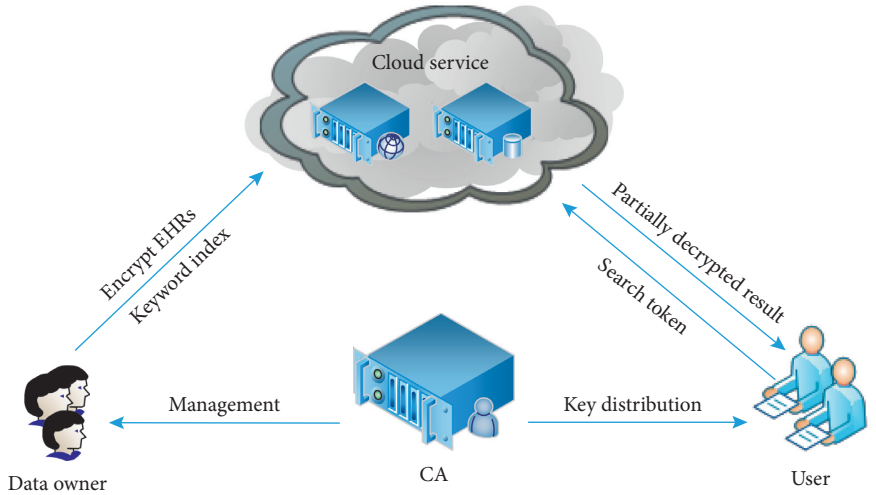

Figure 2: A system model.

Table 1: Notations used in the paper.

Notations	Description
i	The ith attribute, $i = 1, 2, \ldots, k$.
j	The value of the ith attribute $j \in [1, 3k]$
L_{ua}	The attribute set of user U_a
$L_{\text{ua}}[i]$	The ith attribute value of U_a
W	The access policy
$W[i]$	The ith attribute value of W
$\overline{L}_{\text{ua}}$	The hidden attribute set
$\overline{W}$	The hidden access policy

for all $i \in [1, k]$. Then, it computes a production of all A_i as $\text{CT}' = \prod_{i=1}^{k} A_i$ and sends to the data user U_a.

3.1.6. Decrypt. Once the partially decrypted ciphertext T' is received, U_a computes the following:

$$B_i = e\left(C_0, \left(\prod_{s \in W, s \neq j} g_{K+1-s+j}\right)^{D'} \cdot D_i\right) = e(g,g)^{\left(tD'\sum_{s \in W}\alpha^{K+1-s+j} + t\gamma\left(\alpha^j D' + \left(r_i/\left(\mu_a + c\right)\right)\right)\right)}, \tag{9}$$

where $D_i = \begin{cases} D_{1i}, L_{\text{ua}}[i] \in [1, 2k] \\ D_{2i}, L_{\text{ua}}[i] \in [2k+1, 3k] \end{cases}$, and obtains $B = \prod_{i=1}^{k} B_i$; next, U_a carries the decryption as follows:

$$\frac{\text{CT}'}{B} \cdot e(D, C_0) = e(g,g)^{\alpha^{K+1} k t D'},$$

$$\left(\frac{\text{CT}'}{B} \cdot e(D, C_0)\right)^{1/D'} = e(g,g)^{\alpha^{K+1} k t}$$

$$= e\left(g^{\alpha^K}, g^{\alpha}\right)^{kt} = e(g_K, g_1)^{kt} = \text{Key}. \tag{10}$$

Then, U_a decrypts $\{M\}_{\text{Key}}$.

3.1.7. Trace. SK_{ua} is well-formed if the following conditions hold:

$$e\left(D_a \cdot g^{D'}, D\right) = e\left(v, D''\right) \neq 1. \tag{11}$$

If SK_{ua} is well-formed, CA searches D' in T. If D' is in T, then it can output the corresponding identity id_{ua}.

3.2. Analysis of the Scheme. Problems in the scheme are observed. A detailed analysis is given follows.

3.2.1. Destroy One-to-Many Mechanism. The data owner U_o must know exactly the identity id_{ua} of data user in advance, wherein U_o can decide if he/she will provide access to the target user U_a. In this case, the identity id_{ua} of data user is sent to CS with ciphertext. CS can confirm which users can view or access shared messages by providing access rules, which may threaten the security of data users. As a result, this feature is not in line with the feature of developed in the attribute encryption mechanism, given that it cannot guarantee the anonymity of data users.

3.2.2. Identity Leakage. Before constructing the search token, the data user U_a firstly obtains C_2 of the data owner U_o from the cloud service provider (CSP) firstly. In this case, data users may know the identity of the data owner who shared the information he/she is interested in. Therefore, the anonymity of the data owner cannot be guaranteed and the application scope of the scheme is limited.

3.2.3. Interaction Problem. While generating the token, an interaction exists between data user and the CSP. The data user submits the identity U_o of the data owner he/she wants to access. Then, the CS feeds back C_2 corresponding to the given identity U_o, which increases the communication load.

4. Improved Scheme

In this section, we propose an improved scheme that can overcome the defects in [27] by introducing new features without weakening security or setting any particular conditions. (1) Public pseudo-identity is introduced, wherein the real identity is only known by the CA. (2) The access policy is hidden, and the user attribute set is made complicated by eliminating the bilinear pairings to reduce the calculation load. As a result, users will not apply to CS for aid information of the data owner when generating the token. (3) Access validity is added to the ciphertext.

4.1. Concrete Scheme

4.1.1. Access Validity. In order to introduce the temporal factor, we give a mechanism to determine the access validity.

T is the access validity of shared data, T_{stamp} is the time stamp, and T is divided into $(T_1, T_2, \ldots, T_\tau)$ based on different time units of application requirements. Then $H_T = (h_1, h_2, \ldots, h_\tau, h_1', h_2', \ldots, h'\tau_{,})$, where $h_i = h(T_i \mid T_{\text{stamp}})$, $h_{\tau}^{,}$ is redundant data, and$h'i_{,} \neq h_i$. Let tt be the current time; if $h(tt - T_{\text{stamp}} \mid T_{\text{stamp}}) \in H_T$, then tt satisfies T (Figure 3).

For example, access validity $T = 9$ days, which can be expressed as $H_9 = (1, 2, \ldots, 9)$ and time stamp $T_{\text{stamp}} =$ 2020.02.15. A request occurred at timett = 2020.02.19 and then $h_4 = h(4 \mid 2020.02.15) \in H_9$, so the request is within the validity period.

4.1.2. Concrete Construction. Figure 4 shows the overview of the improved scheme, which is described below.

(1) Setup. The same as in [27]. H_1: $\{0,1\}^* \longrightarrow Z_p^*$, h: $\{0,1\}^* \times G_1 \longrightarrow Z_p^*$.

(2) KeyGen. Assume an user U_t with identity id_{ut} and attribute set L_{ut}, where t can be either o or a. U_o is the data owner and U_a is the data user. $D' = c = H_1(\text{id}_{\text{ut}})$, $D_i = H(L_{\text{ua}}[i] \mid i)^\beta$, and the private key and public identity are

$$\begin{aligned} \text{SK}_{\text{ut}} &= \left(D, \{D_{i1}, D_{i2}, D_i\}, D', D'', D_t\right), \\ \text{HID}_{\text{ut}} &= h\left(\text{id}_{\text{ut}} \,\|\, D_t\right). \end{aligned} \tag{12}$$

At last, put a tuple $(\text{id}_{\text{ut}}, \text{HID}_{\text{ut}}, c)$ into the table T and upload $(\text{HID}_{\text{ut}}, \{g_j^{D'}\})$ to CS.

Instantly, having U_o with identity id_{uo} and attribute set L_{uo}, the private key and public identity are

$$\begin{aligned} \text{SK}_{\text{uo}} &= \left(D, \{D_{i1}, D_{i2}, D_i\}, D', D'', D_o\right), \\ \text{HID}_{\text{uo}} &= h\left(\text{id}_{\text{uo}} \,\|\, D_o\right). \end{aligned} \tag{13}$$

Having U_a with identity id_{ua} and attribute set L_{ua}, the private key and public identity are

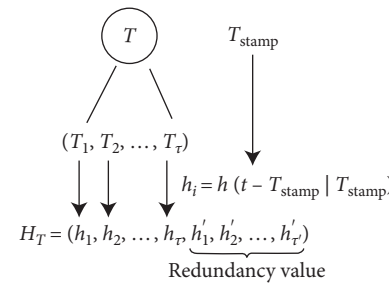

Figure 3: Access validity.

$$\begin{aligned} \text{SK}_{\text{ua}} &= \left(D, \{D_{i1}, D_{i2}, D_i\}, D', D'', D_a\right), \\ \text{HID}_{\text{ua}} &= h\left(\text{id}_{\text{ua}} \,\|\, D_a\right). \end{aligned} \tag{14}$$

(3) Encrypt. Only the differences are shown. U_o computes $C_2 = \prod_{j \in [1,2k]} H(j \mid i)^t$ and obfuscates W as follows: if $\text{W}[i] = \text{A}_i^*$, $\overline{W}[i] = *$, else $\overline{W}[i] = 0$, so only $*$ and 0 are in the hidden access policy $\overline{W}$. Next, U_o sets access validity T of M and computes $H_T = (h_1, h_2, \ldots, h_\tau, h_1', h_2', \ldots, h'\tau')$ as shown in Section 4.1.1. Then the cipher is

$$\text{CT} = \left(\overline{W}, C, C_0, C_1, C_2, \text{HID}_{\text{uo}}, H_t, T_{\text{stamp}}\right). \tag{15}$$

(4) GenToken. U_a chooses $x \in Z_p^*$ randomly and computes $\overline{D_i} = H(L_{\text{ua}}[i] \mid i)^{(\beta/x)}$, $tk_a = h^{1/x} = g^{\beta/x}$. The token is

$$\text{TK}_{\text{ua}} = \left(\{\overline{D_i}\}, \text{tk}_a, \text{HID}_{\text{ua}}\right). \tag{16}$$

(5) PDecrypt. Once the query is received, CS gets current time tt and computes whether $h(tt - T_{\text{stamp}} \mid T_{\text{stamp}}) \in H_T$. If it holds, CS goes on to judge whether the attributes set $\overline{L}_{\text{ua}}$ satisfies the access policy $\overline{W}$ by verifying equation (17). Then, CS performs the same operations as in the literature [27] and returns results to U_a:

$$e\left(\prod_{\overline{W}[i]=0} \overline{D}_i, C_0\right) = e\left(C_2, \text{tk}_a\right). \tag{17}$$

(6) Decryp. The difference in this step is $D_i' = \begin{cases} D_{1i}, \text{if } \overline{W}[i] = 0 \\ D_{2i}, \text{if } \overline{W}[i] = * \end{cases}$; as long as U_a is a legitimate user, it can correctly meet the requirements of the hidden access policy $\overline{W}$ at one time.

(7) Trace. When something goes wrong, only pseudoidentity HID_{ut} is submitted to CA who can find the true identity id_{ut} from $(\text{id}_{\text{ut}}, \text{HID}_{\text{ut}}, c)$.

4.2. Security Model.

The security model of the improved scheme is similar to that of the scheme in [27]. The data confidentiality of the improved scheme is considered to be guaranteed if there is no probabilistic polynomial-time adversary A with nonnegligible advantages in the following security game.

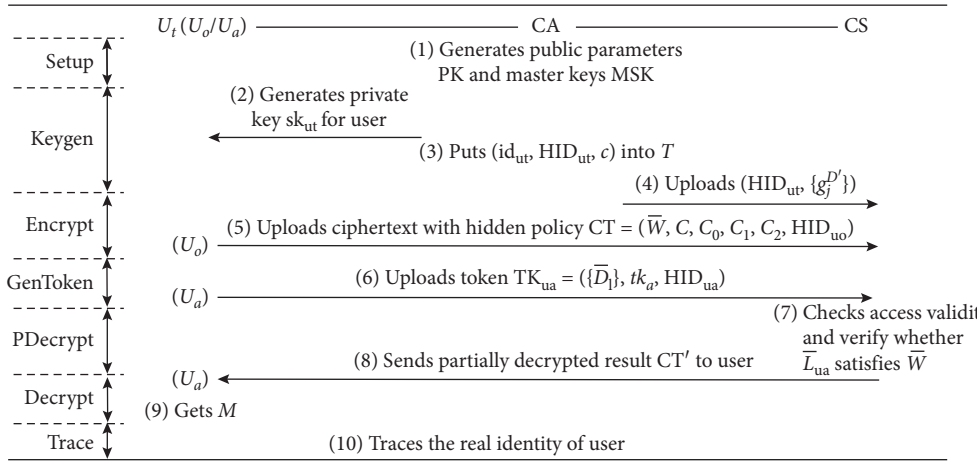

Figure 4: Overview of the improved scheme.

4.2.1. Init. The adversary A chooses a challenge access policy W^* and sends it to the challenger B.

4.2.2. Setup. The challenger B runs the *Setup* algorithm and publishes the public parameter PK.

Phase 1. The adversary A submits $(\mathrm{id}_{\mathrm{ua}}, L_{\mathrm{ua}})$ to query decryption keys where $L_{\mathrm{ua}} \mid \neq W^*$. The challenger B answers with a decryption key $\mathrm{SK}_{\mathrm{ua}}$. A repeats this phase adaptively.

4.2.3. Challenge. The challenger B runs *Encrypt* algorithm to obtain $(\langle C_0^*, C_0^* \rangle, \mathrm{Key})$. Next, B sets $\mathrm{Key}_0 = \mathrm{Key}$ and picks a random Key_1 of same length as Key_0. It then flips a random coin $b \in \{0, 1\}$ and gives $(\langle C_0^*, C_0^* \rangle, \mathrm{Key}_b)$ to the adversary.

Phase 2. The adversary A repeats Phase 1.

4.2.4. Guess. The adversary A outputs a guess $b' \in \{0, 1\}$.

The adversary A wins the game if $b' = b$ under the restriction that $L_{\mathrm{ua}} \mid \neq W^*$. The advantage of an adversary in this game is defined as

$$\left| \Pr[b' = b] - \frac{1}{2} \right|. \tag{18}$$

4.3. Security Analysis

4.3.1. Data Confidentiality. The security of improved scheme is still based on the k – DBDH problem.

Theorem 1. *If a probabilistic polynomial-time adversary A can break our scheme with a nonnegligible advantage, then we can construct a simulator B to solve k − BDHE problem with a nonnegligible advantage.*

Proof. A is an adversary who can break our scheme, and then we can construct a simulator B which solves the k – BDHE problem. □

Then, A and B play the interactive game in Figure 5.

(1) Init. A submits a challenged access policy W^* to B.

(2) Setup. The simulator B runs the *Setup* algorithm to generate the public parameter PK. B chooses random $d \in Z_p$ and generates

$$v = g^d \left(\prod g_{K+1-j} \right)^{-1} = g^{\left(d - \sum_{j \in W} \alpha^{K+1-j} \right)} = g^{\gamma}, \tag{19}$$

where B outputs $\mathrm{PK} = (g, Y_{g,\alpha,K}, v) \in G^{2K+1}$.

Phase 3. The adversary A submits $(\mathrm{id}_{\mathrm{ua}}, L_{\mathrm{ua}})$ to query private keys, where $L_{\mathrm{ua}} \mid \neq W^*$. The challenger B first selects k random numbers $r_i \in Z_p$ for $i = 1, 2, \ldots, k$ and sets $k = r_1 + r_2 + \cdots + r_k$. Then, B randomly chooses $a, c \in Z_p^*$ and computes

$$D = \left(g^d \left(\prod g_{K+1-j} \right)^{-1} \right)^{(r/(a+c))} = g^{(r\gamma/(a+c))}. \tag{20}$$

For $i \in [1, k]$,

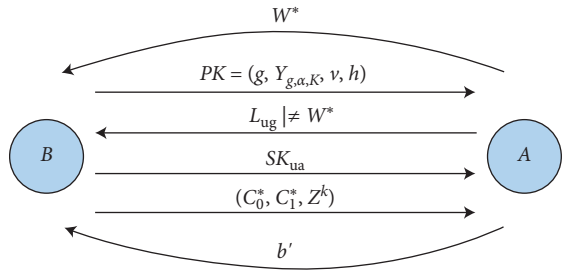

Figure 5: The game between the adversary and simulator.

$$D_{i1} = \begin{cases} g^{d}\left(\prod g_{K+1-j}\right)^{-c}\prod\left(g_{K+1-j}\right)^{\left(-r_i'/(a+c)\right)} = \left(g^{d}\left(\prod g_{K+1-j}\right)^{-1}\right)^{\left(c\alpha^{i}+\left(r_j/(a+c)\right)\right)}, & \text{if } j \in [1,k], \\ g^{d}\left(\prod g_{K+1-j}\right)^{-c}\prod\left(g_{K+1-j}\right)^{\left(-r_i'/(a+c)\right)} = \left(g^{d}\left(\prod g_{K+1-j}\right)^{-1}\right)^{\left(c\alpha^{i}+\left(\left(r_{j-k}\right)/\left(\mu_a+c\right)\right)\right)}, & \text{if } j \in [k+1,2k], \end{cases} \tag{21}$$

$$D_{i2} = g\left(g^{d}\left(\prod g_{K+1-j}\right)^{-1}\right)^{\left(c\alpha^{i}+\left(\left(r_{j-2k}\right)/\left(\mu_a+c\right)\right)\right)}, \quad j \in [2k+1,3k].$$

(3) Challenge. B sets $C_0^* = h = g^t$ for some t and $C_1^* = h^d$ and gives $(\langle C_0^*, C_0^*\rangle, Z^k)$ to A. Thus $h^d = (g^d)^t = (g^d(\prod g_{K+1-j})^{-1}\prod g_{K+1-j})^t = (v\prod g_{K+1-j})^t$, and $Z^k = \text{Key}$ if $Z = e(g,h)^{\alpha^{K+1}}$.

Phase 4. The adversary A repeats Phase 1.

(4) Guess. The adversary A outputs a guess $b' \in \{0,1\}$. If the adversary A outputs $b' = 1$, Z is a random element; if $b' = 0$, Z is $e(g,h)^{\alpha^{K+1}}$. When $b' = 0$, B breaks the $k-$BDHE problem.

Note that in attribute-based cryptography, collusion attack is an important discussion point. In order to model the collusion attacks, a decrypting proxy is presented. Each decryption proxy $p_i(r)$ simulates a legal decryption key component with a random r. The definition of decryption proxy and the detail of model collusion attacks are given in [27], which will not be discussed here.

If there is 0-collusion, B has at least $\in/2$ advantage in breaking the $k-$BDHE problem.

If there is 1-collusion, B has at least $(1-(q/p))^l\ (\in/2)$ advantage in breaking the $k-$BDHE problem.

If there is m-collusion, B has at least $(1-(1-(q/p))^l)^m\ (\in/2)$ advantage in breaking $k-$BDHE problem.

So, the advantage of B to solve the $k-$BDHE problem is $\max\{(\in/2),\ (1-(q/p))^l(\in/2), (1-(1-(q/p))^l)^m\ (\in/2)\}$.

4.3.2. Policy Privacy. In the proposed scheme, no extra computation is needed for policy hiding, given that $\{0,*\}$ exist in $\overline{W}$. Hence, when the CS receives the encrypted sharing data with $\overline{W}$, it can obtain nothing about the content and the access policy. The CS carries out partial decryption and sends the result to U_a. Likewise, unauthorised users, as adversaries, either from the server or users, can only obtain hidden access policies (Figure 6). Thus, our scheme provides policy privacy.

4.3.3. Two-Way Anonymity. When an entity joins the system, the CA generates a pseudo-identity HID_{ut} instead of his/her true identity id_{ut}. Firstly, the data owner shares information under a pseudo-identity HID_{uo}, and thus the CS and data users cannot obtain the true identity of the data owner. Secondly, potential data users' id_{ua} can have an access without revealing their true identity, given that their attribute set satisfies the access policy. In this way, the data owner and the CS cannot know who access the encrypted information uploaded in the system. Therefore, the improved scheme ensures the two-way anonymity whilst realising flexible authorisation.

4.3.4. Traceability. The security of improved scheme is still based on $l-$SDH problem. When a user identity HID_{ut} is questioned, only the CA can trace his/her true identity id_{ut} by detecting the corresponding key and querying $(\text{id}_{\text{ut}}, \text{HID}_{\text{ut}}, c)$. In particular, $D' = c = H_1(\text{id}_{\text{ut}})$, where true identity id_{ut} is used to generate c, which further enhances the accuracy of tracking.

4.3.5. Access Validity. We also include the temporal factor with ciphertext in the improved scheme by giving a judgment mechanism, given that some EHRs have no shared value after a certain period of time. To prevent attack, access validity T and time stamp T_{st} are hashed, which provides evidence for CS verification. For example, the emergency message uploaded by a patient is invalid after his/her treatment time.

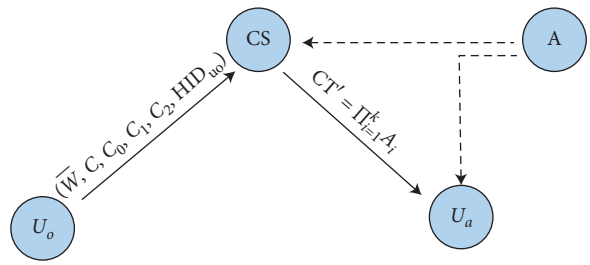

Figure 6: The scheme ensures the policy privacy.

Table 2: Comparison of computation and communication.

Location	Enc.	Token	Dec.	Ciphertext length
[15]	$(n+2)ex$	—	$(2n+1)p + nex$	$(n+1)\|G_0\| + \|G_1\|$
[20]	$(2n+3)ex$	—	$(2n+1)p + nex$	$2(n+1)\|G_0\| + \|G_1\|$
[27]	$3ex + np$	$n(p+h)$	$(n+1)(p+ex)$	$3\|G_0\| + \|G_1\|$
Proposed	$3ex$	$(n+1)ex$	$(n+1)(p+ex)$	$2\|G_0\| + \|G_1\|$

Table 3: Comparison of features.

Location	One-to-many	Anonymity		Noninteraction	Access validity
		Owner	User		
[15]	✓	×	✓	✓	×
[20]	✓	×	×	✓	×
[27]	×	×	✓	×	×
Proposed	✓	✓	✓	✓	✓

4.4. Performance Analysis. In this section, we conduct a performance analysis on the improved scheme compared with that on the existing schemes [15, 20, 27]. For the sake of simplicity, we define some notations on the main operation: n: the number of attributes; ex: modular exponentiation operation; p: pairing operation; h: hash operation; $|G_0|$ and $|G_1|$: the length of element in G_0 and G_1, respectively.

4.4.1. Computation and Communication Costs. As shown in Table 2, computation cost during encryption phase is constant in our scheme, whereas it increases with the number of attributes in other schemes. During the decryption phase, the computation cost is equal to scheme [27], whereas in scheme [15, 20], it is relatively higher. Compared with the computation in the token generation phase, $(n+1)ex$ is observed in our scheme, whereas $n(p+h)$ in scheme [27]. Therefore, the calculation amount of our scheme is low.

In the case of ciphertext, $2|G_0| + |G_1|$ is used in our scheme and $3|G_0| + |G_1|$ in [27]. However, communication cost in other two schemes are $(n+1)|G_0| + |G_1|$ and $2(n+1)|G_0| + |G_1|$, respectively. Thus, the proposed scheme has lesser communication cost, which is independent of the number of attributes.

4.4.2. Features. Table 3 shows the comparison amongst the features of different schemes. The proposed scheme provides one-to-many application requirement and two-way anonymity of data owner and data user, supports noninteractive relationship in token generation, and considers access validity.

Next, we give the thorough experimental evaluation of our scheme. Our simulation experiment is on Intel(R) Core(TM) i7-6500U CPU at 2.5 GHz and 8.00 GB RAM. The algorithms are implemented using the pairing-based cryptography (PBC) library version 0.4.7-vc.zip [41]. Concretely, we select the Type A elliptic curve parameter with the 160-bit order in PBC library. For comparison convenience, we set $n \in [1, 25]$, and all of the experimental results are averages of 200 trials. Meanwhile, we just show the experimental results of *Encrypt*, *Decrypt*, and *GenToken* algorithms.

As shown in Figure 7(a), the encryption time in the proposed scheme is constant $3ex$, whereas in other three schemes, they are $(n+2)ex$, $(2n+3)ex$, and $3ex + np$, respectively; they increase with the number of attributes in the access policy. In the decryption phase, the time cost of our scheme is almost the same as that in scheme [27], while the time cost of the other two schemes is relatively high, as shown in Figure 7(b). Figure 7(c) shows that the token generation time of the improved scheme is slightly lower than that in [27], given that no bilinear pairing exists in our scheme. Thus, the improved scheme is efficient without reducing the security.

4.4.3. Further Efficiency Comparison. In order to show the efficiency of improved scheme, we also simulate the main phase of our scheme on the laptop with Intel(R) Core(TM) i7-8550U CPU at 1.80 GHz and 8.00 GB RAM. Figure 8 shows the results on different devices.

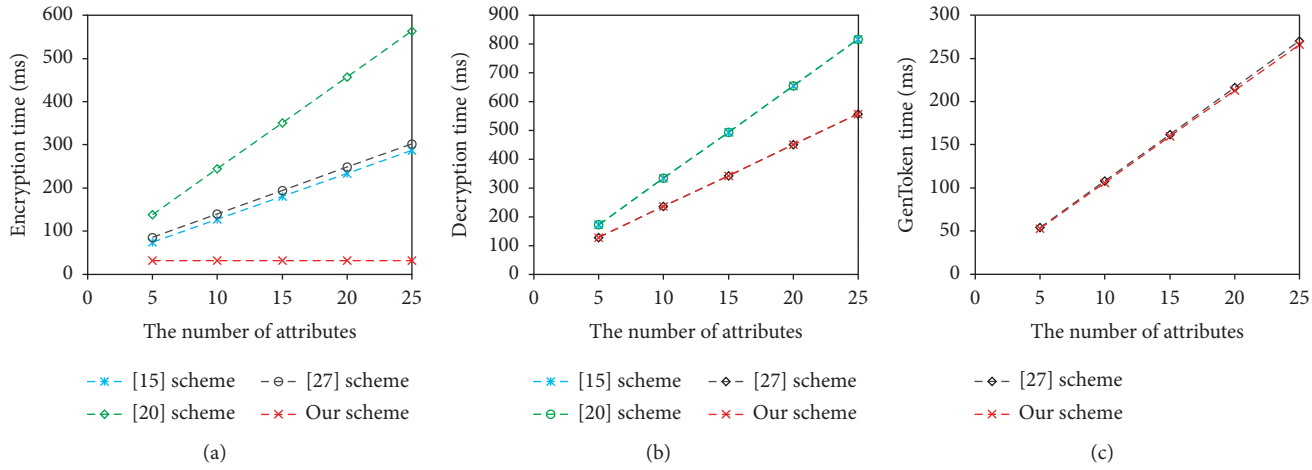

Figure 7: Time costs of different phases. (a) Encryption time. (b) Decryption time. (c) GenToken time.

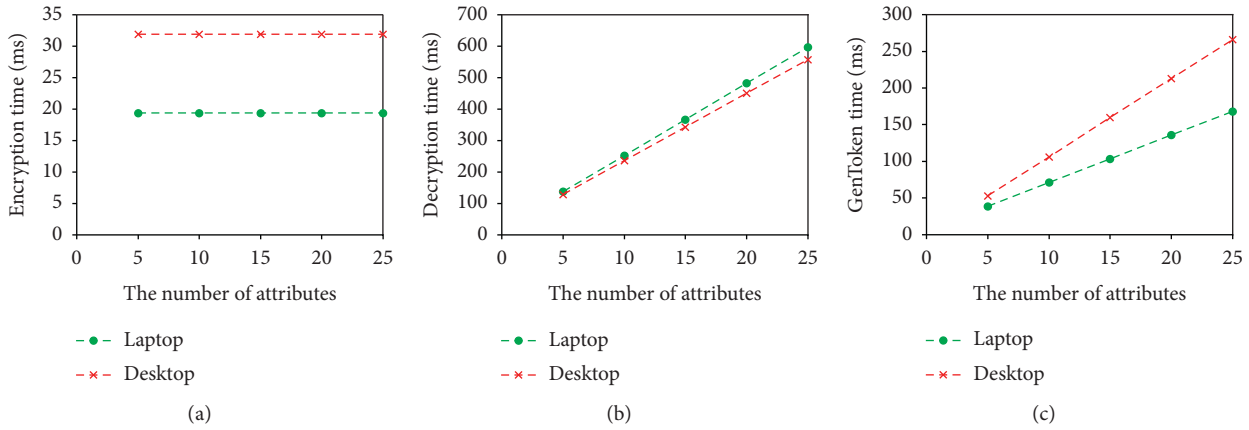

Figure 8: Time costs on different devices. (a) Encryption time. (b) Decryption time. (c) GenToken time.

5. Conclusions

In this study, we propose an improved secure sharing scheme using ABE for mHealth. Our improved scheme has advantages of two-way anonymity of data owner and data user, noninteractive relationship, and low computation costs without weakening security or setting any particular conditions. The improved scheme helps to protect EHRs from the unauthorised online entities in mHealth. The proposed scheme also considers access validity of EHRs. Through security and evaluative results of comparison, our scheme is found more efficient in terms of computational cost and energy consumption than three of the existing schemes.

As part of our future work, we aim to design efficient attribute-based signcryption schemes for mHealth. Additionally, we aim to provide different access rights for different users.

Data Availability

All relevant data are included within the article.

Conflicts of Interest

All the authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under grants 61662071 and 61562077 and the Young Teacher's Scientific Research Ability Promotion Program of Northwest Normal University (NWNU-LKQN-14-1).

References

[1] J. H. Park, J. A. Seol, and Y. H. Oh, "Design and Implementation of an Effective Mobile Healthcare System Using Mobile and RFID Technology," in *Proceedings of the 7th International Workshop on Enterprise Networking and Computing in Healthcare Industry*, June 2005.

[2] P. Gope and T. Hwang, "BSN-care: a secure iot-based modern healthcare system using body sensor network," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1368–1376, 2016.

[3] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based iot: challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.

[4] J. Zhou, Z. Cao, X. Dong, N. Xiong, and A. V. Vasilakos, "4s: a secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks," *Information Sciences*, vol. 314, pp. 255–276, 2015.

[5] C. C. Y. Poon, Y. T. Shu-Di Bao, and S. D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[6] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[7] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: a fuzzy attribute-based signcryption scheme," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 37–46, 2013.

[8] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption," *Future Generation Computer Systems*, vol. 52, pp. 67–76, 2015.

[9] L. Guo, C. Zhang, J. Sun, and Y. Fang, "PAAS: a privacy-preserving attribute-based authentication system for eHealth networks," in *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems (ICDCS '12)*, IEEE, Macau, China, pp. 224–233, June 2012.

[10] A. Kapadia, P. P. Tsang, and S. W. Smith, "Attribute-based publishing with hidden credentials and hidden policies," in *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS'07)*, pp. 179–192, San Diego, CA, USA, 2007.

[11] J. Hur, "Attribute-based secure data sharing with hidden policies in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2171–2180, 2013.

[12] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 126–138, 2015.

[13] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7–11, 2008, Proceedings, Part II, vol. 5126 of Lecture Notes in Computer Science*, pp. 579–591, Springer, Berlin, Germany, 2008.

[14] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Phoabe: securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot," *Computer Networks*, vol. 133, pp. 141–156, 2018.

[15] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," *Lecture Notes in Computer Science*, Springer, vol. 5735, pp. 347–362, , Berlin, Heidelberg, 2009.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *In Advances in Cryptology—EUROCRYPT 2005, vol. 3494 of Lecture Notes in Computer Science*, pp. 457–473, Springer, Berlin, Germany, 2005.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine–grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, Alexandria, VA, USA, October 2006.

[18] S. Belguith, A. Jemai, and R. Attia, "Enhancing data security in cloud computing using a lightweight cryptographic algorithm," in *Proceedings of ICAS 2015: The Eleventh International Conference on Autonomic and Autonomous Systems*, IARIA, Pune, Indiapp, pp. 98–103, February 2015.

[19] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.

[20] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2013.

[21] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied Cryptography and Network Security*, vol. 5037, , pp. 111–129, Springer, 2008.

[22] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant–size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.

[23] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. De Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theoretical Computer Science*, vol. 422, pp. 15–38, 2012.

[24] L. Cheung and C. Newport, ""Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 456–465, ACM, Alexandria, VA, USA, November 2007.

[25] K. Frikken, M. Atallah, and J. Jiangtao Li, "Attribute-based access control with hidden policies and hidden credentials," *IEEE Transactions on Computers*, vol. 55, no. 10, pp. 1259–1270, 2006.

[26] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 18-19, ACM, Seoul, South Korea, May 2012.

[27] C. Hahn, H. Kwon, and J. Hur, "Efficient attribute-based secure data sharing with hidden policies and traceability in mobile health networks," *Mobile Information Systems*, vol. 2016, Article ID 6545873, 13 pages, 2016.

[28] R. Xu and B. Lang, "A CP-ABE scheme with hidden policy and its application in cloud computing," *International Journal of Cloud Computing*, vol. 4, no. 4, pp. 279–298, 2015.

[29] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Information Security Practice and Experience, 5th International Conference, ISPEC 2009, Xi'an, China, April 13-15, 2009, Proceedings*, pp. 13–23, Springer, Berlin, Heidelberg, 2009.

[30] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant–size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.

[31] J. Li, X. Chen, J. Li et al., "Fine-grained access control system based on outsourced attribute-based encryption," in *European Symposium on Research in Computer Security*, vol. 8134, pp. 592–609, Springer, Berlin, Heidelberg, 2013.

[32] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Generation Computer Systems*, vol. 78, no. 2, pp. 730–738, 2016.

[33] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length," *Security and Communication Networks*, vol. 2017, Article ID 3596205, 11 pages, 2017.
[34] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, pp. 487–497, 2015.
[35] H. Zhong, W. Zhu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," *Soft Computing*, vol. 22, no. 1, pp. 1–9, 2016.
[36] J. Li, Y. Zhang, J. Ning et al., "Attribute based encryption with privacy protection and accountability for CloudIoT," *IEEE Transactions on Cloud Computing*, vol. 8, p. 1, 2020.
[37] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Information Sciences*, vol. 484, pp. 113–134, 2019.
[38] J. Cui, H. Zhou, Y. Xu, and H. Zhong, "OOABKS: online/offline attribute-based encryption for keyword search in mobile cloud," *Information Sciences*, vol. 489, pp. 63–77, 2019.
[39] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," *Lecture Notes in Computer Science*, vol. 3494, pp. 440–456, 2005.
[40] D. Boneh and X. Boyen, "Short signatures without random oracles," *Advances in Cryptology-EUROCRYPT 2004*, vol. 3027, pp. 56–73, 2004.
[41] http://crypto.stanford.edu./pc/.