Decentralizing Multi-Authority Attribute-Based Access Control Scheme with Fully Hidden Policy

Leyou Zhang[1], Juan Ren[1,2], Li Kang[1], and Baocang Wang[3,4]

(Corresponding author: Juan Ren)

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China[1]

Science and Technology on Communication Security Laboratory, Chengdu 610041, China[2]

School of Information Engineering, Xuchang University, Xuchang 461000, China[3]

State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China[4]

Email: juaner_r@126.com

(Received Jan. 4, 2020; Revised and Accepted Sept. 8, 2020; First Online May 30, 2021)

Abstract

Decentralized multi-authority attribute-based encryption (MA-ABE) is considered a potential method to protect users' privacy in the cloud. However, most of the existing works cannot provide a complete solution since there are some vulnerabilities to be found in users' collusion, global identity (GID) leakage-resilience, and access policy hiding. In this paper, we focus on overcoming these shortcomings. At first, we investigate the recent works and give a summary of them. Then an efficient decentralized MA-ABE scheme with a fully hidden access policy is presented. To implement the hidden access policy, we use the technique of Inner-Product Encryption (IPE). Under this technique, the Viète's Formulas is used to convert the access policy into a vector, which results in an efficient decentralized MA-ABE scheme with the shortened ciphertext and secret keys, which are only concerned with the number of wildcards. To further improve efficiency, the decryption is partially outsourced. The security of the proposed scheme is reduced to the standard decisional bilinear Diffie-Hellman (DBDH) assumption and the Decisional Linear (DLIN) assumption instead of other strong assumptions. Finally, performance analysis and numerical experiments confirm the scalability and flexibility of our approach.

Keywords: Decentralizing ABE; Fully Hidden Policy; IPE; Resistant-Collusion

1 Introduction

Cloud computing has been widely concerned, and continually developed at present because of its low cost, strong computing capacity, large storage capacity and high data security performance, which makes it convenient and profitable for data owners to share data on third-party cloud storage servers. Therefore, more individuals and enterprises upload application data to cloud storage servers. However, in many applications, the data owners hope that only authorized users can share their data. Additionally, the data owners cannot obtain the prior knowledge of who will share their data. Hence an access control policy is required for encrypted data in the cloud [6,16]. The attribute-based encryption (ABE) due to Sahai and Waters [24] provided a solution to the above problem, which it supported the fine-grained access control by encrypting data with various access policies.

1.1 Motivations

It can be found that attribute-based encryption is applied in cloud computing by summarizing the relevant work, but there are still many problems that need to be solved urgently as stated below:

Firstly, the most existing schemes are based on one authority. However, in real life, it is impractical and overburdened for one authority to authenticate and certificate all attributes. Therefore, single authority has been a bottleneck in a large system. With the development of cloud storage, there is more than one party to act as an authority. Hence, MA-ABE addresses this problem. However, in recent years, some MA-ABE schemes have been successfully attacked repeatedly as in [8,26,29] by means of collusion attacks, test attacks or logical attacks. How to further enhance the collusion-resistance of a decentralized MA-ABE scheme is still a subject worth studying.

Secondly, in a real cloud storage environment, the access policy itself could be sensitive information about users' attributes and be showed in the ciphertext, which will result in the leakage of users' sensitive information when the users want to upload the file encrypted by the access policy to the cloud storage server. Considering this example: An enterprise may release a number of specific files encrypted by the access policy: *(Leader* $\wedge$ *apartment A)* $\vee$ *(Secretary* $\wedge$ *apartmet B)*, notice that itself reveals user's private attributes. It is significant to hide the access policy since it may lead to the privacy leakage. In the study of hiding access policy, most ABE schemes only realize partially hidden access policy. In this paper, another method to hide access policy is considered by combining IPE technology with ABE scheme, in which the user's attribute set is sent to the attribute authorities (AAs) in form of fuzzy vector based on IPE technology, so that AAs cannot know the specific information about attribute names or attribute values. But it is difficult to combine the IPE technology with ABE schemes.

Thirdly, in the existing MA-ABE schemes, exponentiation and pairing operations increase linearly with the number of attributes in the decryption phase, which leads to the increase of decryption costs. Improving the decryption efficiency is also a considerable challenge.

1.2 Our Contributions

As mentioned above, in recent years, many MA-ABE schemes have been attacked repeatedly and successfully by means of collusion attacks or test attacks. In addition, most of the existing MA-ABE schemes only realize partially hidden access policy, and there are compromises in efficiency simultaneously. In this paper, an efficient decentralized MA-ABE scheme with fully hiding access policy and collusion-resistant strongly is presented. Main contributions are summarized as follows:

Strong resistance to attacks. We propose a decentralized MA-ABE scheme with strong resistance to attacks from potential malicious users. Specifically, GID is coupled non-linearly with parameters f_1, f_2 and η_k in the secret keys to resist the attacks mentioned in [23] and [29].

Fully hidden policy. In order to achieve fully policy-hiding, we build the decentralized MA-ABE scheme using the technique of IPE. Based on Viète's Formulas, the access policy, consisting of the position of the symbols, is fully hidden by converting it into a vector.

Low overhead. The length of ciphertext and secret keys is shortened in our scheme, due to it is only related to the number of wildcards in the access policy. To further improve efficiency, the decryption is partially outsourced.

1.3 Paper Organization

We present the related works in Section 2. In Section 3, some preliminaries including the statements of bilinear map, complexity assumptions, access structure and the Viète's formulas are provided. Then the formal definition and its security model are given in Section 4. Section 5 presents the construction of our scheme in detail. The security analysis and performance analysis are proposed in Section 6 and Section 7 respectively. Finally, we give a brief conclusion in Section 8.

2 Related Works

We analyze related researches from three aspects: Multi-authority ABE, policy-hiding ABE and outsourcing ABE. The details are given as follows.

2.1 Multi-authority ABE

It started with the one by Chase [3] with a central authority (CA) and global identify (GID), which GID prevented the collusion attacks from malicious user. But it is limited to the AND-gate policy. Müller *et al.* [18] proposed the other one with CA and could be expressed by the LSSS access structure. However, the CA is required must be honest in [3, 18]. Then Chase and Chow introduced a new scheme that the center was removed [4]. However, The cooperation among multiple authorities is necessary during the system initialization phase. Later, Lewko and Waters [15] proposed a decentralized MA-ABE, in which the CA was removed so that any authorities could join or leave the system freely without reinitializing the system.

In addition, for the MA-ABE scheme, the most basic requirement is the resistance to collusion attacks. Hence, Han *et al.* [8] proposed a decentralized KP-ABE scheme that GID was non-linearly embedded into the user's private keys for enhancing the resistance to collusion attacks. Soon, Ge *et al.* [10] showed a new method of user's collusion attack, and proved the scheme [8] was vulnerable to this collusion attack. Compared with the previous proposed schemes, Han *et al.* [8] proposed a more powerful privacy protection MA-ABE scheme. However, Wang *et al.* [26] pointed out the security weaknesses above scheme and proposed a collusion attack method to Han's scheme. Qian *et al.* [22] constructed another multi-authority ciphertext-policy ABE (MA-CP-ABE) scheme that based on AND-gates access policy on multi-valued attributes. For this method of collusion attack mentioned in [22], Rahulamathavan *et al.* [23] proposed a decentralized ABE scheme that resisted it in 2016, such that the key generation algorithm was improved for breaking the linear relationship between keys. However, this scheme [23] was found that it could not resist the user collusion attacks, and the improved algorithm was given by Zhang *et al.* [29] in 2018. However, these schemes do not consider the feature of hiding policy.

2.2 Policy-Hiding ABE

Hiding policy (or attribute) means that the privacy in access policy is protected in the applications. Fully policy-hiding means that anyone could not know the sensitive attribute information from the access policy, even authorized users who could decrypt successfully.

Nishide *et al.* [19] introduced firstly the concept of policy-hiding by AND-gate access policy on multi-valued attributes with wildcards in 2008. However, the scheme is only proven in a weak model. Later, To protect sensitive information included in the access policy, several

Table 1: The comparison of our scheme and related works

Scheme	Multi-authority	Hidden policy	Way of policy-hiding	Outsource
[26]	Multi	✗	✗	✗
[29]	Multi	✗	✗	✗
[28]	Single	Partially Hidden	Hide attribute values	✗
[20]	Single	Fully Hidden	attribute values as: $+,-,*$	✗
[31]	Single	Fully Hidden	Multi-valued attributes	✗
[21]	Single	Fully Hidden	IPE	✗
[7]	Multi	Partially Hidden	Multi-valued attributes	✗
[30]	Multi	Fully Hidden	One-way anonymous key agreement	✗
[1]	Multi	Fully Hidden	One-way anonymous key agreement	✓
[17]	Multi	Fully Hidden	Randomizing-polynomial encodings	✗
[25]	Multi	✗	✗	✓
Ours	Multi	Fully Hidden	IPE + position of attribute$(+,-,*)$	✓

[1] $+$ or $-$ respectively refers to whether an attribute exists on the access policy or not.
[2] $*$ means that an attribute can be either positive or negative attributes.

ABE schemes with partially hidden access policy were proposed [5, 14, 28]. In most of them, each attribute in the access policy is represented as a couple: The attribute name and the attribute value. Generically, the attribute values contain more sensitive information. For example, the attribute values "secretary" and "CN2019" are more sensitive than the attribute names "Position" and "ID Number", respectively. The above ABE schemes protect the sensitive information by hiding the attribute values. However, the attribute names are revealed in the access policy $(Position : \star) \wedge (ID\ Number : \star)$. Therefore, there are a set of security issue in [5, 14, 28], especially the offline dictionary attacks on partially hidden access policy.

To address the security issues raised by ABE schemes with partially hidden access policy, ABE schemes with fully hidden access policy were introduced in [20, 21, 27, 31]. Xu *et al.* [20] extended the ABE scheme due to Bethencourt *et al.* [2], and proposed an ABE scheme with hidden access policy based on the tree-like access policy for cloud applications, in which the value of each attribute could be represented by three kinds of symbols: $+$, $-$, $*$. However, this scheme relies on only one authority to manage the private keys, so the center authority must be honest and overburdened. In 2015, Zhou *et al.* [31] introduced a privacy preserving attribute-based broadcast encryption scheme with an expressive hidden access policy. However, this construction introduces a high computation because of much pairing operations. In 2016, Phuong *et al.* [21] proposed a new hidden access policy ABE scheme under standard assumptions. Their scheme is based on the IPE and realizes the policy hiding by representing the attributes in the access policy with the position of symbols. Later, Jin *et al.* [12] extends Phuong's scheme to be fully secure one.

Most of the mentioned schemes either fail to consider the feature of hiding policy or are single-authority ABE. Recently, to solve these problems, some MA-ABE schemes with hidden access policy were presented [1, 7, 17, 30]. In 2016, Zhong *et al.* [30] proposed the first policy hidden ABE scheme using multiple attribute authorities architecture. However, the exponential computing cost is required during the decryption due to pairing operations. In 2017, Fan *et al.* [7] presented a MA-CP-ABE access control scheme with hidden policy and constant length ciphertext. But this scheme relies on a weaker model which is called weakly policy (attribute)-hiding. Under this model, a party might decrypt the received ciphertext but the policy is remained unknown to any users, which means the policy may be leaked only upon the final successful decryption. In 2018, Belguith *et al.* [1] proposed a securely outsourcing MA-ABE scheme based on LSSS with hidden policy for cloud assisted IoT. However, it is proven be selectively secure. Recently, Michalevsky et al proposed a full policy-hiding ABE based on IPE [17]. It supports conjunctions, disjunctions and threshold policies and protects the access policies from any user and party that are not authorized to recover the messages. However, this scheme needs coordinations among the authorities at the beginning of Setup algorithm. Additionally, their scheme relies on the random oracle and is reduced to the SXDH assumption and k-Lin assumption.

2.3 Outsourcing ABE

In most of the existing policy-hiding ABE schemes, The decryption computation costs grow proportionally with complexity of the access policy. Hence, many works solves them by using the outsourcing decryption method [9, 11, 13]. In 2017, Shao *et al.* introduced this method to decentralized MA-ABE to decrease the decryption cost [25]. However, their scheme relies on the random oracle and do not consider the hiding policy.

In conclusion, it is very urgent to propose a MA-ABE scheme with strong resistance to attacks, which can achieve the optimal compromise between privacy and efficiency. To evaluate the motivations given in introduction, we introduce a comparison of our scheme with other ABE constructions of recent years in Table 1, that are most closely-related to our scheme.

3 Techniques Preliminaries

To make the description concise, we first give some symbols used in this paper and their meanings. The details are shown in Table 2.

Table 2: Symbols used in this scheme

Symbol	Implication
U	The attribute universe in this system
S	The user's attribute list
W	The access policy
AA_k	The k-th attribute authority
AA_k^*	The k-th corrupted attribute authority
U_k	The attribute list managed by AA_k
S^k	$U_k \cap S$
N_1, N_2, N_3	The number of the symbols +,−,∗ respectively
$\overrightarrow{x_{V^k}}, \overrightarrow{x_{Z^k}}$	Two vectors converted by S^k
$\overrightarrow{v}$	A vector converted by W
n	The length of the vector $\overrightarrow{v}$, $\overrightarrow{x_{V^k}}$ or $\overrightarrow{x_{Z^k}}$
pp	The system public parameters
PK_k/SK_k	The public key/ secret key of AA_k
GID	The user's global identity
$SK_{GID,k,i}$	The attribute secret key of the user GID from AA_k

3.1 Bilinear Map and Complexity Assumptions

Definition 1. *(Bilinear map): Let $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ be two multiplicative cycle groups of same prime order p, g is the generator of $\mathbb{G}$. $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ is a bilinear map with the following properties:*

1) *Bilinearity: $\forall a, b \in Z_p$ and $e(g^a, g^b) = e(g, g)^{ab}$.*

2) *Non-Degeneracy: $e(g, g) \neq 1$;*

3) *Computability: $e(g, g)$ is polynomial-time computable.*

Assumption 1. *(DBDH Assumption): Let $a, b, c, z \in_R Z_p$. Given the tuple (A, B, C)=(g^a, g^b, g^c), the DBDH assumption holds when no polynomial-time algorithm $\mathcal{B}$ can distinguish $e(g, g)^{abc}$ and $e(g, g)^z$ with non-negligible advantage. The advantage of algorithm $\mathcal{B}$ is*

$$\begin{aligned} Adv_{\mathcal{B}}^{DBDH} =& \mid \Pr[\mathcal{B}(A, B, C, e(g, g)^{abc}) = 1] \\ & - \Pr[\mathcal{B}(A, B, C, e(g, g)^z) = 1] \mid \leq \epsilon. \end{aligned} \quad (1)$$

Assumption 2. *(DLIN Assumption): Let z_1, z_2, z_3, z_4, $z \in_R Z_p$. Given the tuple (Z_1, Z_2, Z_3, Z_4) = $(g^{z_1}, g^{z_2}, g^{z_3+z_4}, g^{z_2 z_4})$, the DLIN assumption holds when no polynomial-time algorithm $\mathcal{B}$ can distinguish $g^{z_1 z_3}$ and g^z with non-negligible advantage. The advantage of algorithm $\mathcal{B}$ is*

$$\begin{aligned} Adv_{\mathcal{B}}^{DLIN} =& \mid \Pr[\mathcal{B}(Z_1, Z_2, Z_3, Z_4, g^{z_1 z_3}) = 1] \\ & - \Pr[\mathcal{B}(Z_1, Z_2, Z_3, Z_4, g^z) = 1] \mid \leq \epsilon. \end{aligned} \quad (2)$$

3.2 Access Policy

Consider the access policy based on AND-gates with wildcards. Let the attribute universe descriptions be $U = \{Att_1, Att_2, ..., Att_L\}$. The user's attribute list is denoted as $S = \{S_1, S_2, ..., S_L\}$ where each attribute S_i could be: $+$ or $-$. Let $W = \{S_1^\star, S_2^\star, ..., S_L^\star\}$ be an AND-gate access policy with wildcards where each attribute $S_i^\star$ could be: $+$, $-$ or $*$. The notation $S \models W$ means that the user's attribute list satisfies the access policy.

3.3 The Viète's Formulas

Consider two vectors $\vec{p} = (p_i)$ and $\vec{q} = (q_i), i = 1, ..., L$, where p_i could be alphabets or wildcards, and q_i is alphabets. $H = \{h_1, ..., h_n\} \subset \{1, ..., L\}$ is defined by the positions of the wildcards in vector $\vec{p}$.

Let $\prod_{h \in H}(i - h) = \sum_{k=0}^{n} \lambda_k i^k$, where λ_k are the coefficients dependent on H. If $p_i = q_i \vee p_i = *$:

$$\sum_{i=1, i \notin H}^{L} p_i \prod_{h \in H} (i - h) = \sum_{k=0}^{n} \lambda_k \sum_{i=1}^{L} q_i i^k \quad (3)$$

The coefficient λ_k can be constructed by the Viète's Formulas as follows, where $n = |H|$.

$$\lambda_{n-k} = (-1)^k \sum_{1 \leq i_1 < i_2 < ... < i_k \leq n} h_{i_1} h_{i_2} ... h_{i_k}, 0 \leq k \leq n$$

4 Formal Definition and Security Model

4.1 System Model

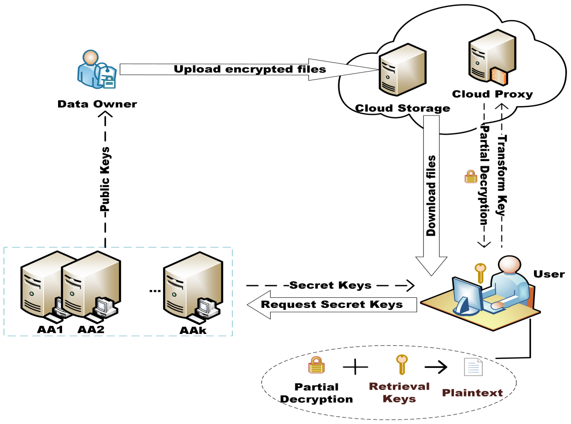


Figure 1: System model

There are five entities: Data owners (DO), data users (DU), several attribute authorities (AAs), and cloud server including cloud storage server(CS) and cloud proxy server(CP) in the system as showed in Figure 1, the details are as follows:

Step 1: In the system initialization stage, the public parameters are generated, and each AA generates the public keys and sends to DO.

Step 2: DO specifies the access policy and encrypts files using the public keys and the access policy, then uploads the encrypted files to CS, in which CS is used to store encrypted files and provide access services for DU.

Step 3: After DU downloads the encrypted file from CS, if DU wants to decrypt it, DU needs to request the secret keys to AAs. Notice that the encrypted file can be decrypted successfully by DU, if and only if DU's attribute list satisfies the access policy.

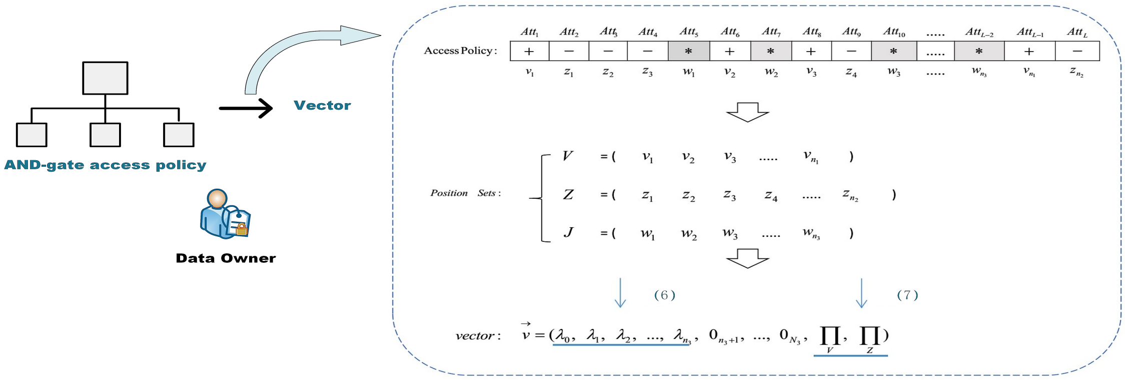

AND-gate access policy
Data Owner
Vector
Access Policy:
Att_1 Att_2 Att_3 Att_4 Att_5 Att_6 Att_7 Att_8 Att_9 Att_{10} ..... Att_{L-2} Att_{L-1} Att_L
+ − − − * + * + − * ..... * + −
v_1 z_1 z_2 z_3 w_1 v_2 w_2 v_3 z_4 w_3 ..... w_{n_3} v_{n_1} z_{n_2}
Position Sets:
$V = (v_1\ v_2\ v_3\ .....\ v_{n_1})$
$Z = (z_1\ z_2\ z_3\ z_4\ .....\ z_{n_2})$
$J = (w_1\ w_2\ w_3\ .....\ w_{n_3})$
(6)
(7)
vector: $\vec{v} = (\underline{\lambda_0, \lambda_1, \lambda_2, ..., \lambda_{n_3}}, 0_{n_3+1}, ..., 0_{N_3}, \underline{\prod_V, \prod_Z})$

Figure 2: convert the access policy into an vector

Step 4: After receiving the request of DU, each AA verifies DU's identity, then distributes the secret keys for legitimate DU.

Step 5: To reduce the burden of calculation, DU converts the secret keys into the transform keys to CP, and remains the retrieval keys. Then CP is responsible for partial decryption.

Step 6: Finally, DU can recover the plaintext using retrieval keys and the information of the partial decryption.

4.2 Scheme Definition

The scheme consists of seven algorithms as follows:

Global setup $(1^\lambda \to \boldsymbol{pp})$**:** The system is produced at this stage. It inputs security parameters λ, and returns the public parameters pp.

Authority setup $(\boldsymbol{pp}, \boldsymbol{k} \to \boldsymbol{PK}_k, \boldsymbol{SK}_k)$**:** It inputs pp, and the authority index k, then it outputs the authority's public keys PK_k and secret keys SK_k.

Encryption $(\boldsymbol{pp}, W, M, \boldsymbol{PK}_k \to \boldsymbol{CT})$**:** It inputs pp, the public key PK_k, the message M, and the access policy W, then outputs the ciphertext CT to CS.

KeyGen $(\boldsymbol{pp}, \boldsymbol{SK}_k, \mathbf{GID}, S \to \boldsymbol{SK}_{GID,k,i})$**:** It takes SK_k, GID, attributes set S as input, returns the secret keys $SK_{GID,k,i}$ to DU.

TransKeyGen$(pp, SK_{GID,k,i} \to TK_{GID,k,i}, RK_{GID,k,i})$**:** It takes pp and $SK_{GID,k,i}$ as input, then returns transformation keys $TK_{GID,k,i}$ to CP and retains a retrieval key $RK_{GID,k,i}$ to DO.

Out.Decryption $(\boldsymbol{pp}, \boldsymbol{CT}, \boldsymbol{TK}_{GID,k,i} \to \widehat{CT})$**:** It inputs pp, CT, and $TK_{GID,k,i}$, then returns $\widehat{CT}$ to DU.

User.Decryption $(\boldsymbol{pp}, \widehat{CT}, \boldsymbol{RK}_{GID,k,i} \to M)$**:** It takes $\widehat{CT}$ and $RK_{GID,k,i}$ as input, then outputs the recovered M.

4.3 Security Model

Based on DBDH and DLIN assumption, the scheme is proven to be the selective IND-CPA security by the security game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$. The details are as following:

Initialization: $\mathcal{A}$ submits two challenge access structures W_0, W_1 and a series of corrupted authorities AA_k^* to $\mathcal{C}$, where $|AA_k^*| < \mathcal{K}$.

Global Setup: $\mathcal{C}$ runs the *Global Setup* algorithm and outputs pp to $\mathcal{A}$.

Authorities Setup:

1) For the corrupted authorities, $\mathcal{C}$ sends PK_k and SK_k to $\mathcal{A}$.
2) For the honest authorities, $\mathcal{C}$ sends PK_k to $\mathcal{A}$.
3) For the half-honest authorities, $\mathcal{C}$ sends PK_k and parts of SK_k to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ submits the attribute set S and GID for querying secret keys. If $(S \models W_0 \wedge S \models W_1)$ or $(S \nvDash W_0 \wedge S \nvDash W_1)$, $\mathcal{C}$ sends SK_S to the adversary. $\mathcal{A}$ can query polynomially.

Challenge: $\mathcal{A}$ submits two equal length messages M_0 and M_1. $\mathcal{C}$ flips a random coin ξ and runs the Encryption algorithm. $\mathcal{C}$ sends CT_ξ to $\mathcal{A}$. Note that if $\mathcal{A}$ obtains SK_S under the condition $(S \models W_0 \wedge S \models W_1)$ in Phase 1, then it is needed that $M_0 = M_1$.

Phase 2: Phase 1 is repeated. If $M_0 \neq M_1$, $\mathcal{A}$ can't submit $S^{'}$ such that $S^{'} \models W_0 \wedge S^{'} \models W_1$.

Guess: Finally, $\mathcal{A}$ outputs his guess $\xi^{'}$ on ξ.

Definition 2. *The decentralized ABE scheme with fully hidden policy is selective IND-CPA security if against any probabilistic polynomial-time adversary* $\mathcal{A}$*,*

$$Adv_{\mathcal{A}}^{IND-CPA}(\lambda) = |\Pr[\xi^{'} = \xi] - \frac{1}{2}| \qquad (4)$$

is negligible in the security parameter λ*.*

5 Our Construction

5.1 Extending Phuong's Technology

In our scheme, we also extend Phoung's technique [21] to convert an access policy into a vector $\overrightarrow{v}$ which is combined with the technique of IPE to encrypt the data. In addition, for each authority, the attribute set S^k is converted into two vectors $\overrightarrow{x_{V^k}}, \overrightarrow{x_{Z^k}}$ which is used in key generation. Again, the conversion is performed by combining with the Viète's formulas and the positions of symbols. The details are showed as following:

5.1.1 Convert the Access Policy into an Vector

Firstly, the access policy W that consists of $+$, $-$, and $*$ can be separated into three position sets: V, Z, and J, which contains the positions of $+$, $-$, and $*$ in W respectively, where let $V = \{v_1, ..., v_{n_1}\}, Z = \{z_1, ..., z_{n_2}\}, J = \{w_1, ..., w_{n_3}\}$ $(n_i \leq N_i,\ i$=1, 2, 3). Next, based on the position set J and the Viète's formulas, we can calculate the coefficients $(\lambda_0, \lambda_1, ..., \lambda_{n_3})$, as $\lambda_{n_3} = 1, \lambda_{n_3-1} = -(w_1 + w_2 ... + w_{n_3}), \lambda_{n_3-2} = (w_1w_2 + w_1w_3 + ... + w_{n_3-1}w_{n_3}), ..., \lambda_0 = -(w_1w_2...w_{n_3})$.

And construct a polynomial $\sum_{k=0}^{n_3} \lambda_k i^k$, where i is the position of $+$ or $-$. Then we combine V and Z respectively as follows:

$$\begin{aligned}\prod V &= +\sum{}_{v_i \in V} \prod{}_{w_j \in J}(v_i - w_j), \\ \prod Z &= -\sum{}_{z_i \in Z} \prod{}_{w_j \in J}(z_i - w_j).\end{aligned} \tag{5}$$

Finally, we can convert the access policy W into a vector

$$\begin{aligned}\overrightarrow{v} =&(v_1, v_2, ..., v_n), \\ =&(\lambda_0, \lambda_1, ..., \lambda_{n_3}, 0_{n_3+1}, ..., 0_{N_3}, \prod V, \prod Z).\end{aligned} \tag{6}$$

where N_1, N_2, $N_3 \leq L$ show the maximum number of $+$, $-$, and $*$ in an access policy respectively. The process is shown in Figure 2.

5.1.2 Convert the Attributes set S^k into Two Vectors

In user key generation, attributes set S^k containing $+$ and $-$ attributes also need to be separated into two sets V^k and Z^k which contains respectively positions of positive and negative attributes. Then calculate:

$$v_l^* = -\sum_{v_i^k \in V^k} {v_i^k}^l,\ z_l^* = +\sum_{z_i^k \in Z^k} {z_i^k}^l\ (l = 0, ..., N_3). \tag{7}$$

Finally, the attributes set S^k is converted into two vectors:

$$\begin{aligned}\overrightarrow{x_{V^k}} =&(x_{V_1^k}, x_{V_2^k}, ..., x_{V_n^k}) = (v_0^*, v_1^*, ..., v_{N_3}^*, 1/\mathcal{K}, 0), \\ \overrightarrow{x_{Z^k}} =&(x_{Z_1^k}, x_{Z_2^k}, ..., x_{Z_n^k}) = (z_0^*, z_1^*, ..., z_{N_3}^*, 0, 1/\mathcal{K}).\end{aligned} \tag{8}$$

The process is shown in Figure 3, in which we assume $(Att_1,\ Att_2,\ Att_3) \subseteq\ U_1$; $(Att_4,\ Att_5,\ Att_6,\ Att_7) \subseteq\ U_2$; $(Att_8,\ Att_9,\ Att_{10}) \subseteq\ U_3$; and so on; $(Att_{L-2},\ Att_{L-1},\ Att_L) \subseteq\ U_{\mathcal{K}}$;where U_k is the attribute set be managed by authority A_k. In summary, $(\overrightarrow{v},\ \sum_{k=1}^{\mathcal{K}}\ \overrightarrow{x_{V^k}}) = 0$, $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{Z^k}}) = 0$ iff $v_i = v_i^* \vee v_i = *$ and $z_i = z_i^* \vee z_i = *$, since combining Figure 2 and Figure 3, calculating:

$$\begin{aligned}&(\overrightarrow{v}, \sum{}_{k=1}^{\mathcal{K}} \overrightarrow{x_{V^k}}) \\ =& -({v_1^*}^0 + {v_2^*}^0 \cdots + {v_{n_1}^*}^0) \cdot \lambda_0 - ({v_1^*}^1 + {v_2^*}^1 \cdots \\ &+ {v_{n_1}^*}^1) \cdot \lambda_1 \cdots - ({v_1^*}^{N_3} + {v_2^*}^{N_3} \cdots + {v_{n_1}^*}^{N_3}) \\ &\cdot \lambda_{N_3} + \prod V \cdot (\sum{}_{k=1}^{\mathcal{K}} 1/\mathcal{K}) \\ =& -({v_1^*}^0 + {v_2^*}^0 \cdots + {v_{n_1}^*}^0) \cdot \lambda_0 - ({v_1^*}^1 + {v_2^*}^1 \cdots \\ &+ {v_{n_1}^*}^1) \cdot \lambda_1 \cdots - ({v_1^*}^{N_3} + {v_2^*}^{N_3} \cdots + {v_{n_1}^*}^{N_3}) \\ &\cdot \lambda_{N_3} + (v_1^0 + v_2^0 \cdots + v_{n_1}^0) \cdot \lambda_0 + (v_1^1 + v_2^1 \cdots \\ &+ v_{n_1}^1) \cdot \lambda_1 \cdots + (v_1^{N_3} + v_2^{N_3} \cdots + v_{n_1}^{N_3}) \cdot \lambda_{N_3}.\end{aligned} \tag{9}$$

$$\begin{aligned}&(\overrightarrow{v}, \sum{}_{k=1}^{\mathcal{K}} \overrightarrow{x_{Z^k}}) \\ =& +({z_1^*}^0 + {z_2^*}^0 \cdots + {z_{n_2}^*}^0) \cdot \lambda_0 + ({z_1^*}^1 + {z_2^*}^1 \cdots \\ &+ {z_{n_2}^*}^1) \cdot \lambda_1 \cdots + ({z_1^*}^{N_3} + {z_2^*}^{N_3} \cdots + {z_{n_2}^*}^{N_3}) \\ &\cdot \lambda_{N_3} - \prod Z \cdot (\sum{}_{k=1}^{\mathcal{K}} 1/\mathcal{K}) \\ =& +({z_1^*}^0 + {z_2^*}^0 \cdots + {z_{n_2}^*}^0) \cdot \lambda_0 + ({z_1^*}^1 + {z_2^*}^1 \cdots \\ &+ {z_{n_2}^*}^1) \cdot \lambda_1 \cdots + ({z_1^*}^{N_3} + {z_2^*}^{N_3} \cdots - {z_{n_2}^*}^{N_3}) \\ &\cdot \lambda_{N_3} - (z_1^0 + z_2^0 \cdots + z_{n_2}^0) \cdot \lambda_0 - (z_1^1 + z_2^1 \cdots \\ &+ z_{n_2}^1) \cdot \lambda_1 \cdots - (z_1^{N_3} + z_2^{N_3} \cdots + z_{n_2}^{N_3}) \cdot \lambda_{N_3}\end{aligned}$$

5.2 Decentralizing Attribute-Based Access Control Scheme With Fully Hidden Policy

The algorithm of our scheme is presented as follows:

Global Setup: Given the security parameter λ, the algorithm returns a bilinear group $param$ = (p, g, e, $\mathbb{G}$, $\mathbb{G}_{\mathbb{T}}$). Let $H : \{0,1\}^* \to \mathbb{G}$ be a hash function, and $n = N_3 + 3$. Defining that there are $\mathcal{K}$ authorities in the system, and each authority AA_k manages disjoint attribute set $U_k = \{Att_1, Att_2, ..., Att_{n_k}\}$, where$|U_k|$ = n_k. Later, it selects randomly $\{\Delta,\ f_1,\ f_2,\ \mu_1,\ \mu_2,\ \theta_1,\ \theta_2\} \in Z_p$, $g_2 \in \mathbb{G}$, then publishes the public parameters pp as follows:

$$\begin{aligned}pp =&\{param, V_1 = g^{\mu_1}, V_2 = g^{\mu_2}, \\ &X_1 = g^{\theta_1}, X_2 = g^{\theta_2}, g_1 = g^{\Delta}\}\end{aligned} \tag{10}$$

Authority Setup: The algorithm is run by AA_k as Algorithm 1.

Encryption: The algorithm is run by DO, the detailed process is as Algorithm 2.

KeyGen: DU submits $u = H(GID)$ and S to AA_k for requesting the secret keys. Each AA_k runs Algorithm 3 and distributes the secret keys to DU.

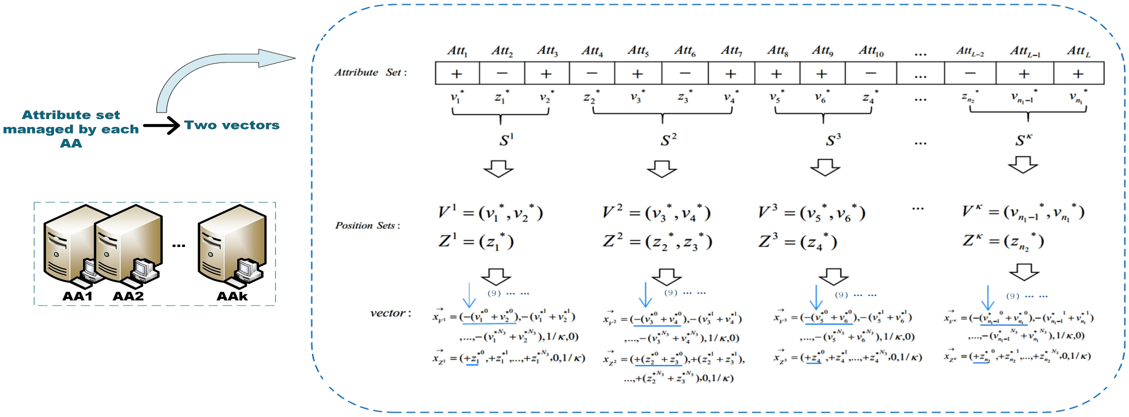

Attribute set managed by each AA
Two vectors
AA1
AA2
...
AAk
Attribute Set :
Att1 Att2 Att3 Att4 Att5 Att6 Att7 Att8 Att9 Att10 ... AttL−2 AttL−1 AttL
+ − + − + − + + + − ... − + +
v1* z1* v2* z2* v3* z3* v4* v5* v6* z4* ... zn2* vn1−1* vn1*
S1 S2 S3 ... Sκ
Position Sets :
V1 = (v1*, v2*)
Z1 = (z1*)
V2 = (v3*, v4*)
Z2 = (z2*, z3*)
V3 = (v5*, v6*)
Z3 = (z4*)
Vκ = (vn1−1*, vn1*)
Zκ = (zn2*)
(9) ... ...
vector :
xV1 = (−(v1*0 + v2*0), −(v1*1 + v2*1), ..., −(v1*N3 + v2*N3), 1/κ, 0)
xZ1 = (+z1*0, +z1*1, ..., +z1*N3, 0, 1/κ)
xV2 = (−(v3*0 + v4*0), −(v3*1 + v4*1), ..., −(v3*N3 + v4*N3), 1/κ, 0)
xZ2 = (+(z2*0 + z3*0), +(z2*1 + z3*1), ..., +(z2*N3 + z3*N3), 0, 1/κ)
xV3 = (−(v5*0 + v6*0), −(v5*1 + v6*1), ..., −(v5*N3 + v6*N3), 1/κ, 0)
xZ3 = (+z4*0, +z4*1, ..., +z4*N3, 0, 1/κ)
xVκ = (−(vn1−1*0 + vn1*0), −(vn1−1*1 + vn1*1), ..., −(vn1−1*N3 + vn1*N3), 1/κ, 0)
xZκ = (+zn2*0, +zn2*1, ..., +zn2*N3, 0, 1/κ)

Figure 3: Convert the attribute set into two vectors

Algorithm 1 Authority Setup

```
Require: $pp$, $k$
Ensure: $SK_k$, $PK_k$
  for each authority $AA_k$ in system do
      select $\alpha_k$, $\gamma_k$, $\beta_k$, $\zeta_k$, $\varsigma_k$, $\eta_k$;compute: $T_k = g^{\gamma_k}$, $Z_k = g^{\beta_k}$, $M_k = g^{\zeta_k}$, $N_k = g^{\varsigma_k}$, $Y_k = e(g,\ g_2)^{\alpha_k}$;
      for $i$ in [1,$n$] do
          select successively $u_{1,i,k}$,$w_{1,i,k}$,$u_{2,i,k}$,$w_{2,i,k}$
          under following condition:
          $\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$;
          compute: $U_{1,i,k} = g^{u_{1,i,k}}$, $U_{2,i,k} = g^{u_{2,i,k}}$,
           $W_{1,i,k} = g^{w_{1,i,k}}$, $W_{2,i,k} = g^{w_{2,i,k}}$;
      end for
  end for
  return $PK_k$=($\{Y_k,\ T_k,\ Z_k,\ M_k,\ N_k\}$, $\{U_{1,i,k},\ U_{2,i,k},\ W_{1,i,k},\ W_{2,i,k}\}_{i=1}^{n}$) and $SK_k$=($\{\alpha_k,\ \gamma_k,\ \beta_k,\ \zeta_k,\ \varsigma_k,\ \eta_k\}$, $\{u_{1,i,k},\ u_{2,i,k},\ w_{1,i,k},\ w_{2,i,k}\}_{i=1}^{n})_{k=1}^{\mathcal{K}}$;
```

TransKeyGen: DU chooses a random number $z \in Z_p$, and constructs the transformation keys $TK_{GID,k,i}$ and the retrieval keys $RK_{GID,k,i}$ as follows. Note that $TK_{GID,k,i}$ is sent to CP, and $RK_{GID,k,i} = z$ is remained.

$$\begin{aligned} TK_{GID,k,i} =&({K_{A_k}}^{\frac{1}{z}}, {K_{B_k}}^{\frac{1}{z}}, {K_{1,i,k}}^{\frac{1}{z}}, \\ &{K_{2,i,k}}^{\frac{1}{z}}, {K_{3,i,k}}^{\frac{1}{z}}, {K_{4,i,k}}^{\frac{1}{z}})_{i=1}^{n}{}_{k=1}^{\mathcal{K}} \\ =&({K_{A_k}}', {K_{B_k}}', {K_{1,i,k}}', \\ &{K_{2,i,k}}', {K_{3,i,k}}', {K_{4,i,k}}')_{i=1}^{n}{}_{k=1}^{\mathcal{K}}. \end{aligned} \tag{11}$$

Out.Decryption: CP runs the Out.Decryption algorithm and calculates as follow:

$$CT_1 = \prod_{k=1}^{\mathcal{K}} \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i,k}, {K_{j,i,k}}') \tag{12}$$

$$CT_2 = \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} [e(C_A, {K_{A_k}}') \cdot e(C_B, {K_{B_k}}')].$$

then returns $\widehat{CT} = \{CT_1, CT_2\}$ to DU.

Algorithm 2 Encryption

```
Require: $pp$, $PK_k$, $W$, $M \in \mathbb{G}_{\mathbb{T}}$
Ensure: the ciphertext $CT$
  for each data owner in system do
      convert $W$ into the vector $\overrightarrow{v}$ as subsection 5.1.1;
      select $s_1, s_2, \beta \in Z_p$, compute $C_A = g^{s_2}$, $C_B = g_1^{s_1}$;
      for $k$ in [1,$\mathcal{K}$] do
          compute:$C_0 = \prod_{k=1}^{\mathcal{K}} M \cdot e(g, g_2)^{\alpha_k s_2}$;
          for $i$ in [1,$n$] do
              compute: $C_{1,i,k} = {U_{1,i,k}}^{s_1} \cdot {V_1}^{v_i \beta} \cdot T_k^{s_2}$,
              $C_{2,i,k} = {U_{2,i,k}}^{s_1} \cdot {V_2}^{v_i \beta} \cdot Z_k^{s_2}$,
              $C_{3,i,k} = {W_{1,i,k}}^{s_1} \cdot {X_1}^{v_i \beta} \cdot M_k^{s_2}$,
              $C_{4,i,k} = {W_{2,i,k}}^{s_1} \cdot {X_2}^{v_i \beta} \cdot N_k^{s_2}$;
          end for
      end for
  end for
  return the ciphertext $CT$=($C_0$,$C_A$,$C_B$,$\{C_{1,i,k},\ C_{2,i,k},\ C_{3,i,k},\ C_{4,i,k}\}_{i=1}^{n}{}_{k=1}^{\mathcal{K}}$);
```

User.Decryption: After obtaining $\widehat{CT}$ from CP, DU runs the *User.Decryption* algorithm and calculates as follows: $C_0/(CT_1 \cdot CT_2)^z = M$.

5.3 Correction Analysis

Calculate firstly as follow:

$$\begin{aligned} &e(C_{1,i,k}, K_{1,i,k}) \\ =&e({U_{1,i,k}}^{s_1} V_1^{v_i \beta} T_k^{s_2}, V_2^{-r_{1,i,k}} {U_{2,i,k}}^{\frac{x_{V_i^k}}{u+f_1}+\eta_k}) \\ =&e(g, K_{1,i,k})^{s_2 \gamma_k} \cdot e(g,g)^{s_1(-u_{1,i,k}\mu_2) r_{1,i,k}} \\ &\cdot e(g,g)^{(\frac{x_{V_i^k}}{u+f_1}+\eta_k) \cdot s_1 u_{1,i,k} u_{2,i,k}} \cdot e(g,g)^{-\mu_1 \mu_2 \beta v_i r_{1,i,k}} \\ &\cdot e(g,g)^{\beta(u_{2,i,k}\mu_1) v_i \cdot (\frac{x_{V_i^k}}{u+f_1}+\eta_k)} \end{aligned} \tag{13}$$

Algorithm 3 Key Generation

```
Require: $SK_k$, $S$, GID
Ensure: user's secret key $SK_{GID,k,i}$
  for each authority $AA_k$ in system do
    convert $S^k$ into vectors: $\overrightarrow{x_{V^k}}$,$\overrightarrow{x_{Z^k}}$ as subsection 5.1.2
    for $i$ in [1,$n$] do
      select randomly $r_{i,k,1}, r_{i,k,2}$;
      compute $K_{1,i,k} = {V_2}^{-r_{1,i,k}} \cdot {U_{2,i,k}}^{\frac{x_{V_i^k}}{u+f_1}+\eta_k}$,
      $K_{2,i,k} = {V_1}^{r_{1,i,k}} \cdot {U_{1,i,k}}^{-(\frac{x_{V_i^k}}{u+f_1}+\eta_k)}$,
      $K_{3,i,k} = {X_2}^{-r_{2,i,k}} \cdot {W_{2,i,k}}^{\frac{x_{Z_i^k}}{u+f_2}-\eta_k}$,
      $K_{4,i,k} = {X_1}^{r_{2,i,k}} \cdot {W_{1,i,k}}^{-(\frac{x_{Z_i^k}}{u+f_2}-\eta_k)}$,
      $K_{A_k} = {g_2}^{\alpha_k} \prod_{i=1}^{n} ({K_{1,i,k}}^{-\gamma_k} {K_{2,i,k}}^{-\beta_k} {K_{3,i,k}}^{-\zeta_k}$
      ${K_{4,i,k}}^{-\varsigma_k})$, $K_{B_k} = \prod_{i=1}^{n} g^{-(r_{1,i,k}+r_{2,i,k})\triangle}$;
    end for
  end for
  return the user's secret key $SK_{GID,k,i}$=($K_{A_k}$, $K_{B_k}$, $\{K_{1,i,k}, K_{2,i,k}, K_{3,i,k}, K_{4,i,k}\}_{i=1}^{n}{}_{k=1}^{\mathcal{K}}$);
```

Then we have:

$$
\begin{aligned}
CT_1 = & \prod_{k=1}^{\mathcal{K}} \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i,k}, K_{j,i,k})^{\frac{1}{z}} \\
=[& \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{1,i,k})^{\gamma_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{2,i,k})^{\beta_k s_2} \\
& \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{3,i,k})^{\zeta_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{4,i,k})^{\varsigma_k s_2} \\
& \cdot e(g,g)^{\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} (r_{1,i,k}+r_{2,i,k}) s_1 \Delta} \\
& \cdot e(g,g)^{\sum\limits_{k=1}^{\mathcal{K}} \sum\limits_{i=1}^{n} (\frac{x_{V_i^k} v_i \beta \Delta}{f_1+u} + \frac{x_{Z_i^k} v_i \beta \Delta}{f_2+u})}]^{\frac{1}{z}}
\end{aligned}
$$

Also have:

$$
\begin{aligned}
CT_2 = & \prod_{k=1}^{\mathcal{K}} [e(C_A, K_{A_k})^{\frac{1}{z}} \cdot e(C_B, K_{B_k})^{\frac{1}{z}}] \\
=[& e(g^{s_2}, g_2^{\sum_{k=1}^{\mathcal{K}} \alpha_k}) \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{1,i,k})^{-\gamma_k s_2} \\
& \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{2,i,k})^{-\beta_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, \\
& K_{3,i,k})^{-\zeta_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{4,i,k})^{-\varsigma_k s_2} \\
& \cdot e(g,g)^{-\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} (r_{1,i,k}+r_{2,i,k}) s_1 \Delta}]^{\frac{1}{z}}
\end{aligned}
$$

Finally, we have:

$$
\begin{aligned}
& \frac{C_0}{(CT_1 \cdot CT_2)^z} \qquad (14) \\
= & \frac{M}{e(g,g)^{\frac{(\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} x_{V_i^k} v_i)\beta\Delta}{f_1+u}} \cdot e(g,g)^{\frac{(\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} x_{Z_i^k} v_i)\beta\Delta}{f_2+u}}}
\end{aligned}
$$

Therefore, the message M will be recovered iff $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{V^k}}) = 0$ and $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{Z^k}}) = 0$, meaning that users' attributes list satisfies the access policy.

5.4 Security Against Attack

A basic requirement of the decentralized ABE scheme is to prevent collusion between users, meaning that any two or more users who are not authorized to decrypt individually can successfully decrypt by combining their keys. In our scheme, GID is introduced to solve this problem as [3], and GID is coupled non-linearly with f_1 and f_2 in the secret keys to resist the attack mentioned by Rahulamathavan *et al.* in [23].

In addition, our scheme is proven to resist the collusion attack mentioned in [29] as follows. Suppose that there are three attribute authorities: AA_1, AA_2, AA_3, which monitor respectively attribute att_1, att_2, att_3. The access policy is specified as $W = \{att_1, att_2, att_3\}$. Consider that two users u_1 and u_2, with attribute sets $S_1 = \{att_1, att_2\}$ and $S_2 = \{att_2, att_3\}$ respectively, hope to decrypt the ciphertext by collusion.

$$
\begin{aligned}
u_1 : & K_{j,i,1}(u_1), K_{j,i,2}(u_1); (where j = 1,2,3,4) \\
u_2 : & K_{j,i,2}(u_2), K_{j,i,3}(u_2); \\
CT : & C_0 = M \cdot e(g, g_2)^{s_2(\alpha_1+\alpha_2+\alpha_3)}, \\
& C_A, C_B, \{C_{j,i,1}, C_{j,i,2}, C_{j,i,3}\}. \qquad (15)
\end{aligned}
$$

Then we use the secret keys of u_1 and u_2 to decrypt the ciphertext CT. When calculate:

$$
\begin{aligned}
& \prod_{j=1}^{4} [e(C_{j,i,1}, K_{j,i,1}(u_1)) \cdot e(C_{j,i,2}, K_{j,i,2}(u_1)) \\
& \cdot e(C_{j,i,3}, K_{j,i,3}(u_2))] \qquad (16)
\end{aligned}
$$

we find that the collusion is prevented by two special items:

$$
\begin{aligned}
& e(g,g)^{(\frac{x_{V_i^1}}{f_1+u_1} + \frac{x_{V_i^2}}{f_1+u_1} + \frac{x_{V_i^3}}{f_1+u_2}) v_i \beta \Delta} \neq e(g,g)^0 \qquad (17) \\
& e(g,g)^{(\frac{x_{Z_i^1}}{f_2+u_1} + \frac{x_{Z_i^2}}{f_2+u_1} + \frac{x_{Z_i^3}}{f_2+u_2}) v_i \beta \Delta} \neq e(g,g)^0
\end{aligned}
$$

So the message M can not be recovered.

Moreover, the existence of η_k prevents the leakage when the malicious user lets attribute vectors $\overrightarrow{x_{V^k}}$ and $\overrightarrow{x_{Z^k}}$ equal to $\overrightarrow{0}$. If $\eta_k = 0$, the secret value $r_{i,k,1}$ and $r_{i,k,2}$ associated with the attribute will be leaked.

In conclusion, the decentralizing MA-ABE scheme resists strongly attacks from potential malicious users.

6 Security Analysis

Theorem 1. *If the DBDH and DLIN assumption hold in group $\mathbb{G}$, then our decentralizing ABE scheme is selective IND-CPA secure and policy hiding.*

Proof. The proof technique is similar to that of the scheme in [21] except that the corrupted authorities need

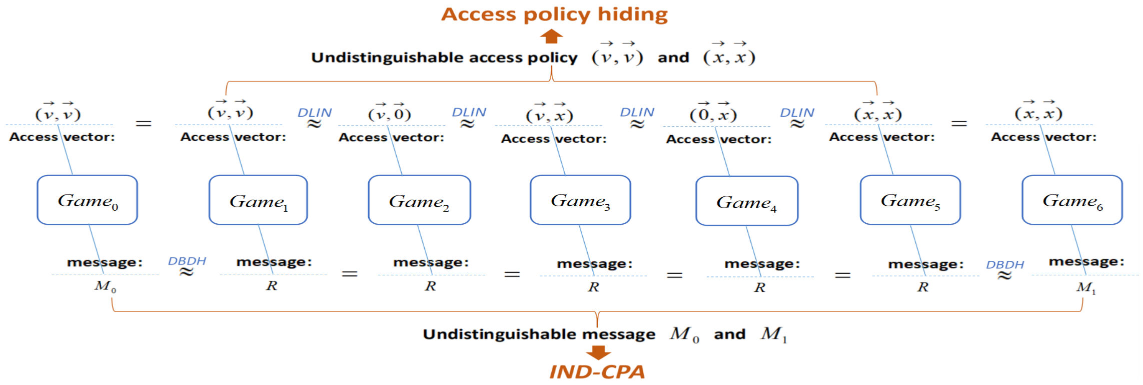

Access policy hiding
Undistinguishable access policy $(\overrightarrow{v},\overrightarrow{v})$ and $(\overrightarrow{x},\overrightarrow{x})$
$(\overrightarrow{v},\overrightarrow{v})$ Access vector:
=
$(\overrightarrow{v},\overrightarrow{v})$ Access vector:
DLIN ≈
$(\overrightarrow{v},\overrightarrow{0})$ Access vector:
DLIN ≈
$(\overrightarrow{v},\overrightarrow{x})$ Access vector:
DLIN ≈
$(\overrightarrow{0},\overrightarrow{x})$ Access vector:
DLIN ≈
$(\overrightarrow{x},\overrightarrow{x})$ Access vector:
=
$(\overrightarrow{x},\overrightarrow{x})$ Access vector:
$Game_0$
$Game_1$
$Game_2$
$Game_3$
$Game_4$
$Game_5$
$Game_6$
message: M_0
DBDH ≈
message: R
=
message: R
=
message: R
=
message: R
=
message: R
DBDH ≈
message: M_1
Undistinguishable message M_0 and M_1
IND-CPA

Figure 4: The analysis of the security proof

to be considered in ours. Suppose there is at least one honest center to distribute the private keys in the system. Since the message M is encrypted by an vector that is transformed by the access policy in our scheme. To prove that the policy is hidden, it is required only to prove that the two vectors $\overrightarrow{v}$ and $\overrightarrow{x}$ cannot be distinguished by the adversary, where $\overrightarrow{v}$ and $\overrightarrow{x}$ are corresponding to W_0 and W_1 respectively. □

The following two cases $M_0 = M_1$ and $M_0 \neq M_1$ will be considered. For $M_0 = M_1$, we only prove the property of policy hiding by discussing games from $Game_1$ to $Game_5$ in sequence. For $M_0 \neq M_1$, we need to discuss the whole proof from $Game_0$ to $Game_6$. This specific process is shown in Figure 4.

Firstly, a high level description of each game is given as follows, where $i = 1, ..., n$; $k = 1, ..., \mathcal{K}$.

- $Game_0$: The access policy $(\overrightarrow{v}, \overrightarrow{v})$ is used to encrypt the message M_0. The ciphertext CT_0 is as follows:
$$(M_0\prod_{k=1}^{\mathcal{K}}Y_k^{s_2}, C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}V_1^{v_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2}V_2^{v_i\beta}, W_{1,i,k}^{s_1}M_k^{s_2}X_1^{v_i\beta}, W_{2,i,k}^{s_1}N_k^{s_2}X_2^{v_i\beta}\})$$
- $Game_1$: The access policy $(\overrightarrow{v}, \overrightarrow{v})$ is used to encrypt a random message $R \in \mathbb{G}_{\mathbb{T}}$. The ciphertext CT_1 is as follows:
$$(R', C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}V_1^{v_i\beta}, U_{2,i,k}^{s_1}Z_k^{s_2} V_2^{v_i\beta}, W_{1,i,k}^{s_1}M_k^{s_2}X_1^{v_i\beta}, W_{2,i,k}^{s_1}N_k^{s_2}X_2^{v_i\beta}\})$$
- $Game_2$: The access policy $(\overrightarrow{v}, \overrightarrow{0})$ is used to encrypt a random message R. The ciphertext CT_2 is as follows:
$$(R', C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}V_1^{v_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2}V_2^{v_i\beta}, W_{1,i,k}^{s_1}M_k^{s_2}, W_{2,i,k}^{s_1}N_k^{s_2}\})$$
- $Game_3$: The access policy $(\overrightarrow{v}, \overrightarrow{x})$ is used to encrypt a random message R. The ciphertext CT_3 is as follows:
$$(R', C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}V_1^{v_i\beta}, U_{2,i,k}^{s_1}Z_k^{s_2} V_2^{v_i\beta}, W_{1,i,k}^{s_1}M_k^{s_2}X_1^{x_i\beta}, W_{2,i,k}^{s_1}N_k^{s_2}X_2^{x_i\beta}\})$$
- $Game_4$: The access policy $(\overrightarrow{0}, \overrightarrow{x})$ is used to encrypt a random message R. The ciphertext CT_4 is as follows:
$$(R', C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}, U_{2,i,k}^{s_1}Z_k^{s_2}, W_{1,i,k}^{s_1}M_k^{s_2}X_1^{x_i\beta}, W_{2,i,k}^{s_1}N_k^{s_2}X_2^{x_i\beta}\})$$
- $Game_5$: The access policy $(\overrightarrow{x}, \overrightarrow{x})$ is used to encrypt a random message R. The ciphertext CT_5 is as follows:
$$(R', C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}V_1^{x_i\beta}, U_{2,i,k}^{s_1}Z_k^{s_2} V_2^{x_i\beta}, W_{1,i,k}^{s_1}M_k^{s_2}X_1^{x_i\beta}, W_{2,i,k}^{s_1}N_k^{s_2}X_2^{x_i\beta}\})$$
- $Game_6$: The access policy $(\overrightarrow{x}, \overrightarrow{x})$ is used to encrypt the message $M_1 \in \mathbb{G}_{\mathbb{T}}$. The ciphertext CT_6 is as follows:
$$(M_1\prod_{k=1}^{\mathcal{K}}Y_k^{s_2}, C_A, C_B, \{U_{1,i,k}^{s_1}T_k^{s_2}V_1^{x_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2}V_2^{x_i\beta}, W_{1,i,k}^{s_1}M_k^{s_2}X_1^{x_i\beta}, W_{2,i,k}^{s_1}N_k^{s_2}X_2^{x_i\beta}\})$$

6.1 Indistinguishability Between $Game_0$ and $Game_1$

Lemma 1. *For any adversary$\mathcal{A}$, $Game_0$ and $Game_1$ could be distinguished with a non-negligible advantage, then there exists algorithm $\mathcal{B}$ that could solve the DBDH assumption with a non-negligible advantage, i.e.*

$$| Adv_{Game_0}(\lambda) - Adv_{Game_1}(\lambda) | \leq Adv_{\mathcal{B}}^{DBDH}(\lambda) \tag{18}$$

Proof. Let $\overrightarrow{y} = \{g, A = g^a, B = g^b, C = g^c\}$. The challenger $\mathcal{C}$ generates the bilinear group $(e,\ p,\ g,\ \mathbb{G},\ \mathbb{G}_{\mathbb{T}})$, then flips an unbiased cion to obtain a bit $\mu \in \{0, 1\}$. If $\mu = 0$, then $\mathcal{C}$ sends $(\overrightarrow{y},\ e(g,\ g)^{abc})$ to $\mathcal{B}$; If $\mu = 1$, then $\mathcal{C}$ sends $(\overrightarrow{y},\ R)$ to $\mathcal{B}$, where $R \in_R \mathbb{G}_{\mathbb{T}}$. □

Global setup: $\mathcal{A}$ submits an access vector $(\overrightarrow{v}, \overrightarrow{v})$ corresponding to W_0 and a series of corrupted authorities AA_k^* to $\mathcal{B}$. $\mathcal{B}$ selects randomly μ_1, μ_2, θ_1, θ_2, λ, f_1, f_2, $\Delta \in_R Z_p$ and sets $V_1 = g^{\mu_1}$, $V_2 = g^{\mu_2}$, $X_1 = g^{\theta_1}$, $X_2 = g^{\theta_2}$.

Authority setup: Let I_C be universe authority. There should be three kinds of authority, the corrupted authorities AA_k^*, the honest ones AA_k^{**}, and at least one half-honest authority AA_δ that can only get partial secret key.

1) For corrupted authorities AA_k^*, $\mathcal{B}$ selects randomly α_k, γ_k, β_k, ζ_k, ς_k, η_k, $\{u_{1,i,k}$, $w_{1,i,k}$, $u_{2,i,k}$, $w_{2,i,k}\}\,_{i=1\ k\in AA_k^*}^{n}$ as secret keys under the condition:$\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$, then calculates $g_2 = g$, $g_1 = g^{\Delta}$, $Y_k = e(g,\ g)^{\alpha_k}$, for i=1 to n computes:

$$\begin{aligned} U_{1,i,k} &= g^{u_{1,i,k}}, U_{2,i,k} = g^{u_{2,i,k}} \\ W_{1,i,k} &= g^{w_{1,i,k}}, W_{2,i,k} = g^{w_{2,i,k}} \\ T_k &= g^{\gamma_k}, Z_k = g^{\beta_k}, M_k = g^{\zeta_k}, N_k = g^{\varsigma_k} \end{aligned}$$

as the attribute public keys. $\mathcal{B}$ sends authorities AA_k^*'s secret keys $SK_k = (\alpha_k$, γ_k, β_k, ζ_k, ς_k, η_k, $\{u_{1,i,k}$, $u_{2,i,k}$, $w_{1,i,k}$, $w_{2,i,k}\}\,_{i=1}^{n})_{k\in AA_k^*}$ and public keys $PK_k = (g_1$, Y_k, T_k, Z_k, M_k, N_k, $\{U_{1,i,k}$, $U_{2,i,k}$, $W_{1,i,k}$, $W_{2,i,k}\}_{i=1}^{n}\}_{k\in AA_k^*}$ to $\mathcal{A}$.

2) For the honest authorities AA_k^{**}, $\mathcal{B}$ selects randomly α_k, γ_k, β_k, ζ_k, ς_k, η_k,$\{u_{1,i,k}$, $w_{1,i,k}$, $u_{2,i,k}$, $w_{2,i,k}\ \}_{i=1 k\in AA_k^{**}}^{n}$ as secret keys under the condition: $\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$, then lets $g_2 = g^b$, and calculates $g_1 = g^{\Delta}$, $Y_k = e(g,g)^{b\alpha_k}$, $T_k = g^{\gamma_k}$, $Z_k = g^{\beta_k}$, $M_k = g^{\zeta_k}$, $N_k = g^{\varsigma_k}$ as public keys. $\mathcal{B}$ calculates the attribute public keys for $j = 1, 2$ as follows:

$$U_{j,i,k} = \begin{cases} g^{u_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) \\ g^{bu_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) \end{cases} \qquad (19)$$

$$W_{j,i,k} = \begin{cases} g^{w_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) \\ g^{bw_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) \end{cases}$$

$\mathcal{B}$ sends honest authority AA_k^{**}'s public keys $PK_k = (g_1$, $\{Y_k$, T_k, Z_k, M_k, $N_k\}$, $\{U_{1,i,k}$, $U_{2,i,k}$, $W_{1,i,k}$, $W_{2,i,k}\}_{i=1}^{n})_{k\in AA_k^{**}}$ to $\mathcal{A}$.

3) For the half-honest authority AA_δ, it is same as the second case except that $\mathcal{B}$ calculates g_1=g^{Δ}, Y_k=$e(g,\ g)^{ab} \cdot \prod_{k\in AA_k^*} e(g{,}g)^{-\alpha_k} \cdot \prod_{k\in AA_k^{**}} e(g,\ g)^{-b\alpha_k}$.

Phase 1: $\mathcal{A}$ submits the attributes list S and GID for secret keys queries. $\mathcal{B}$ chooses random $u^{'} \in Z_p$ for H(GID). $\mathcal{A}$ can query polynomially. Consider a query with two vectors $\overrightarrow{x_{V^k}} = (x_{V_1^k}, x_{V_2^k}, ..., x_{V_n^k})$ and $\overrightarrow{x_{Z^k}} = (x_{Z_1^k}, x_{Z_2^k}, ..., x_{Z_n^k})$, which is related to attributes in $S^k = U_k \bigcap S$. $\mathcal{A}$ can query the secret keys as long as $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = (\overrightarrow{v}, \overrightarrow{x_{Z^k}}) \neq 0$.

1) For corrupted authorities AA_k^*: $\mathcal{B}$ computes secret keys $SK_{L_k^*}$ for attributes in $S^{k^*} = U_k^* \bigcap S$ to $u^{'}$, where U_k^* is AA_k^*'s attributes set.

2) For the honest authorities AA_k^{**}: $\mathcal{B}$ picks random exponents $\{r_{1,i,k},\ r_{2,i,k}\}\,_{i=1,k\in AA_k^{**}}^{n} \in_R Z_p$, then $\mathcal{B}$ computes

$$\begin{aligned} K_{1,i,k} =& g^{-\mu_2 r_{1,i,k}} \cdot U_{2,i,k}^{\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k} \qquad (20) \\ K_{2,i,k} =& g^{\mu_1 r_{1,i,k}} \cdot U_{1,i,k}^{-(\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k)} \\ K_{3,i,k} =& g^{-\theta_2 r_{2,i,k}} \cdot W_{2,i,k}^{\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k} \\ K_{4,i,k} =& g^{\theta_1 r_{2,i,k}} \cdot W_{1,i,k}^{-(\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k)} \end{aligned}$$

Then K_{A_k}, K_{B_k} is calculated as:

$$\begin{aligned} K_{B_k} =& \prod_{i=1}^{n} g^{-(r_{1,i,k}+r_{2,i,k})\triangle} \qquad (21) \\ K_{A_k} =& B^{\alpha_k} \cdot \prod_{i=1}^{n} K_{1,i,k}{}^{-\gamma_k} \cdot K_{2,i,k}{}^{-\beta_k} \\ & \cdot K_{3,i,k}{}^{-\zeta_k} \cdot K_{4,i,k}{}^{-\varsigma_k} \end{aligned}$$

3) For the half-honest authority AA_δ: $\mathcal{B}$ selects random $\{r_{1,i,\delta}, r_{2,i,\delta}\}_{i=1}^{n} \in_R Z_p$, then $\mathcal{B}$ computes K_{A_δ} as follows:

$$\begin{aligned} K_{A_\delta} =& B^{-\lambda} \prod_{i=1}^{n} (K_{1,i,\delta}{}^{-\gamma_k} \cdot K_{2,i,\delta}{}^{-\beta_k} \cdot K_{3,i,\delta}{}^{-\zeta_k} \\ & \cdot K_{4,i,\delta}{}^{-\varsigma_k}) \cdot \prod_{k\in A_k^*} g^{-\alpha_k} \cdot \prod_{k\notin A_k^*} B^{-\alpha_k} \end{aligned}$$

We claim that K_{A_δ} is a valid secret key as follows:

$$\begin{aligned} K_{A_\delta} =& B^{-\lambda} \prod_{i=1}^{n} (K_{1,i,\delta}{}^{-\gamma_k} K_{2,i,\delta}{}^{-\beta_k} K_{3,i,\delta}{}^{-\zeta_k} \\ & K_{4,i,\delta}{}^{-\varsigma_k}) \cdot \prod_{k\in A_k^*} g^{-\alpha_k} \cdot \prod_{k\notin A_k^*} B^{-\alpha_k} \\ =& g^{ab-(\sum_{k\in A_k^*} \alpha_k + \sum_{k\notin A_k^*} b\alpha_k)} \prod_{i=1}^{n} [\\ & g^{\mu_2 (r_{1,i,\delta}-b)\gamma_k} \cdot g^{-\mu_1 (r_{1,i,\delta}-b)\beta_k} \\ & \cdot g^{\theta_2 (r_{2,i,\delta}-b)\zeta_k} \cdot g^{-\theta_1 (r_{2,i,\delta}-b)\varsigma_k} \\ & \cdot U_{2,i,\delta}^{-(\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k)\gamma_k} \cdot U_{1,i,\delta}^{(\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k)\beta_k} \\ & \cdot W_{2,i,\delta}^{-(\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k)\zeta_k} \cdot W_{1,i,\delta}^{(\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k)\varsigma_k}] \\ =& g^{ab-(\sum_{k\in A_k^*} \alpha_k + \sum_{k\notin A_k^*} b\alpha_k)} \cdot \prod_{i=1}^{n} (K_{1,i,\delta}^{'}{}^{-\gamma_k} \\ & \cdot K_{2,i,\delta}^{'}{}^{-\beta_k} \cdot K_{3,i,\delta}^{'}{}^{-\zeta_k} \cdot K_{4,i,\delta}^{'}{}^{-\varsigma_k}) \end{aligned}$$

Where lets $r_{1,i,\delta}^{'} = r_{1,i,\delta} - b$, $r_{2,i,\delta}^{'} = r_{2,i,\delta} - b$, and implicitly sets: $\mu_2\gamma_k - \mu_1\beta_k + \theta_2\zeta_k - \theta_1\varsigma_k = a + \lambda$. Note that $K_{1,i,\delta}$,$K_{2,i,\delta}$,$K_{3,i,\delta}$,$K_{4,i,\delta}$ and K_{B_δ} is same as equations (21) and (22).

$\mathcal{B}$ gives $\mathcal{A}$ the secret keys $SK_{GID,k,i} = (K_{A_k}$, K_{B_k}, $\{K_{1,i,k}$, $K_{2,i,k}$, $K_{3,i,k}$, $K_{4,i,k}\}_{i=1}^{n})_{k\in AA_k^*, k\in AA_k^{**}, k\in AA_\delta}$ for the queried attributes set S.

Challenge: $\mathcal{A}$ submits two equal length messages M_0 and M_1 to $\mathcal{B}$. $\mathcal{B}$ selects a random bit $\xi \in_R \{0,1\}$ and runs *Encryption*$(PK_k,\ M_\xi)$. $\mathcal{B}$ selects randomly $s_1^{'}, \beta^{'} \in Z_p$, and implicitly sets $s_1 = s_1^{'}, s_2 = c, \beta = \beta^{'}$. For i from 1 to n, $\mathcal{B}$ computes as:

$$C_0 = M_\xi \cdot Z;\ C_A = g^c;\ C_B = (g^{\triangle})^{s_1^{'}}; \tag{22}$$

$$C_{1,i,k} = U_{1,i,k}^{s_1^{'}}(g^{\gamma_k})^c g^{\mu_1 v_i \beta^{'}}, C_{2,i,k} = U_{2,i,k}^{s_1^{'}}(g^{\beta_k})^c g^{\mu_2 v_i \beta^{'}}$$

$$C_{3,i,k} = W_{1,i,k}^{s_1^{'}}(g^{\zeta_k})^c g^{\theta_1 v_i \beta^{'}}, C_{4,i,k} = W_{2,i,k}^{s_1^{'}}(g^{\varsigma_k})^c g^{\theta_2 v_i \beta^{'}}$$

$\mathcal{B}$ sends the ciphertext $CT = (\ C_0,\ \mathrm{C}_A,\ C_B,\ \{C_{1,i,k},\ C_{2,i,k},\ C_{3,i,k},\ C_{4,i,k}\}_{i=1}^{n}{}_{k=1}^{\mathcal{K}})$ to $\mathcal{A}$.

If $\mu = 0$, then $Z = e(g,g)^{abc}$, we can show CT is a valid ciphertext of message M_ξ by computing

$$\prod_{k \in I_C} {Y_k}^c = \prod_{k \in A_k^*} e(g,g)^{\alpha_k} \prod_{k \in A_k^{**}} e(g,g)^{b\alpha_k}[e(g,g)^{abc} \cdot \prod_{k \in A_k^*} e(g,g)^{-\alpha_k} \cdot \prod_{k \in A_k^{**}} e(g,g)^{-b\alpha_k}] = Z$$

Phase 2: Phase 1 is repeated.

Guess: Finally $\mathcal{A}$ returns his guess $\xi^{'}$ on ξ. If $\xi^{'} = \xi$, $\mathcal{B}$ returns his guess $\mu^{'} = 0$ on μ, otherwise, $\mathcal{B}$ returns his guess $\mu^{'} = 1$ on μ.

$\mathcal{A}$ can get nothing about his guess on ξ when $\mu = 1$ since the input of Z is a random number z. Therefore, $\mathcal{A}$ cannot distinguish ξ with non-negligible advantage, so $\Pr[\xi^{'} \neq \xi | \mu = 1] = \frac{1}{2}$, $\mathcal{B}$ returns his guess $\mu^{'} = 1$ when $\xi^{'} \neq \xi$, thus we have $\Pr[\mu^{'} = \mu | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$, according to the definition of DBDH complexity assumption, the advantage of adversary $\mathcal{A}$ in outputing $\xi^{'} = \xi$ is at least ϵ. Therefore, we have $\Pr[\xi^{'} = \xi | \mu = 0] \geq \frac{1}{2} + \epsilon$. When $\xi^{'} = \xi$, $\mathcal{B}$ returns $\mu^{'} = 0$ on μ, so we have $\Pr[\mu^{'} = \mu | \mu = 0] \geq \frac{1}{2} + \epsilon$.

In conclusion, $\mathcal{B}$'s advantage to break the DBDH assumption is $|\frac{1}{2}\Pr[\mu^{'} = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu^{'} = \mu | \mu = 1] - \frac{1}{2}| \geq \frac{\epsilon}{2}$. Hence, if adversary $\mathcal{A}$ can distinguish these two games, $\mathcal{B}$ can solve the DBDH problem.

6.2 Indistinguishability Between Game$_1$ and Game$_2$

Lemma 2. *For any adversary $\mathcal{A}$, Game$_1$ and Game$_2$ could be distinguished with a non-negligible advantage, then there exists algorithm $\mathcal{B}$ that could solve the DLIN assumption with a non-negligible advantage, i.e.*

$$|\ Adv_{Game_1}(\lambda) - Adv_{Game_2}(\lambda)\ | \leq Adv_{\mathcal{B}}^{DLIN}(\lambda) \tag{23}$$

Proof. Let $\overrightarrow{y} = \{g, Z_1 = g^{z_1}, Z_2 = g^{z_2}, Z_3 = g^{z_2 z_4}, Z_4 = g^{z_3 + z_4}\}$. The challenger $\mathcal{C}$ generates the bilinear group $(e,\ p,\ g,\ \mathbb{G},\ \mathbb{G}_{\mathbb{T}})$, and flips an unbiased cion with $\{0,1\}$ to obtains a bit μ. If $\mu = 0$, $\mathcal{C}$ sends $(\overrightarrow{y},\ g^{z_1 z_3})$ to $\mathcal{B}$; If $\mu = 1$, then $\mathcal{C}$ sends $(\overrightarrow{y},\ R)$ to $\mathcal{B}$, where $R \in_R \mathbb{G}_{\mathbb{T}}$. □

Global setup: $\mathcal{A}$ submits two access vectors $(\overrightarrow{v}, \overrightarrow{v})$ and $(\overrightarrow{v}, \overrightarrow{0})$ which is corresponding to W_0 and W_1, and sends a list of corrupted authorities AA_k^* to $\mathcal{B}$. Then $\mathcal{B}$ selects randomly $\mu_1,\ \mu_2,\ \theta_1,\ \theta_2,\ f_1,\ f_2,\ \Delta \in_R Z_p$, and sets $V_1 = g^{\mu_1}$, $V_2 = g^{\mu_2}$, $X_1 = g^{\theta_1}$, $X_2 = g^{\theta_2}$.

Authorities setup: Let I_C be universe authority. There should be three kinds of authorities, the corrupted authorities AA_k^*, the honest ones AA_k^{**} and at least one half-honest authority AA_δ for which $\mathcal{A}$ can only get parts of the secret keys.

1) For corrupted authorities AA_k^*: $\mathcal{B}$ is same as the algorithm of corruption authorities during *Authorities Setup* in the proof of *lemma 1*.

2) For the honest authorities AA_k^{**}: $\mathcal{B}$ selects randomly $\alpha_k, \gamma_k, \beta_k, \zeta_k, \varsigma_k, \eta_k, \{u_{1,i,k}, w_{1,i,k}, u_{2,i,k}, w_{2,i,k}\}_{i=1}^{n}{}_{k \in AA_k^{**}}$ as secret keys under the condition: $\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$, then lets $g_2 = g^{z_2}$, and calculates $g_1 = g^{\Delta}$, $Y_k = e(g,g)^{z_2 \alpha_k}$, $T_k = g^{\gamma_k}$, $Z_k = g^{\beta_k}$, $M_k = g^{\zeta_k}$, $N_k = g^{\varsigma_k}$ as public keys. $\mathcal{B}$ calculates the attribute public keys for $j = 1$ to 2 as follows:

$$U_{j,i,k} = \begin{cases} g^{u_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \\ g^{z_1 u_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \end{cases}$$

$$W_{j,i,k} = \begin{cases} g^{w_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \\ g^{z_1 w_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \end{cases}$$

$\mathcal{B}$ sends honest authority AA_k^{**}'s public keys $PK_k = (g_1,\ \{Y_k,\ T_k,\ Z_k,\ M_k,\ N_k\},\ \{U_{1,i,k},\ U_{2,i,k},\ W_{1,i,k},\ W_{2,i,k}\}_{i=1}^{n})_{k \in AA_k^{**}}$ to $\mathcal{A}$.

3) For the half-honest authority AA_δ, it is same as the second case except that $\mathcal{B}$ calculates $g_1 = g^{\Delta}$, $Y_k = e(g,g)^{z_2} \cdot \prod_{k \in AA_k^*} e(g,g)^{-\alpha_k} \cdot \prod_{k \in AA_k^{**}} e(g,g)^{-z_2 \alpha_k}$.

Phase 1: $\mathcal{A}$ submits the attributes list S and GID for secret keys queries. $\mathcal{B}$ chooses random $u^{'} \in Z_p$ for H(GID). $\mathcal{A}$ can query polynomially. Consider a query with two vectors $\overrightarrow{x_{V^k}} = (x_{V_1^k}, x_{V_2^k}, ..., x_{V_n^k})$ and $\overrightarrow{x_{Z^k}} = (x_{Z_1^k}, x_{Z_2^k}, ..., x_{Z_n^k})$, which is related to attributes in $L_k = U_k \bigcap L$. $\mathcal{A}$ can query the secret keys as long as $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = (\overrightarrow{v}, \overrightarrow{x_{Z^k}}) \neq 0$.

1) For corrupted authorities AA_k^*: $\mathcal{B}$ computes secret keys $SK_{L_k^*}$ for attributes in $S^{k^*} = U_k^* \bigcap S$ to $u^{'}$, where U_k^* is the attribute set of the authorities AA_k^*.

2) For the honest authorities AA_k^{**}: $\mathcal{B}$ is same as the algorithm of honest authorities during *Phase 1* in the proof of *lemma 1*.

3) For the half-honest authority AA_δ: $\mathcal{B}$ selects random $\{r_{1,i,\delta}, r_{2,i,\delta}\}_{i=1}^{n} \in_R Z_p$, then $\mathcal{B}$ computes K_{A_δ} as fol-

Table 3: The comparison of storage costs at different phases

Scheme	**Authority Setup**	**Encryption**	**KeyGen**	**TransKeyGen**
[7]	$I(\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_{\mathbb{T}}\rvert)$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_{\mathbb{T}}\rvert$	$p_i N_S\lvert\mathbb{G}\rvert$	$-$
[30]	$(I+\mathcal{K})\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$3\lvert\mathbb{G}\rvert + 3N_W\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$2N_S\lvert\mathbb{G}\rvert$	$-$
[1]	$(I+\mathcal{K})\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$4\lvert\mathbb{G}\rvert + 3N_W\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$2N_S\lvert\mathbb{G}\rvert$	$2N_S\lvert\mathbb{G}\rvert + 3\lvert\mathbb{G}\rvert$
[17]	$2I\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$(1+N_W)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_{\mathbb{T}}\rvert$	$N_S\lvert\mathbb{G}\rvert$	$-$
[25]	$(I+\mathcal{K})\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$(2N_W+1)\lvert G\rvert + (N_W+1)\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$2N_S\lvert\mathbb{G}\rvert$	$N_S(\lvert\mathbb{G}\rvert + \lvert\mathcal{O}(\mathcal{H})\rvert)$
Ours	$(4+5\mathcal{K})\lvert\mathbb{G}\rvert + \mathcal{K}\lvert\mathbb{G}_{\mathbb{T}}\rvert$	$(2+4\rho N_W)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_{\mathbb{T}}\rvert$	$(2+4\rho N_W)\lvert\mathbb{G}\rvert$	$(2+4\rho N_W)\lvert\mathbb{G}\rvert$

[1] $|\mathbb{G}|$: the size of one element in the group $\mathbb{G}$. $|\mathbb{G}_{\mathbb{T}}|$: the size of one element in the group $\mathbb{G}_{\mathbb{T}}$.
[2] N_W: the number of attributes in the access policy. N_S: The number of attributes in user's attribute set. I: The number of attributes in the system.
[3] $|\mathcal{O}(\mathcal{H})|$: The size of a hash function. $\rho \in (0,1)$: The coefficient associated with the number of wildcards.
[4] $\mathcal{K}$: The number of the attribute authority.

lows:

$$K_{A_\delta} = Z_2^{-\lambda'} \cdot \prod_{i=1}^{n} ({K_{1,i,\delta}}^{-\gamma_k} {K_{2,i,\delta}}^{-\beta_k} {K_{3,i,\delta}}^{-\zeta_k} {K_{4,i,\delta}}^{-\varsigma_k}) \prod_{k\in A_k^*} g^{-\alpha_k} \prod_{k\notin A_k^*} Z_2^{-\alpha_k} \tag{24}$$

We claim that K_{A_δ} is a valid secret key as follows:

$$\begin{aligned}
K_{A_\delta} =& Z_2^{-\lambda'} \prod_{i=1}^{n} ({K_{1,i,\delta}}^{-\gamma_k} {K_{2,i,\delta}}^{-\beta_k} {K_{3,i,\delta}}^{-\zeta_k} \\
& {K_{4,i,\delta}}^{-\varsigma_k}) \cdot \prod_{k\in A_k^*} g^{-\alpha_k} \cdot \prod_{k\notin A_k^*} Z_2^{-\alpha_k} \\
=& g^{z_2 - (\sum_{k\in A_k^*} \alpha_k + \sum_{k\notin A_k^*} z_2\alpha_k)} \\
& \cdot \prod_{i=1}^{n} g^{\mu_2(r_{1,i,\delta}-z_2)\gamma_k} \cdot g^{-\mu_1(r_{1,i,\delta}-z_2)\beta_k} \\
& \cdot g^{\theta_2(r_{2,i,\delta}-z_2)\zeta_k} \cdot g^{-\theta_1(r_{2,i,\delta}-z_2)\varsigma_k} \\
& \cdot {U_{2,i,\delta}}^{-(\frac{x_{V_i^k}}{u'+f_1}+\eta_k)\gamma_k} \cdot {U_{1,i,\delta}}^{(\frac{x_{V_i^k}}{u'+f_1}+\eta_k)\beta_k} \\
& \cdot {W_{2,i,\delta}}^{-(\frac{x_{V_i^k}}{u'+f_2}-\eta_k)\zeta_k} \cdot {W_{1,i,\delta}}^{(\frac{x_{V_i^k}}{u'+f_2}-\eta_k)\varsigma_k}] \\
=& g^{z_2 - (\sum_{k\in A_k^*} \alpha_k + \sum_{k\notin A_k^*} z_2\alpha_k)} \cdot \prod_{i=1}^{n} ({K'_{1,i,\delta}}^{-\gamma_k} \\
& \cdot {K'_{2,i,\delta}}^{-\beta_k} \cdot {K'_{3,i,\delta}}^{-\zeta_k} \cdot {K'_{4,i,\delta}}^{-\varsigma_k})
\end{aligned}$$

Where lets $r'_{1,i,\delta} = r_{1,i,\delta} - z_2$, $r'_{2,i,\delta} = r_{2,i,\delta} - z_2$, $\lambda' = \lambda - 1$, and implicitly sets: $\mu_2\gamma_k - \mu_1\beta_k + \theta_2\zeta_k - \theta_1\varsigma_k = \lambda$. Note that $K_{1,i,\delta}$,$K_{2,i,\delta}$,$K_{3,i,\delta}$,$K_{4,i,\delta}$ and K_{B_δ} is same as *lemma 1*.

Finally, $\mathcal{B}$ gives $\mathcal{A}$ the secret keys $SK_{GID,k,i} = (K_{A_k}, K_{B_k}, \{K_{1,i,k}, K_{2,i,k}, K_{3,i,k}, K_{4,i,k}\}_{i=1\ k\in A, A_k^*, k\in AA_k^{**}, k\in AA_\delta}^{n})$ for the queried attribute set L.

Challenge: $\mathcal{A}$ submits two equal length messages M_0 and M_1. $\mathcal{B}$ selects a random bit $\xi \in_R \{0,1\}$ and runs *Encryption*$(PK_k,\ W_\xi,\ M_\xi)$. $\mathcal{B}$ selects random s_1', β' $\in Z_p$, and implicitly sets $s_1 = s_1'$, $s_2 = z_3 + z_4$, $\beta = \beta'$.

For i from 1 to n, $\mathcal{B}$ computes as:

$$\begin{aligned}
C_{1,i,k} =& g^{u_{1,i,k} z_1 s_1'} (g^{\gamma_k})^{z_3+z_4} g^{\mu_1 v_i \beta'} = U_{1,i,k}^{s_1} T_k^{s_2} V_1^{v_i\beta'} \\
C_{2,i,k} =& g^{u_{2,i,k} z_1 s_1'} (g^{\beta_k})^{z_3+z_4} g^{\mu_2 v_i \alpha'} = U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{v_i\beta'}
\end{aligned}$$

It implies $s_1' = z_3$, if $\mu = 0$, then $Z = g^{z_1 z_3}$, then $\mathcal{B}$ is simulating $Game_1$:

$$\begin{aligned}
C_{3,i,k} =& g^{w_{1,i,k} z_1 s_1'} (g^{\zeta_k})^{z_3+z_4} g^{\theta_1 v_i \beta'} = W_{1,i,k}^{s_1} M_k^{s_2} X_1^{v_i\beta'} \\
C_{4,i,k} =& g^{w_{2,i,k} z_1 s_1'} (g^{\varsigma_k})^{z_3+z_4} g^{\theta_2 v_i \beta'} = W_{2,i,k}^{s_1} N_k^{s_2} X_2^{v_i\beta'}
\end{aligned}$$

It implies $s_1' = z_3 \cdot z (z \in_R \mathbb{G})$, if $\mu = 1$, then $Z = g^{z_1 z_3 \cdot z} = R$, then $\mathcal{B}$ is simulating $Game_2$:

$$\begin{aligned}
C_{3,i,k} &= g^{w_{1,i,k} z_1 s_1'} (g^{\zeta_k})^{z_3+z_4} g^{\theta_1 v_i \beta'} = W_{1,i,k}^{s_1} M_k^{s_2} \\
C_{4,i,k} &= g^{w_{2,i,k} z_1 s_1'} (g^{\varsigma_k})^{z_3+z_4} g^{\theta_2 v_i \beta'} = W_{2,i,k}^{s_1} N_k^{s_2}
\end{aligned}$$

Finally, $\mathcal{B}$ calculates:

$$\begin{aligned}
C_0 =& M_\xi \cdot \prod_{k\in I_C} Y_k^{s_2} \\
=& M_\xi \cdot e(g,g)^{(z_3+z_4)z_2} \cdot \prod_{k\in A_k^*} e(g,g)^{(z_3+z_4)\alpha_k} \cdot \\
& \prod_{k\in A_k^{**}} e(g,g)^{(z_3+z_4)z_2\alpha_k} \prod_{k\in A_k^*} e(g,g)^{-(z_3+z_4)\alpha_k} \\
& \cdot \prod_{k\in A_k^{**}} e(g,g)^{-(z_3+z_4)z_2\alpha_k} \\
=& M_\xi \cdot e(g,g)^{(z_3+z_4)z_2} \\
C_A =& g^{z_3+z_4}, C_B = (g^{\triangle})^{z_3}
\end{aligned} \tag{25}$$

$\mathcal{B}$ sends the ciphertext $CT = (C_0,\ C_A,\ C_B,\ \{C_{1,i,k},\ C_{2,i,k},\ C_{3,i,k},\ C_{4,i,k}\ \}_{\ i\in[1,n], k\in AA_k^{**}, k\in AA_\delta})$ to $\mathcal{A}$.

Phase 2: Phase 1 is repeated.

Guess: Finally $\mathcal{A}$ returns his guess ξ' on ξ. If $\xi' = \xi$, $\mathcal{B}$ returns his guess $\mu' = 0$ on μ, otherwise, $\mathcal{B}$ returns his guess $\mu' = 1$ on μ.

Similarly, $\mathcal{B}$'s advantage to break the DLIN assumption is $|\frac{1}{2}\Pr[\mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu' = \mu | \mu = 1] - \frac{1}{2}| \geq \frac{\epsilon}{2}$. Hence, if $\mathcal{A}$ can distinguish these two games, $\mathcal{B}$ can solve the DLIN assumption.

Table 4: The comparison of computation costs at different phases

Scheme	Encryption	User.Decryption	Out.Decryption
[7]	$3E_{\mathbb{G}}$	$2\hat{e}$	$-$
[30]	$2\hat{e} + (1+2N_W)E_{\mathbb{G}_{\mathbb{T}}} + 3N_W E_{\mathbb{G}}$	$(1+2N_S)\hat{e} + N_S E_{\mathbb{G}_{\mathbb{T}}}$	$-$
[1]	$5E_{\mathbb{G}} + E_{\mathbb{G}_{\mathbb{T}}} + 3\mathcal{O}(\mathcal{H})$	$E_{\mathbb{G}_{\mathbb{T}}} + 3\mathcal{O}(\mathcal{H})$	$3N_S\hat{e} + E_{\mathbb{G}_{\mathbb{T}}}$
[17]	$\hat{e} + (1+N_W)E_{\mathbb{G}} + E_{\mathbb{G}_{\mathbb{T}}}$	$2\hat{e} + N_S E_{\mathbb{G}}$	$-$
[25]	$E_{\mathbb{G}_{\mathbb{T}}} + E_{\mathbb{G}}$	$E_{\mathbb{G}_{\mathbb{T}}}$	$2N_S\hat{e} + N_S(E_{\mathbb{G}} + E_{\mathbb{G}_{\mathbb{T}}})$
Ours	$\hat{e} + 4\rho N_W E_{\mathbb{G}} + E_{\mathbb{G}_{\mathbb{T}}}$	$E_{\mathbb{G}_{\mathbb{T}}}$	$(2+4\rho N_S)\hat{e}$

[1] $E_{\mathbb{G}}$: The time of an exponential operation in the group $\mathbb{G}$. $E_{\mathbb{G}_{\mathbb{T}}}$: The time of an exponential operation in the group $\mathbb{G}_{\mathbb{T}}$.

[2] $\hat{e}$: The time of computing a pairing function e. $\mathcal{O}(\mathcal{H})$: The time of computing a hash function.

6.3 Indistinguishability Between Game$_2$ and Game$_3$

Lemma 3. *For any adversary $\mathcal{A}$, $Game_2$ and $Game_3$ could be distinguished with a non-negligible advantage, then there exists algorithm $\mathcal{B}$ that could solve the DLIN assumption with a non-negligible advantage,* i.e.

$$| \, Adv_{Game_2}(\lambda) - Adv_{Game_3}(\lambda) \, | \leq Adv_{\mathcal{B}}^{DLIN}(\lambda)$$

Proof. Except for *phase 1*, the rest is the same as the above proof in Lemma 2. There are two cases in *phase 1*:

- $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = (\overrightarrow{v}, \overrightarrow{x_{Z^k}}) = 0 \pmod{p}$.
- $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = c_v \neq 0$, $(\overrightarrow{v}, \overrightarrow{x_{Z^k}}) = c_x \neq 0$.

□

Similarly, we can prove the indistinguishability between $Game_3$ and $Game_4$ in the similar way as for that of $Game_2$ and $Game_3$. The proof of indistinguishability between $Game_4$ and $Game_5$ is similar to that of $Game_1$ and $Game_2$. The proof of indistinguishability between $Game_5$ and $Game_6$ is similar to that of $Game_0$ and $Game_1$.

7 Performance Analysis

In this section, we evaluate the storage costs and computation costs of our scheme. For this purpose, we introduce the size of the system's public keys, the ciphertext, the user's secret keys and the transform-keys. In addition, we consider the computation costs related to execution of Encryption, User.Decryption and Out.Decryption Algorithm, which those algorithms are performed by DO, DU and CP, respectively.

To compare the performance of those schemes more intuitively, we give here an empirical comparison of storage costs and computation costs in ours, and the results with the latest the work of Belguith *et al.* [1], the work of Michdevsky *et al.* [17] and the work of Shao *et al.* [25]. We conduct our experiments on a Windows machine with 3.40 GHz Intel(R) Core(TM) i3-3240 CPU and 4 GB RAM. The code uses Pairing Based Cryptography library to achieve the access control scheme, which supports pairing operation. Type A pairings are used in the simulation, which are constructed on the curve over the field for some prime q. The pairing is symmetric, where the order of groups is 160 bits, the base field size is 512 bits. All that the length of an element in each group G and the target group G_T is set to 512 bits.

7.1 Storage Costs

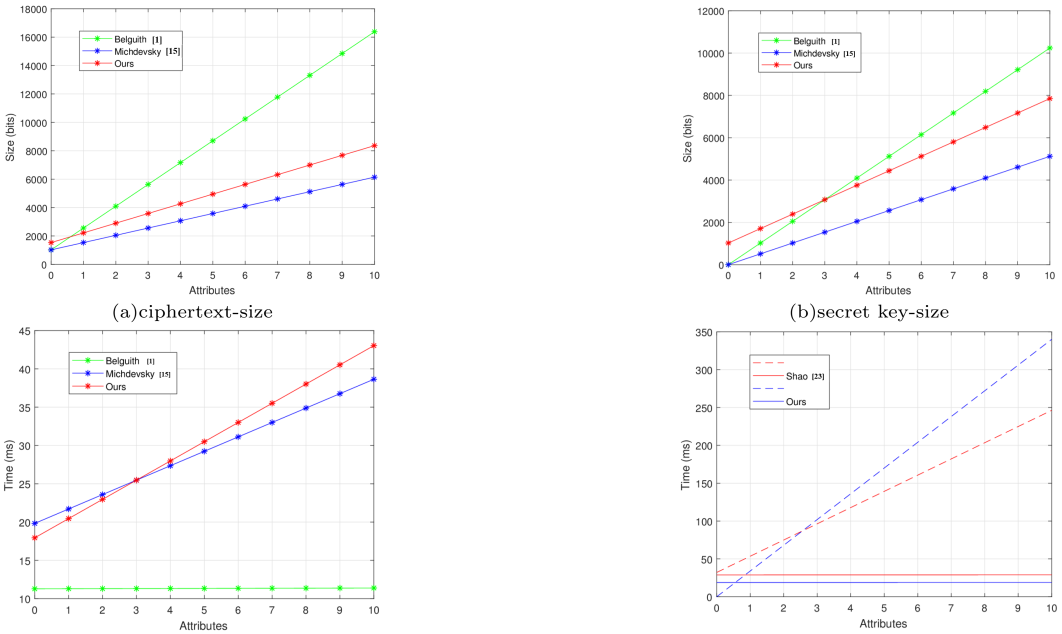
Based on Table 1, we find that our scheme achieve the optimal compromise between policy-hiding fully and the efficiency on the user side. Compared with [1, 21, 25] and [17], our scheme is more flexible in multi-authority environments. We make a comparison between latest closely MA-ABE schemes and our scheme with regard to the size of public keys, ciphertext, secret keys and transform keys in Table 3. From Table 3, we can know that the size of the ciphertext significantly is shorter than that in [1, 30] and [25] when $\rho < 0.8$, and the size of the secret key is shorter than that in [1, 7, 30] and [25] when $\rho < 0.5$, which depends on the number of the wildcards instead that of the attributes.

In addition, the results in Figure 5-(a) and Fifure 5-(b) reveal the storage costs of the ciphertext and the secret keys, which the size of an encrypted file and secret keys grows linearly with the number of attributes involved in the access policy and the user's attribute set in ours.

7.2 Computation Costs

We make a theoretical analysis about the time of encryption and decryption in Table 4. The performance is analyzed under three aspects which are the computational costs in terms of multiplication, exponentiation and pairing in $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$. Compared with [7, 30] and [17] in Table 4, the proposed scheme takes less time in user.decryption phase than others, because most bilinear pairing calculation is transferred to CP in ours. The existence of three hash functions $\mathcal{O}(\mathcal{H})$ in [1] results in our scheme being more efficient in user.decryption phase.

In addition, the computation costs of the encryption operation and user.decryption operation are presented in Figure 5-(c) and Figure 5-(d). From Figure 5-(c), it can be known that our scheme has some performance disadvantages, such as encryption time. However, this is a compromise to achieve fully policy-hiding. The time of user's decryption in our scheme is greatly short than others due to the outsource operation. Our scheme reduces the overhead on the user side online computation and has a clear



(a)ciphertext-size

(b)secret key-size

(c)encryption-time

(d)user.decryption-time

Figure 5: Evaluation of algorithms

advantage over ABE scheme without outsourcing. However, it can be found that comparing between a scheme that outsources most of its heavy computational operations and other that doesn't is not fair enough. So the comparison of the decryption cost is performed between our scheme and [25] in Figure 5-(d).

8 Conclusion

In this paper, a scheme with fully hiding access policy is studied in the cloud storage system. The policy is hidden by converting the access policy and attribute set into vectors, and the scheme is constructed based on IPE technique. In addition, this decentralized MA-ABE scheme with strong resistance to attacks mentioned in [23] and [29] from potential malicious users. Moreover, the decryption is partially outsourced to the third party proxy services which results in a more efficient decentralizing MA-ABE. Then, the security of the presented scheme is reduced to the DBDH assumption and the DLIN assumption instead of others strong assumptions. We also confirm the scalability and flexibility of the proposed scheme by numerical experiments. However, our scheme only achieves selectively security. It is left as the future work to construct the decentralizing MA-ABE with adaptive security and full hiding policy.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant (61872087), the National Cryptography Development Fund under Grant (MMJJ20180209), the National Key Research and Development Program of China under Grants No. 2017YFB0802002, International S&T Cooperation Program of Shaanxi Province No. 2019KW-056, the Plan For Scientific Innovation Talent of Henan Province under Grant 184100510012, and the Program for Science and Technology Innovation Talents in the Universities of Henan Province under Grant 18HASTIT022.

References

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

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, pp. 321–334. IEEE, 2007.

[3] M. Chase, "Multi-authority attribute based encryption," *Proceedings of Theory Cryptography Conference (TCC'07)*, vol. 4392, pp. 515–534, 2007.

[4] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," *The 16th ACM Conference on Computer and Communications Security*, vol. 14, pp. 121–130, 2009.

[5] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of 14th ACM Conference Security in Computing and Communications*, pp. 456–465, 2007.

[6] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments", *International Journal of Network Security*, vol. 16, no. 1, pp. 1-13, 2014.

[7] Y. Fan, X. Wu, and J. Wang, "Multi-authority attribute-based encryption access control scheme with hidden policy and constant length ciphertext for cloud storage," in *IEEE Second International Conference on Data Science in Cyberspace*, pp. 205–212, 2017.

[8] A. Ge, J. Zhang, R. Zhang, C. Ma, and Z. Zhang, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2319–2321, 2013.

[9] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proceedings of the 20th USENIX Conference on Security*, pp. 523–538, 2011.

[10] J. Han, W. Susilo, Y. Mu, and J. Yan, "Security analysis of a privacy-preserving decentralized key- policy attribute-based encryption scheme," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[11] S. Honhenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography (PKC'14)*, pp. 293–310, 2014.

[12] C. Jin, X. Feng, and Q. Shen, "Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size," in *ACM International Conference on Advanced Computing, Networking and Security*, vol. 11, no. 6, pp. 91–98, 2016.

[13] J. Lai, R. H. Deng, C. Guan, and J. Wang, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 1343–1354, 2013.

[14] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *ACM Symposium on Information Computer and Communications Security*, pp. 18–19, 2012.

[15] A. Lewko, and B. Waters, "Decentralizing attribute-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 6632, pp. 568–588, 2011.

[16] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[17] Y. Michalevsky and M. Joye, "Decentralized policy-hiding abe with receiver privacy," *European Symposium on Research in Computer Security Springer*, vol. 11099, pp. 548–567, 2018.

[18] S. Muller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *International Conference on Information Security and Cryptology*, vol. 5461, pp. 20–36, 2009.

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

[20] T. Okamoto and K. Takashima, "Adaptively attribute-hiding (hierarchical) inner product encryption," *Advances in Cryptology*, vol. 7237, pp. 591–608, 2012.

[21] T. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute based encryption under standard assumptions," *IEEE Transactions on Information Forensics Security*, vol. 11, pp. 35–45, 2016.

[22] H. Qian, J. Li, and Y. Zhang, "Privacy-preserving decentralized ciphertext-policy attribute-based encryption with fully hidden access structure," in *International Conference on Information and Communications Security*, vol. 8233, pp. 363–372, 2013.

[23] Y. Rahulamathavan, S. Veluru, J. Han, F. Li, M. Rajarajan, and R. Lu, "User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Computers*, vol. 65, no. 9, pp. 2939–2946, 2016.

[24] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Advances in Cryptology*, vol. 3494, pp. 457–473, 2005.

[25] J. Shao, Y. Zhu, and Q. Ji, "Efficient decentralized attribute-based encryption with outsourced computation for mobile cloud computing," *IEEE International Symposium on Parallel and Distributed Processing with Applications and IEEE International Conference on Ubiquitous Computing and Communications*, vol. 1, pp. 417–422, 2017.

[26] M. Wang, Z. Zhang, and C. Chen, "Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1237–1245, 2016.

[27] R. Xu and B. Lang, "A cp-abe scheme with hidden policy and its application in cloud computing," *International Journal of Cloud Computing*, vol. 4, pp. 279–298, 2015.

[28] Z. Ying, J. Ma, and J. Cui, "Partially policy hidden CP-ABE supporting dynamic policy updating," *Journal on Communications*, vol. 36, pp. 178–189, 2015.

[29] L. Zhang, P. Liang, and Y. Mu, "Improving privacy-preserving and security for decentralized key-policy attributed-based encryption," in *IEEE Access*, pp. 1–1, 2018.

[30] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," in *Soft Computing*, pp. 243–251, 2016.

[31] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 126–138, 2015.

Biography

Leyou Zhang received the M.S. and Ph.D. degrees from Xidian University, in 2002 and 2009, respectively. He is currently a Professor with Xidian University. His current research interests include cryptography, network security, cloud security, and computer security.

Juan Ren received the B.S. degree in mathematics from Nanchang Hangkong University, China, in 2017. He is currently pursuing the Ph.D. degree in applied mathematics with Xidian University, China. His current interests include applied cryptography and cloud security.

Li Kang is a master degree student in the school of mathematics and statistics, Xidian University. Her research interests focus on computer and network security.

Baocang Wang received the B.S. degree in Computational Mathematics and the Application Software, the M.S and the Ph.D. degrees in cryptography from Xidian University in 2001, 2004, and 2006, respectively. He is currently a professor with the School of Telecommunications Engineering, Xidian University. His main research interests include public key cryptography, wireless network security, and cloud computing security.