# A Secure EMR Sharing System With Tamper Resistance and Expressive Access Control

Shengmin Xu, Jianting Ning, Yingjiu Li, Yinghui Zhang, Guowen Xu, Xinyi Huang, and Robert H. Deng, *Fellow, IEEE*

**Abstract**—To reduce the cost of human and material resources and improve the collaborations among medical systems, research laboratories and insurance companies for healthcare researches and commercial activities, electronic medical records (EMRs) have been proposed to shift from paperwork to friendly shareable electronic records. To take advantage of EMRs efficiently and reduce the cost of local storage, EMRs are usually outsourced to the remote cloud for sharing medical data with authorized users. However, cloud service providers are untrustworthy. In this paper, we propose an efficient, secure, and flexible EMR sharing system by introducing a novel cryptosystem called dual-policy revocable attribute-based encryption and tamper resistance blockchain technology. Our proposed system enables EMRs to be shared at a fine-grained level and allows data users to detect any unauthorized manipulation. Moreover, the key generation center can revoke malicious users without affecting the honest users. We provide the formal security model as well as the concrete scheme with security analysis. The experimental simulation and experimental analysis of our proposed scheme demonstrate that our proposed system has superior performances to the most relevant solutions.

**Index Terms**—Dual-polocy attribute-based encryption, user revocation, blockchain, electronic medical records

## 1 Introduction

Electronic medical records (EMRs) have been widely used in the current medical systems to improve collaborations among medical staffs, patients, and even research laboratories as well as insurance companies for sharing patients' information [2], [23]. EMRs not only convenience to technology exchange and medical case study for patients, their relatives, and medical researchers, but also bring a significant improvement in practices of healthcare professionals and a large number of benefits to the current medical industry. According to the report from SlectHub [41], EMRs adoption rates reach at around 87% in 2019. A report from the University of Michigan [9] also pointed out that the cost of outpatient care is reduced by 3% in savings per patient each month by changing from paper documents to EMRs. The foundation of EMRs is to record and maintain medical documents, including lab tests, images, and prescriptions, to provide flexible data sharing to medical staffs, patients and insurance companies.

Many EMRs have been outsourced to the cloud system, such as Amazon EMR [7], AdvancedMD [1], and DrChrono EHR [19]. Fig. 1 presents a basic model of the cloud-based EMR system. Typical entities are (1) data owners who collect medical data from various devices; (2) a remote cloud to store EMRs from data owners; and (3) data users fetch EMRs from the remote cloud. Such a cloud-based EMR sharing system allows convenient data management and reduces the cost of local storage. However, it also faces many security issues, such as availability, accessibility, and standardization.

*Tamper resistance*. EMRs usually contain sensitive personal data that should be immutable. The alteration and falsification of medical records are considered crimes in most countries, according to the general data protection regulation (GDPR) in the EU, the personal data protection act (PDPA) in Singapore, and the health insurance portability accountability act (HIPPAA) in the US. However, the integrity of cloud-based EMRs is vulnerable since EMR providers can upload a modified version to replace the original one and

- *Shengmin Xu is with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian 350117, China, and also with the Secure Mobile Center, Singapore Management University, Singapore 188065, Singapore. E-mail: smxu1989@gmail.com.*
- *Jianting Ning is with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian 350117, China, and also with the State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, Chian. E-mail: jtning88@gmail.com.*
- *Yingjiu Li is with the Computer and Information Science Department, University of Oregon, Eugene, OR 97403 USA. E-mail: yingjiul@uoregon.edu.*
- *Yinghui Zhang is with the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi 710121, China. E-mail: yhzhaang@163.com.*
- *Guowen Xu is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore. E-mail: guowen.xu@ntu.edu.sg.*
- *Xinyi Huang is with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian 350117, China. E-mail: xyhuang81@gmail.com.*
- *Robert H. Deng is with the School of Computing and Information Systems, Singapore Management University, Singapore 188065, Singapore. E-mail: robertdeng@smu.edu.sg.*

*Manuscript received 15 Dec. 2019; revised 21 Sept. 2021; accepted 4 Nov. 2021. Date of publication 9 Nov. 2021; date of current version 16 Jan. 2023.*
*This work was supported in part by the Natural Science Foundation of China under Grants 62102090, 62072369, 62032005, 61972094, and 61872089, in part by the Science Foundation of Fujian Provincial Science and Technology Agency under Grant 2020J02016, and in part by the Innovation Capability Support Program of Shaanxi under Grant 2020KJXX-052. Jianting Ning's work was supported by the young talent promotion project of Fujian Science and Technology Association. Yingjiu Li's work was supported by the Ripple University Blockchain Research Initiative. Yinghui Zhang's work was supported by the Shaanxi Special Support Program Youth Top-notch Talent Program.*
*(Corresponding author: Jianting Ning.)*
*Digital Object Identifier no. 10.1109/TDSC.2021.3126532*

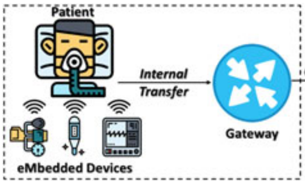


Fig. 1. Basic Cloud-Based EMR Data Sharing Model.

untrustworthy cloud service providers (CSPs) physically control the outsourced EMRs. When a medical mistake and any medical malpractice case occur, doctors and related healthcare providers have great temptations to change the medical records for preventing the exposure of medical malpractices and maintaining their reputations [20]. CSPs also have strong motivations to modify the outsourced EMRs for various reasons. For example, CSPs may delete merely used data to save the cost of data storage, and even collude with EMR providers to cover medical malpractices. Determining whether the EMR is modified or not is one of the significant challenges in current EMR sharing systems. Fortunately, blockchain, as a growing list of records with immutability, has been widely used to ensure data integrity [29]. However, it is inadequate to directly apply blockchain in the EMR sharing system due to the lack of data confidentiality [16].

*EMRs Confidentiality*. Data confidentiality is one of the most foundational requirements in EMR sharing systems, especially preventing untrustworthy CSPs and malicious users from learning any sensitive information. EMRs are outsourced to CSPs, and the medical center cannot physically manage these outsourced EMRs. To achieve data confidentiality, many cryptographic tools have been applied to build secure data-sharing mechanisms. However, many encryption mechanisms only provide access control with a coarse-grained level which is unscalable and not suitable to the cloud environment.

*Expressive Access Control*. To realize fine-grained access control over encrypted data, attribute-based encryption (ABE) [40] has been widely applied in various EMR sharing systems [31], [32], [44], [51], [52], [54], [55]. There are two primary flavors of ABE: Key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE [22], users' secret keys are depended on access trees (as access structures “*Name* **OR**(*Weight* **AND***Sex*)” and “(*Name* **AND**#*ID*) **OR***Disease*” in Fig. 2), and ciphertexts are encrypted over an attribute sets. Hence, KP-ABE provides content-based access control, where medical records are encrypted with a set of attributes, such as “*Name*”, “*Weight*”, “*Sex*” and “*#ID*”, to summarize the content of the record. In CP-ABE [10], access trees are specified for ciphertexts, and secret keys are associated with a set of attributes. Hence, CP-ABE focuses on role-based access control, where medical records are encrypted with an access policy (e.g., “*123-45-678* **OR** (*Alice Hospital* **AND***Cardiologist*)”) describing who is allowed to access the EMR. Therefore, KP-ABE conveniences to research laboratories for medical researchers and insurance companies to analyze and expect business development, and CP-ABE is easy to help medical staffs and patients to record and analyze the healthcare information. However, KP-ABE and CP-ABE cannot offer role-based and content-based access control simultaneously. To address this issue, dual-policy ABE (DP-ABE) [5], [46] can be applied, which provides content-based and role-based access control simultaneously. Although DP-ABE provides more flexible access control than KP-ABE and CP-ABE, straightforward applying DP-ABE in the cloud-based environment still suffers many threats.

*Dynamic User Groups*. Dynamic user groups are a challenging problem in the cloud-based EMR sharing system. In the real world, commercial EMR systems usually stop the data accessibility when users fail to renew their memberships and change of positions (e.g., promotion or retirement), whose decryption privileges should be revoked. Hence, how to manage the revoked users is an essential issue in many cloud-based EMRs systems. There are two strategies to manage user revocation: instant revocation and indirect revocation. In instant revocation [4], users are revoked by a fully trusted party immediately once their credentials are no longer valid. However, this method is impractical since requiring a fully trusted party always online to issue the latest revocation list when the user revocation happens, and the data owners have to keep the revocation list up to date. In indirect revocation [11], the validation of data users is based on a negotiated revocation epoch (e.g., one hour/day/month). Data owners only need to know the negotiated revocation epoch rather than keeping the revocation list up to date. Existing DP-ABE solutions [5], [6], [46] provide neither instant revocation nor indirect revocation. Therefore, how to design a secure, efficient, and flexible EMR sharing system with user revocation in the cloud is a challenging problem and should be urgently solved.

### 1.1 Related Work

Some EMR sharing systems [44], [46], [49], [51], [52], [53], [55] and cloud-based systems [8], [27], [33], [50] have been proposed to solve some of the above problems. We give a comparison between them and ours, as shown in Table 1.

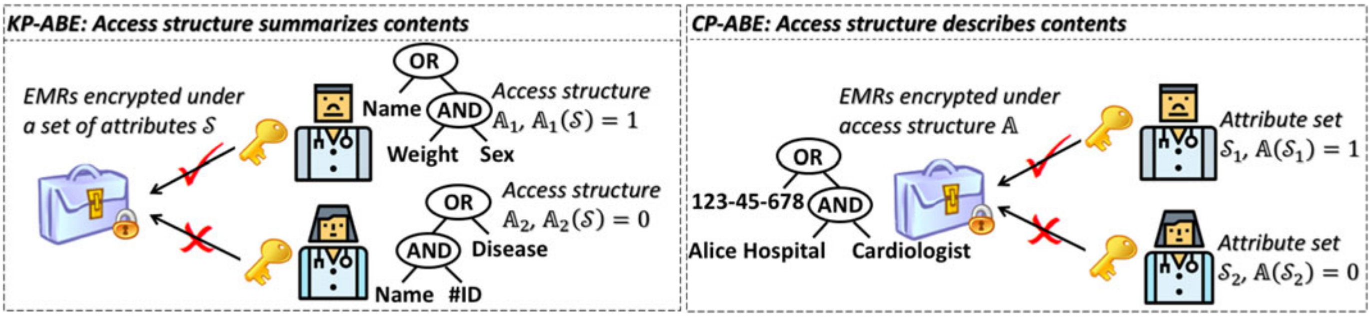

KP-ABE: Access structure summarizes contents
EMRs encrypted under a set of attributes $\mathcal{S}$
OR
Name
AND
Weight
Sex
Access structure $\mathbb{A}_1$, $\mathbb{A}_1(\mathcal{S}) = 1$
OR
AND
Disease
Name
#ID
Access structure $\mathbb{A}_2$, $\mathbb{A}_2(\mathcal{S}) = 0$
CP-ABE: Access structure describes contents
EMRs encrypted under access structure $\mathbb{A}$
OR
123-45-678
AND
Alice Hospital
Cardiologist
Attribute set $\mathcal{S}_1$, $\mathbb{A}(\mathcal{S}_1) = 1$
Attribute set $\mathcal{S}_2$, $\mathbb{A}(\mathcal{S}_2) = 0$

Fig. 2. Examples of KP-ABE and CP-ABE in EMR Sharing System.

TABLE 1
Comparison Among Existing Electronic Healthcare Medical and Cloud-Based Systems

| | Fine-Grained Control | Manipulation Detection | Dynamic User Group | Revocation Complexity | Forward and Backward Secrecy | Ciphertext Update | Security Model |
|---|---|---|---|---|---|---|---|
| *[44]* | × | × | ✓ | *Linear* | × | *N/A* | *Random Oracle* |
| *[51]* | × | × | × | *N/A* | *N/A* | *N/A* | *Random Oracle* |
| *[53]* | × | × | ✓ | *Linear* | × | *Re-Encryption* | *Random Oracle* |
| *[55]* | × | × | ✓ | *N/A* | × | *N/A* | *Random Oracle* |
| *[52]* | × | × | ✓ | *Linear* | × | *Re-Encryption* | *Random Oracle* |
| *[49]* | × | × | ✓ | ✓ | ✓ | *Public Update* | *Standard Model* |
| *[46]* | ✓ | ✓ | × | *N/A* | *N/A* | *N/A* | *Standard Model* |
| *[33]* | × | × | × | *N/A* | × | *N/A* | *Random Oracle* |
| *[27]* | × | × | ✓ | *N/A* | × | *Re-Encryption* | *Random Oracle* |
| *[8]* | × | × | ✓ | *N/A* | × | *SGX* | *Random Oracle* |
| *[50]* | ✓ | × | ✓ | *Linear* | × | *N/A* | *Standard Model* |
| **Ours** | ✓ | ✓ | ✓ | *Logarithmic* | ✓ | *Public Update* | *Standard Model* |

"Fine-Grained Control" *refers to role-based and content-based access control simultaneously.*
"Revocation Complexity" *means the overhead for generating and distributing secret key or key-updating material to all the non-revoked data users.*
"✓" *means the corresponding property is achieved.*
"×" *means the corresponding property is not achieved.*
"N/A" *means not applicable.*

Wan *et al.* [44] proposed a variant ABE with user revocation in the random oracle model, but it requires an extra attribute to record expiration time and re-encryption to ensure data confidentiality against revoked users, which incurs linear revocation complexity since the key generation center needs to distribute an updated secret key to non-revoked users in each revocation epoch. Yang *et al.* [51] combined symmetric key encryption and asymmetric key encryption to introduce an EMR sharing system without user revocation. Yeh *et al.* [53] applied the standard ABE to realize user revocation based on the Merkle hash tree and also considered data confidentiality against revoked users. However, they applied the ciphertext re-encryption approach for protecting data confidentiality to prevent revoked users, which leads to a significant overhead of data owners who has to re-encrypt data in each revocation epoch. Zhang *et al.* [55] proposed an EMR sharing system with user revocation and the constant-size system parameter based on hash functions, which is simulated in the random oracle model. However, the random oracle is an ideal model which is nonexistent in the real world, and the efficiency of user revocation is proportional to the number of users rather than logarithmic compared to [53] and ours. To reduce the cost in the data sharing phase, Yang *et al.* [52] applied server-aided revocable ABE to build an EMR sharing system in the random model, where the CSP controls the user revocation. However, the CSP cannot be fully trusted and suffers from a variety of attacks. Besides, the non-revoked data user cannot decrypt the ciphertext without assistance from the third party since the data user only has partial decryption ability. Xu *et al.* [49] considered the dynamic user revocation with logarithmic complexity, but it only has role-based access control. Xu *et al.* [46] proposed a lightweight and expressive access control ABE in the standard model with role-based and access-based simultaneously, while the user revocation is not considered and the method to detect data manipulation is weak. They applied the collision-resistance hash function to detect data manipulation, which only allows the data users to know the data is compromised rather than finding the manipulator. Wan

*et al.* [44] proposed a variant ABE with user revocation in the random oracle model, but it requires an extra attribute to record expiration time and re-encryption to ensure data confidentiality against revoked users, which incurs linear revocation complexity since the key generation center needs to distribute an updated secret key to non-revoked users in each revocation epoch. Yang *et al.* [51] combined symmetric key encryption and asymmetric key encryption to introduce an EMR sharing system without user revocation. Yeh *et al.* [53] applied the standard ABE to realize user revocation based on the Merkle hash tree and also considered data confidentiality against revoked users. However, they applied the ciphertext re-encryption approach for protecting data confidentiality to prevent revoked users, which leads to a significant overhead of data owners who has to re-encrypt data in each revocation epoch. Zhang *et al.* [55] proposed an EMR sharing system with user revocation and the constant-size system parameter based on hash functions, which is simulated in the random oracle model. However, the random oracle is an ideal model which is nonexistent in the real world, and the efficiency of user revocation is proportional to the number of users rather than logarithmic compared to [53] and ours. To reduce the cost in the data sharing phase, Yang *et al.* [52] applied server-aided revocable ABE to build an EMR sharing system in the random model, where the CSP controls the user revocation. However, the CSP cannot be fully trusted and suffers from a variety of attacks. Besides, the non-revoked data user cannot decrypt the ciphertext without assistance from the third party since the data user only has partial decryption ability. Xu *et al.* [49] considered the dynamic user revocation with logarithmic complexity, but it only has role-based access control. Xu *et al.* [46] proposed a lightweight and expressive access control ABE in the standard model with role-based and access-based simultaneously, while the user revocation is not considered and the method to detect data manipulation is weak. They applied the collision-resistance hash function to detect data manipulation, which only allows the data users to know the data is compromised rather than finding the manipulator.

Ning *et al.* [33] introduced a cloud-based sharing system by applying ABE and searchable encryption. They focus on the performance of keyword search and privacy of the searching pattern during the searching phase, and the property of user revocation has not been taken into consideration. Michalas [27] additionally offers the user revocation by presenting an attribute-based data sharing system with keyword search and user revocation. To prevent the revoked user from fetching the data generated before user revocation, the re-encryption method is offered to realize the ciphertext updating, which leads to a significant overhead for data owners to re-encrypt data in each revocation epoch. To optimize the updating performance, Bakas *et al.* [8] designed a revocation mechanism and an access control mechanism by exploiting the functionalities offered by SGX, and presented a cloud-based data sharing system with user revocation and keyword search. Recently, Xu *et al.* [50] introduced a novel concept of ABE, called ElGamal-type ABE, which is based on the property of linear master secret sharing. They introduced a generic construction of revocable attribute-based encryption, which several candidates of ABE schemes [22], [37] can be used to instantiate it. However, the proposed generic construction does not provide any backward secrecy or ciphertext update.

Therefore, there is no formal treatment to build an EMR sharing system with fine-grained access control, manipulation detection, dynamic user group, and forward and backward secrecy simultaneously via the public cloud.

### 1.2 Contributions

In this paper, we introduce a dual-policy revocable attribute-based encryption (DP-RABE). By applying this cryptographic tool and blockchain technique, we design an efficient, secure, and flexible EMR sharing system with dynamic user groups to solve the above problem simultaneously.

*Non-Manipulated EMRs.* The previous solutions [44], [51], [52], [53], [55] focus on data confidentiality in the remote cloud storage rather than data integrity. Another research field called data of storage [3], [43] concentrates on data integrity and requires a third party to verify the remote data. To prevent EMRs manipulation, we apply the blockchain technique rather than the third party to check data integrity.

*Expressive Flexible Access Control With Dynamic User Groups.* Many existing EMRs management systems only have either content-based access control or role-based access control. To offer them simultaneously, we design a dual-policy mechanism with a constant-size system parameter and supporting the large universe. To manage revocation, we apply tree-based data structure [30] to decrease the overhead of user revocation from linear [13] to logarithmic.

*Publicly Updatable Ciphertext.* The previous solutions for user revocation [4], [11], [13], [17], [34], [35] only consider the revoked users cannot access any subsequent ciphertext after being revoked. However, the old ciphertexts still can be accessed by these revoked users. To overcome this problem, we design the mechanism that allows ciphertext to be updatable publicly, which allows ciphertexts can be updated by the CPS without any delegation key. Hence, users cannot access the data before being revoked.

*Decryption Exposure Resistance.* Decryption exposure attack [42] is a practical attack in indirectly revocable cryptosystems. The frequently used decryption key faces many threats, such as key leakage attacks and side-channel attacks. Decryption exposure attack enables the adversary to learn some decryption keys to break forward and backward secrecy. To prevent this attack, our proposed scheme utilizes key re-randomization technology to remove the relationship of decryption in each revocation epoch. Therefore, the forward and backward secrecy is preserved even some decryption keys are compromised.

### 1.3 Roadmap

Section 2 recalls some preliminaries about our proposed scheme. Section 3 introduces the system architecture, including definition, system model, threat model, and corresponding security model. Section 4 proposes the concrete DP-RABE scheme and corresponding security analysis. Section 5 gives the efficiency analysis to illustrate the practice of our proposed scheme. The paper is concluded in Section 6.

## 2 Preliminaries

Before giving an accurate description and definition of our proposed EMR sharing system, we introduce some notations and necessary preliminaries used in the proposed system.

### 2.1 Notations

Let $\mathbb{N}$ be the set of all natural numbers, and for $n \in \mathbb{N}$, we define $[n] : \{1, ..., n\}$. If $s$ is a string, then $s[i]$ denotes the $i^{th}$ bit of $s$. Let $\vec{u} := (u_1, u_2, ...u_\ell)$ denote a vector of dimension $\ell$ in $\mathbb{Z}_p^\ell$.

### 2.2 Assumptions

The decisional Bilinear Diffie-Hellman (BDH) assumption [45] defines the following game: The challenger takes as input a security parameter $\lambda$ to run the group generation algorithm $\mathcal{G}(\lambda)$ to derive the description of bilinear map $(\mathbb{G}, \mathbb{G}_T, g, p)$. Next, it picks three random terms $a, b, c \in \mathbb{Z}_p$ and sends $g, g^a, g^b, g^c, e(g, g)^z$ to distinguish the value $z$ is $abe$ or a random value.

The modified $q$ assumption [46] defines the following game: The challenger takes as input a security parameter $\lambda$ to run the group generation algorithm $\mathcal{G}(\lambda)$ to derive the description of bilinear map $(\mathbb{G}, \mathbb{G}_T, g, p)$. Next, it picks $2q + 2$ random exponents $a, s, b_1, b_2, ..., b_q, c_1, \ c_2, ..., c_q \in \mathbb{Z}_p$ and sends the following terms

$$\begin{array}{ll}
g, g^s, g^{(sa)^2} & \\
g^{a^i}, g^{b_j}, g^{sb_j}, g^{a^i b_j}, g^{a^i/b_j^2} & \forall (i,j) \in [q,q] \\
g^{a^i b_j/b_{j'}^2} & \forall (i,j,j') \in [2q,q,q] \text{ with } j \neq j' \\
g^{a^i/b_j} & \forall (i,j) \in [2q,q] \text{ with } i \neq q+1 \\
g^{sa^i b_j/b_{j'}}, g^{sa^i b_j/b_{j'}^2} & \forall (i,j,j') \in [q,q,q] \text{ with } j \neq j' \\
g^{c_k}, g^{sac_k}, g^{sa/c_k} & \forall k \in [q] \\
g^{sa^2 c_k}, g^{a^q/c_i^2}, g^{a^{q^2}/c_k^2} & \forall k \in [q] \\
g^{sac_k/c_{k'}}, g^{a^q c_k/c_{k'}^2} & \forall (k,k') \in [q,q] \text{ with } k \neq k' \\
g^{sa^{q+1} c_k/c_{k'}^2}, g^{(sa)^2 c_k/c_{k'}} & \forall (k,k') \in [q,q] \text{ with } k \neq k'
\end{array}$$

to distinguish $e(g,g)^{sa^{q+1}}$ from a random term $R \in \mathbb{G}_T$. In Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2021.3126532., we give the rigorous security analysis of the modified $q$ assumption.

### 2.3 Linear Secret Sharing Scheme (LSSS) [22]

Let $\mathbb{M}$ be an $\ell \times n$ matrix over the base field $\mathbb{F}$ and $\rho$ denote a mapping function from the set $[\ell]$ to the universe of attributes. An LSSS policy $(\mathbb{M}, \rho)$ satisfies an attribute set $\psi$ if $(1, 0, ..., 0) \in \mathbb{F}^n$ is contained in $\mathsf{Span}_{\mathbb{F}}(\mathbb{M}_i : \rho(i) \in \psi)$, where $\mathbb{M}_i$ is the $i^{th}$ row of $\mathbb{M}$.

### 2.4 Tree-Based Revocation Mechanism

The subset-cover algorithm $\mathsf{KUNode}(st, rl, t)$ [30] as shown in Algorithm 1 was introduced to fetch the minimum set related to non-revoked data users to derive key-updating material, which takes as input state $st$ as a binary tree, a revocation list $rl$ and a revocation timestamp $t$, and outputs a set of nodes. Specifically, let $\mathsf{Path}(v)$ be the set of nodes on the path from the root node of the tree to the node $v$, and $(v_l, v_r)$ denote the left and right child of $v$ if $v$ is a non-leaf node, the details of the tree-based revocation mechanism are given below:

**Algorithm 1.** $\mathsf{KUNode}(st, rl, t)$ [30]

```
$\mathsf{X}, \mathsf{Y} \leftarrow \emptyset$;
for $(v_i, t_i) \in rl$ do
    if $t_i \leq t$ then
        X$\leftarrow$ X$\cup$Path$(v_i)$
for $x \in \mathsf{X}$ do
    if $x_l \notin \mathsf{X}$ thenY $\leftarrow \mathsf{Y} \cup x_l$
    if $x_r \notin \mathsf{X}$ then
        Y $\leftarrow \mathsf{Y} \cup x_r$
if $\mathsf{Y} = \emptyset$ then
    Y $\leftarrow$ root
return Y.
```

### 2.5 Time Encoding Mechanism

The time encoding algorithm $\mathsf{CTEncode}(t, \mathcal{T})$ [49] as shown in Algorithm 2 was introduced to encode the timestamp $t$ to a binary timestamp $\tilde{t} \leq t$, where $\mathcal{T}$ is the system bounded lifetime. By replacing identity to $\tilde{t}$ in Waters' identity-based encryption (IBE) [45], the timestamp is allowed to be updatable without delegated information. The details of time encoding mechanism are given below:

**Algorithm 2.** $\mathsf{CTEncode}(t, \mathcal{T})$ [49]

```
chk $\leftarrow$ false;
len $\leftarrow \log_2 \mathcal{T}$;
for $i = \log_2 \mathcal{T}$ to $i = 1$ do
    if $t[i] = 1$ and $i =$ len and chk $=$ false then
        $\tilde{t}[i] = 1$, len $=$ len $- 1$
    else
        chk $\leftarrow$ true, $\tilde{t} = 0$
return $\tilde{t}$.
```

### 2.6 Symmetric Encryption

**Definition 1 (Symmetric Encryption).** *A symmetric encryption scheme $\Pi$ consists of two deterministic algorithms operated between senders and receivers.*

$\Pi.\mathsf{Enc}(K, m) \rightarrow c$: *The encryption algorithm is run by senders. The algorithm takes as input a key $K$ and a message $m$, and outputs a ciphertext $c$.*

$\Pi.\mathsf{Dec}(K, c) \rightarrow m$: *The decryption algorithm is run by receivers. The algorithm takes as input the key $K$ and ciphertext $c$, and outputs the message $m$.*

A symmetric-key encryption scheme is said to be one-time semantically secure (SS) if for any probabilistic polynomial time adversary $\mathcal{A}$, the following advantage is negligible:

$$\mathbf{Adv}^{\mathsf{SS}}_{\mathcal{A},\Pi}(\lambda) = \left| \Pr\left[ b = b' : \begin{array}{c} (m_0, m_1) \leftarrow \mathcal{A}(\lambda) \\ b \in \{0,1\}, K \in \mathcal{K} \\ c^* \leftarrow \Pi.\mathsf{Enc}(K, m_b) \\ b' \leftarrow \mathcal{A}(c^*) \end{array} \right] - \frac{1}{2} \right|.$$

### 2.7 Collision-Resistant Hash Function

**Definition 2 (Collision-Resistant Hash Function).** *A hash function $h : \{0,1\}^n \rightarrow \{0,1\}^m$ is said to be collision resistant if it has following properties: (1) Length-compressing: $m < n$; and (2) Hard to find collisions: For all probabilistic polynomial time adversaries, the following probability is negligible:*

$$\Pr[(x_0, x_1) \leftarrow \mathcal{A}(1^n, h) : x_0 \neq x_1 \cap h(x_0) = h(x_1)].$$

**Remark 1.** To support the authorized party to modify outsourced data and blockchain information, we can replace collision-resistant hash function to chameleon hash [28], which has been widely used in blockchain to modify the block information [14], [18].

## 3 System Architecture

In this section, we present the definition of DP-RABE scheme, and a system model as well as a threat model of EMR sharing systems. We then provide a formal security model to simulate all the possible attacks in the threat model.

### 3.1 Definition

In our proposed DP-RABE, we consider flexible access control and design an efficient revocation mechanism to handle dynamic user groups. The definition of DP-RABE is given below.

**Definition 3 (DP-RABE).** *A DP-RABE scheme $\Lambda$ with a subjective attribute universe $\Omega_s$ and an objective universe $\Omega_o$, which supports subjective policies $\mathcal{P}_s$ and objective policies $\mathcal{P}_o$ with an identity space $\mathcal{I}$ and a message space $\mathcal{M}$ consists of the following algorithms:*

$\Lambda.\mathsf{Setup}(\lambda, \mathcal{N}, \mathcal{T}) \rightarrow (pp, msk, rl, st)$: *The probabilistic setup algorithm takes as input a security parameter $\lambda$, a number of users $\mathcal{N}$ and a system lifetime $\mathcal{T}$, and outputs a parameter parameter $pp$, a master secret key $msk$, a revocation list $rl$ and a state $st$.*

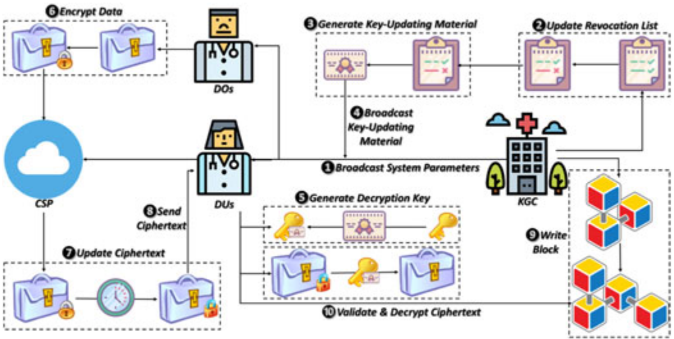

❻Encrypt Data
DOs
❸Generate Key-Updating Material
❷Update Revocation List
❹Broadcast Key-Updating Material
❶Broadcast System Parameters
CSP
DUs
KGC
❽Send Ciphertext
❺Generate Decryption Key
❾Write Block
❼Update Ciphertext
❿Validate & Decrypt Ciphertext

Fig. 3. System Model of EMR Data Sharing via Cloud.

$\Lambda.\mathsf{KeyGen}(pp, msk, st, id, (\psi, \mathbb{O})) \to (sk_{id}, st)$: *The probabilistic key generation algorithm takes as input a public parameter* $pp$, *a master secret key, a state* $st$, *an identity* $id \in \mathcal{I}$, *a set of subjective attributes* $\psi \in \Omega_s$ *and an objective access structure* $\mathbb{O} \in \mathcal{P}_o$, *and outputs a secret key* $sk_{id}$ *and a state* $st$.

$\Lambda.\mathsf{KeyUpdate}(pp, rl, st, t) \to ku_t$: *The probabilistic key update algorithm takes as input a public parameter* $pp$, *a revocation list* $rl$, *a state* $st$ *and a timestamp* $t$, *and outputs a key-updating material* $ku_t$.

$\Lambda.\mathsf{DKGen}(pp, sk_{id}, ku_t) \to dk_{id,t/}\perp$: *The probabilistic decryption key generation algorithm takes as input a public parameter* $pp$, *a secret key* $sk_{id}$ *and key-updating material* $ku_t$, *and outputs a decryption key* $dk_{id,t}$ *or a failure symbol* $\perp$.

$\Lambda.\mathsf{Enc}(pp, t, m, (\mathbb{S}, \omega)) \to c$: *The probabilistic encryption algorithm takes as input a public parameter* $pp$, *a timestamp* $t$, *a message* $m \in \mathcal{M}$, *a subjective access structure* $\mathbb{S} \in \mathcal{P}_s$ *and a set of objective attribute* $\omega \in \Omega_o$, *and outputs a ciphertext* $c$.

$\Lambda.\mathsf{CTUpdate}(pp, c, t') \to c'$: *The probabilistic ciphertext update algorithm takes as input a public parameter* $pp$, *a ciphertext* $c$ *and a timestamp* $t'$, *and outputs an updated ciphertext* $c'$.

$\Lambda.\mathsf{Dec}(pp, dk_{id,t}, c') \to m$: *The deterministic decryption algorithm takes as input a public parameter* $pp$, *a decryption key* $dk_{id,t}$ *and a updated ciphertext* $c'$, *and outputs a message* $m \in \mathcal{M}$.

$\Lambda.\mathsf{Rev}(rl, id, t) \to rl$: *The deterministic revocation algorithm takes as input a revocation list* $rl$, *an identity* $id \in \mathcal{I}$ *and a timestamp* $t$, *and outputs a revocation list* $rl$.

### 3.2 System Model

Our proposed EMR sharing system has four typical parties: a key generation center (KGC), data owners (DOs), data users (DUs) and a CSP as shown in Fig. 3. The details of each entity are given below.

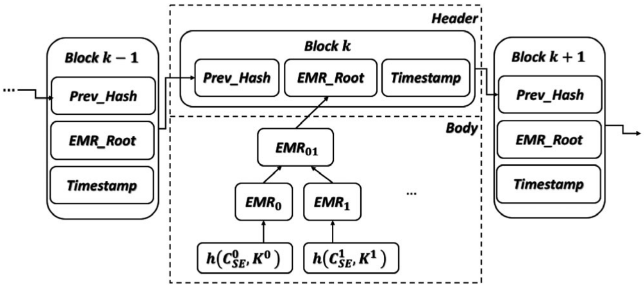


Fig. 4. Blockchain Maintains EMR to Prevent Manipulation.

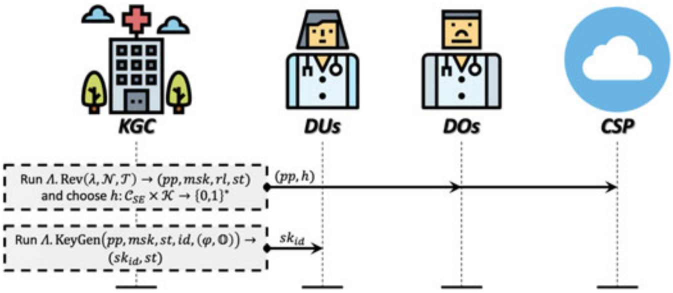

KGC
DUs
DOs
CSP
Run Λ. Rev(λ, N, T) → (pp, msk, rl, st) and choose h: C_SE × K → {0,1}*
(pp, h)
Run Λ. KeyGen(pp, msk, st, id, (φ, O)) → (sk_id, st)
sk_id

Fig. 5. System Initialization.

*KGC*. The KGC acts as the medical center to initialize system and broadcast system parameters to other entities, and also maintains credentials of medical staffs and patients when they join the system. Besides, the KGC can revoke the invalid medical staffs and patients by publicly broadcasting key-updating materials in each revocation epoch. Moreover, to prevent manipulation of EMRs, the KGC appends new block in the blockchain, as shown in Fig. 4 in each revocation epoch. In the real-world applications, the manager of blockchain could be a trusted third party, e.g., government and insurance company, etc.

*DOs*. DOs represents a set of medical staffs and patients who have confidential data to be shared with DUs by uploading the corresponding ciphertexts to the CSP.

*DUs*. DUs is a set of medical staffs and patients who have valid secrets key issued by the KGC. They can derive valid decryption keys if their credentials are not revoked.

*CSP*. The CSP offers a large amount of data storage to accommodate ciphertexts from DOs and unlimited computational power to evolve ciphertexts to the updated ciphertext in current revocation epoch.

Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme as in Section 2.6, $\Lambda = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{CTUpdate}, \mathsf{Dec}, \mathsf{Rev})$ represent a DP-RABE scheme as in Section 3.1 and $h$ be a collision resistant hash function as in Section 2.7. The workflow as shown in Fig. 3 includes the following four steps:

- *System initialization*: Fig. 5 shows the system initialization phase. The KGC runs $\Lambda.\mathsf{Setup}(\lambda, \mathcal{N}, \mathcal{T}) \to (pp, msk, rl, st)$ and chooses a collision resistant hash function $h : \mathcal{C}_{SE} \times \mathcal{K} \to \{0,1\}^*$ to initialize the system, and broadcasts $(pp, h)$ to DOs and CSP, where $\mathcal{C}_{SE}$ is the ciphertext of symmetric encryption scheme and $\mathcal{K}$ is key space of symmetric key. For each DUs, the KGC runs $\Lambda.\mathsf{KeyGen}(pp, msk, st, id, (\psi, \mathbb{O})) \to (sk_{id}, st)$ and sends the secret key $sk_{id}$ to the corresponding DU (see ❶).
- *User managing*: Fig. 6 shows the user managing phase. The KGC runs $\Lambda.\mathsf{Rev}(rl, id, t) \to rl$ to add invalid data receivers to the revocation list $rl$ (see ❷) and runs $\Lambda.\mathsf{KeyUpdate}(pp, rl, st, t) \to ku_t$ (see ❸) to broadcast public key-updating material in each revocation epoch (see ❹). For all DUs, they runs $\Lambda.\mathsf{DKGen}(pp, sk_{id}, ku_t) \to dk_{id,t}/\perp$ to derive the valid decryption key $dk_{id,t}$ if they are non-revoked (see ❺); otherwise, they obtain a failure symbol $\perp$.
- *Data sharing*: Fig. 7 shows the data sharing phase. DOs pick a random symmetric key $K \in \mathcal{K}$ and run $\Pi.\mathsf{Enc}(K,$

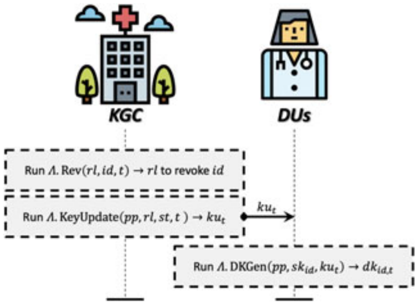

KGC
DUs
Run $\Lambda$.Rev$(rl, id, t) \rightarrow rl$ to revoke $id$
Run $\Lambda$.KeyUpdate$(pp, rl, st, t) \rightarrow ku_t$
$ku_t$
Run $\Lambda$.DKGen$(pp, sk_{id}, ku_t) \rightarrow dk_{id,t}$

Fig. 6. User Managing.

$EMR) \rightarrow C_{SE}$, $\Lambda.\mathsf{Enc}(pp, t, K, (\mathbb{S}, \omega)) \rightarrow C_{ABE}$ and $H = h(C_{SE}, K)$ (see ❻), then upload ciphertexts $C = (C_{SE}, C_{ABE}, H)$ to the CSP. The CSP then runs $\Lambda.\mathsf{CTUpdate}(pp, C_{ABE}, t') \rightarrow C'_{ABE}$ to update ciphertexts $C = (C_{SE}, C_{ABE}, H)$ to $C' = (C_{SE}, C'_{ABE}, H)$ with the current revocation epoch $t'$ (see ❼). The KGC and DUs are allowed to request the updated ciphertext $C'$ (see ❽). The KGC extracts $H$ in newly derived ciphertexts $C'$ and then writes them into the blockchain as in Fig. 4 to prevent data manipulation (see ❾).

- *Data revealing*: Fig. 8 shows the data revealing phase. DUs first run $\Lambda.\mathsf{Dec}(pp, dk_{id,t}, C'_{ABE}) \rightarrow K$ and generate $H' = h(C_{SE}, K)$, and then check the validation depended on blockchain. If $H = H'$, DUs run $\Pi.\mathsf{Dec}(K, C_{SE}) \rightarrow EMR$ to get retrieve the data EMR; otherwise, it aborts this ciphertext since it is invalid message (see ❿).

**Remark 2.** Our proposed EMR sharing system includes key encapsulation mechanism (KEM) to improve the performance of data transmissions and blockchain technology for achieving immutability, where the security of KEM is protected by the symmetric encryption as shown in Definition 1 and the security of blockchain technology is based on ECDSA [24] and collision-resistant hash function as shown in Definition 2.

### 3.3 Threat Model

In our system, the fully trusted entities are the KGC and DOs. The KGC issues credentials to DUs, broadcasts public key-updating materials and appends valid block in blockchain in each revocation epoch. DOs encrypt EMR honestly following our proposed mechanism.

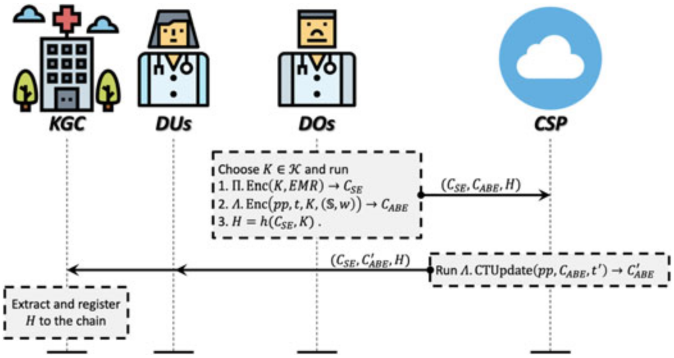

KGC
DUs
DOs
CSP
Choose $K \in \mathcal{K}$ and run
1. $\Pi$.Enc$(K, EMR) \rightarrow C_{SE}$
2. $\Lambda$.Enc$(pp, t, K, (\mathbb{S}, w)) \rightarrow C_{ABE}$
3. $H = h(C_{SE}, K)$ .
$(C_{SE}, C_{ABE}, H)$
$(C_{SE}, C'_{ABE}, H)$
Run $\Lambda$.CTUpdate$(pp, C_{ABE}, t') \rightarrow C'_{ABE}$
Extract and register $H$ to the chain

Fig. 7. Data Sharing.

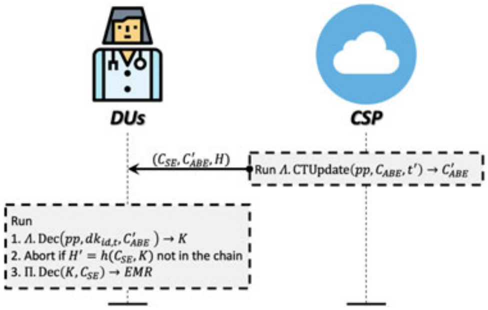

DUs
CSP
$(C_{SE}, C'_{ABE}, H)$
Run $\Lambda$.CTUpdate$(pp, C_{ABE}, t') \rightarrow C'_{ABE}$
Run
1. $\Lambda$.Dec$(pp, dk_{id,t}, C'_{ABE}) \rightarrow K$
2. Abort if $H' = h(C_{SE}, K)$ not in the chain
3. $\Pi$.Dec$(K, C_{SE}) \rightarrow EMR$

Fig. 8. Data Revealing.

The CSP is semi-trusted who follows our scheme but tries to learn the sensitive data with any possible passive attacks. Moreover, the CSP may dishonestly store the outsourced data from DOs for reducing the resource cost. To prevent the data manipulation attack from the semi-trusted CSP, we apply blockchain to check data integrity and cryptosystems to ensure data confidentiality.

DUs are untrusted who can be either non-revoked and revoked. They all try to learn the unauthorized data and even collude with the other DUs. For example, unauthorized and DUs, and authorized but revoked DUs to reveal sensitive data. To prevent attacks from untrusted DUs, we apply cryptosystems to ensure data confidentiality.

From the above, the security of our proposed scheme depends on blockchain and cryptosystems. Blockchain has been well known to secure and against manipulation. The cryptosystems in our scheme are hash function, symmetric encryption, and DP-RABE scheme. We apply the collision-resistant hash function to prevent the adversary from breaking data integrity. AES is the instantiation of symmetric encryption, which is well known secure against varieties of attacks. For the security of DP-RABE, we present the formal security model called selectively indistinguishable against chosen plaintext attacks ($\mathsf{sIND-CPA}$) in the next subsection to demonstrate the security of our DP-RABE.

### 3.4 Security Model

**Definition 4 ($\mathsf{sIND-CPA}$ in CP-RABE)** *A DP-RABE consists of eight algorithms* $\Lambda = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyUpdate}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{CTUpdate}, \mathsf{Dec}, \mathsf{Rev})$. *For an adversary $\mathcal{A}$, we define the following experiment:*

$$
\begin{array}{l}
\text{Experiment } \mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\Lambda}(\lambda, \mathcal{N}, \mathcal{T}) \\
\quad (\mathbb{S}^*, \omega^*, t^*) \leftarrow \mathcal{A}(\lambda); \\
\quad (pp, msk, rl, st) \leftarrow \Lambda.\mathsf{Setup}(\lambda, \mathcal{N}, \mathcal{T}); \\
\quad (m_0, m_1, t) \leftarrow \mathcal{A}^{\mathcal{O}}(pp); \\
\quad b \leftarrow \{0, 1\}; \\
\quad c \leftarrow \Lambda.\mathsf{Enc}(pp, t^*, m_b, (\mathbb{S}^*, \omega^*)); \\
\quad c' \leftarrow \Lambda.\mathsf{CTUpdate}(pp, c, t^*); \\
\quad b' \leftarrow \mathcal{A}(c'); \\
\quad return\ 1\ if\ b = b', else\ return\ 0.
\end{array}
$$

*$\mathcal{O}$ denotes a set of oracles, $\{\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot,\cdot), \mathcal{O}_{\mathsf{KeyUpdate}}(\cdot), \mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot), \mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot,\cdot,\cdot)\}$. The definition of above oracles are described as follows:*

- $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot,\cdot)$ *is a key generation oracle that allows* $\mathcal{A}$ *to query an identity* $id \in \mathcal{I}$*, a set of subjective attributes* $\psi \in \mathbf{\Omega}_s$ *and an objective access structure* $\mathbb{O} \in \mathcal{P}_o$*. For each query, it runs* $\Lambda.\mathsf{KeyGen}(pp, msk, st, id, (\psi, \mathbb{O}))$ *to return the secret key* $sk_{id}$*.*
- $\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot)$ *is a key update oracle that allows* $\mathcal{A}$ *to query a timestamp* $t \in \mathcal{T}$*. For each query, it runs* $\Lambda.\mathsf{KeyUpdate}(pp, rl, st, t)$ *to output the key update* $ku_t$*.*
- $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ *is a revocation oracle that allows* $\mathcal{A}$ *to query an identity* $id \in \mathcal{I}$ *and a timestamp* $t$*. For each query, it runs* $\Lambda.\mathsf{Rev}(rl, id, t)$ *to update the revocation list* $rl$*.*
- $\mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot,\cdot,\cdot)$ *is a decryption key generation oracle that allows* $\mathcal{A}$ *to query an identity* $id \in \mathcal{I}$*, a set of subjective attributes* $\psi \in \mathbf{\Omega}_s$*, an objective access structure* $\mathbb{O} \in \mathcal{P}_o$ *and a timestamp* $t \in \mathcal{T}$*. For each query, it runs* $\Lambda.\mathsf{DKGen}(pp, sk_{id}, ku_t)$ *to output the decryption key* $dk_{id,t}$ *if the corresponding secret key* $sk_{id}$ *and key update* $ku_t$ *are available; otherwise, it runs* $\mathcal{O}_{\mathsf{KeyGen}}(id, \psi, \mathbb{O})$ *and* $\mathcal{O}_{\mathsf{KeyUpdate}}(t)$ *first to obtain the secret key* $sk_{id}$ *and key update* $ku_t$*. The above oracles are allowed to query adaptively with the following restrictions:*

1) $\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot)$ *and* $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ *can be queried at the time* $t$ *which is greater than or equal to that of all previous queries.*
2) $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ *cannot be queried at the time* $t$ *if* $\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot)$ *was queried at the time* $t$*.*
3) *If* $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot,\cdot)$ *was queried on an identity* $id \in \mathcal{I}$ *with a set of subjective attributes* $\psi \in \mathbf{\Omega}_s$ *and an objective access structure* $\mathbb{O} \in \mathcal{P}_o$*, s.t.* $\mathbb{S}^*(\psi) = \mathbb{O}(\omega^*) = 1$*, then* $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ *must be queried on this identity* $id$ *at the time* $t \le t^*$*.*
4) $\mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot,\cdot,\cdot)$ *cannot be queried on any identity* $id \in \mathcal{I}$ *with a set of subjective attributes* $\psi \in \mathbf{\Omega}_s$ *and an objective access structure* $\mathbb{O} \in \mathcal{P}_o$*, s.t.* $\mathbb{S}^*(\psi) = \mathbb{O}(\omega^*) = 1$ *at the challenge time* $t^*$ *or any identity* $id \in \mathcal{I}$ *has been revoked. A DP-RABE scheme is said to be* sIND-CPA *secure if for any probabilistic polynomial time adversary* $\mathcal{A}$*, the following advantage is negligible:*

$$
\begin{aligned}
&\mathbf{Adv}_{\mathcal{A},\Lambda}^{\mathsf{sIND\text{-}CPA}}(\lambda, \mathcal{N}, \mathcal{T}) \\
&= \left| \Pr[\mathbf{Exp}_{\mathcal{A},\Lambda}^{\mathsf{sIND\text{-}CPA}}(\lambda, \mathcal{N}, \mathcal{T}) = 1] - 1/2 \right|.
\end{aligned}
$$

**Remark 3.** Our security model also captures the security threats as we mentioned in the threat model. By querying $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot,\cdot,\cdot)$, the adversary can obtain multiple secret keys and decryption keys, which allow the adversary to play as untrusted revoked DUs. By querying $\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot)$ and $\mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot,\cdot,\cdot)$, the adversary also obtains multiple secret keys and key-updating materials, which enable the adversary to play as untrusted non-revoked DUs. Combing the above two cases, the adversary can launch the collude attack on behave of untrusted DUs. The adversary also can play as the semi-trusted cloud by obtaining the challenge ciphertext.

**Remark 4.** Our security model is derived from the previous RABE schemes [25], [39], [48] with the advanced access control policy. These previous solutions only consider key-policy and ciphertext-policy access control individually; in contrast, our model focuses on dual-policy access control. Note that the initial ciphertext $c$ must be protected. In previous solutions [25], [39], [48], the ciphertext is generated for the future ($t^* > t$ for all $t$ having been queried) or is updated before sending to $\mathcal{A}$. Our security model follows the latter case.

## 4 Proposed DP-RABE Scheme

In this section, we give the concrete scheme of DP-RABE with the security proof based on the decisional BDH assumption and the $q$-type assumption.

### 4.1 Concrete Scheme

Let $\Lambda$ be a DP-RABE scheme with a subjective attribute universe $\mathbf{\Omega}_s$ and an objective universe $\mathbf{\Omega}_o$ that supports subjective policies $\mathcal{P}_s$ and objective policies $\mathcal{P}_o$ with an identity space $\mathcal{I}$ and a message space $\mathcal{M}$. The concrete construction of DP-RABE are given below:

$\Lambda.\mathsf{Setup}(\lambda, \mathcal{N}, \mathcal{T}) \rightarrow (pp, msk, rl, st)$: The setup algorithm generates the description of bilinear map $(\mathbb{G}, \mathbb{G}_T, g, p)$ by running the group generation algorithm $\mathcal{G}(\lambda)$, then randomly picks $\alpha \in \mathbb{Z}_p$ and $w, v, u, h, \tilde{u}, \tilde{h}, u_0, u_1, \ldots, u_\ell \in \mathbb{G}$, where $\ell = \log_2 \mathcal{T}$. The algorithm returns the public parameter $pp$ and the master secret key $msk$ as:

$$pp = (g, w, v, u, h, \tilde{u}, \tilde{h}, u_0, u_1, \ldots, u_\ell, e(g,g)^\alpha), msk = \alpha,$$

and two empty sets $rl, st \leftarrow \emptyset$ denoted a revocation list and a state.

$\Lambda.\mathsf{KeyGen}(pp, msk, st, id, (\psi, \mathbb{O})) \rightarrow (sk_{id}, st)$: Parse the set of subjective attributes $\psi = (\psi_1, \psi_2, \ldots, \psi_{k_s}) \in \mathbf{\Omega}_s$ and the objective access structure $\mathbb{O} = (\mathbb{M}, \rho) \in \mathcal{P}_o$, where $\mathbb{M}$ is an $\ell_o \times n_o$ matrix and $\rho$ is a mapping function $\rho : [\ell_o] \rightarrow \mathbb{Z}_p$. The key generation algorithm assigns $id$ to a unassigned leaf node. For all nodes $\theta \in \mathsf{Path}(id)$, it retrieves $\alpha_\theta$ if available; otherwise, it chooses $\alpha_\theta \in \mathbb{Z}_p$ and updates the state $st \leftarrow st \cup (\theta, \alpha_\theta)$. The algorithm chooses $\vec{x} = (\alpha, x_2, .., x_{n_o})^\top \in \mathbb{Z}_p^{n_o \times 1}$ to derive $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{\ell_o}) = \mathbb{M}\vec{x}$ and picks $r$, $\{r_i\}_{i \in [\ell_o]}, \{r_j\}_{j \in [k_s]} \in \mathbb{Z}_p$, then computes $sk_{id,\theta} = (\{sk_{1,i},\ sk_{2,i}, sk_{3,i}\}_{i \in [\ell_o]}, sk_4, \{sk_{5,j}, sk_{6,j}\}_{j \in [k_s]})$ as:

$$
\begin{aligned}
sk_{1,i} = g^{\lambda_i - \alpha_\theta} w^{r_i} w^r, sk_{2,i} = (\tilde{u}^{\rho(i)} \tilde{h})^{-r_i}, sk_{3,i} = g^{r_i}, sk_4 = g^r, sk_{5,j} \\
= g^{r_j}, sk_{6,j} = (u^{\psi_j} h)^{r_j} v^{-r}.
\end{aligned}
$$

The algorithm returns $sk_{id} = ((\psi, \mathbb{O}), \{sk_{id,\theta}\}_{\theta \in \mathsf{Path}(id)})$ and the updated state $st$.

$\Lambda.\mathsf{KeyUpdate}(pp, rl, st, t) \rightarrow ku_t$: The key update algorithm encodes $t$ to the bit representation. Let $\mathcal{V} \subseteq [\ell]$ be a set of $k$ with $t[k] = 0$. For $\theta \in \mathsf{KUNode}(st, rl, t)$, it retrieves $\alpha_\theta$, randomly picks $\tau \in \mathbb{Z}_p$ and computes $ku_{t,\theta} = (ku_1, ku_2)$ as:

$$ku_1 = g^{\alpha_\theta} (u_0 \textstyle\prod_{k \in \mathcal{V}} u_k)^\tau, ku_2 = g^\tau.$$

The algorithm returns $ku_t = (t, \{ku_{t,\theta}\}_{\theta \in \mathsf{KUNode}(st,rl,t)})$.

$\Lambda.\mathsf{DKGen}(pp, sk_{id}, ku_t) \rightarrow dk_{id,t}/\bot$: The decryption key generation algorithm returns a failure symbol $\bot$ if $\mathsf{Path}(id) \cap \mathsf{KUNode}(st, rl, t) = \emptyset$; otherwise, it randomly chooses $r', \{r'_i\}_{i\in[\ell_o]}, \{r'_j\}_{j\in[k_s]}, \tau' \in \mathbb{Z}_p$ and computes $dk_{id,t} = (\{dk_{1,i}, dk_{2,i}, dk_{3,i}\}_{i\in[\ell_o]}, dk_4, \{dk_{5,j}, dk_{6,j}\}_{i\in[k_s]}, dk_7)$ as:

$$\begin{aligned} dk_{1,i} &= sk_{1,i} \cdot ku_1 \cdot w^{r'_i} \cdot w^{r'} \cdot (u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{\tau'} \\ &= g^{\lambda_i} w^{r_i+r'_i} w^{r+r'} (u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{\tau+\tau'}, dk_{2,i} = sk_{2,i} \cdot (\tilde{u}^{\rho(i)}\tilde{h})^{-r'_i} \\ &= (\tilde{u}^{\rho(i)}\tilde{h})^{-(r_i+r'_i)}, dk_{3,i} = sk_{3,i} \cdot g^{r'_i} = g^{r_i+r'_i}, dk_4 = sk_4 \cdot g^{r'} \\ &= g^{r+r'}, dk_{5,j} = sk_{5,j} \cdot g^{r'_j} = g^{r_j+r'_j}, dk_{6,j} \\ &= sk_{6,j} \cdot (u^{\psi_j}h)^{r'_j} v^{-r'} = (u^{\psi_j}h)^{r_j+r'_j} v^{-(r+r')}, dk_7 = ku_2 \cdot g^{\tau'} \\ &= g^{\tau+\tau'}. \end{aligned}$$

The algorithm returns $dk_{id,t} = (\{dk_{1,i}, dk_{2,i}, dk_{3,i}\}_{i\in[\ell_o]}, dk_4, \{dk_{5,j}, dk_{6,j}\}_{i\in[k_s]}, dk_7)$.

$\Lambda.\mathsf{Enc}(pp, t, m, (\mathbb{S}, \omega)) \to c$: Parse the set of objective attributes $\omega = (\omega_1, \omega_2, ..., \omega_{k_o}) \in \Omega_o$ and the subjective access structure $\mathbb{S} = (\mathbb{N}, \pi) \in \mathcal{P}_s$, where $\mathbb{N}$ is an $\ell_s \times n_s$ matrix and $\pi$ is a mapping function $\pi : [\ell_s] \to \mathbb{Z}_p$. The encryption algorithm chooses $\vec{y} = (s, y_2, ..., y_n)^\top \in \mathbb{Z}_p^{n_s\times 1}$ to derive $\vec{\lambda'} = (\lambda'_1, \lambda'_2, ..., \lambda'_{\ell_s}) = \mathbb{N}\vec{y}$, then encodes time $t$ as $\tilde{t} \leftarrow \mathsf{CTEncode}(t, \mathcal{T})$. Let $\mathcal{V} \subseteq [\ell]$ be a set of $k$ with $\tilde{t}[k] = 0$. The algorithm picks $\{s_i\}_{i\in[k_o]}, \{s_j\}_{j\in[\ell_s]} \in \mathbb{Z}_p$ and computes:

$$\begin{aligned} c_0 &= m \cdot e(g,g)^{\alpha s}, c_1 = g^s, c_{2,i} = g^{s_i}, c_{3,i} = (\tilde{u}^{\omega_1}\tilde{h})^{s_i} w^{-s}, c_{4,j} \\ &= w^{\lambda'_j} v^{s_j}, c_{5,j} = (u^{\pi(j)}h)^{-s_j}, c_{6,j} = g^{s_j}, c_7 = u_0^s, c_{8,k} = u_k^s. \end{aligned}$$

The algorithm returns $c = ((t, \mathbb{S}, \omega), c_0, \;\; c_1, \{c_{2,i}, c_{3,i}\}_{i\in[k_o]}, \{c_{4,j}, c_{5,j}, c_{6,j}\}_{j\in[\ell_s]}, c_7, \{c_{8,k}\}_{k\in\mathcal{V}})$.

$\Lambda.\mathsf{CTUpdate}(pp, c, t') \to c'$: The ciphertext update algorithm encodes the time $t'$ to the bit representation. Let $\mathcal{V} \subseteq [\ell]$ be a set of $k$ with $t'[k] = 0$. It chooses $\vec{y'} = (s', y'_2, ..., y'_n)^\top \in \mathbb{Z}_p^{n_s\times 1}$ to derive $\vec{\lambda''} = (\lambda''_1, \lambda''_2, ..., \lambda''_{\ell_s}) = \mathbb{N}\vec{y'}$. The algorithm picks $\{s'_i\}_{i\in[k_o]}, \{s'_j\}_{j\in[\ell_s]} \in \mathbb{Z}_p$ and computes the ciphertext $c'$:

$$\begin{aligned} c'_0 &= c_0 \cdot e(g,g)^{\alpha s'} = m \cdot e(g,g)^{\alpha(s+s')}, c'_1 = c_1 \cdot g^{s'} = g^{s+s'}, c'_{2,i} \\ &= c_{2,i} \cdot g^{s'_i} = g^{s_i+s'_i}, c'_{3,i} \\ &= c_{3,i} \cdot (\tilde{u}^{\omega_1}\tilde{h})^{s'_i} w^{-s'} = (\tilde{u}^{\omega_1}\tilde{h})^{s_i+s'_i} w^{-(s+s')}, c'_{4,j} \\ &= c_{4,j} \cdot w^{\lambda''_j} v^{s'_j} = w^{\lambda'_j+\lambda''_j} v^{s_j+s'_j}, c'_{5,j} \\ &= c_{5,j} \cdot (u^{\pi(j)}h)^{-s'_j} = (u^{\pi(j)}h)^{-(s_j+s'_j)}, c'_{6,j} \\ &= c_{6,j} \cdot g^{s'_j} = g^{s_j+s'_j}, c'_7 \\ &= c_7 \cdot u_0^{s'} \cdot \textstyle\prod_{k\in\mathcal{V}} c_{8,k} \cdot u_k^{s'} = (u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{s+s'}. \end{aligned}$$

The algorithm returns $c' = ((t', \mathbb{S}, \omega), c'_0, c'_1, \{c'_{2,i}, c'_{3,i}\}_{i\in[k_o]}, \{c'_{4,j}, c'_{5,j}, c'_{6,j}\}_{j\in[\ell_s]}, c'_7)$.

$\Lambda.\mathsf{Dec}(pp, dk_{id,t}, c') \to m$: Let $I = \{i : \rho(i) \in \psi\}$ and $J = \{j : \pi(j) \in \omega\}$. The decryption algorithm computes two vectors:

$$\begin{aligned} \vec{v} &= \{v_i \in \mathbb{Z}_p\}, s.t. \textstyle\sum_{i\in I} \mathbb{M}_i v_i = (1, 0, ..., 0), \vec{w} \\ &= \{w_j \in \mathbb{Z}_p\}, s.t. \textstyle\sum_{j\in J} \mathbb{N}_j w_j = (1, 0, ..., 0). \end{aligned}$$

To decrypt the ciphertext $c$, it computes $den$ as:

$$\begin{aligned} den &= \textstyle\prod_{j\in J} (e(c'_{4,j}, dk_4) \cdot e(c'_{5,j}, dk_{5,j}) \cdot e(c'_{6,j}, dk_{6,j}))^{w_j} \cdot \\ &\quad e(c'_7, dk_7) = e(g,w)^{(r+r')(s+s')} \cdot e(g, u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{(\tau+\tau')(s+s')}. \end{aligned}$$

Next, it computes $num$ as:

$$\begin{aligned} num &= \textstyle\prod_{i\in I} (e(c'_1, dk_{1,i}) \cdot e(c'_{2,i}, dk_{2,i}) \cdot e(c'_{3,i}, dk_{3,i})/den)^{v_i} \\ &= e(g,g)^{\alpha(s+s')}. \end{aligned}$$

The algorithm returns $m = c'_0/num$.

$\Lambda.\mathsf{Rev}(rl, id, t) \to rl$: The revocation algorithm updates the revocation list as $rl \leftarrow rl \cup (id, t)$.

**Remark 5.** To optimize the performance, once the ciphertext is updated, the updated version will replace the original version except the components related to timestamp, e.g., $c_7$ and $c_{8,k}$. In our threat model, we assume that CSP is semi-trusted, hence, this strategy is secure and the adversary cannot gain any advantage from it. Several promising solutions [25], [39] have been introduced to update the whole storage including the components related to timestamp. They use various cryptographic tools to manage the timestamp, e.g., ABE [26] in [39] and hierarchical IBE [12] in [25].

**Remark 6.** We consider the security notion of $\mathsf{sIND-CPA}$ in our proposed DP-RABE scheme, which offers semantic security. To achieve notions like selectively indistinguishable against chosen ciphertext attacks ($\mathsf{sIND-CCA}$), the well-known Fujisaki-Okamoto transformation [21] can be applied, which provides a generic transformation approach from $\mathsf{IND-CPA}$-like notions to $\mathsf{IND-CCA}$-like notions. Besides the widely adopted Fujisaki-Okamoto transformation [21], many promising candidates [15], [38] can be leveraged.

## 4.2 Correctness

The message $m$ can be recovered if the decryption key associated with $(\psi, \mathbb{O}, t)$ satisfies the ciphertext $(\mathbb{S}, \omega, t')$, s.t. $\mathbb{S}(\psi) = \mathbb{O}(\omega) = 1$ and $t = t'$. In particular, if $\mathbb{O}(\omega) = 1$, the decryption key holder can compute $dec$ as:

$$\begin{aligned} &\textstyle\prod_{j\in J} (e(c'_{4,j}, dk_4) \cdot e(c'_{5,j}, dk_{5,j}) \cdot e(c'_{6,j}, dk_{6,j}))^{w_j} \cdot e(c'_7, dk_7) \\ &= \textstyle\prod_{j\in J} (e(w^{\lambda'_j+\lambda''_j} v^{s_j+s'_j}, g^{r+r'}) e((u^{\pi(j)}h)^{-(s_j+s'_j)}, g^{r_j+r'_j}) \cdot \\ &\quad e(g^{s_j+s'_j}, (u^{\psi_j}h)^{r_j+r'_j} v^{-(r+r')}))^{w_j} e((u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{s+s'}, g^{\tau+\tau'}) \\ &= \textstyle\prod_{j\in J} (e(w^{\lambda'_j+\lambda''_j}, g^{r+r'})^{w_j} e((u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{s+s'}, g^{\tau+\tau'}) \\ &= e(g,w)^{(r+r')\sum_{j\in J} w_j(\lambda'_j+\lambda''_j)} e(g, u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{(\tau+\tau')(s+s')} \\ &= e(g,w)^{(r+r')(s+s')} \cdot e(g, u_0 \textstyle\prod_{k\in\mathcal{V}} u_k)^{(\tau+\tau')(s+s')}. \end{aligned}$$

If $\mathbb{S}(\psi) = 1$ and $t = t'$, the decryption key holder can compute $num$ as:

$$
\begin{aligned}
&\prod_{i\in I}(e(c'_1, dk_{1,i}) \cdot e(c'_{2,i}, dk_{2,i}) \cdot e(c'_{3,i}, dk_{3,i})/dec)^{v_i}\\
&= \prod_{i\in I}(e(g^{s+s'}, g^{\lambda_i} w^{r_i+r'_i} w^{r+r'}(u_0 \prod_{k\in\mathcal{V}} u_k)^{\tau+\tau'})e(g^{s_i+s'_i},\\
&\quad (\tilde{u}^{\rho(i)}\tilde{h})^{-(r_i+r'_i)})e((\tilde{u}^{\omega_1}\tilde{h})^{s_i+s'_i} w^{-(s+s')}, g^{r_i+r'_i})/den)^{v_i}\\
&= \prod_{i\in I}(e(g^{s+s'}, g^{\lambda_i} w^{r+r'}(u_0 \prod_{k\in\mathcal{V}} u_k)^{\tau+\tau'})/den)^{v_i}\\
&= \prod e(g,g)^{\sum_{i\in I} v_i\lambda_i(s+s')}\\
&= e(g,g)^{\alpha(s+s')}.
\end{aligned}
$$

The message $m$ can be recovered as:

$$c'_0/num = m \cdot e(g,g)^{\alpha(s+s')}/e(g,g)^{\alpha(s+s')} = m.$$

### 4.3 Security Analysis

**Theorem 1.** *The proposed DP-RABE scheme is secure if the decisional BDH assumption and the modified q assumption hold.*

Before giving a brief sketch of our security proof, we suggest readers to revisit RABE [47], [48] and DP-ABE [46]. Our security proof combines the proofs from the above scheme. The security proof in our proposed scheme can be divided into two components: attribute-based component and time-based component. The attribute-based component is based on DP-ABE [46] from [36], [37], and this component is secure if the modified $q$ assumption holds. The time-based component is based on RABE schemes [47], [48], and this component is secure if the decisional BDH assumption holds. Hence, our proposed scheme is secure if the modified $q$ and the decisional BDH assumptions hold.

The sketch of the proof is that we can construct a simulator $\mathcal{B}$ to break decisional BDH assumption (referring assumption in Waters's IBE [45]) or modified $q$ assumption (referring assumption in KP-ABE and CP-ABE [47], [48]) with interacting of $\mathcal{A}$ which can break our proposed scheme. Before the beginning of reduction, $\mathcal{B}$ chooses a random bit $rec \in \{0,1\}$ to guess the role of $\mathcal{A}$.

If $rev = 0$, $\mathcal{A}$ plays the non-revoked users. For each key generation query, $\mathcal{B}$ simulates the ABE component as [47], [48] except the embedding master secret key is $\alpha - \alpha_\theta$ rather than $\alpha$ in the original proof, where $\alpha_\theta$ is the secret information in $st$. For each key update query, $\mathcal{B}$ runs the key update algorithm since $\alpha_\theta$ is known. For decryption key generation oracle, $\mathcal{B}$ simulates the decryption key based on key generation oracle and key update oracle. For revocation oracle, $\mathcal{B}$ runs the revocation algorithm to update the revocation list. In the challenge phase, $\mathcal{B}$ forwards challenge messages $(m_0, m_1)$ to the underlying ABE schemes and simulates the time-based component itself to return the challenge ciphertext to $\mathcal{A}$. $\mathcal{A}$ then submits a bit $b'$ as the guessing of challenge message. $\mathcal{B}$ forwards this bit $b'$ to underlying ABE schemes.

If $rev = 1$, $\mathcal{A}$ acts as the revoked users. For each key generation query, $\mathcal{B}$ runs the key generation algorithm using $sk_\theta$ as the master secret key. For each key update query, $\mathcal{B}$ encodes the timestamp to the bit representation then forwards to an underlying IBE scheme to answer the query. For decryption key generation oracle, $\mathcal{B}$ simulates the decryption key based on key generation oracle and key update oracle. For revocation oracle, $\mathcal{B}$ runs the revocation algorithm to update the revocation list. In the challenge phase, $\mathcal{B}$ forwards challenge messages $(m_0, m_1)$ to the underlying IBE scheme and simulates the attribute-based component itself to return the challenge ciphertext to $\mathcal{A}$. $\mathcal{A}$ then submits a bit $b'$ as the guess of challenge message. $\mathcal{B}$ forwards this bit $b'$ to underlying IBE schemes.

Therefore, if there exists an adversary $\mathcal{A}$ can break our proposed DP-RABE scheme, we can simulate an algorithm $\mathcal{B}$ to break $\mathcal{C}_{ibe}$ with the decisional BDH assumption, and $\mathcal{C}_{kp}$ and $\mathcal{C}_{cp}$ with the modified $q$ assumption. Please refer to Appendix B, available in the online supplemental material, for the details of the security proof.

## 5 Efficiency Analysis

In this section, we give the theoretical analysis and experimental simulation between previous solutions and ours. For theoretical analysis, we focus on the comparison between the existing solutions [44], [46], [49], [51], [52], [53], [55] and ours. For experimental simulation, we analyze the recent works [46], [49] and ours since they are sharing system with either efficient user revocation or expressive fine-grained access control in the standard model. Specifically, [49] provides an efficient revocation mechanism with logarithmic user revocation and no secure channel, and [46] applies role-based and content-based access control simultaneously with comparable performances among the other solutions.

### 5.1 Theoretical Analysis

Table 2 shows the theoretical analysis between the relevant EMR sharing systems and ours from two angles: computational complexity and space complexity. For computational complexity, we focus on setup, key generation, encryption, and key update generation algorithms. For the space complexity, we focus on the system parameter, secret key, and ciphertext.

In the system setup phase, only [52], [55] and [46] have the constant computational complexity to generate the system parameter, and other solutions base on either the set of attributes or the policies depend on different access policy (e.g., KP-ABE and CP-ABE). Besides, [52], [55] and [46] also have the constant-size system parameter. To achieve constant costs, [55] and [52] use hash functions as random oracles to achieve the constant-size system parameter, where the random oracle is an ideal model and does not exist in the real world. [46] applies the constant-size ABE [37] to build DP-ABE in the standard model without user revocation. Hence, they are not desirable solutions in EMR sharing system. Our solution applies DP-ABE as in [46] and efficient user revocation without any secure channel as in [49]. Thus we have an additional factor of $\log \mathbb{N}$ due to the tree-based revocation list as shown in Algorithm 1.

In the secret key generation phase, all existing sharing systems have computational complexity and space complexity based on either the set of attributes or the policies except [46] and ours since the DP-ABE has been applied to manage the key attribute-based component. Besides, to achieve efficient revocation mechanism, [49] and ours slightly increase the key size in the order of $\log \mathcal{N}$ since all users are mapped to the leaves of a tree-based structure and the users' secret key

TABLE 2
Theoretical Analysis Among Existing Electronic Medical Sharing Systems

| | Computational Complexity | | | | Space Complexity | | |
|---|---|---|---|---|---|---|---|
| | System Setup | Secret Key Generation | Encryption | Key Update Generation | System Parameter | Secret Key | Ciphertext |
| *[44]* | $\mathcal{O}(d \cdot \psi)$ | $\mathcal{O}(d \cdot \psi)$ | $\mathcal{O}(2^d \cdot \mathbb{S})$ | $\mathcal{O}(\mathcal{N})$ | $\mathcal{O}(d \cdot \psi)$ | $\mathcal{O}(d \cdot \psi)$ | $\mathcal{O}(2^d \cdot \mathbb{S})$ |
| *[51]* | $\mathcal{O}(\psi)^{[*]}$ | $\mathcal{O}(\psi)^{[*]}$ | $\mathcal{O}(\mathbb{S})^{[*]}$ | $N/A$ | $\mathcal{O}(\psi)^{[*]}$ | $\mathcal{O}(\psi)^{[*]}$ | $\mathcal{O}(\mathbb{S})^{[*]}$ |
| *[53]* | $\mathcal{O}(n_{\mathsf{mk}} \cdot \mathbb{S})$ | $\mathcal{O}(n_{\mathsf{mk}} \cdot \psi)$ | $\mathcal{O}(\mathbb{S})$ | $\mathcal{O}(\mathcal{N})$ | $\mathcal{O}(n_{\mathsf{mk}} \cdot \mathbb{S})$ | $\mathcal{O}(n_{\mathsf{mk}} \cdot \psi)$ | $\mathcal{O}(\mathbb{S})$ |
| *[55]* | $\mathcal{O}(1)^{[\mathsf{P}]}$ | $\mathcal{O}(n_a \cdot \psi)^{[\mathsf{P}]}$ | $\mathcal{O}(\mathbb{S})^{[\mathsf{P}]}$ | $N/A$ | $\mathcal{O}(1)$ | $\mathcal{O}(n_a \cdot \psi)$ | $\mathcal{O}(\mathbb{S})$ |
| *[52]* | $\mathcal{O}(1)$ | $\mathcal{O}(\psi)^{[*]}$ | $\mathcal{O}(\mathbb{S})$ | $\mathcal{O}(\mathcal{N})$ | $\mathcal{O}(1)$ | $\mathcal{O}(\psi)^{[*]}$ | $\mathcal{O}(\mathbb{S})$ |
| *[49]* | $\mathcal{O}(\log \mathcal{T})$ | $\mathcal{O}(\log \mathcal{N} \cdot \psi)$ | $\mathcal{O}(\mathbb{S})$ | $\mathcal{O}(\log \mathcal{N})$ | $\mathcal{O}(\log \mathcal{T})$ | $\mathcal{O}(\log \mathcal{N} \cdot \psi)$ | $\mathcal{O}(\mathbb{S})$ |
| *[46]* | $\mathcal{O}(1)$ | $\mathcal{O}(\psi + \mathbb{O})$ | $\mathcal{O}(\mathbb{S} + \omega)$ | $N/A$ | $\mathcal{O}(1)$ | $\mathcal{O}(\psi + \mathbb{O})$ | $\mathcal{O}(\mathbb{S} + \omega)$ |
| *Ours* | $\mathcal{O}(\log \mathcal{T})$ | $\mathcal{O}(\log \mathcal{N} \cdot (\psi + \mathbb{O}))$ | $\mathcal{O}(\mathbb{S} + \omega)$ | $\mathcal{O}(\log \mathcal{N})$ | $\mathcal{O}(\log T)$ | $\mathcal{O}(\log \mathcal{N} \cdot (\psi + \mathbb{O}))$ | $\mathcal{O}(\mathbb{S} + \omega)$ |

*$d$ denotes the depth of the key structure, which is used for access control based on tree structure rather than LSSS.*
*$n_{\mathsf{mk}}$ denotes the number of nodes in the Merkle hash tree.*
*$n_a$ denotes the number of key authorities.*
*$\mathcal{N}$ denotes the number of system users.*
*$\mathcal{T}$ denotes the system bounded lifetime.*
*$\psi$ and $\mathbb{O}$ are the set of attributes and the access policies associated with the secret key.*
*$\mathbb{S}$ and $\omega$ are the access policies and the set of attributes associated with the ciphertext.*
[P] *means protocol which requires multiple runs or many interactions among different entities.*
[$*$] *means unknown accurate complexity since only the (semi-)generic construction rather than the concrete scheme is provided.*

depends on the nodes from the root node to the corresponding leaf node [11].

In the data encryption phase, the complexity of all existing solutions is based on the performances in the secret key generation phase. It is worth to notice that the coefficient $\log \mathcal{N}$ in [49] and ours is eliminated since the time-based component is a constant-size parameter (e.g., ciphertext to a timestamp in the form of a binary string).

For user revocation, we only consider the computational cost since the existing solutions (expect [49]) with user revocation apply either the secure channel to distribute the new secret key or secret key based on re-encryption methodology. Hence, the overhead for key distribution is high, such as additional cost to build and maintain the secure channel. [49] and our solution use directly revocation [11], which have no overhead and the key-updating material on the public channel. Besides, by applying a tree-based revocation list, our solution only requires logarithmic complexity depends on the number of users with sacrificing the performance of secret key.

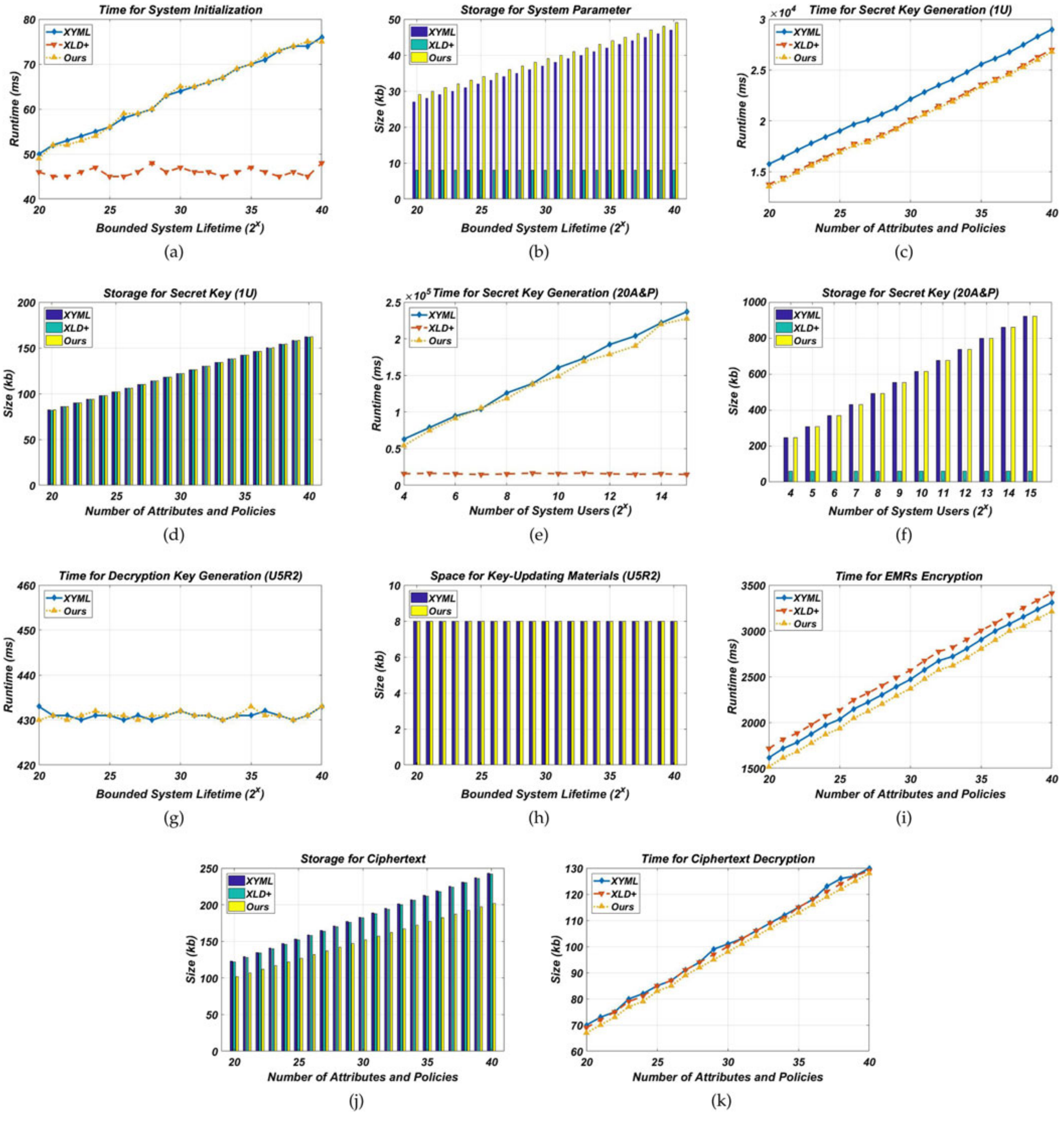
**Remark 7.** The secret key is distributed from KGC when the user joins the system. The key-updating material is distributed in each revocation epoch. Hence, the logarithmic key-updating materiel improves overall performance significantly, even sacrificing the performance of secret key from the constant cost to the logarithmic cost.

### 5.2 Experimental Simulation

We have implemented very recent works [46], [49] (refer to XYML and XLD+) and our scheme on a personal computer with 64-bit Windows 10, 3.60 GHz Intel(R) Core(TM) i7-490 CPU and 24GB memory with JPBC library. Type A elliptic curve in the symmetric setting is used in our implementation. Specifically, we set the prime number $p$ to be 160-bit, the elements in $\mathbb{G}$ to be 512-bit, and the items in $\mathbb{G}_T$ to be 1024-bit. The result of experimental performance is given in Fig. 9. Overall, our scheme has comparable performance to very recently EMR sharing systems [46], [49] with additionally desirable functionalities, such as dual-policy access control and user revocation.

Figs. 9a and 9b give the performance about system initialization by increasing the system bounded time from $2^{20}$ to $2^{40}$. In Fig. 9a, the running time of XYML and ours are similar since the efficient revocation mechanism is applied. The efficient revocation sacrifices the system parameter based on the factor of $\log \mathcal{T}$ for generating parameters for the time-based component. Although the computational cost of XYML and ours as the growth of the bounded system lifetime, the overhead is still small, even for the large case. For EMR sharing systems, we think the $2^{20}$ bounded system lifetime is enough since it supports about 119 years if the revocation epoch is 1 hour ($2^{20}/365/24 \approx 119.7$ years). Alternatively, XLD+ has almost constant time since the constant-size ABE has been used without user revocation. In Fig. 9b, the size of the system parameter of XYML and ours are the growth of the bounded system lifetime, and XLD+ keeps stable, where XYML has less storage than ours since our scheme requires two more elements in $\mathbb{G}$ to support DP-ABE rather than CP-ABE only. It is worth to notice that XYML, XLD+ and ours base on constant-size ABE supporting the large universe with universe domain $\mathbb{Z}_p$, which means that the size of attribute universe does not affect the running time to initialize the system. Therefore, we do not provide the performances based on the number of attributes and policies for system initialization.

Figs. 9c and 9f present the performance about secret key by increasing the number of attributes and policies from $2^{20}$ to $2^{40}$ as well as the number of users from $2^4$ to $2^{15}$, where *"1U"* in Figs. 9c and 9d means the number of key for a single to eliminate the factor of number of system users for neutral results and *"20A&P"* in Figs. 9e and 9f means the number of attributes and policies are 20. In Fig. 9c, the computational cost of all three schemes are the growth of attributes and policies since the underlying scheme of them is based on ABE with linear complexity based on the size of attributes and policies. The cost of ours is the same as XLD+ since we all based on DP-ABE. XYML takes a little longer time since CP-ABE takes extra time to generate the secret key. In Fig. 9d, the data storage of all three schemes are the growth of the number of



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Fig. 9. Experimental performances.

attributes and policies, and each of them has almost similar size secret key. In Fig. 9e, XYML and ours are growing as the increasing of the number of system users since every user is required to store additional keys about $\log \mathcal{N}$ keys for efficient revocation mechanism. XLD+ keeps stable since it has no user revocation. In Fig. 9f, XYML and ours have a similar performance that is growing as the increasing of the number of system users, and XLD+ has the same size of secret key no matter how many users in the system. Note that the key generation in XLD+ includes the time in the data user side and the time in the edge server, we sum them as the running time for secret key generation.

Figs. 9g and 9h illustrate the performance of user revocation by increasing the bounded system lifetime from $2^{20}$ to $2^{40}$ between XYML and ours since XLD+ cannot provide user revocation. In Fig. 9g, we can find that XYML and ours have the almost same time for decryption key generation. In Fig. 9h, we also have similar key-updating materials for distribution in the public channel. Note that we do not provide the performance based on the different number of revoked users since it is not accurate. The revocation mechanism is based on tree-based revocation mechanism, as shown in Algorithm 1, which leads to different performances, even the number of revoked users is the same. Specifically, the performance of user revocation depends on the location of the user's identity in the tree structure. For example, a case with a single subtree root, including a large number of revoked users, has better performance than a case with multiple subtree roots containing the small number of revoked users. We suggest the readers to the paper [11] for the details.

Figs. 9i and 9j display the performance about EMRs encryption by increasing the bounded system lifetime from $2^{20}$ to $2^{40}$. In Fig. 9i, the trends of all three schemes are the same since the underlying ABE scheme requires the linear time depending on the number of attributes and policies. Although the gaps among XYML, XLD+ and ours are tiny, our scheme takes less time than others since CP-ABE in XYML takes more time and XLD+ has overhead to outsourced decryption. In Fig. 9j, the performance is quite similar in Fig. 9i. Our scheme has less ciphertext size since CP-ABE in XYML takes more space to ciphertexts and XLD+ has overhead to outsourced ciphertexts. Note that the ciphertext decryption in XLD+ includes the time in the data user side and the time in the edge server, we sum them as the running time for secret key generation.

Fig. 9k presents the performance about decryption by increasing the bounded system lifetime from $2^{20}$ to $2^{40}$. The trends of all three schemes are quite similar, and the gaps among them are very tiny. Our scheme takes less time than others since our solution based on DP-ABE rather than CP-ABE in XYML and has no overhead for outsourced decryption rather than overhead for outsourcing in XLD+.

Therefore, our scheme is comparable to very recently EMR sharing systems [46], [49] with extra functionalities, such as dual-policy access control and user revocation.

**Remark 8.** The number of attributes and policies in Fig. 9 means the number of attributes and policies associated with a secret key or a ciphertext, respectively. For example, "20" in x-axis means 20 attributes and 20 policies associated with a secret key or a ciphertext. To provide a neutral result to XYML (CP-ABE), we consider "20" in x-axis means 40 attributes to a key or 40 policies to a ciphertext.

## 6 Conclusion

In this paper, we proposed an EMR sharing system by introducing a dual-policy revocable attribute-based encryption scheme with decryption key exposure resistance and revocable storage. We provided the formal definition and security model for our proposed scheme. The security analysis and experimental evaluation confirm that the proposed scheme is secure and practical, which is suitable for real-world EMR systems.

## References

[1] AdvancedMD. Accessed: Dec. 15, 2019. [Online]. Available: https://www.advancedmd.com/

[2] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. J. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," in *Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices*, 2011, pp. 75–86.

[3] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.

[4] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Proc. IMA Int. Conf. Cryptogr. Coding*, 2009, pp. 278–300.

[5] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2009, pp. 168–185.

[6] N. Attrapadung and S. Yamada, "Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings. in *Proc. Cryptograph. Track RSA Conf.*, 2015, pp. 87–105.

[7] AWS, "Amazon EMR." Accessed: Dec. 15, 2019. [Online]. Available: https://aws.amazon.com/emr/

[8] A. Bakas, H.-V. Dang, A. Michalas, and A. Zalitko, "The cloud we share: Access control on symmetrically encrypted data in untrusted clouds," *IEEE Access*, vol. 8, pp. 210 462–210 477, 2020.

[9] BIANCA BANOVA, "The impact of technology on healthcare." Accessed: Dec. 15, 2019. [Online]. Available: https://www.aimseducation.edu/blog/the-impact-of-technology-on-healthcare/

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, 2007, pp. 321–334.

[11] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 417–426.

[12] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2005, pp. 440–456.

[13] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," In *Proc. Annu. Int. Cryptol. Conf.*, 2001, vol. 2139, pp. 213–229.

[14] J. Camenisch, D. Derler, S. Krenn, H. C. Pöhls, K. Samelin, and D. Slamanig, "Chameleon-hashes with ephemeral trapdoors - and applications to invisible sanitizable signatures," in *Proc. IACR Int. Workshop Public Key Cryptogr.*, 2017, pp 152–182.

[15] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proc. IACR Int. Workshop Public Key Cryptogr.*, 2004, pp. 207–222.

[16] J. Y. Chen, H. Xu, P. Shi, A. Culbertson, and E. M. Meslin, "Ethics and privacy considerations for systems biology applications in predictive and personalized medicine," in *Proc. Handbook Res. Comput. Syst. Biol. - Interdiscipl. Appl.*, 2011, pp. 1–27.

[17] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2016, pp. 570–587.

[18] D. Derler, K. Samelin, D. Slamanig, and C. Striecks, "Fine-grained and controlled rewriting in blockchains: Chameleon-hashing gone attribute-based," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019.

[19] DrChrono EHR. Accessed: Dec. 15, 2019. [Online]. Available: https://www.drchrono.com/

[20] M. Frakes and A. B. Jena, "Does medical malpractice law improve health care quality?," *J. Public Econ.*, vol. 143, pp. 142–158, 2016.

[21] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Proc. Annu. Int. Cryptol. Conf.*, 1999, pp. 537–554.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data. in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[23] L. Ibraimi, M. Petkovic, S. Nikova, P. H. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Proc. Int. Workshop Inf. Secur. Appl.*, 2009, pp. 309–323.

[24] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, 2001.

[25] K. Lee, S. G. Choi, D. H. Lee, J. H. Park, and M. Yung, "Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2013, pp. 235–254.

[26] A. B. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2011, pp. 547–567.

[27] A. Michalas, "The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Chih-Cheng Hung and George A. Papadopoulos, Eds, 2019, pp. 146–155.

[28] P. Mohassel, "One-time signatures and chameleon hash functions," in *Proc. Int. Workshop Sel. Areas Cryptogr.*, 2010, pp. 302–319.

[29] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” *Decentralized Bus. Rev.*, p. 21260, 2008.

[30] D. Naor, M. Naor, and J. Lotspiech, “Revocation and tracing schemes for stateless receivers,” in *Proc. Annu. Int. Cryptol. Conf.*, 2001, pp. 41–62.

[31] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, “Auditable $\sigma$-time outsourced attribute-based encryption for access control in cloud computing,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 1, pp. 94–105, Jan. 2018.

[32] J. Ning, Z. Cao, X. Dong, and L. Wei, “White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively,” *IEEE Trans. Dependable Sec. Comput.*, vol. 15, no. 5, pp. 883–897, Sep.–Oct. 2018.

[33] J. Ning, J. Chen, K. Liang, J. K. Liu, C. Su, and Q. Wu, “Efficient encrypted data search with expressive queries and flexible update,” *IEEE Trans. Serv. Comput.*, early access, Jun. 25, 2020, doi: 10.1109/TSC.2020.3004988.

[34] B. Qin, R. H. Deng, Y. Li, and S. Liu, “Server-aided revocable identity-based encryption,” in *Proc. Eur. Symp. Res. Comput. Secur.*, 2015, pp. 286–304.

[35] B. Qin, Q. Zhao, D. Zheng, and H. Cui, “Server-aided revocable attribute-based encryption resilient to decryption key exposure,” in *Proc. Int. Conf. Cryptol. Netw. Securi.*, 2017, pp. 504–514.

[36] Y. Rouselakis and B. Waters,. “New constructions and proof methods for large universe attribute-based encryption,” *IACR Cryptol. ePrint Arch.*, vol. 2012, 2012, Art. no. 583.

[37] Y. Rouselakis and B. Waters, “Practical constructions and new proof methods for large universe attribute-based encryption,” in *Proc, ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 463–474.

[38] A. Sahai, “Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security,” in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, 1999, pp. 543–553.

[39] A. Sahai, H. Seyalioglu, and B. Waters, “Dynamic credentials and ciphertext delegation for attribute-based encryption,” in *Proc. Annu. Cryptol. Conf.*, 2012, pp. 199–217.

[40] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Technol.*, 2005, vol. 3494, pp. 457–473.

[41] SelectHub, “Future of electronic medical records: Experts predict EMR trends in 2019.” Accessed: Dec. 15, 2019. [Online]. Available: https://selecthub.com/medical-software/emr/electronic-medical-records-future-emr-trends/

[42] J. H. Seo and K. Emura, “Revocable identity-based encryption revisited: Security model and construction. in *Proc. Int. Workshop Public Key Cryptogr.*, 2013, pp. 216–234.

[43] H. Shacham and B. Waters, “Compact proofs of retrievability,” in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2008, pp. 90–107.

[44] Z. Wan, J-. Liu, and R. H. Deng, “HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing,” *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 743–754, Apr. 2012.

[45] B. Waters, “Efficient identity-based encryption without random oracles,” in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2005, pp. 114–127.

[46] S. Xu, Y. Li, R. H. Deng, Y. Zhang, X. Luo, and X. Liu, “Lightweight and expressive fine-grained access control for healthcare internet-of-things,” *IEEE Trans. Cloud Comput.*, early access, Aug. 20, 2019, doi: 10.1109/TCC.2019.2936481.

[47] S. Xu, G. Yang, and Y. Mu, “Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation,” *Inf. Sci.*, vol. 479, pp. 116–134, 2018.

[48] S. Xu, G. Yang, Y. Mu, and R. H. Deng, “Secure fine-grained access control and data sharing for dynamic groups in the cloud. *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 8, pp. 2101–2113, Aug. 2018.

[49] S. Xu, G. Yang, Y. Mu, and X. Liu, “A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance,” *Future Gener. Comp. Syst.*, vol. 97. pp. 284–294, 2019.

[50] Sh. Xu, Y. Zhang, Y. Li, X. Liu, and G. Yang, “Generic construction of elgamal-type attribute-based encryption schemes with revocability and dual-policy,” in *Proc. Int. Conf. Secur. Privacy Commun. Syst.*, 2019, pp. 184–204.

[51] J.-J. Yang, J. Li, and Y. Niu, “A hybrid solution for privacy preserving medical data sharing in the cloud environment,” *Future Gener. Comp. Syst.*, vol. 43-44, pp. 74–86, 2015.

[52] Y. Yang, X. Liu, R. H. Deng, and Y. Li, “Lightweight sharable and traceable secure mobile health system,” *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 78–91, Jan./Feb. 2020.

[53] L.-Y. Yeh, P.-Y. Chiang, Y.-L. Tsai, and J.-L. Huang, “Cloud-based fine-grained health information access control framework for lightweight IoT devices with dynamic auditing and attribute revocation,” *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 532–544, 2018.

[54] Y. Zhang, J. Shu, X. Liu, J. Li, and D. Zheng, “Comments on a large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1287–1290, Feb. 2019.

[55] Y. Zhang, D. Zheng, and R. H. Deng, “Security and privacy in smart health: Efficient policy-hiding attribute-based access control,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

**Shengmin Xu** received the PhD degree in cryptography from the University of Wollongong, Australia, in 2018. He is currently a research scientist with Singapore Management University, Singapore. He was a research fellow with Singapore University of Technology and Design, Singapore. He has authored or coauthored more than 30 research papers in top international conferences and journals, including ESORICS, ACM ACSAC, ACM ASIACCS, *IEEE Transactions on Information Forensics and Security*, and the IEEE *Transactions on Dependable and Secure Computing*. His research interests include information security, cloud computing, and blockchain.

**Jianting Ning** received the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2016. He is currently a professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Computer and Cyber Security, Fujian Normal University, China. He was a research scientist with the School of Information Systems, Singapore Management University and a research fellow with the Department of Computer Science, National University of Singapore. He has authored or coauthored papers in major conferences or journals such as ACM CCS, Asiacrypt, ESORICS, ACSAC, *IEEE Transactions on Information Forensics and Security*, and the *IEEE Transactions on Dependable and Secure Computing*. His research interests include applied cryptography and information security.

**Yingjiu Li** is currently a ripple professor with the Computer and Information Science Department, University of Oregon. He has authored or coauthored more than 140 technical papers in international conferences and journals, and served in the program committees for over 80 international conferences and workshops, including top-tier cybersecurity conferences and journals. His research interests include IoT security and privacy, mobile and system security, applied cryptography and cloud security, and data application security and privacy.

**Yinghui Zhang** is currently a professor with the National Engineering Laboratory for Wireless Security (NELWS), Xi’an University of Posts and Telecommunications since 2018 and a research fellow with Singapore Management University. He has authored or coauthored more than 100 research articles including AsiaCCS, ACM CSUR, *IEEE Transactions on Dependable and Secure Computing*, and the *IEEE Transactions on Services Computing*, *Computer Networks*, *Computers & Security*. His research interests include public key cryptography, cloud security, and wireless network security.

**Guowen Xu** received the PhD degree in cyberspace security from the University of Electronic Science and Technology of China (UESTC) in 2020. He is currently a research fellow with Nanyang Technological University, Singapore. As the first author, he has authored or coauthored more than 20 papers in top international conferences and journals, including ACM CCS, ACM ACSAC, ACM ASIACCS, *IEEE Transactions on Dependable and Secure Computing,* and the *IEEE Transactions on Information Forensics and Security*. His research interests include Secure Outsourcing Computing and privacy-preserving issues in Deep Learning. He was the recipient of Best Paper Award of the 26th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2020) , and the IEEE INFOCOM Student Conference Award.

**Xinyi Huang** received the PhD degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2009. He is currently a professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Computer and Cyber Security, Fujian Normal University, China. His research interests include cryptography and information security. He has authored or coauthored more than 130 research papers in refereed international conferences and journals. His work has been cited more than 9000 times at Google Scholar. He on the editorial board of *International Journal of Information Security*. He was the program or general chair or program committee member in more than 120 international conferences.

**Robert H. Deng** (Fellow, IEEE) is currently the AXA chair professor of cybersecurity and the director of the Secure Mobile Centre, School of Information Systems, Singapore Management University (SMU). His research interests include data security and privacy, cloud security and Internet of Things security. His professional contributions include an extensive list of positions in several industry and public services advisory boards, editorial boards and conference committees. These include the editorial boards of IEEE Security & Privacy Magazine, the *IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology*, and the Steering Committee chair of the ACM Asia Conference on Computer and Communications Security. He was the recipient of the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium.