Blockchain-Based Hierarchical Access Control with Efficient Revocation in mHealth System

Ting Liang[1], Yaorui He[1], Pei Huang[1], and Zhe Xia[1,2(✉)]

[1] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
[2] Hubei Key Laboratory of Transportation Internet of Things, Wuhan University of Technology, Wuhan 430071, China
xiazhe@whut.edu.cn

Abstract. With the development of information technology, people can share their health records (PHRs) through the Internet and obtain rapid medical services, which makes mobile health become a promising field. PHRs are collected from wireless body area networks (WBANs) and will be shared with people in different fields through public channels, increasing the risk of leaking personal privacy. Ciphertext-policy attribute-based encryption (CP-ABE) is a popular solution for fine-grained access control, but most existing schemes cannot be directly applied to the WBANs with limited resources and dynamic changes in user roles. In this paper, to meet the requirement of the mHealth System, we propose blockchain-based hierarchical access control with efficient revocation in the mHealth system. We use the Extendable Hierarchical attribute-based encryption (EH-ABE), a file-related hierarchical access control scheme, to encrypt PHRs, which reduces the repetitive computation and storage overhead. The proposed scheme adds the function of offline/online encryption, which can greatly save the energy consumption of the sensors in the WBANs. In addition, this scheme can provide attribute-level user revocation and is proven to be IND-CCA secure.

Keywords: Attribute level user revocation · Hierarchical Access Control · Offline/Online Encryption

1 Introduction

With the development of information technologies, wireless body area networks (WBANs) have been used to provide higher quality medical services [15]. WBAN is the fundamental component in mobile health system consisting of smart medical sensors. It can collect physiological parameters, such as body temperature, blood glucose level, blood pressure, heart rate, Electrocardiogram (ECG), or Electromyogram (EMG) [9]. These body parameters can be used to monitor

J. Chen and Z. Xia (Eds.), BlockTEA 2023, LNICST 577, pp. 135–153, 2024.
https://doi.org/10.1007/978-3-031-60037-1_8

chronic diseases. The mobile device receives the personal health records (PHRs) from the body area networks and encrypts them before uploading them to the cloud service provider (CSP). The PHR user, such as a researcher in a medical college, or a doctor in the hospital, can access these data if he/she is a valid user [17]. Then, the data could be used for better medical research or to provide users with more effective treatments.

PHR involves the user's private information and is shared with people in different fields through public channels. There may be malicious users in the system who steal data to obtain economic benefits, but the leakage of PHR can pose a serious threat to the life and health of patients. Therefore, ensuring data confidentiality is particularly crucial [22]. According to the multi-receiver characteristic of PHR sharing, CP-ABE is the best solution because it can provide fine-grained access control. However, the original CP-ABE schemes [2,28], are not suitable for direct application in the mHealth system. So far many CP-ABE schemes [13,23,26,29,32] have been proposed for the mHealth system. But none of them offer the function of user revocation. There are also some schemes with user revocation [1,8,24,30,31] which only provide user revocation but not user's attribute revocation. The work on attribute-level user revocation [4] is relatively less.

In the existing work, access policies are isolated and fixed during encryption, resulting in a complicated and repetitive encryption process, which is not suitable for resource-constrained body area networks. For example, a patient may have high blood pressure and heart disease. To treat these chronic diseases, physical signs information needs to be monitored, such as body temperature, blood pressure, heart rate, Electrocardiogram (ECG), and Electromyogram (EMG). Different data may require different access policies, such as Fig. 1. Body temperature can be accessed by physicians and nurses in the Department of Cardiology and Orthopedics. T_2 is an access policy for blood pressure, and doctors who treat hypertension and heart disease need to be allowed to access these data. T_4 is the access strategy for heart rate, and T_5 is the access strategy for EMG. However, hierarchical relationships may exist between these policies. For instance, doctors treating high blood pressure also need to monitor the patient's blood pressure and temperature. Doctors who treat cervical spondylosis need to monitor their patients' EMG and heart rate. The cardiologist needs blood pressure, temperature, ECG, EMG, and heart rate. In traditional schemes, they are encrypted separately, and users need to have an access policy for each data provider that provides services to them. In the hierarchical access control scheme, some access policies with hierarchical relationships can be aggregated into one, and data is simultaneously encrypted under the same access policy. This method is helpful to improve the efficiency of encryption and reduce computation and storage costs.

In this paper, we propose a revocable CP-ABE scheme with hierarchical access control and offline/online encryption, named RH-ABE, which simultaneously solves the problems of encryption efficiency, repetitive computation, and attribute-level user revocation. Inspired by the Key Encrypting Key(KEK) technique in the efficient revocation [11], we propose a revocable hierarchical access

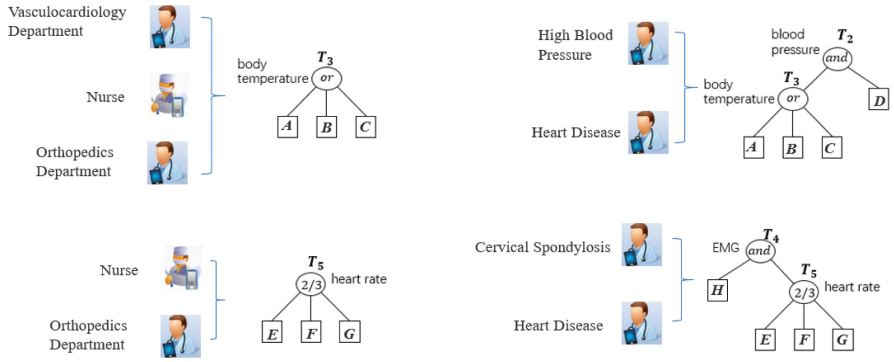


Fig. 1. Access Policies

control scheme for the mHealth system based on EH-ABE in [28]. RH-ABE enables the PHR owner(PO) to precompute policy-related ciphertext under self-defined fine-grained access policies when the body area network device is offline. When receiving PHR from sensors, the PO performs the online encryption process to compute the ciphertext associated with the message before uploading it to the cloud service provider (CSP). Unauthorized users cannot get the plaintext PHRs. By the revocation method, the PHR is preserved from being abused and exposed to revoked users. Our main contributions can be summarized as follows.

- **Offline/Online encryption:** Various sensor devices in body area networks have limited storage capacity, small battery capacity, and low computing power. In our scheme, when the device is offline, the PO encrypts the ciphertext related to the policy in advance. When the device is online, it encrypts the received ciphertext related to the PHR. This approach is helpful to reduce the sensor devices' energy consumption, hence it is suitable to be used in the resource-constrained environment.
- **Hierarchical access control:** In our scheme, we use a hierarchical access control structure that can encrypt a series of files with hierarchical access relationships. It can reduce the cost of complicated and repetitive encryption.
- **Attribute-level user revocation:** In this paper, we can implement fine-grained user revocation, where some attributes of a user are revoked without affecting the permissions of other attributes. This is suitable for a situation where the user's role is changing dynamically, with some attributes being removed while others remain, such as the change of the doctor's position. Once the attribute of a user is revoked, if his attributes do not match the access policy, he cannot continue to access the previous ciphertext.

2 Related Works

Attribute-based encryption (ABE) was first introduced by Sahai and Waters [21] in 2003. Then, Goyal et al. [6] proposed the first key-policy attribute-based

encryption (KP-ABE) scheme in 2006, which embedded policies into keys and attributes into ciphertext. KP-ABE can be used for various scenarios, such as video websites and log encryption management. Bethencourt et al. [2] proposed the first CP-ABE scheme in 2007, which embedded attributes into keys and policies into ciphertext. CP-ABE is found useful in the encrypted storage and fine-grained sharing of data on public clouds. In both of them, the monotonic access structure is expressed using a tree structure. Thanks to this feature, some improvements have been proposed for different requirements, such as ABE with hidden policy [19], ABE with revocable storage [12], ABE with user revocation [18], ABE with outsourced decryption [7].

2.1 Attribute-Based Hierarchical Encryption (HABE)

Attribute-based hierarchical control schemes can be divided into two classes according to the different construction ideas. One is the hierarchical management of attributes to reduce the workload of central attribute authority. The other is to aggregate and encrypt files with hierarchical access relationships. There are many schemes [14,20,25] followed the first idea by setting a central authority to govern a series of hierarchical domain authorities and the key generation could be performed by all of the authorities which can avoid key abuse. In 2016, Wang et al. [27] first proposed a hierarchical attribute-based encryption scheme in which encrypted files are hierarchical access control relationships. This kind of HABE can reduce the burden on data owners by encrypting files with hierarchical access control relationships. Following this idea, the access policies associated with the hierarchical files can be integrated into a single access structure with several levels. It is suitable for the mHealth system which has High requirements for resources and efficiency. In the existing work, there are several efficient schemes are proposed [5,28]. But the [5] is not flexible and only considers an AND-gate access structure.

2.2 User Revocation

A series of fine-grained access control constructions with user revocation for the mHealth system was proposed.

Based on the different ways to implement the revocation, user revocation can be done either by direct revocation or by indirect revocation. The first approach is achieved by maintaining a list of revocation users. Yang et al. [30] used this way by embedding the identity in the secret key and storing the revocation list in the public cloud. Tan et al. [24] proposed a blockchain-empowered approach for data sharing. In their scheme, the malicious user's ID will be added to the revocation list, and the revocation list will be sent to the medical practitioner. The medical practitioner re-encrypts the data. Li et al. [17] proposed a traceable and revocable scheme in which user revocation is complicated. The valid user in their scheme needs to be pre-chosen, but it's not suitable for most situations the PO doesn't know the users who can match the access policy. The second approach is realized by updating the secret key. Only the updated user can decrypt successfully, and

the revoked user cannot be updated. Guo et al. [8] proposed a revocable scheme in the cloud-assisted IoMT system. In the revocation process, a TTP updates the attribute set for the revoked user. The revoked user cannot get the information for a valid secret key.

Based on the scope of influence, attribute revocation can be divided into three categories: user revocation, user partial attribute revocation, and system attribute revocation. They are described as follows:

- **User revocation**: All attributes of a specific user are revoked without affecting other users.
- **Partial attributes revocation of a user**: Revocation of the attributes of a specific user without affecting other users and other attributes of the user.
- **System attribute revocation**: A specific attribute is revoked, and all users no longer have permission on the attribute.

Most of the existing works are user revocation, which is not the best solution for the changeable environment. In the scheme [16], there are two kinds of users: public domain users and individual domain users. The revocation of a public domain user can achieve attribute revocation and user revocation. The scheme [4] is also an attribute-level revocable scheme that used the Key Encrypting Key(KEK) techniques proposed in [11], but the access structure is the ordered binary decision diagram (OBBD).

3 Preliminaries

In this section, we review some basic notions and definitions. They are the basic knowledge of cryptography and effective tools for building the scheme.

3.1 Bilinear Maps

Definition 1 (Bilinear Maps [3]). Let G_0 and G_T be two finite cyclic groups of prime order p. And g is denoted as a generator of G_0. Select a map $e : G_0 \times G_0 \rightarrow G_T$ should satisfy:

- Bilinear: $\forall P, Q \in G_0$ and $\forall a, b \in Z_p^*$, there is $e(P^a, Q^b) = e(P, Q)^{ab}$,
- Non-degenerate: $e(g, g) \not\equiv 1$,
- Computable: $\forall P, Q \in G_0$, $e(P, Q)$ can be computed efficiently.

3.2 Access Structure

Definition 2 (Access Structure [28]). Denote $\{P_1, P_2, \cdots P_n\}$ as a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots P_n\}}$ is monotone if that $\forall B, C$, $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \cdots P_n\}$ i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in A are called the unauthorized sets.

3.3 Hierarchical Access Tree

Definition 3 (Hierarchical Access Tree [28]**).** The hierarchical access tree defined in [6] is used that is an extension of a tree structure in [28]. Denote $\mathcal{T}$ as a hierarchical access tree with an access structure that can be further represented as a series of hierarchical access structures. The x represents the non-leaf node of the tree, and y represents the leaf node of the tree. Each sub-tree with root x in $\mathcal{T}$ is a sub-structure. For example in Fig. 2, in the integrated structure T_1, there are four sub-trees that the trees T_2, T_3, T_4 and T_5 rooted at node 2, 3, 4 and 5 represent different sub-structures. Each node x of $\mathcal{T}$ that is not a leaf stands for a threshold gate i.e. an AND, OR, or OUTOF-gate, according to the number of its children and a threshold value. Denote num_x as the number of children of x, and z_x as the threshold value, then we have $0 < z_x \leq num_x$. In particular, the threshold gate is OR when $z_x = 1$, and the threshold gate is AND when $z_x = num_x$.

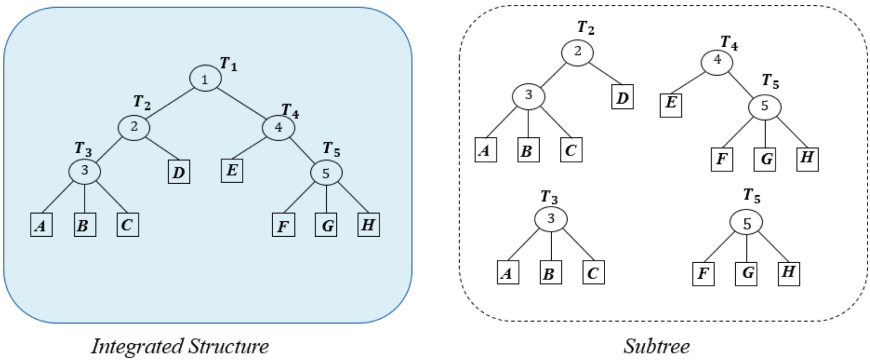


Fig. 2. Hierarchical Access Tree

4 Models and Definitions

In this section, we outline the models and definitions, including the system model, the communication model, the adversary model, the security requirements, and the security assumptions.

4.1 System Model

As shown in Fig. 3, RH-ABE consists of five types of participants: trusted authority (TA), PHR owner (PO), mHealth cloud service provider (mHealth CSP), PHR user (PU), and blockchain. The characteristics and functions of each party are described below:

- **TA**: TA is a trusted authority that is responsible for managing users and distributing secret keys. Specifically, it produces the system's public parameters and the system master secret key. It also issues the keys for PUs which grants fine-grained access to them. The keys are associated with their respective attributes, and none of them can collude with others for invalid entry. To effectively manage users and implement attribute-level user revocation, TA needs to generate a list of valid users for each attribute. When a user needs to be removed from the system, or the user's attribute set changes, the valid user list of attributes needs to be updated immediately.
- **PO**: PO integrates the PHRs that are collected by WBSN. WBSN employs smart wireless sensors embedded inside or on the skin of a patient. These sensors monitor the patient's vital physiological parameters who is suffering from chronic diseases. The collected data is aggregated and transmitted to a mobile device via wireless communication [31]. Before uploading the PHR to the mHealth CSP, the PO should encrypt it according to the access policy.
- **mHealth CSP**: In our system, CSP has two main functions: transform and store. In this scheme, attribute-level user revocation is implemented with the help of CSP. It runs algorithms to generate KEKs for all users in the system. When receiving the data uploaded by the PO, it first performs ciphertext transformation with the attribute group key. Then, the mHealth CSP stores the transformed PHRs. It also needs to delete the Previous ciphertext. When receiving a download request from the PU, the CSP constructs a ciphertext header that binds the information of the attribute group key. If the user is not revoked, he/she can recover the correct group key by using the valid KEKs. Furthermore, the CSP needs to keep the list of unrevoked users up to date.
- **PU**: PU is a PHR user, PU can access the shared PHRs if and only if he/she is not revoked and his/her attributes match the embedded access policy.

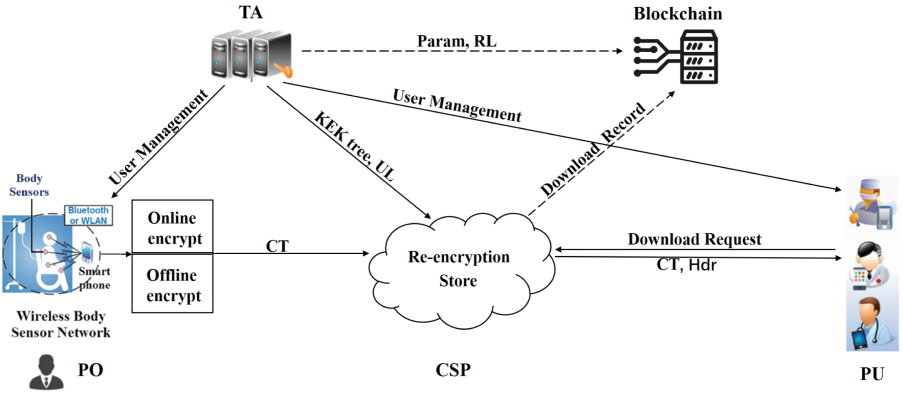


Fig. 3. System Architecture

4.2 Communication Model and Adversary Model

In our scheme, TA is assumed to send the secret keys to the users over secure channels. Information transmitted between users and the CSP is assumed to use an authentication channel. TA is fully trusted, it can't disclose any secrets. CSP is deemed as semi-honest, it keeps group keys secret and follows pre-defined operations. Meanwhile, it is curious about the PHRs and attempts to get secret information about the original PHRs as much as possible. We assume that CSP does not collude with revoked users to obtain unauthorized data. PUs are not trusted, and they can collude with each other for decrypting PHRs which none of them can decrypt alone.

4.3 Security Requirements

The following security requirements are considered in our proposed scheme.

- **Data Confidentiality.** PHRs are related to the life, health, and safety of users. The confidentiality of outsourced data should be protected. There are two main aspects of confidentiality: unauthorized users and revoked users. Unauthorized users, such as CSP or other unauthorized users, cannot access plaintext information of the outsourced PHRs. In this paper, our scheme is secure against the chosen ciphertext attacks to ensure data security.
- **Collusion Resistance.** In the system, different users' attribute sets are not identical. Some malicious users may try to collude by combining their secret keys to gain more privilege. In our scheme, this kind of attack defined as collusion can't succeed. Since TA is honest and user revocation is performed with the help of CSP. We assume that CSP can't collude with the revoked users.

4.4 Security Assumptions

CBDH Assumption [28]: Computational Bilinear Diffie-Hellman Assumption. According to the **Definition 1**, G_0 and G_T are denoted as finite cyclic groups with prime order p. Denote g as a generator of G_0. And $e : G_0 \times G_0 \rightarrow G_T$ is a bilinear pairing. a, b, c are randomly chosen from $\mathbb{Z}_p$. The assumption is that for any PPT adversary $\mathcal{A}$, with inputs $(G_0, p, g, g^a, g^b, g^c)$, it is infeasible to output $e(g, g)^{abc}$ with a non-negligible advantage, that is:

$$Adv_{\mathcal{A}} = Pr[\mathcal{A}(G_0, p, g, g^a, g^b, g^c) \rightarrow arrowe(g, g)^{abc}]. \quad (1)$$

5 The Proposed Scheme

In this section, we describe our proposed blockchain-based hierarchical access control with efficient revocation in the mHealth system. We divide the program into the following six phases: System initialization, User authorization, PHR

outsourcing, Re-encryption, PHR access, and Attribute revocation. Let's explain it in detail.

System Initialization. In this phase, TA initializes the protocol and generates the public parameters. First, TA chooses the security parameter λ as input and denotes U as the attribute universe. According to the *Definition 1*, TA chooses a finite cyclic group G_0 with order p, where g and h are two generators. Denote e as a bilinear map $e : G_0 \times G_0 \rightarrow G_T$. $\Delta_{i,S} = \prod_{j \in s, j \neq i} \frac{x-j}{i-j}$ is denoted as the Lagrange coefficient. (E, D) is denoted as a symmetric cipher, and two exponents $\alpha, \beta \in \mathbb{Z}_p$ are randomly chosen. Furthermore, two cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow G_0$, $H_2 : G_T \times G_0 \times G_T \rightarrow \{0,1\}^l$ are chosen, where l is the symmetric key length. The system parameter $Param = \{G_0, G_T, e, p, g, h = g^\beta, e(g,g)^\alpha, H_1, H_2, (E, D)\}$. And the master secret key $MSK = \{\beta, g^\alpha\}$. Then, TA generates a binary KEK tree with u leaf nodes, where u is the maximum number of users in the system. For each node in the KEK tree, TA selects a random number as the KEK, we call it KEK_i. It creates a user Revocation List (RL), initialized to empty, that records the revoked users of each attribute i. It also creates a legitimate user list $UL = G_i, (i \in U)$, initialized to empty, that records the legitimate users of each attribute i. Finally, TA uploads the system parameter $Param$ and the RL to the blockchain so that all of the entities in the system can access it. The master key MSK is held by the authority secretly for user authorization. It sends the KEK tree, UL, and all of the KEK_i to the CSP over a secure channel.

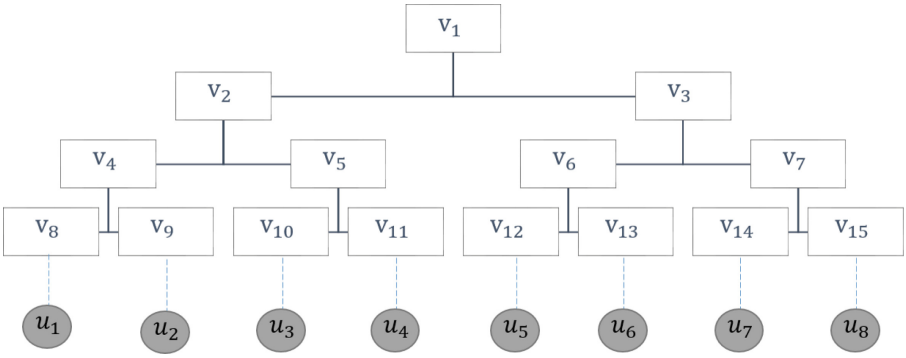


Fig. 4. hierarchical access control

User Authorization. In this phase, each user registers with TA to obtain the keys of the ABE algorithm and the KEK keys. At first, TA checks the user's attributes, adds the user to the UL, and generates the keys of the ABE algorithm as follows. Input $Param, MSK$, the set γ attributes of this user. And it chooses a random exponent: $r \in \mathbb{Z}_p$ and computes $D = g^\alpha h^r$. For each attribute $i \in \gamma$, randomly choose $r_i \in \mathbb{Z}_p$, and computes $D_i = g^r \cdot H_1(i)^{r_i}, D_i^{'} = h^{r_i}$. The

keys of ABE algorithm $SK = \{D, \{D_i, D_i^{'}\}, \forall i \in \gamma\}$. Then, TA chooses a null leaf node i for this user and assigns all KEK keys along the path from the root node to the leaf node u_i to the user. The set of KEK keys of the path is called path keys (PK_i), such as $PK_5 = \{KEK_1, KEK_3, KEK_6, KEK_{12}\}$. As shown in Fig. 4, every user is assigned to the leaf nodes of the tree $(u_1, u_2, u_3, \cdots, u_8)$. Random keys ($KEK_i$) are generated and assigned to each leaf node and internal node $(v_1, v_2, v_3, \cdots, v_{15})$.

PHR Outsourcing. PHR is encrypted before uploading to CSP. Data encryption consists of two parts: offline encryption and online encryption. Before PHR is collected or WBSN is offline, PO performs the offline encryption algorithm in advance which is combined with the algorithm in [28]. When WBSN is online, the PO receives the PHRs and performs the online encryption algorithm under the chosen access policy. The process is explained in detail below.

– **OfflineEnc**$(Param, \mathcal{T}) \rightarrow (\{C_x^{1^{'}}, C_x^2\}_{\forall x \in \mathcal{T}}, \{C_y, C_y^{'}\}_{\forall y \in \mathcal{T}})$
  - Provided with the system parameter $Param$, the PO chooses a hierarchical access tree $\mathcal{T}$ according to the related policies. A is the root node, x is a non-leaf node and y is a leaf node.
  - It generates a polynomial q_x for each node x in the tree $\mathcal{T}$ in a top-down manner starting from root node A.
  - For each node x, set its degree as $d_x = z_x - 1$, where z_x is the threshold value of x.
  - For the polynomial q_A of the root node A, we randomly choose $q_A(0) \epsilon \mathbb{Z}_p$ as well as d_A additional points satisfy q_A. Furthermore, set $\sigma_A = q_A(0)$.
  - For the other nodes x, $q_x(0)$ is computed as $q_x(0) = q_{parent(x)}(index(x))$ ($parent(x)$ is parent node of x, $index(x)$ represents that the order number of x in all children). d_x other points of polynomial q_x are chosen randomly to define it completely. Set $\sigma_x = q_x(0)$.
  - For each leaf node y, compute: $C_y = h^{q_y(0)}$,$C_y^{'} = H_1(att(y))^{q_y(0)}$.
  - Compute: $C_x^{1^{'}} = e(g,g)^{\alpha \sigma_x}$, $C_x^2 = g^{\sigma_x}$.

– **OnlineEnc** $(Param, \mathcal{T}, \{C_X^{1'}\}_{x \in \mathcal{T}}, \{M_x\}_{x \in X}) \rightarrow (\{C_X^1\}_{x \in \mathcal{T}}, \{C_X^3\}_{x \in \mathcal{T}})$
  - For each $x \in X$, choose a random: $R_x \in Z_p$. X is the set of x that will be used in the encryption.
  - Compute: $C_x^1 = R_x \cdot C_x^{1^{'}}$, $k_x = H_2(C_x^1 \parallel C_x^2 \parallel R_x)$, $C_x^3 = E_{k_x(M_X)}$.

Finally, PO sends the $CT = \{\mathcal{T}, \{C_x^1, C_x^2, C_x^3\}_{\forall x \in X}, \{C_y, C_y^{'}, \}_{\forall y \in Y}\}$ to the CSP for sharing.

Re-encryption. To achieve attribute-level user revocation, the CSP needs to re-encrypt the ciphertext so that only legitimate users can decrypt the ciphertext. It needs to generate and update the secret key for the attribute group. Using the UL, it selects the root nodes with the minimum cover sets in the KEK tree. Such a selection should cover all leaf nodes associated with the users in G_i. The group key of the attribute is denoted as $KEK(G_i)$. For example,

if $G_i = (u_1, u_3, u_7, u_8)$ in Fig. 4, then $KEK(G_i) = \{KEK_8, KEK_{10}, KEK_7\}$ because v_8, v_{10} and v_7 are the root nodes with the minimum cover sets for all members in G_i. Therefore, it covers all users in G_i. Users not in G_i have no knowledge of KEK in $KEK(G_i)$. Once CT is received from the PO, CSP first randomly chooses $\delta_i \in Z_P$, and computes $C_y^{''} = C_y^{'1/\delta_i}$. Then, a header message $Hdr = (\forall y \in Y : \{E_K(\delta_i)\}_{K \in KEK(G_y)})$ is generated, where $E_K(M)$ is an encryption of M using the key K. Finally, the CSP stores the ciphertext $CT = \{Hdr, \mathcal{T}, \{C_x^1, C_x^2, C_x^3\}_{\forall x \in X}, \{C_y, C_y^{''}, \}_{\forall y \in Y}\}$.

PHR Access. After the PU requests to obtain the ciphertext, it needs to perform two steps to recover the PHRs, decryption for the attribute group key and decryption for the message. When PU receives the ciphertext $(Hdr, CT^{'})$, he first obtains the attribute group keys for all attributes which will be used from Hdr. If he is the valid user in G_i, he has a valid KEK which can be used in the symmetric decryption. For example, if $G_i = (u_1, u_3, u_7, u_8)$ in Fig. 4, then $KEK(G_i) = \{KEK_8, KEK_{10}, KEK_7\}$. The PK of u_8 is $\{KEK_1, KEK_3, KEK_7, KEK_{15}\}$. The only common key is KEK_7, then u_8 can decrypt Hdr for the group key. The PK of u_2 is $\{KEK_1, KEK_2, KEK_4, KEK_9\}$ which can not be used for a valid key. PU can compute its decryption key with the attribute group keys as follows: $\{D = g^{\alpha} h^r, \{D_i = g^r \cdot H_1(i)^{r_i}, D_i^{'} = h^{r_i \delta_i}\}, \forall i \in \gamma\}$

Then, PU Performs the decrypt algorithm which is defined in the same way as [28] with the decryption key, and the message can be recovered as follows:

– For each leaf node y, let $i = attt(y)$, compute:

$$DecNode(y) = \frac{e(D_i, C_y)}{e(D_i^{'}, C_y^{'})} = \frac{e(g^r \cdot H_1(i)^{r_i}, h^{q_y(0)})}{e(h^{r_i \delta_i}, H_1(i)^{(1/\delta_i) \cdot q_y(0)})} = e(g,g)^{r \beta q_y(0)} \quad (2)$$

– For each non-leaf node x, compute the algorithm DecNode(x) recursively:

$$\begin{aligned} DecNode(x) &= \prod_{x^{'} \epsilon S_x} DecNode(x^{'})^{\Delta_{j, S_x^{'}}(0)} \\ &= \prod_{x^{'} \epsilon S_x} (e(g,g)^{r \beta q_{x^{'}}(0)})^{\Delta_{j, S_x^{'}}(0)} \\ &= \prod_{x^{'} \epsilon S_x} (e(g,g)^{r \beta q_{parent(x^{'})} t(indext(x^{'}))})^{\Delta_{j, S_x^{'}}(0)} \\ &= \prod_{x^{'} \epsilon S_x} e(g,g)^{r \beta q_x(j) \cdot \Delta_{j, S_x^{'}}(0)} \\ &= e(g,g)^{r \beta q_x(0)} \end{aligned} \quad (3)$$

– Compute:

$$\frac{C_x^1}{e(C_x^2, D) / DecNode(x)} = \frac{R_x \cdot e(g,g)^{\alpha \sigma_x}}{e(g,g)^{\alpha \sigma_x}} = R_x \quad (4)$$

– Compute:

$$k_x = H_2(C_x^1 \parallel C_x^2 \parallel R_x), M_x = D_{k_x}(C_x^3) \tag{5}$$

Finally, the download of PU is recorded on the blockchain as a transaction record for review by other parties.

Attribute Revocation. If the list of revoked RL is changed, somebody leaving or joining, and TA needs to upload the latest RL to the blockchain. The CSP can perform as follows for CT updating and key updating.

– It randomly selects $s^{'} \in Z_p$, and a different group key $\delta_i^{'}$.
– The ciphertext CT is re-encrypted. For $\mathcal{T}, \forall x \in X$, it computes $C_x^1 = R_x \cdot e(g,g)^{\alpha\sigma_x + s^{'}}, C_x^2 = g^{\sigma_x + s^{'}}, C_x^3 = E_{k_x(M_X)}$, $C_i = h^{q_i(0)+s^{'}}$, $C_i^{'} = (H_1(att(y))^{q_i(0)+s^{'}})^{\frac{1}{\delta_i^{'}}}$. For $\mathcal{T}, \forall y \in Y/\{i\}$, it computes $C_y = h^{q_y(0)+s^{'}}, C_y^{'} = (H_1(att(y))^{q_y(0)+s^{'}})^{1/\delta_i}$.
– It selects new minimum cover sets for G_i, and sets the group key is $KEK(G_i)$. And it computes $Hdr = (\{E_K(\delta_i^{'})\}_{K \in KEK(G_i)}, \forall y \in Y/\{i\} : \{E_K(\delta_i)\}_{K \in KEK(G_y)})$.

6 Security Analyses

In this section, we prove that the proposed scheme satisfies the desired security requirements, based on the CBDH assumption.

Theorem 1 *Our proposed scheme is IND-CCA secure in the random oracle model based on the CBDH assumption and assuming the symmetric cipher (E, D) is IND-CCA secure. (Note that the protocol itself is only CPA secure, but using Fujisaki-Okamoto conversion, it can be made CCA secure.)*

Proof. Recall that $\mathcal{A}$ is a PPT adversary, who makes q_k key queries and q_d decryption Queries in time t, can win the IND-CCA game in our scheme with an advantage ϵ. B is a simulator that is used to solve the CBDH problem, and it controls H_1 and H_2. With the given instance (g, g^a, g^b, g^c), they interact with each other as follows:

– **Setup**. B performs the simulated system initialization algorithm to generate a set of parameters. According to the instance, it first chooses a random $\alpha^{'} \in Z_p$, and sets $\alpha = \alpha^{'} + ab$, $\beta = b$, computing $e(g,g)^{\alpha} = e(g,g)^{\alpha^{'}+ab} = e(g,g)^{\alpha^{'}} e(g,g)^{ab}$. The public parameter $Param = \{G_0, G_T, e, H_1, H_2, g, h = g^b, e(g,g)^{\alpha^{'}} e(g,g)^{ab}, (E,D)\}$. It constructs a KEK tree with k leaf nodes which means that at most k different users can be asked for their keys.
H_1 queries: For each H_1 query of attribute i, if there is a i to make $H_1(i) = s_i$ in the H_1 list, it returns the corresponding value; otherwise, it chooses a random $s \in G_0$ as the return value and adds the $(i, s_i = s)$ to the list.
H_2 queries: For each H_2 query, if there is a (m, v_m) in the H_2 list, it returns the corresponding value v_m, otherwise it choose a random value v_m from $\{0,1\}^l$ as the result and add the (m, v_m) to the list.

– **Phase 1**. In this phase, the adversary $\mathcal{A}$ performs two kinds of queries as follows:
Key Queries: $\mathcal{A}$ makes key queries for user with attribute set I in an adaptive way. B randomly chooses $r^{'} \in Z_p$ and sets $r = r^{'} - a$, computing: $D = g^{\alpha}h^{r} = g^{\alpha^{'}+ab}g^{b(r^{'}-a)} = g^{\alpha^{'}+br^{'}}$, $D_i = \frac{g^{r^{'}}}{g^{a}} \cdot H_1(i)^{r_i}$. It chooses path keys from the KEK tree as PK for the user. Note that the PK was not distributed to others and the different user has different PK. Add the user to the user list. The decryption keys $SK = \{D = g^{\alpha^{'}+br^{'}}, \forall i \in I : D_i = \frac{g^{r^{'}}}{g^{a}} \cdot H_1(i)^{r_i}, D_i^{'} = h^{r_i}\}$, $PK = \{KEK(i)\}_{i \in path}$.

Decryption Queries: $\mathcal{A}$ requests the decryption query of the ciphertext $CT = \{\mathcal{T}, C^1, C^2, C^3, C_y, C_y^{'}, Hdr\}$. B first decrypts the Hdr with the KEK_i in the tree, if there is no δ_i that can be found, it outputs $\perp$. Then, B searches hash list H_1, H_2 for the value (i, s_i) and (m, v_m). It extracts R from $v_i = C^1 \parallel C^2 \parallel R$ and checks if the pairs (i, s_i), and (m, v_m) satisfy the following equations:$K = V_i = H_2(C^1 \parallel C^2 \parallel R)$, $M = D_K(C^3)$, $C^1 = R \cdot e(g, g)^{\alpha^{'}\sigma} e(g, g)^{ab\sigma}$, $C^2 = g^{\sigma}$, $C_y^{'} = s_i^{q_y(0)\frac{1}{\delta_i}}$. If there exist hash queries satisfying the equations:

$$DecNode(y) = \frac{e(D_i, C_y)}{e(D_i^{'}, C_y^{'})} = \frac{e(\frac{g^{r^{'}}}{g^{a}} \cdot H_1(i)^{r_i}, h^{q_y(0)})}{e(h^{r_i\delta_i}, s_i^{q_y(0)\frac{1}{\delta_i}})} = (\frac{e(g, g)^{r^{'}b}}{e(g, g)^{ab}})^{q_y(0)} \quad (6)$$

$$DecNode(x) = (\frac{e(g, g)^{r^{'}b}}{e(g, g)^{ab}})^{\sigma} \quad (7)$$

B returns the result to $\mathcal{A}$, otherwise it outputs $\perp$.

– **Challenge.** $\mathcal{A}$ chooses a challenge access structure $\mathcal{T}^*$ for the the user u_i. It selects two messages M_0, M_1 with the same length. For $\forall y \in Y$, it chooses a random number $\delta_i \in Z_p$, and computes the Hdr that $Hdr = (\forall y \in Y : \{E_K(\delta_i)\}_{K \in KEK(G_y)})$, where $EKE(G_j)$ is not available for u_i. Then, B randomly chooses $R_1 \in G_T$, $R_3, R_4 \in G_0$, and R_2 from the symmetric cipher's ciphertext space. Then, $CT^* = \{\mathcal{T}^*, R_1, g^c, R_2, R_3, R_4\}$ is computed. If $\mathcal{A}$ does not query $att(y)$, with the random number σ^*, it is the correct CT^* which can be seen as $R_1 = R \cdot e(g, g)^{\alpha^{'}c} e(g, g)^{abc}$, $R_2 = E_K(M_\theta)$, $R_3 = h^{q_y(0)}$, $R_4 = H_1(att(y))^{q_y(0)\delta_i}$.
– **Phase 2.** In this phase, $\mathcal{A}$ can make queries in the same way as in phase 1 with two exceptions: First, it cannot query the access structure that matches T^* in the key query. And second, it cannot query the challenge ciphertext CT^* in the decryption query.
– **Guess.** $\mathcal{A}$ guesses the bit θ or outputs $\perp$.

Discussion: For all H_2 queries, the simulator chooses a random one as the challenge hash query which is defined as $H_2^* = C^1 \parallel C^2 \parallel R = R \cdot e(g, g)^{\alpha^{'}c} e(g, g)^{abc} \parallel g^c \parallel R$. In this way, if the H_2^* is used for the challenge by $\mathcal{A}$, the simulator can

use it to solve the CBDH problem. In the above game, the decryption simulation is correct except with a negligible probability:

- According to the description in *Phase 1*, B can answer the decryption query on $CT = \{\mathcal{T}, C^1, C^2, C^3, C_y, C_y^{'}, Hdr\}$.
- If there is a required CT which satisfy $C^2 = g^c$ (g^c is the challenge ciphertext). B must return $\perp$ to $\mathcal{A}$. Because B can determine the corresponding ciphertext, and the CT which is different from CT^* is an invalid ciphertext.
- If $C^2 \not\equiv g^c$, but $\mathcal{A}$ has queried (i, s_i) and (m, v_m) which the challenge M_θ can be satisfied as following:

$$K = V_i = H_2(C^1 \parallel C^2 \parallel R), M_\theta = D_K(C^3),$$
$$C^1 = R \cdot e(g,g)^{\alpha^{'}\sigma} e(g,g)^{ab\sigma}, C^2 = g^\sigma, C_y^{'} = s_i^{q_y(0)\frac{1}{\delta_i}}$$
$$DecNode(y) = \frac{e(D_i, C_y)}{e(D_i^{'}, C_y^{'})} = (\frac{e(g,g)^{r^{'}b}}{e(g,g)^{ab}})^{q_y(0)}, DecNode(x) = (\frac{e(g,g)^{r^{'}b}}{e(g,g)^{ab}})^{\sigma}$$

In this situation, B returns the corresponding result to $\mathcal{A}$. However, it has a negligible probability to hold this kind of equation, because s_i and v_m are selected randomly. Furthermore, $(a, b, r, r_i, R, s_i, v_m, R_1, R_2, R_3, R_4)$ are chosen randomly in the simulation game so that the simulation is indistinguishable in the view of adversary $\mathcal{A}$.

7 Efficiency Analyses

In this part, we analyze the performance of our RH-ABE with related works in this field, TRAC in [17], data sharing scheme in [24], TR-AP-CPABE in [10].

Features. As shown in Table 1, We compare their features in seven dimensions: Access Model(AC), Hierarchical Access Control(HAC), Offline/Online Encryption(Off/On En), Traceable(Tra), User Revocation(UR), Attribute-level Revocation(Attr UR), Security Level(SL). Li et al. [17] proposed a traceable and revocable access control scheme for mHealth which is an And gate access structure scheme with IND-CPA security. Tan et al. [24] proposed a data-sharing scheme for COVID-19 Medical Records. It is a blockchain-empowered approach that can achieve malicious user traceability and revocation. It is also IND-CPA secure. Han et al. [10] proposed a traceable and revocable ABE scheme with a hidden policy. Both of them are user revocation schemes with IND-CPA, and they can not achieve attribute-level revocation. In our scheme, we proposed a hierarchical access control scheme with offline/online encryption and attribute-level user revocation which can achieve IND-CCA security. In terms of features, our scheme is more suitable for mobile health systems.

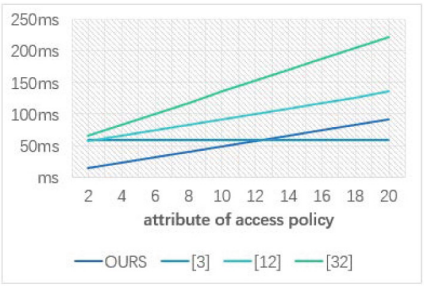
7.1 Computation Costs

In this part, we ignore the computational costs of operations in symmetric cipher as well as hash functions, as they are much lighter compared with the costs in

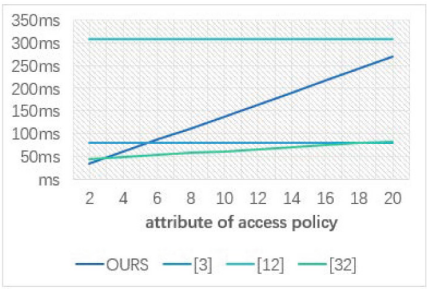
Table 1. Comparison of Features.

Work	AM	HAC	Off/On En	Tra	UR	Attr UR	SL
[17]	And gate	No	No	Yes	Yes	No	$IND-CPA$
[24]	Tree	No	No	Yes	Yes	No	$IND-CPA$
[10]	Tree	No	No	Yes	Yes	No	$IND-CPA$
Ours	Tree	Yes	Yes	No	Yes	Yes	$IND-CPA$

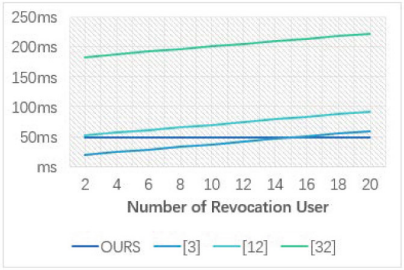
attribute-based encryption. Table 2 shows the number of exponential and paring operations in the four schemes. E represents an exponential operation and P represents a bilinear pair operation. $|U|$ is the number of users in the system. $|I|$ is the number of attributes in the system. $|R|$ is the max number of revocation user and $|S|$ represents a user's attribute number. r is the number of attributes in the access policy.



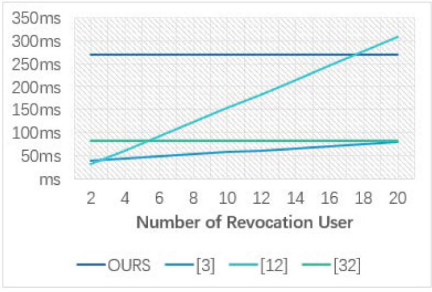
(a) Encryption ($|R|$=20)



(b) Decryption ($|R|$=20)



(c) Encryption ($|r|$=20)



(d) Decryption ($|r|$=20)

Fig. 5. Comparisons of Computation Cost

In the *Setup* phase, the computation costs in [17] and [10] are linear to the number of users, but they are constant in [24] and our scheme. The computation

Table 2. Comparison of Computation cost.

	Setup	KeyGen	Encrypt	Decrypt
[17]	$(4\|U\|+2)E+2P$	$(3+2\|I\|)E$	$(7+\|R\|)E$	$(4+\|R\|)E+4P$
[24]	$2E+P$	$(3+\|S\|)E$	$(2+r+\|R\|)E$	$\|R\|E+2\|R\|P$
[10]	$2\|U\|E+P$	$(4+2\|S\|)E$	$(2+4r+\|R\|)E$	$(3+r)E+5P$
Ours	$2E+P$	$(2+2\|S\|)E$	$(2X+2r)E$	$(1+2r)P$

cost of *KeyGen* in [17] is linear to the number of system attributes, but it relates to the number of user's attributes in [10,24] and our scheme. Considering that the user's attribute set is generally smaller than the system attribute set. The computation cost of encryption in [17] is related to the number of revocation users. The larger the user revocation list, is more computationally expensive. In our scheme, The encryption costs are related to the access structure. The larger the access policy, the higher computational cost in our scheme. In general, the size of the access policy does not change regularly, but the number of revoked users may increase. In [10,24], the encryption costs are related to the size of $|R|$ and r. The decryption costs in [17] and [24] are only related to the $|R|$, while the work in [10] and our scheme are related to r.

In Fig. 5, we graphically depict how the computational cost varies with the size of $|R|$ and r. The time of a pairing is 6.59 ms, and the time of exponentiation is 2.18 ms. In Fig. 5 (a) and (b), we assumed the number of revocation users is 20, and the computation cost of encryption and decryption varies with the number of attributes of the access policy. In Fig. 5(c) and (d), we assumed the size of r is 20, and the computation cost of encryption and decryption varies with the number of revocation users. In both cases, the computation encryption cost of our scheme is relatively lower. The decryption efficiency is not very good, since the pairing operation is related to the size of the access structure. But, as the number of revoked users in the system increases, the cost of other schemes will become larger and larger.

7.2 Communication Costs

As shown in Table 3, we compare the communication cost of each scheme from three aspects, size of the public parameter, secret key, and ciphertext.

Table 3. Comparison of Communication Cost.

	Size of Public Parameter	Size of Secret Key	Size of Ciphertext
[17]	$(3\|I\|+2\|U\|+2)G_0+G_T$	$(3+2\|I\|)G_0$	$(5+\|R\|)G_0+G_T$
[24]	$2G_0+G_T$	$(3+\|S\|)G_0$	$(1+r+\|R\|)G_0+G_T$
[10]	$(3+2\|U\|)G_0+G_T$	$(3+\|S\|)G_0$	$(2+3r+\|R\|)G_0+G_T$
Ours	$2G_0+G_T$	$(1+2\|S\|)E$	$(1+2r)G_0+G_T$

In the scheme [17], the size of the public parameter is related to the number of system users and system attributes. And public parameter in [10] is related to the number of system users. Both of them are more than $2G_0 + G_T$. The size of the secret key in [17] is related to the number of system attributes while others are related to the user's attributes. And communication cost of ciphertext is related to the size of revoked user list except for our scheme. Totally, the communication cost in our scheme is the least.

8 Conclusion

In this paper, a blockchain-based hierarchical access control with efficient revocation is proposed for the mHealth system. Our scheme is based on the file-related hierarchical access control scheme, we use the KEK technique for attribute-lever user revocation. PO can encrypt a series of data with hierarchical access control relationships at the same time to improve efficiency which is suitable for a resource-constrained environment. We also use the offline/online encryption model, since the access policy can be chosen in advance. Fine-grained user revocation is desirable because users' roles may change regularly. It is also proved that our scheme is IND-CCA secure. The comparison with some related works demonstrates that RH-ABE is generally efficient for the mHealth system.

Note that the pairing operation of decryption grows linearly with the scale of access policy, in our scheme. There are many schemes that outsource decryption computing to cloud providers. But some of them do not use hierarchical control, and others do not provide attribute-level user revocation. It makes sense to build solutions that are efficient in all aspects.

References

1. Bao, Y., Qiu, W., Tang, P., Cheng, X.: Efficient, revocable, and privacy-preserving fine-grained data sharing with keyword search for the cloud-assisted medical iot system. IEEE J. Biomed. Health Inform. **26**(5), 2041–2051 (2021)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP'07), pp. 321–334. IEEE (2007)
3. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
4. Edemacu, K., Jang, B., Kim, J.W.: Collaborative ehealth privacy and security: An access control with attribute revocation based on OBDD access structure. IEEE J. Biomed. Health Inform. **24**(10), 2960–2972 (2020)
5. Fu, J., Wang, N.: A practical attribute-based document collection hierarchical encryption scheme in cloud computing. IEEE Access **7**, 36218–36232 (2019)
6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: 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 (2006)
7. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of {ABE} ciphertexts. In: 20th USENIX Security Symposium (USENIX Security 11) (2011)

8. Guo, R., Yang, G., Shi, H., Zhang, Y., Zheng, D.: O 3-R-CP-ABE: an efficient and revocable attribute-based encryption scheme in the cloud-assisted iomt system. IEEE Internet Things J. **8**(11), 8949–8963 (2021)
9. Hajar, M.S., Al-Kadri, M.O., Kalutarage, H.K.: A survey on wireless body area networks: architecture, security challenges and research opportunities. Comput. Secur. **104**, 102211 (2021)
10. Han, D., Pan, N., Li, K.C.: A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. IEEE Transactions on Dependable and Secure Computing (2020)
11. Hur, J., Noh, D.K.: Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans. Parallel Distrib. Syst. **22**(7), 1214–1221 (2010)
12. Lee, K., Choi, S.G., Lee, D.H., Park, J.H., Yung, M.: Self-updatable encryption: time constrained access control with hidden attributes and better efficiency. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013, pp. 235–254. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_13
13. Li, H., Yu, K., Liu, B., Feng, C., Qin, Z., Srivastava, G.: An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things. IEEE J. Biomed. Health Inform. **26**(5), 1949–1960 (2021)
14. Li, J., Yu, Q., Zhang, Y.: Hierarchical attribute based encryption with continuous leakage-resilience. Inf. Sci. **484**, 113–134 (2019)
15. Li, M., Lou, W., Ren, K.: Data security and privacy in wireless body area networks. IEEE Wirel. Commun. **17**(1), 51–58 (2010)
16. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. **24**(1), 131–143 (2012)
17. Li, Q., Xia, B., Huang, H., Zhang, Y., Zhang, T.: Trac: traceable and revocable access control scheme for mhealth in 5G-enabled iiot. IEEE Trans. Industr. Inf. **18**(5), 3437–3448 (2021)
18. Liang, X., Lu, R., Lin, X., Shen, X.S.: Ciphertext policy attribute based encryption with efficient revocation. TechnicalReport, University of Waterloo **2**(8) (2010)
19. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) Applied Cryptography and Network Security: 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings, pp. 111–129. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_7
20. Riad, K., Huang, T., Ke, L.: A dynamic and hierarchical access control for Iot in multi-authority cloud storage. J. Netw. Comput. Appl. **160**, 102633 (2020)
21. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
22. Shen, J., Gui, Z., Chen, X., Zhang, J., Xiang, Y.: Lightweight and certificateless multi-receiver secure data transmission protocol for wireless body area networks. IEEE Transactions on Dependable and Secure Computing (2020)
23. Sun, J., Xiong, H., Liu, X., Zhang, Y., Nie, X., Deng, R.H.: Lightweight and privacy-aware fine-grained access control for Iot-oriented smart health. IEEE Internet Things J. **7**(7), 6566–6575 (2020)
24. Tan, L., Yu, K., Shi, N., Yang, C., Wei, W., Lu, H.: Towards secure and privacy-preserving data sharing for Covid-19 medical records: a blockchain-empowered approach. IEEE Trans. Netw. Sci. Eng. **9**(1), 271–281 (2021)

25. Tang, W., Zhang, K., Ren, J., Zhang, Y., Shen, X.: Lightweight and privacy-preserving fog-assisted information sharing scheme for health big data. In: GLOBECOM 2017-2017 IEEE Global Communications Conference, pp. 1–6. IEEE (2017)
26. Wang, S., et al.: A fast CP-ABE system for cyber-physical security and privacy in mobile healthcare network. IEEE Trans. Ind. Appl. **56**(4), 4467–4477 (2020)
27. Wang, S., Zhou, J., Liu, J.K., Yu, J., Chen, J., Xie, W.: An efficient file hierarchy attribute-based encryption scheme in cloud computing. IEEE Trans. Inf. Forensics Secur. **11**(6), 1265–1277 (2016)
28. Xiao, M., Li, H., Huang, Q., Yu, S., Susilo, W.: Attribute-based hierarchical access control with extendable policy. IEEE Transactions on Information Forensics and Security (2022)
29. Xu, S., Li, Y., Deng, R., Zhang, Y., Luo, X., Liu, X.: Lightweight and expressive fine-grained access control for healthcare internet-of-things. IEEE Transactions on Cloud Computing (2019)
30. Yang, Y., Liu, X., Deng, R.H., Li, Y.: Lightweight sharable and traceable secure mobile health system. IEEE Trans. Dependable Secure Comput. **17**(1), 78–91 (2017)
31. Zhang, L., Zhao, C., Wu, Q., Mu, Y., Rezaeibagha, F.: A traceable and revocable multi-authority access control scheme with privacy preserving for mhealth. J. Syst. Architect. **130**, 102654 (2022)
32. Zhong, H., Zhou, Y., Zhang, Q., Xu, Y., Cui, J.: An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare. Futur. Gener. Comput. Syst. **115**, 486–496 (2021)