Published in IEEE Transactions on Cloud Computing (2023) 11 (3). DOI: 10.1109/TCC.2022.3208168.

Balancing Privacy and Flexibility of Cloud-Based Personal Health Records Sharing System

Yudi Zhang, Fuchun Guo, Willy Susilo, *Fellow, IEEE*, and Guomin Yang, *Senior Member, IEEE*

Abstract—The Internet of Things and cloud services have been widely adopted in many applications, and personal health records (PHR) can provide tailored medical care. The PHR data is usually stored on cloud servers for sharing. Weighted attribute-based encryption (ABE) is a practical and flexible technique to protect PHR data. Under a weighted ABE policy, the data user's attributes will be "scored", if and only if the score reaches the threshold value, he/she can access the data. However, while this approach offers a flexible access policy, the data owners have difficulty controlling their privacy, especially sharing PHR data in collaborative e-health systems. This article aims to find a balance between privacy and flexibility and proposes an AND-weighted ABE scheme in cloud-based personal health records sharing systems. The proposed scheme can meet both privacy and flexibility. Only when the data user satisfies the scored-based policy and is in the specified organization(s), can the data user access the PHR data. Besides, we give the security proof and the performance evaluation of the proposed scheme. The security proof and performance analysis show that the proposed scheme can efficiently and securely share PHR data in cloud service.

Index Terms—Attribute-based encryption, privacy-preserving and flexible access control, secure PHR date sharing, weighted attribute

1 Introduction

In the post-pandemic era, people are increasingly reliant on digital health (especially e-health), and digital healthcare is gaining popularity. Since most personal health record (PHR) data is stored on cloud servers, security concerns cannot be ignored while enjoying its convenience. A survey by Statista [1] shows that approximately 90% of all users who experienced a cloud storage security issue had problems with a permissive storage policy. To share the PHR data to multiple data users, the patient usually utilizes attribute-based access policies to restrict user access to the data. Therefore, the data user can access the PHR data only if his/her attributes satisfy the access policy.

In many practical applications, different attributes may have different weight values, while evaluating whether a user is eligible is a score-based mechanism. For example, before a bank issues the credit card to the user, the bank will consider various factors such as age, occupation, deposit, assets, and housing. It scores each item according to certain standards, then aggregates it into a cumulative credit score. If the score has reached the minimum standard, the bank will issue a credit card to the user. Similarly, some Internet auction systems also score sellers based on their history and other factors. Therefore, this mechanism will bring more flexibility and convenience to data sharing. Suppose that a patient wants to share their PHR data according to the access policy shown in Table 1. There are several attributes, where different attributes are given different weights. To access the PHR data, the data user (e.g., a doctor, a nurse, or a researcher) should hold some of the following attributes, and the sum of the weight must be greater or equal to a specific weight. Without losing generality, in the following example, we set this specific weight to be 8.

At first sight, the patient can utilize the weighted attribute-based encryption (ABE) scheme proposed by Herranz et al. [2] to accomplish the above purpose. The patient first encrypts the PHR data under the specific attributes, the weight values of each attribute, and the threshold weight which are shown in Table 1. Then they upload the ciphertext to the cloud server. The authority will issue the secret keys to valid users according to their attributes. Suppose there are three valid data users as depicted in Table 2, who want to access the protected PHR data. As we can see from Table 2, Alice's attributes: "Nurse," "Female," and "Surgery" are in Table 1, and the weight sum (score) is 10. Therefore, Alice is eligible to decrypt the ciphertext and obtain the PHR data. Similarly, we can conclude that Bob can access the data, but Carl cannot do so.

However, the weighted ABE only requires the sum of the weights of the intersection of the attributes of the data user and those in the policy is greater than the threshold value. This kind of access policy is too powerful, which may lead to the misuse of the PHR data. In Fig. 1, Bob and Andy are in different organizations, but their attributes are the same. If one of them can satisfy the access policy, another one can decrypt the ciphertext too. For privacy reasons, the patient may just want to share the PHR data to one of the organizations.

An intuitive way is to set relative high weight values of the target organization(s). Nevertheless, it cannot control the policy precisely. For some users who are not in the specified target organization(s), their weight value may reach

- *The authors are with the Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia. E-mail: {yudi, fuchun, wsusilo, gyang}@uow.edu.au.*

Manuscript received 25 April 2022; revised 1 August 2022; accepted 17 September 2022. Date of publication 20 September 2022; date of current version 6 September 2023.
This work was supported by Australian Research Council Discovery Project under Grant DP200100144.
(Corresponding author: Yudi Zhang.)
Recommended for acceptance by Z. Wang.
Digital Object Identifier no. 10.1109/TCC.2022.3208168

TABLE 1
A Score-Based Access Policy

Attribute	Weight	Attribute	Weight
Doctor	5	Internal medicine	3
Researcher	5	Surgery	5
Nurse	2	ENT Clinic	2
Male	2	Dermatology	3
Female	3	Neurosurgery	2
		Minimal value: **8**	

The data user can access the PHR data if his/her attributes score can achieve the minimum score according to the weight specification in this table.

the threshold value, or the weight of some users in the target organization(s) cannot reach the threshold value.

The goal of our work in this paper is to balance the privacy and the flexibility of ABE for cloud-based PHR sharing system. This scheme utilizes an AND-gate policy to protect user's privacy, and a threshold policy to achieve the flexible access control. The patient can not only specify the weight values of each attribute and a threshold summation, but also specify some specific attributes to limit the permission of the data users. We use the following example to further illustrate our approach.

As shown in Fig. 2, in a collaborative e-health system, the patient shares the PHR data to the doctors, nurses via the cloud server. Suppose the weight values of each attribute are ranged in $[1, 2, \ldots, 10]$. There is a PHR data encrypted under the policy *{Doctor: 6, Nurse: 3, Internal medicine: 5, ENT Clinic: 3, Epidemiology: 5}; threshold value: 8;* **AND***Hospital A;* **AND***Institute*. For the data users, only the doctor in hospital A with ENT clinic and also in the institute can access the data. Doctors or nurses in other hospitals cannot decrypt it even if the weight of the attributes satisfies the policy. However, if we delete *Institute* from the access policy, then the nurse in hospital A can decrypt it.

1.1 Our Contributions

The existing weighted ABE schemes are too powerful for the data users, especially in the collaborative e-health system. The patients' PHR data is stored in the cloud storage. If the PHR data is encrypted under the conventional weighted ABE, there are many data users from different organizations who can satisfy the access policy, since there will be users with the same attributes in different organizations. Like the example in Fig. 2, if the ciphertext is encrypted under the policy *{Doctor: 6, Nurse: 3, Internal medicine: 5, ENT Clinic: 3, Female: 3, Male: 2}; threshold value: 8*, both Andy and Bob can decrypt the ciphertext. However, the patient may only want to share the PHR data to the data users in the hospital other than in the institute. To balance the privacy and the flexibility, we design an AND-weighted ABE scheme for PHR sharing system, which allows the patient to choose the organization(s) that wants to share. In our proposed system, the access policy not only depends on the weight values of the attributes, but also depends on some attributes which must be possessed by the eligible data users.

TABLE 2
A User Set in the PHR Sharing System

User\Attribute	Occupation	Gender	Department	
Alice	Nurse	Female	Surgery	√
Bob	Doctor	Male	Internal medicine	√
Carl	Researcher	Male	Pediatrics	×

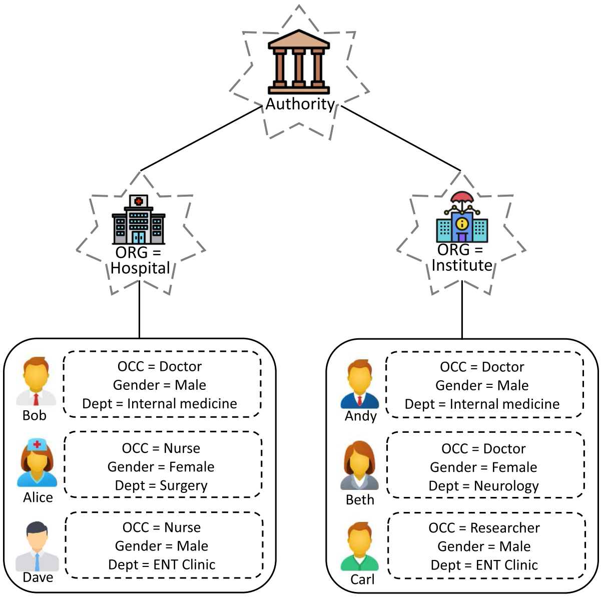


Fig. 1. Diverse attributes in the collaborative e-health system.

In summary, the main contribution of our work is to balance the privacy and the flexibility of the PHR sharing system in cloud environment. The patient can fix the organization(s) and choose different weight values for an attribute in the ciphertext during encryption according to different scenarios. Then, we give the security analysis of our proposed scheme. We prove that our scheme is CPA (chosen-plaintext attacks) secure under augmented multi-sequence of exponents decisional Diffie-Hellman ($\mathsf{aMSE-DDH}$) assumption. Finally, we implement our scheme by using the RELIC cryptographic meta-toolkit [3] on a personal computer (PC) and an Android device. The experimental results demonstrate that our scheme is efficient and practical.

1.2 Organization of This Paper

The rest of this paper is organized as follows. In Section 2, we show the related work about attribute-based encryption and weighted ABE schemes briefly. In Section 3, we describe the system model and the threat model. In Section 4, we show the algorithm definitions and introduce our construction in detail. In Section 5, we present the security analysis. In section 6, we present our evaluation results. Finally, section 7 concludes this paper.

2 Related Work

2.1 Attribute-Based Encryption

The first ABE scheme was designed by Sahai and Waters [4] which is actually a fuzzy identity-based encryption (IBE) scheme. Over that last decades, many scholars focused on

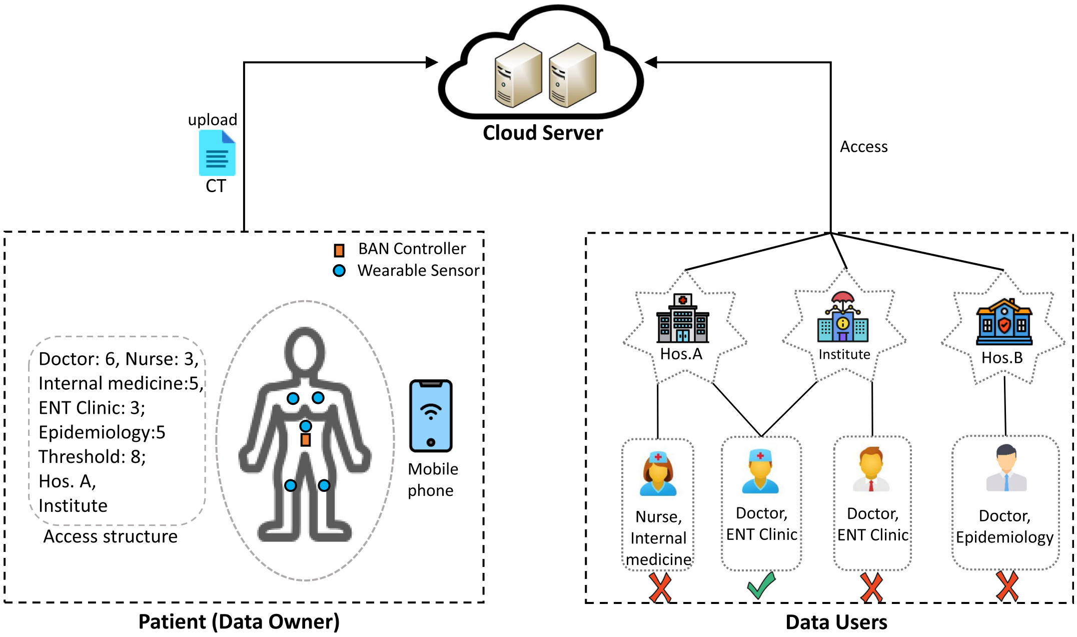


Fig. 2. A functional overview of our proposed system.

it, and a lot of ABE schemes have been proposed. Currently, there are two typical ABE schemes [5], [6]: (1) KP-ABE (key-policy ABE) [7], [8], the access policy is encoded into a user's secret key, and the ciphertext is generated with a related attribute list; and (2) CP-ABE (ciphertext-policy ABE) [9], [10], [11], [12], user's secret key is associated with an attribute list, and the ciphertext specifies an access policy. As mentioned by Bethencourt [13], unlike CP-ABE, in the KP-ABE scheme, the data owner exerts no control over who has access to the shared data. Therefore, in the cloud-based sharing system, CP-ABE gets more attention. Ambrona et al. [14] a new ABE scheme which is proved secure under generic group model.

The first CP-ABE scheme was introduced by Bethencourt et al. [13] in 2007, their scheme supports tree-based access policy, they use a two-level random masking methodology to construct a private key randomization technique. Cheung and Newport [15] proposed CP-ABE scheme in which AND gate policies which access structures are AND gates on positive and negative attributes. They proved their scheme is CPA (chosen-plaintext attacks) secure under DBDH assumption. However, these schemes either support a limited access structure or the security proof should in the generic group model.

Goyal et al. [16] introduced the first construction of CP-ABE scheme, the security proof is based on a number theoretic assumption, and it supports advanced access structures. Following this work, Liang et al. [17] designed a new bounded CP-ABE scheme which can reduce the computation cost during the encryption and decryption phase. Lewko et al. [18] proposed a fully secure ABE scheme, which is proved secure from three static assumptions.

In order to keep the size of the ciphertext from growing linearly with the number of attributes. Herranz et al. [2] proposed an ABE scheme with constant size ciphertexts. Susilo et al. [19] introduced a more efficient ABE scheme with constant size ciphertexts which removes the dummy attributes. In 2020, Yang et al. [20] proposed a modified CP-ABE scheme which the length of ciphertext are constant. Recently, many CP-ABE schemes with different functional features have been proposed [21], [22], [23], [24]. These features can make CP-ABE schemes more practical in the real applications.

2.2 Weighted ABE

Wang et al. [25] proposed a tree-based weighted ABE scheme, which can enhance the expression of attribute. In their scheme, each attribute of the data receiver is given a different weight. When encrypting a message, the data owner assigns a minimum weight value to each attribute, the data receiver can decrypt the ciphertext only when the weight of each attribute are greater than the minimum weight value in the ciphertext. Li et al. [26] introduced an unified weighted attribute-based encryption scheme, it can optimize the encryption process if there is a mutual sub-policy.

Based on Lin et al.'s idea [27], Xue et al. [28] presented a comparable attribute-based encryption scheme, it is also a weighted ABE scheme. They utilized the notion of 0-encoding and 1-encoding to construct a scheme that requires less computational overhead, communication overhead, and storage overhead. After that, Li et al. [29] proposed a flexible and efficient ciphertext-policy weighted ABE scheme for the Internet of Health Things. Instead of specifying a minimum value, in their scheme, the user can specify that the attribute's weight value falls within a certain range.

However, the above schemes are either too restrictive. The weight values of each attribute is fixed by the PKG (Private Key Generator) in the Setup algorithm, and cannot be changed. The scheme proposed by Herranz et al. [2] can extend to a threshold weighted ABE scheme. Nevertheless, it is too powerful, especially in the cloud-based collaborative e-health system.

3 Outline

3.1 System Architecture

As shown in Fig. 2, our proposed system consists of four different entities: an authority, a cloud server, a patient (data owner), and data users.

- *The authority* is a trusted entity in our system. It holds the master secret key, conducts the entire system and distributes the system parameters to all the entities. The authority generates the secret keys to the data users according to the users' attributes, and issues the secret keys to the data users via a secure channel. Note that, we assume that the authority neither colludes with any other entities nor is compromised.
- *The cloud server* is a semi-honest entity in our system. The encryption of the data is stored on the cloud server. It may try to decrypt the ciphertext to obtain the PHR data, but it will not tamper any ciphertext stored in the cloud server.
- *The patient (data owner)* encrypts the PHR data M under two sets of attributes S_n, S_t (S_n denotes all the attributes in this set are "needed," S_t denotes that the set is a "threshold" set), the weights of each attribute $\{w_{\mathtt{at}}\}_{\mathtt{at}\in S_t}$, and the threshold weight value w. The patient generates a corresponding ciphertext CT. Then he/she uploads the ciphertext to the cloud server.
- *The data user* downloads ciphertext from the cloud server. Then, he/she checks whether his/her attributes and weight satisfy the policy in the ciphertext. If it satisfies, the data user decrypts the ciphertext and outputs the PHR data M. Note that, the data user may try to collude with other data users, and try to decrypt the ciphertext that is not authorized to them.

3.2 Threat Model

In our system, the authority is a trusted party which cannot be colluded by any adversary. It honestly generates all the public parameters and secret keys.

The cloud server is a semi-honest party, and it is an internal adversary. The cloud server stores the ciphertext honestly, but it may try to collect more information than the permitted leakage, and wants to obtain the sensitive information, such as the plaintext.

The data owner is assumed to be honest. It will generate the ciphertexts according the algorithms, and upload the ciphertexts to the cloud server.

All the data users (include the ineligible data users) in the system are trying to access the plaintext. We assume there is an external adversary who can obtain all the information between the patient (data owner) and the cloud server. The goal of both the internal adversary and the external adversary is to decrypt the ciphertext and obtain the PHR data. Meanwhile, the data users can collude with each other, they may try to combine their secret keys to decrypt unauthorized ciphertext. The external adversaries do not have the valid decryption key, they can only obtain the ciphertext. Due to the external adversaries have less capability than the internal adversaries, we only consider the powerful (internal) adversaries in the security proof since they cover the attack capability of the other entities.

We now briefly introduce the ability of the adversary. The adversary is allowed to register into the system, it can access the public information (e.g., the system public parameters, the universe of attributes, and the value range of the weight), all the data which is transferred in the public channel, and the ciphertext stored on the cloud server. The adversary can also submit an access policy, and can be assigned a secret key not satisfying the target ciphertext which has a same form with a real secret key.

3.3 AND-Weighted Access Policy

Unlike traditional ABE scheme, in our AND-weighted ABE scheme, a data owner encrypts the plaintext under two attributes sets, the weight values of each attribute and a threshold weight value. It can achieve the score-based access control, and can also add more access restrictions to the policy. Suppose that, the universe attributes set is U, the weight of each attribute is ranged from 1 to n. A data owner chooses two sets of attributes $S_n, S_t \subset U$. The set S_n specifies the attributes which must be included in the policy, S_t can specify the attributes and the weight. Besides, the data owner specifies the weight values $w_{\mathtt{at}}$ for each attribute in S_t, and selects a threshold weight value w. Then encrypt a plaintext under the access policy $(S_n, S_t, w_{\mathtt{at}}, w)$.

A data user who holds a set of attributes A can access the protected data if and only if:

1) $S_n \subset A$: The data user must have all the attributes listed in S_n.
2) $\sum_{\mathtt{at}\in A\cap S_t} w_{\mathtt{at}} \geq w$: The data user first sets A_{S_t} as the intersection of A and S_t. Then computes $\sum_{\mathtt{at}\in A_{S_t}} w_{\mathtt{at}}$, this value must equals or greater than w.

3.4 Design Goals

The proposed system is aimed to achieve the following goals.

- *Attribute with Dynamic Weight*. The proposed system introduces weights into attributes, and allows the patient to set each attribute's weight values dynamically. According to different scenarios, the same attribute can be assigned different weight values by the patient.
- *Stronger Privacy-Preserving.* The proposed system ensures stronger privacy-preserving. It guarantees that only the valid data users in the specific organization(s) can access the protected PHR data. Any other users who are not in the organization(s), even if he/she has the same attributes, cannot decrypt the ciphertext.

3.5 Scheme Workflow

The proposed scheme consists of the following phases:

- *System Initialization.* This phase is run by the authority. In this phase, the authority should initialize the system, and generate the public parameters and the master secret key. Every entity in the system can access the public parameters, while the authority keeps the master secret key secret.

- *User Registration.* This phase is mainly run by the authority. First, the user sends a registration request to the authority. If the user is valid, the authority will generate a secret key for the user according to his/her attributes. Then, it returns the secret key to the data user securely.
- *Data Sharing.* This phase is run by the patient. The patient encrypts the PHR data under a specific access policy, then uploads the ciphertext to the cloud server. Only a registered user whose attributes and weight value match the policy can decrypt the ciphertext.
- *Data Access.* This phase is run by the data user. The data user downloads the ciphertext from the cloud server, and utilizes the secret key to decrypt it. If the attributes in the secret key satisfy the access policy in the ciphertext, then, the data user can run the decryption algorithm to obtain the PHR data.

4 Proposed Scheme

In this section, we first show the algorithm definition of the proposed *AND-weighted ABE* scheme. Then give the mathematical tools which have been utilized in our scheme. Finally, we present the designed system in details.

4.1 Algorithm Definition

The proposed *AND-weighted ABE scheme* contains the following algorithms:

- Setup: ($1^\lambda, U, W \rightarrow MPK, MSK$): Takes a security parameter λ and a universe of attributes $U = \mathtt{at}_1, \ldots, \mathtt{at}_m$, a universe of weight $W = \{1, 2, \ldots, n\}$ as inputs and outputs some public parameters MPK, containing in particular the set U and W, which will be common to all the users of the system, along with a secret key MSK for the master entity. The public parameters will be an input of all the following algorithms.
- KeyGen: ($MPK, MSK, A \rightarrow sk_A$): Takes a subset $A \subset U$ of attributes, master secret key MSK as inputs. It generates a secret key sk_A.
- Encrypt: ($MPK, S_n, S_t, \{w_{\mathtt{at}}\}_{\mathtt{at} \in S_t}, w, M \rightarrow CT$): Takes a subset of attributes $S_n \subset U$, a subset of attributes $S_t \subset U$, together the weight value of each attribute $(\mathtt{at}, w_{\mathtt{at}})_{\mathtt{at} \in S_t}$, a weight threshold value w such that $1 \leq w \leq \sum_{\mathtt{at} \in S_t} w_{at}$, and a message M as inputs. It outputs a ciphertext CT.
- Decrypt: ($MPK, CT, sk_A \rightarrow M$): Takes the ciphertext CT and a secret key sk_A corresponding to some subset A of attributes as inputs. It outputs a message M.

Correctness. The correctness of this scheme is as follows. For all MPK, MSK, S_n, S_t, A such that the attributes set A satisfies the access policies S_n, S_t (i.e., $S_n \subset A$ and $\sum_{\mathtt{at} \in A_{S_t}} w_{\mathtt{at}} \geq w$). If $sk_A \leftarrow$KeyGen$(MPK, MSK, A \rightarrow sk_A)$, and $CT \leftarrow$Encrypt$(MPK,\ S_n, S_t, \{w_{\mathtt{at}}\}_{\mathtt{at} \in S_t}, w, M)$, we have $M =$Decrypt$(MPK, CT, sk_A \rightarrow M)$.

TABLE 3
Notations in the Proposed Scheme

Notation	Description
MPK	System public parameters
MSK	Master secret key
m	The total number of attributes in the system
n	The maximum value of the weight
A, S_n, S_t	Attributes sets
$w_{\mathtt{at}}$	Weight value of the attribute $\mathtt{at}$
w	Threshold weight value
sk_A	Private key for the data user
M	PHR data in plaintext
CT	Ciphertext

4.2 Mathematical Tools

4.2.1 Aggregate Algorithm

The Decrypt process in our proposed scheme should utilize an `Aggregate` algorithm to aggregate several elements in $\mathbb{G}_1$ to an element. This algorithm is proposed in [30], [31] to aggregate the elements in a cyclic group. In our scheme, we use it to aggregate the elements in $\mathbb{G}_1$. On input a set of values $\{g^{\frac{r}{x+x_i}}, x_i\}_{1 \leq i \leq n} \in \mathbb{G}_1$, where $r, x \in \mathbb{Z}_p$ are secret unknown values, the `Aggregate` can output $g^{\frac{r}{\prod_{i=1}^{n}(x+x_i)}} \in \mathbb{G}_1$:

$$\mathtt{Aggregate}(\{g^{\frac{r}{x+x_i}}, x_i\}_{1 \leq i \leq n}) = g^{\frac{r}{\prod_{i=1}^{n}(x+x_i)}}.$$

The time complexity of this algorithm is $O(n^2)$.

4.2.2 Bilinear Pairing

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ denotes a bilinear map satisfies the following properties:

1) Given any two elements $a, b \in \mathbb{Z}_p$ and $\forall g \in \mathbb{G}_1$, $\forall h \in \mathbb{G}_2$, $e(g^a, h^b) = e(g, h)^{ab}$.
2) For all $g \in \mathbb{G}_1$, there exists $h \in \mathbb{G}_2$ such that $e(g, h) \neq 1$.
3) Given any two elements $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$, there exists at least one efficient algorithm to compute $e(g, h)$.

4.3 Designed System

This section shows the designed system in details. This system employs our *AND-weighted ABE scheme*. The patient (data owner) encrypts the plaintext under several attributes, weight values of each attribute, and a threshold weight. The main challenge is that the traditional weighted ABE schemes cannot support both threshold weight and stricter restrictions on access policy. To solve this, we use an AND-gate policy in C_3 to protect the privacy, and a threshold policy in C_4 to achieve the flexibility. Before the detailed construction, we give the notations of our scheme in Table 3.

4.3.1 System Initialization

First, the authority initializes the system to generate the private and public parameters, where the input security parameter is λ. It runs the Setup algorithm to generate the master secret key MSK and the public parameters MPK. Assume there are m attributes $\mathtt{at} \in U$ in universe, define $W = \{1, 2 \ldots, n\}$ as the set of weights, $\tau(\mathtt{at}||i) = x \in \mathbb{Z}_p$ denotes that map an attribute $\mathtt{at}$ and its weight i to $\mathbb{Z}_p$. $\mathbb{G}_1$, $\mathbb{G}_2$ are two cyclic groups of prime order p, g and h are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, there exists a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The authority sets $\mathcal{D} = \{d_1, \ldots, d_{nm-1}\}$ consisting of $nm - 1$ pairwise different elements of $\mathbb{Z}_p$, which must be different to the values $x = \tau(\mathtt{at}||i)$. It randomly selects $\alpha, \gamma \xleftarrow{r} \mathbb{Z}_p^*$, computes $u = g^{\alpha\gamma}$, $v = e(g^\alpha, h)$. Keeps the master secret key $MSK = (g, \alpha, \gamma)$ secret, and outputs the public parameter:

$$MPK = (U, W, m, u, v, \{h^{\alpha\gamma^j}\}_{j=0,\ldots,2nm-1}).$$

4.3.2 User Registration

Before a data user (such as a doctor, a nurse, a researcher, etc.) joins the system, he/she should register to the system, and gets a secret key from the authority. The data user sends the attributes A to the authority, where the attributes satisfy $A \subset U$. After receiving the registration request from the data user, the authority checks whether the data user is a valid one. If the data user is valid, then the authority uses his/her attributes A, public parameters MPK, and master secret key MSK to generate the user's secret key. The authority runs the KeyGen algorithm on input (MPK, MKS, A), then, selects randomly $r \xleftarrow{r} \mathbb{Z}_p^*$, sets

$$sk_A = \left\{ \left\{ g^{\frac{r}{\gamma + \tau(\mathtt{at}||i)}} \right\}_{\mathtt{at} \in A, i \in \{1,\ldots,n\}}, \left\{ h^{r\gamma^j} \right\}_{j=0,\ldots,nm-2}, h^{\frac{r-1}{\gamma}} \right\}$$

Then, the authority sends the secret key sk_A to the data user.

4.3.3 Data Sharing

In this phase, the patient (data owner) encrypts the PHR data and shares the ciphertext to the specified data users. First, he/she runs the Encrypt algorithm to encrypt the PHR data, the workflow is as follows:

Taking as input public parameter MPK, the access policy $S_n \subset U$, $S_t \subset U$ (S_n represents the domain set, S_t represents the weighted attributes set), the PHR data M, the weight values $w_{\mathtt{at}} \in \{1, 2 \ldots, n\}$ of each attribute $\mathtt{at} \in S_t$, threshold weight value w, lets $s_1 = \sum_{\mathtt{at} \in S_t} w_{\mathtt{at}}$. The patient selects $k_1, k_2 \xleftarrow{r} \mathbb{Z}_p^*$ randomly, sets $k = k_1 + k_2$ and computes

$$\begin{cases} C_1 = u^{-k}, \\ C_2 = u^{-k_1}, \\ C_3 = h^{k_2 \cdot \alpha \prod_{\mathtt{at} \in S_n} (\gamma + \tau(\mathtt{at}||1))}, \\ C_4 = h^\sigma, \\ K = v^k \end{cases}$$

where

$$\sigma = h^{k_1 \cdot \alpha \prod_{\mathtt{at} \in S_t, i \in \{1,\ldots,w_{\mathtt{at}}\}} (\gamma + \tau(\mathtt{at}||i)) \prod_{d \in D_{nm+w-1-s_1}} (\gamma + d)}.$$

The values C_3 and C_4 can be computed from the public parameters $\{h^{\alpha\gamma^j}\}_{j=0,\ldots,2nm-1}$. Then, the patient sets the ciphertext $C_5 = K \cdot M$. Finally, the patient uploads the ciphertext $CT = (C_1, C_2, C_3, C_4, C_5)$ to the cloud server.

4.3.4 Data Access

All the data stored on the cloud server has been encrypted by different access policy, the data user can access the original PHR data if and only if his/her attributes satisfy the access policy. After the data user downloads the ciphertext, he/she can use a personal computer or a mobile phone runs the Decrypt algorithm to decrypt the ciphertext and obtain the PHR data. First, the data user computes:

$$\mathtt{Aggregate}\left(\left\{ g^{\frac{r}{\gamma + \tau(\mathtt{at}||1)}} \right\}_{\mathtt{at} \in S_n} \right) = g^{\frac{r}{\prod_{\mathtt{at} \in S_n} (\gamma + \tau(\mathtt{at}||1))}}.$$

Computes

$$e(g, h)^{k_2 \cdot r \cdot \alpha} = e\left(g^{\frac{r}{\prod_{\mathtt{at} \in S_n} (\gamma + \tau(\mathtt{at}||1))}}, C_3 \right). \tag{1}$$

Then, the data user lets A_{S_t} be the subset of $A \cap S_t$ with the weight value $\sum_{|A_{S_t}|} w_{\mathtt{at}} = w$, and computes

$$\begin{aligned} &\mathtt{Aggregate}\left(\left\{ g^{\frac{r}{\gamma + \tau(\mathtt{at}||i)}} \right\}_{\mathtt{at} \in A_{S_t}, i \in \{1,\ldots,w_{\mathtt{at}}\}} \right) \\ =& g^{\frac{r}{\prod_{\mathtt{at} \in A_{S_t}, i \in \{1,\ldots,w_{\mathtt{at}}\}} (\gamma + \tau(\mathtt{at}||i))}}. \end{aligned}$$

and

$$L = e\left(g^{\frac{r}{\prod_{\mathtt{at} \in A_{S_t}, i \in \{1,\ldots,w_{\mathtt{at}}\}} (\gamma + \tau(\mathtt{at}||i))}}, C_4 \right)$$

Now, the data user defines $P_{A_S,S}(\gamma)$ as:

$$\begin{aligned} &P_{A_S,S}(\gamma) \\ =& \frac{1}{\gamma} \left(\prod_{\mathtt{at} \in (S_t \cup D_{nm+w-1-s_1}) \setminus A_S, i \in \{1,\ldots,w_{\mathtt{at}}\}} (\gamma + \tau(\mathtt{at}||i) \right) \\ &- \prod_{\mathtt{at} \in (S_t \cup D_{nm+w-1-s_1}) \setminus A_S, i \in \{1,\ldots,w_{\mathtt{at}}\}} \tau(\mathtt{at}||i)). \end{aligned}$$

The data user computes

$$\begin{aligned} &e(C_2, h^{rP_{A_S,S}(\gamma)}) \cdot L \\ =& e(g, h)^{r \cdot k_1 \cdot \alpha \prod_{at \in (S \cup D_{nm+w-1-\sum w_{\mathtt{at}}}) \setminus A_{S_t}, i \in \{1,\ldots,w_{\mathtt{at}}\}} \tau(\mathtt{at}||i)} \end{aligned} \tag{2}$$

and

$$e(C_1, h^{\frac{r-1}{\gamma}}) = e(g, h)^{-k \cdot r \cdot \alpha} e(g, h)^{k \cdot \alpha} \tag{3}$$

The data user can obtain $e(g, h)^{r \cdot k_2 \cdot \alpha}$ from Equation (1), and $e(g, h)^{r \cdot k_1 \cdot \alpha}$ from Equation (2). He/She computes $e(g, h)^{r \cdot k \cdot \alpha} = e(g, h)^{r \cdot k_1 \cdot \alpha} \cdot e(g, h)^{r \cdot k_2 \cdot \alpha}$. Then, the data user computes $K = e(g, h)^{k \cdot \alpha}$ from (3). Finally, outputs the PHR data $M = C_5 / K$.

Note that, in the above scheme, we hide the PHR data in C_5. If the PHR data only contains some data collected by the sensors (such as blood pressure sensor, EEG sensor), it is efficient and secure. However, if the PHR data is a large file (contains several medical images, videos), we will consider using a symmetric encryption algorithm (such as AES, Salsa20) to encrypt the PHR data. Then hide the symmetric secret key in C_5, it will be more efficient to share a large file.

5 Security Analysis

In this section, we first give the mathematical assumptions. Then we show our security model before we prove that our proposed scheme can achieve selective CPA secure.

5.1 Mathematical Assumptions

Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be three groups of the same prime order p (this is called a *bilinear group triple* in the sequel), and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a non-degenerate and efficiently computable bilinear map. Let g_0 be a generator of $\mathbb{G}_1$ and h_0 be a generator of $\mathbb{G}_2$.

Let $\tilde{\ell}$, $\tilde{m}$, $\tilde{t}$ be three integers. The modified $(\tilde{\ell}, \tilde{m}, \tilde{t})$-*augmented multi-sequence of exponents decisional Diffie-Hellman problem* $((\tilde{\ell}, \tilde{m}, \tilde{t}) - \mathsf{aMSE} - \mathsf{DDH})$ related to the group triplet $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is as follows:

Input: The vector $\vec{x}_{\tilde{\ell}+\tilde{m}} = (x_1, \ldots, x_{\tilde{\ell}+\tilde{m}})$ whose components are pairwise distinct elements of $\mathbb{Z}_p^*$ which define the polynomials

$$f(X) = \prod_{i=1}^{\tilde{\ell}} (X + x_i) \quad \text{and} \quad g(X) = \prod_{i=\tilde{\ell}+1}^{\tilde{\ell}+\tilde{m}} (X + x_i),$$

the values

$$\begin{cases} g_0, g_0^{\gamma}, \ldots, g_0^{\gamma^{\tilde{\ell}+\tilde{t}-2}}, \qquad g_0^{\kappa_1 \cdot \gamma \cdot f(\gamma)} g_0^{\kappa_2 \cdot \gamma \cdot f(\gamma)}, & (1.1) \\ g_0^{\kappa_1 \cdot \gamma \cdot f(\gamma)} & (1.2) \\ g_0^{\omega\gamma}, \ldots, g_0^{\omega\gamma^{\tilde{\ell}+\tilde{t}-2}}, & (1.3) \\ g_0^{\alpha}, g_0^{\alpha\gamma}, \ldots, g_0^{\alpha\gamma^{\tilde{\ell}+\tilde{t}}}, & (1.4) \\ h_0, h_0^{\gamma}, \ldots, h_0^{\gamma^{\tilde{m}-2}}, \qquad h_0^{\kappa_1 \cdot g(\gamma)}, & (1.5) \\ h_0^{\kappa_2 \cdot g(\gamma)}, & (1.6) \\ h_0^{\omega}, h_0^{\omega\gamma}, \ldots, h_0^{\omega\gamma^{\tilde{m}-1}}, & (1.7) \\ h_0, h_0^{\alpha\gamma}, \ldots, h_0^{\alpha\gamma^{2(\tilde{m}-\tilde{t}+3)}} & (1.8) \end{cases}$$

where κ_1, κ_2, α, γ, ω are unknown random elements of $\mathbb{Z}_p^*$, and finally an element $T \in \mathbb{G}_T$.

Output : a bit b.

Suppose $\kappa = \kappa_1 + \kappa_2$, the problem is correctly solved if the output is $b = 1$ when $T = e(g_0, h_0)^{\kappa \cdot f(\gamma)}$ or if the output is $b = 0$ when T is a random value from $\mathbb{G}_T$. In other words, the goal is to distinguish if T is a random value or if it is equal to $e(g_0, h_0)^{\kappa \cdot f(\gamma)}$.

More formally, let us denote by **real** the event that T is indeed equal to $T = e(g_0, h_0)^{\kappa \cdot f(\gamma)}$, by **random** the event that T is a random value from $\mathbb{G}_T$ and by $\mathcal{I}(\vec{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)$ the input of the problem. Then, we define the *advantage* of an algorithm $\mathcal{B}$ in solving the $((\tilde{\ell}, \tilde{m}, \tilde{t}) - \mathsf{aMSE} - \mathsf{DDH})$ problem as

$$\begin{aligned} &\mathrm{Adv}_{\mathcal{B}}^{(\tilde{\ell},\tilde{m},\tilde{t})-\mathsf{aMSE}-\mathsf{DDH}} \\ =&|\Pr[\mathcal{B}(\mathcal{I}(\vec{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)) = 1|\textbf{real}] \\ &- \Pr[\mathcal{B}(\mathcal{I}(\vec{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)) = 1|\textbf{random}]| \end{aligned}$$

where the probability is taken over all random choices and over the random coins of $\mathcal{B}$.

5.2 Security Model

Definition 1. *If $\mathcal{A}$ is a* P.P.T *algorithm, $\mathcal{C}$ be a challenger. The definition of indistinguishability against selective ciphertext-policy and chosen-plaintext attacks (IND-sCP-CPA) is as follows:*

- *The challenger specifies a universe of attributes U of size m and a weight set $W = \{1, 2, \ldots, n\}$, then gives it to the attacker $\mathcal{A}$.*
- *$\mathcal{A}$ selects two subsets $S_n, S_t \subset U$, where $|S_n| + |S_t| = s$, each weight value of the attribute $w_{\mathtt{at}}$, and a weight threshold w such that $1 \le w \le \sum_{\mathtt{at} \in S_t} w_{\mathtt{at}}$.*
- *The challenger runs $(MPK, MSK) \leftarrow \mathsf{Setup}(1^{\lambda}, U, W)$ and gives MPK to $\mathcal{A}$.*
- *Secret key queries: $\mathcal{A}$ adaptively sends subsets of attributes $B \subset U$, with the restriction $S_n \not\subset B$ or $\sum_{\mathtt{at} \in B \cap S_t} < w$, and must receive the secret key sk_B as the answer.*
- *$\mathcal{A}$ outputs two messages M_0, M_1 of the same length.*
- *Challenge: The challenger picks a random bit $b \in \{0, 1\}$, computes $CT \leftarrow \mathsf{Encrypt}(MPK, S_n, S_t, \{w_{\mathtt{at}}\}_{\mathtt{at} \in S_t}, w, M_b)$ and gives CT to $\mathcal{A}$.*
- *Same as Step 4.*
- *Guess: Finally, the adversary $\mathcal{A}$ outputs a bit b'.*

The advantage of an adversary $\mathcal{A}$ in the above game is defined as

$$\mathrm{Adv}_{\mathcal{A}} = \Pr[b' = b] - \frac{1}{2}.$$

The scheme is secure against chosen-plaintext attacks if for all P.P.T *algorithms, we have $\mathrm{Adv}_{\mathcal{A}}$ is a negligible function of λ.*

The security of our proposed scheme is conducted in IND-sCP-CPA model. Therefore, the adversary should commit the attributes subsets and the threshold before obtaining the public parameters.

5.3 Security Proof

Theorem 1. *If the $((\tilde{\ell}, \tilde{m}, \tilde{t}) - \mathsf{aMSE} - \mathsf{DDH})$ assumption holds, then our proposed weighted CP-ABE scheme in section 4 is secure. Here $\tilde{\ell} = nm - S_w$, $\tilde{m} = nm + w - 1$ and $\tilde{t} = w + 1$. (We use S_w to denote $\sum_{\mathtt{at} \in S_t} w_{\mathtt{at}}$)*

Proof. We now construct a simulator $\mathcal{B}$ which uses the adversary $\mathcal{A}$ to solve the $(nm - S_w, nm + w - 1, w + 1) - \mathsf{aMSE} - \mathsf{DDH}$ problem. Let $I(\vec{x}_{2nm+w-1-S_w}, \kappa, \alpha, \gamma, \omega, T)$ be the input of the simulator $\mathcal{B}$. At beginning, $\mathcal{B}$ selects a universe of attributes set $P = \{\mathtt{at}_1, \mathtt{at}_2 \ldots, \mathtt{at}_m\}$, where the weight values of each attribute is $[1, \ldots, n]$. Then, the adversary $\mathcal{A}$ chooses two target sets $S_n, S_t \subset P$, and a threshold w such that $1 \le w \subset \sum_{\mathtt{at} \in S_t} w_{\mathtt{at}}$. Suppose that $S_t = \{\mathtt{at}_{m-s+1}, \ldots, \mathtt{at} - m\} \subset P$, A_{S_t} denotes the subset $A \cap S_t$.

Simulation of the Setup. The simulator $\mathcal{B}$ defines the encoding of the attributes as $\tau(at_i||j) = x_i j$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$. Here, the encodings of the first $nm - S_w$ elements are the opposite of the roots of $f(X)$, and the encodings of the attributes in S_t are the opposite of some roots of $g(X)$

The values corresponding to the "dummy" attributes $\mathcal{D} = \{d_1, \ldots, d_{nm-1}\}$ are defined as $d_j = x_{nm+j}$ if $j =$

$1...nm+w-1-S_w$. For $j=nm+w-S_w,\ldots,nm-1$, the d_j's are picked uniformly at random in $\mathbb{Z}_p^*$ until they are distinct from $\{x_1,\ldots,x_{2nm+w-1-S_w},d_{nm+w-S_w},\ldots,d_{j-1}\}$.

The simulator $\mathcal{B}$ defines $g:=g_0^{f(\gamma)}$. Note that $\mathcal{B}$ can compute g with the elements of line (l.1) of its input, since f is a polynomial of degree $\tilde{\ell}$. To complete the setup phase, $\mathcal{B}$ sets $h=h_0$ and computes

- $u=g^{\alpha\gamma}=g_0^{\alpha\cdot\gamma\cdot f(\gamma)}$ with line (l.4) of its input, which is possible since $Xf(X)$ is a polynomial of degree $\tilde{\ell}+1$. Indeed, $\alpha\cdot\gamma\cdot f(\gamma)$ is a linear combination of $\alpha\gamma,\ldots,\alpha\gamma^{\tilde{\ell}+1}$ and the coefficients of this linear combination are known to $\mathcal{B}$, so the value u can be computed from line (l.4).
- $v=e(g,h)^{\alpha}=e(g_0^{f(\gamma)\alpha},h_0)$ with line (l.4) for $g_0^{f(\gamma)\alpha}$. Note that the value g^{α} could be computed by $\mathcal{B}$ and added to the public parameters, in case the verification of the consistency of the secret keys is desired for the scheme.

The simulator $\mathcal{B}$ can compute the values $\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2nm-1}$ from line (l.8) of its input. Eventually, $\mathcal{B}$ gives to $\mathcal{A}$ the resulting

$$MPK=\{P,m,u,v,\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2nm-1},\mathcal{D},\tau\}.$$

Simulation of Key Extraction Queries. Whenever the adversary $\mathcal{A}$ makes a key extraction query for a subset of attributes $A=\{at_{i_1},\ldots,at_{i_q}\}\subset P$ satisfying that $0\le|A_{S_t}|\le w-1$ or $S_n\not\subset B_2$, the simulator $\mathcal{B}$ must produce a tuple of the form

$$sk_A=\left\{\left\{g^{\frac{r}{\gamma+\tau(\mathtt{at}||i)}}\right\}_{\mathtt{at}\in A,i\in\{1,\ldots,n\}},\{h^{r\gamma^j}\}_{j=0,\ldots,nm-2},h^{\frac{r-1}{\gamma}}\right\},$$

for some random value $r\in\mathbb{Z}_p^*$, To do so, $\mathcal{B}$ implicitly defines $r=(\omega y_A\gamma+1)Q_A(\gamma)$, where y_A is randomly picked in $\mathbb{Z}_p^*$, and the polynomial $Q_A(X)$ is defined as $Q_A(\gamma)=1$ when $|A_{S_t}|=0$, or $Q_A(X)=\lambda_A\cdot\prod_{\mathrm{at}\in A_{S_t}}(X+\tau(\mathtt{at}||i))$ otherwise, in which case $\lambda_A=(\prod_{\mathrm{at}\in A_{S_t}}\tau(\mathtt{at}||i))^{-1}$.

The elements which form sk_A are then computed as follows:

- For any $\mathtt{at}\in A_{S_t}$, $\mathcal{B}$ defines
$$\begin{aligned}&Q_{\mathrm{at}}(\gamma)\\&=Q_A(\gamma)/(\gamma+\tau(\mathtt{at}||i))\\&=\lambda_A\cdot\prod_{\tilde{\mathtt{at}}\in A_{S_t},\tilde{\mathtt{at}}\neq\mathrm{at}}(\gamma+\tau(\tilde{\mathtt{at}}||i))\end{aligned}$$
Then $g^{\frac{r}{\gamma+\tau(\mathtt{at}||i)}}=g_0^{f(\gamma)\omega y_A\gamma Q_{\mathrm{at}}(\gamma)}\cdot g_0^{f(\gamma)Q_{\mathrm{at}}(\gamma)}$. The first factor of the product (whose exponent is a polynomial in γ of degree at most $(nm-S_w)+1+w-2)$ can be computed from line (l.3), whereas the second factor (whose exponent is a polynomial in γ of degree at most $(nm-S_w)+w-2)$ can be computed from line (l.1).
- For any $\mathtt{at}\in A\backslash A_{S_t}$, the simulator $\mathcal{B}$ defines the polynomial $f_{\mathtt{at}}(X)=f(X)/(X+\tau(\mathtt{at}||i))$. Then $g^{\frac{r}{\gamma+\tau(\mathtt{at}||i)}}=g_0^{f_{\mathtt{at}}(\gamma)\omega y_A\gamma Q_A(\gamma)}\cdot g_0^{f_{\mathtt{at}}(\gamma)Q_A(\gamma)}$. Similarly, the first factor of this product can be computed from line (l.3), and the second factor can be computed from line (l.1).
- The values $\{h^{r\gamma^i}\}_{i=0,\ldots,nm-2}$ can be computed from line (l.4) and (l.5), since $h^{r\gamma^i}=h^{Q_A(\gamma)\omega y_A\gamma^{i+1}}\cdot h^{Q_A(\gamma)\gamma^2}$.
- Finally, $\mathcal{B}$ has to compute $h^{\frac{r-1}{\gamma}}=h^{Q_A(\gamma)\omega y_A}\cdot h^{\frac{Q_A(\gamma)-1}{\gamma}}$. The first factor of the product can be computed from line (l.7) and the second factor can be computed from line (l.5), since by definition of λ_A, $Q_A(X)$ is a polynomial with independent term equal to 1 and thus $\frac{Q_A(\gamma)-1}{\gamma}$ is a linear combination of $\{1,\gamma,\ldots,\gamma^{(w-2)}\}$.

Note that $Q_A(\gamma)\neq 0$ (otherwise $\gamma=\tau(\mathtt{at}||i)$) for some $\mathtt{at}\in A_{S_t}$ and γ is public), in which case it is not hard to see that r is uniformly distributed in $\mathbb{Z}_p$. If the choice of y_A leads to $r=0$ (which occurs only with negligible probability anyhow), it suffices to pick a different value for y_A. That is, in the simulation r is uniformly distributed in $\mathbb{Z}_p^*$.

Simulation of the Challenge. Once $\mathcal{A}$ sends to $\mathcal{B}$ the two messages M_0 and M_1, $\mathcal{B}$ flips a coin $b\in\{0,1\}$, and sets $C_5^*=T\cdot M_b$. To simulate the rest of the challenge ciphertext, $\mathcal{B}$ implicitly defines the randomness for the encryption as $\kappa'=\kappa/\alpha$, sets $C_3^*=h_0^{\kappa_2\cdot g(\gamma)}$ (given in line (l.6)) and $C_4^*=h_0^{\kappa_1\cdot g(\gamma)}$ (given in line (l.5) of the aMSE-DDH input). To complete the ciphertext, $\mathcal{B}$ computes $C_2^*=(g_0^{\kappa_1\cdot\gamma f(\gamma)})^{-1}$ from line (l.2) and computes $C_1^*=(g_0^{\kappa_1\cdot\gamma f(\gamma)}g_0^{\kappa_2\cdot\gamma f(\gamma)})^{-1}$ from line (l.1) of the input, which is equal to $u^{-\kappa'}$.

After the challenge step, the adversary $\mathcal{A}$ can also make other key extraction queries, this step is same as previous *Simulation of key extraction queries* phase.

Guess. Finally, the adversary $\mathcal{A}$ outputs a bit b'. If $b'=b$, the simulator $\mathcal{B}$ outputs 1 as the solution to the given instance of the aMSE-DDH problem, which means that $T=e(g_0,h_0)^{\kappa\cdot f(\gamma)}$. Otherwise, $\mathcal{B}$ will output 0 to indicate that T is a random element.

We now have to analyze the advantage of the simulator $\mathcal{B}$:

$$\begin{aligned}&\mathrm{Adv}_{\mathcal{B}}^{\mathrm{aMSE-DDH}}(\lambda)\\=&\big|\Pr\big[\mathcal{B}\big(\mathcal{I}\big(\vec{x}_{\tilde{\ell}+\tilde{m}},\kappa,\alpha,\gamma,\omega,T\big)\big)=1|\text{ real }\big]-\\&\Pr\big[\mathcal{B}\big(\mathcal{I}\big(\vec{x}_{\tilde{\ell}+\tilde{m}},\kappa,\alpha,\gamma,\omega,T\big)\big)=1|\text{ random }\big]\big|\\=&\big|\Pr[b=b'|\text{ real }]-\Pr[b=b'|\text{ random }]\big|\end{aligned}$$

When the event `real` occurs, then $\mathcal{A}$ is playing a real attack and therefore $|\Pr[b=b'|\text{ real }]-1/2|=\frac{1}{2}Adv_{\mathcal{A},\Pi}^{IND-sCPA}(\lambda)$. During the random `event`, the view of $\mathcal{A}$ is completely independent of the bit b; in this case, the probability $\Pr[b=b']$ is equal to $1/2$. Therefore, we have

$$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{aMSE-DDH}}(\lambda)\ge\frac{1}{2}\mathrm{Adv}_{\mathcal{A},\Pi}^{\mathrm{IND-sCPA}}(\lambda)$$

□

This concludes the proof of the Theorem 1.

Theorem 2. *If the proposed scheme is s-IND-CPA secure, then the cloud-based PHR sharing system is secure against the following attacks defined in the threat model.*

TABLE 4
A Score-Based Access Policy

Algorithm	Costs
Setup	$\|E_{\mathbb{G}_1}\| + \|BP_e\| + 2nm\|E_{\mathbb{G}_2}\|$
KeyGen	$\|A\| \cdot \|E_{\mathbb{G}_1}\| + nm\|E_{\mathbb{G}_2}\|$
Encrypt	$2\|E_{\mathbb{G}_1}\| + (nm + w - 1 + \|S_n\|)\|E_{\mathbb{G}_2}\| + (nm + w + \|S_n\| - 3)\|M_{\mathbb{G}_2}\| + \|E_{\mathbb{G}_T}\|$
Decrypt	$(w^2 + \|S_n\|^2)\|E_{\mathbb{G}_1}\| + (w^2 + \|S_n\|^2 - 2)\|M_{\mathbb{G}_1}\| + (nm-2)^2\|E_{\mathbb{G}_2}\| + ((nm-2)^2 - 1)\|M_{\mathbb{G}_2}\| + \|E_{\mathbb{G}_T}\| + 3\|M_{\mathbb{G}_T}\| + 4\|BP_e\|$

According to Theorem 1, the security of the proposed system can be concluded by the IND-sCP-CPA secure. Now, we analyze the following attacks which is defined in the threat model.

The *external and internal attacks* can be reduced to the attacks in the IND-sCP-CPA model and the adversary is any party except the data owner and the valid receivers. The internal adversaries hold the invalid decryption keys. They can also try to combine different decryption keys to a new one which may satisfy the access policy. However, the combined one must be invalid, due to each private key has a unique random number. Therefore, the internal adversaries can learn nothing from the ciphertext.

The *collusion attack* can be reduced to the attacks in the IND-sCP-CPA model. Each secret key is derived from unique random number. The combination of different valid secret keys cannot help the adversaries to decrypt the unauthorized ciphertexts.

6 Performance Evaluation

In this section, we show the performance analysis for our proposed scheme in terms of both communication costs and computation costs.

6.1 Communication Costs Analysis

Suppose there are m attributes in the universe, and the maximum weight value is n. We use $|\mathbb{Z}_p|$, $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, $|\mathbb{G}_T|$ to denote the size of the element in groups $\mathbb{Z}_p$, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, respectively. In KeyGen phase (i.e., *User Registration*), the communication cost is $n|A| \cdot |\mathbb{G}_1| + (nm-1)|\mathbb{G}_2|$ where $|A|$ is the number of user's attributes. Due to the size of the ciphertext is constant, the communication costs of the Encrypt phase (i.e., uploading ciphertext in *Data Sharing*) and the Decrypt phase (i.e., download ciphertext in *Data Access*) is the same: $2|\mathbb{G}_1| + 2|\mathbb{G}_2| + |\mathbb{G}_T|$.

6.2 Computational Costs Analysis

We now give the basic analysis of the computational costs of each process. Same as before, we suppose there are m attributes in the universe, and the maximum weight value is n. w is the threshold value, and $|S_n|$ is the number of attributes in the set S_n. We use $|E_{\mathbb{G}_1}|$, $|E_{\mathbb{G}_2}|$ and $|E_{\mathbb{G}_T}|$ to denote exponentiation operations on the cyclic group $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, respectively; use $|M_{\mathbb{G}_1}|$, $|M_{\mathbb{G}_2}|$ and $|M_{\mathbb{G}_T}|$ to denote modular multiplication operations on the cyclic group $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, respectively; and use $|BP_e|$ to denote the operation bilinear operation. The computational costs are show in Table 4.

We use RELIC to evaluate the performance of our proposed scheme on a PC with VMware ESXi (an Intel Xeon E5-2678 v3 @ 12x 2.494 GHz CPU, 32 GB RAM, Ubuntu 20.04 operating system). The curve we used in our experiment is *x64-pbc-bn254.sh* (BN254 curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$, with embedding degree $k = 12$, the bit lengths of elements in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are 64 bytes, 128 bytes and 384 bytes, respectively). The implementation consists of the following parts: Setup, KeyGen, Encrypt, and Decrypt.

First, we set there are 6 attributes in the system, each of them is weighted from 1 to 5. Suppose that, the data user hold all the attributes, and we set the threshold weight values is 10. Besides, we also fix 2 attributes must be included in decryptor's attribute set. Suppose the data user has 4 different attributes and satisfy the access policy. In KeyGen algorithm, the PKG should compute 20 exponential operations on $\mathbb{G}_1$, and 30 exponential operations on $\mathbb{G}_2$. In Encrypt algorithm, the patient needs to compute 41 exponential operations on $\mathbb{G}_2$ and 39 multiplication operations on $\mathbb{G}_2$. To decrypt the ciphertex, the data user needs to compute 4 pairing operations, and over 100 exponential operations on $\mathbb{G}_1$, nearly 900 exponential operations on $\mathbb{G}_2$.

We compare our proposed scheme with [2] by using the same parameters as shown above (we removed the AND-gate attributes). The runtime of each part is shown in Fig. 3 (averages were computed over 1,000 executions).

Furthermore, we utilize the same settings to compare our proposed scheme and [2] on an Android device (Google Pixel 3 XL, with a Qualcomm Snapdragon 845 CPU, 4 GB RAM, and Android 9.0 (Pie) operating system). The curve we used in this implementation is *arm-pbc-bn254.sh*. All the implementations are running 1,000 times, the averaged results are shown in Fig. 4.

As can be seen from Figs. 3 and 4, the computational overhead of our scheme is somewhat higher than [2].

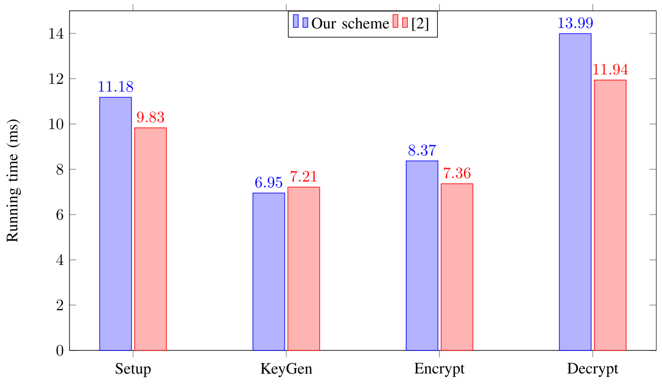


Fig. 3. Comparison of computational overhead on PC.

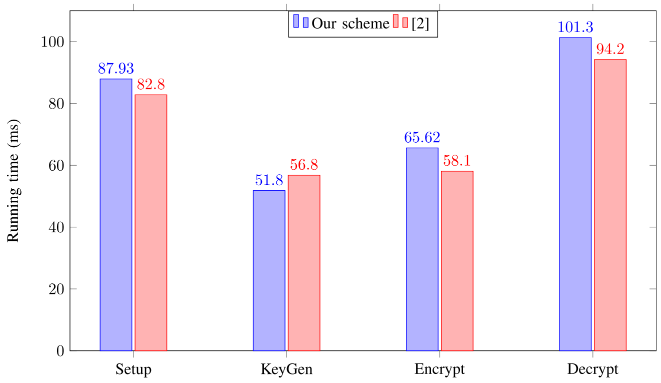


Fig. 4. Comparison of computational overhead on Android phone.

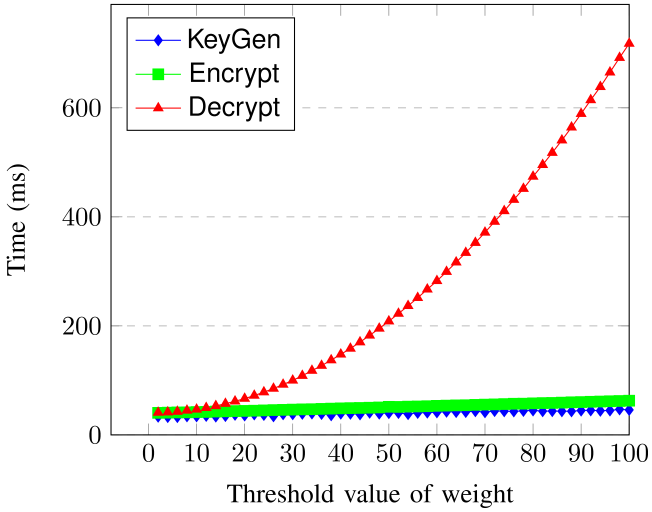


Fig. 5. Benchmark results of different algorithms of our scheme.

Nonetheless, by introducing more restrictions on the access policy, our scheme can ensure the privacy. Moreover, from the comparison, we can conclude that the computational overheads are acceptable even when running the scheme on an android phone.

For more precise evaluation, we set the threshold value in the ciphertext from 2 to 100. In this implementation, there are 40 attributes in the universe, and the weight is ranged from 1 to 10. The Setup algorithm takes about 66.76 ms, the time consuming of other algorithms are shown in Fig. 5.

Due to it needs many modular exponential operations, especially in the `Aggregate` algorithm. If the `Aggregate` algorithm should aggregate n elements, it needs n^2 modular exponential operations. Therefore, as the threshold weight value increases, the data user's computational cost is also greatly increased.

7 Conclusion

In this paper, we proposed a weighted attribute-based encryption scheme in cloud-based personal health records sharing system to balance between the privacy and the flexibility. Specifically, it allows the patient to share the PHR data under different weighted attributes with a threshold weight value. Meanwhile, the patient can fix the specific organization(s) where the data users belong to. According to the performance evaluation, our proposed scheme is potentially useful in the real world applications.

In the future, we intend to explore more restrictions on threshold attribute-based encryption, such as blacklist-based encryption to exclude some attribute sets in the threshold ABE scheme.

References

[1] Statista, "Cloud storage (buckets and blobs) security issues experienced by organizations worldwide between 2020 and 2021," 2021. Accessed: May 2021. [Online]. Available: https://www.statista.com/statistics/1238774/cloud-storage-security-issues-organizational-size/

[2] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proc. 13th Int. Conf. Theory Pract. Public Key Cryptogr.*, 2010, pp. 19–34.

[3] D. F. Aranha, C. P. L. Gouvêa, T. Markmann, R. S. Wahby, and K. Liao, "RELIC is an efficient library for cryptography," 2010. [Online]. Available: https://github.com/relic-toolkit/relic

[4] A. Sahai and B. R. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, R. Cramer, Ed., Berlin, Germany: Springer, 2005, pp. 457–473.

[5] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "Attribute-based encryption for cloud computing access control: A survey," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–41, 2020.

[6] M. Rasori, M. La Manna, P. Perazzo, and G. Dini, "A survey on attribute-based encryption schemes suitable for the Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8269–8290, Jun. 2022.

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

[8] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th Conf. Comput. Commun. Secur.*, 2007, pp. 195–203.

[9] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. 14th Int. Conf. Theory Pract. Public Key Cryptogr.*, 2011, pp. 53–70.

[10] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. Au, "PPDCP-ABE: Privacy-preserving decentralized ciphertext-policy attribute-based encryption," in *Proc. 19th Eur. Symp. Res. Comput. Secur.*, 2014, pp. 73–90.

[11] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Proc. 8th Int. Conf. Provable Secur.*, 2014, pp. 259–273.

[12] Q. M. Malluhi, A. Shikfa, and V. C. Trinh, "A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption," in *Proc. 12th ACM Symp. Inf. Comput. Commun. Secur.*, 2017, pp. 230–240.

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

[14] M. Ambrona, G.R. BartheGay, and H. Wee, "Attribute-based encryption in the generic group model: Automated proofs and new constructions," in *Proc. 24th Conf. Commun. Secur.*, 2017, pp. 647–664.

[15] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.

[16] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. 35th Int. Colloq. Automata Lang. Programm.*, 2008, pp. 579–591.

[17] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proc. 4th ACM Symp. Inf. Comput. Commun. Secur.*, 2009, pp. 343–352.

[18] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology–EUROCRYPT 2010*, H. Gilbert, Berlin, Germany: Springer, 2010, pp. 62–91.

[19] W. Susilo, G. Yang, F. Guo, and Q. Huang, "Constant-size ciphertexts in threshold attribute-based encryption without dummy attributes," *Inf. Sci.*, vol. 429, pp. 349–360, 2018.

[20] W. Yang, R. Wang, Z. Guan, L. Wu, X. Du, and M. Guizani, "A lightweight attribute based encryption scheme with constant size ciphertext for Internet of Things," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[21] H. Xiong, X. Huang, M. Yang, L. Wang, and S. Yu, "Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 3097–3111, Feb. 2022.

[22] P. Zeng, Z. Zhang, R. Lu, and K.-K. R. Choo, "Efficient policy-hiding and large universe attribute-based encryption with public traceability for internet of medical things," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10 963–10 972, Jul. 2021.

[23] R. Zhang, J. Li, Y. Lu, J. Han, and Y. Zhang, "Key escrow-free attribute based encryption with user revocation," *Inf. Sci.*, vol. 600, pp. 59–72, 2022.

[24] H. S. G. Pussewalage and V. Oleshchuk, "A delegatable attribute based encryption scheme for a collaborative E-health cloud," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2022. 3174909.

[25] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1661–1673, Aug. 2016.

[26] W. Li, W. Ni, D. Liu, R. P. Liu, and S. Luo, "Unified ciphertext-policy weighted attribute-based encryption for sharing data in cloud computing," *Appl. Sci.*, vol. 8, no. 12, 2018, Art. no. 2519.

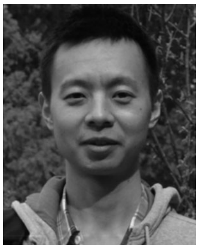
[27] H.-Y. Lin and W.-G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2005, pp. 456–466.

[28] K. Xue, J. Hong, Y. Xue, D. S. Wei, N. Yu, and P. Hong, "CABE: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding," *IEEE Trans. Comput.*, vol. 66, no. 9, pp. 1491–1503, Sep. 2017.

[29] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 5, pp. 1949–1960, May 2021.

[30] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Proc. 1st Int. Conf. Pairing-Based Cryptogr.*, 2007, pp. 39–59.

[31] C. Delerablée and D. Pointcheval, "Dynamic threshold public-key encryption," in *Advances in Cryptology–CRYPTO*, Santa Barbara, CA, USA: Springer, 2008, pp. 317–334.



Yudi Zhang received the master's degree from the Hubei University of Technology, China, in 2017, and the PhD degree from Wuhan University, China, in 2020. He is currently a post-doctor with the School of Computing and Information Technology, University of Wollongong, Australia. His main research interests include cryptography and information security, in particular, and cryptographic protocols.

Fuchun Guo received the PhD degree from the University of Wollongong, Australia, in 2013. He is currently a senior lecturer with the Institute of Cybersecurity and Cryptology, University of Wollongong. Dr. Fuchun received the prestigious Australian Research Council DECRA Fellowship award in 2017. His primary research interests include the public-key cryptography, in particular, protocols, encryption and signature schemes, and security proof.

Willy Susilo (Fellow, IEEE) received the PhD degree in computer science from the University of Wollongong, Australia. He is currently a distinguished professor and the head of the School of Computing and Information Technology and the director of the Institute of Cybersecurity and Cryptology, University of Wollongong. He has published more than 400 research papers in the area of cybersecurity and cryptology. His main research interests include cybersecurity, cryptography, and information security. He was a recipient of the prestigious Australian Research Council (ARC) Future Fellow by the ARC and the researcher of the Year Award by the University of Wollongong in 2016. He is the editor-in-chief of the Elsevier's Computer Standards and Interfaces and MDPI's Information journal. He has served as a program committee member in dozens of international conferences. He is currently serving as an associate editor in several international journals, including the *IEEE Transactions on Dependable and Secure Computin*g and the International Journal of Information Security (Springer). His work has been cited over 19,000 times in Google Scholar. He is also a fellow of the Australian Computer Society (ACS).

Guomin Yang (Senior Member, IEEE) received the PhD degree from the Computer Science Department, City University of Hong Kong in 2009. Formerly, he worked as a research scientist with the Temasek Laboratories with the National University of Singapore. He is currently an associate professor with the School of Computing and Information Technology, University of Wollongong, Australia. He has been awarded a prestigious Australian Research Council DECRA fellowship award. His research interests include cryptography and network security.