An Expressive Decentralizing KP-ABE Scheme with Constant-Size Ciphertext

Qinyi Li[1], Hu Xiong[1,2], Fengli Zhang[1], and Shengke Zeng[1]
(Corresponding author: Qinyi Li)

School of Computer Science and Engineering, University of Electronic Science and Technology of China[1]
No.2006, Xiyuan Avenue, West Hi-tech Zone, Chengdu, 611731, P.R.China
State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences[2]
Beijing, 100080, P.R.China
(Email: leebluedreams@gmail.com)
(Received Apr. 14, 2012; revised and accepted June 27, 2012)

Abstract

Decentralizing attribute based encryption is a variant of multi-authority attribute based encryption which doesn't require a trusted central authority to conduct the system setup. In this paper, we propose an expressive decentralizing KP-ABE scheme with constant ciphertext size. In our construction, the access policy can be expressed as any non-monotone access structure. Meanwhile, the ciphertext size is independent on the number of attributes used in the scheme. We prove that our scheme is semantic secure in so-called Selective-Set model based on the n-DBDHE assumption. To the best of our knowledge, this is the first multi-authority attribute based encryption scheme realizing such expressive access policy and constant ciphertext size.
Keywords: Access structure, attribute based encryption, constant ciphertext size, decentralizing

1 Introduction

Traditional public key encryption (PKE) [20] and identity based encryption (IBE) [22] are useful mechanisms to ensure confidential data storage and transformation. However, one drawback of them is that they can only realize coarse-grained access control in encrypted data. Sahai and Waters proposed a new cryptographic primitive called attribute based encryption (ABE) [21] which provides a new viewpoint of encryption for new one-to-many and fine-grained access control application environment. In ABE, decrypt ability depends on user's attribute. Only the one who has the required attribute can get decryption key from a trustworthy authority. Goyal *et al.* [13] divided ABE into two flavorers, ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). In CP-ABE, encipher specifies an access structure in ciphertext which defines over universal attributes set. User's key is associated with some attributes, if user's attributes satisfy the access structure, then it can decrypt the ciphertext. In KP-ABE, situation is inversed. That is user's key is attached with an access structure and ciphertext is associated with attributes. Only if ciphertext's attributes is the authorized set of the access structure, decryption will proceed.

In reality, it is impossible for attributes to be monitored by one authority. For example, a professor's driving license is managed by Vehicle Management Institutions while his professional titles are authorized by a university. Chase [7] addressed this problem affirmatively by firstly proposing a multi-authority ABE (MA-ABE) scheme. In MA-ABE, universal attributes set are divided into several domains and managed by corresponding authorities. A user will issue his attributes to all the authorities to get his decryption key. Decentralizing ABE is a special MA-ABE and it does not require a trusted central authority to conduct the system setup. In general ABE schemes, the size of ciphertext always grows linearly with universal attributes. When enlarging the number of system attribute, the ciphertext size is also growing. In cloud storage system or other bandwidth-limited transmission system, this problem is noticeable. Another problem for current ABE schemes is that their access formula is not expressive enough. In most of them, any monotone access structure is supported while others only realize AND-Gate structure for the reasons of getting provably secure [10] and some special functions [16]. Therefore, it is meaningful to devise ABE scheme which supports more expressive formula. Ostrovsky *et al.* [19] considered ABE scheme which realizes any non-monotone access structure - the most complex policy in ABE field.

In Chase's work [7], except for attribute authorities, there is a central authority that can decrypt all ciphertext because it masters the system secret key and thus key-escrow problem is aroused. Following Chase's work,

Table 1: Comparisons between MA-ABE

Scheme	Central Authority	Adaptive Secure	Standard Model	Access Structure	Ciphertext Size	CP or KP
[7]	Yes	No	Yes	Any Monotone	$\mathcal{O}(n)$	KP
[17]	No	No	Yes	Any Monotone	$\mathcal{O}(n)$	KP
[8]	No	No	Yes	Any Monotone	$\mathcal{O}(n)$	KP
[15]	No	Yes	No	Any Monotone	$\mathcal{O}(\ell)$	CP
[18]	No	Yes	Yes	Any Monotone	$\mathcal{O}(\ell)$	CP
Our Scheme	No	No	Yes	Any Non-Monotone	$\mathcal{O}(1)$	KP

there are several MA-ABE schemes proposed. Lin *et al.* [17] solve this problem by proposing a MA-ABE which uses distributed key generation (DKG) protocol and joint zero secret sharing (JZSS) protocol. However, both the security and the efficiency of their scheme depends on a special parameter m which has to be fixed in system setup phase. With regard to security, the system will be broken when more than m users are colluded. On the other hand, if m becomes large, efficiency will decrease dramatically. V. Božović *et al.* [4] proposed a MA-ABE scheme. In their scheme, central authority is viewed as "Honest but Curious", and new authorities can be jointed to system whenever there is no system re-initial. In the above MA-ABE schemes, every user has a global identifier (GID) to realize collusion resistance which ensures that multiple users cannot combine their keys to perform decryption if one of them can't decrypt message individually. Every user can only use its GID once to apply for decryption key from the same authority. In general MA-ABE, since each user is bonded with GID, if each attribute authority unites all the attributes summited by users, a comprehensive attribute description about the users will be formed. This will disclose users' privacy in sensitive and private settings, such as medical consultation. In [8], Chase and Chow gave a decentralizing KP-ABE scheme. They considered both the user privacy and key-escrow problem in MA-ABE. In their scheme, distributed pseudorandom function (PRF) and anonymous credential technique are used to fix the problem of key-escrow and user's privacy respectively. Their scheme can also be extended to support any monotone access structure. Lewko *et al.* [15] proposed the first ciphertext-policy Decentralizing ABE. They use the famous proof technology, "Dual system encryption" [23], to prove that their scheme is adaptively secure in the Random Oracle Model. Liu *et al.* [18] give an adaptively secure MA-ABE in Standard Model.

For the aspect of constant size ciphertext ABE, Emura *et al.* [12] first proposed a constant ciphertext size ABE scheme. However, it requires a one-to-one correspondence among user's attributes and AND-Gate access structure and this is equivalent to ID-based encryption. Zhou *et al.* [25] proposed a constant ciphertext size CP-ABE scheme by using Boneh *et al.*'s broadcast encryption scheme [3]. This scheme support monotone AND-Gate with wildcard access policy. Herranz *et al.*[14] proposed a threshold attribute based encryption scheme with constant-size ciphertext. Its security based on non-standard aMSH-DDH assumption. Chen *et al.* [9] proposed an AND-Gate with wildcard CP-ABE scheme. In their scheme, both ciphertext size and computational consuming are constant magnitude. By utilizing public inner product technique, Attrapadung *et al.* [1] proposed the first expressive KP-ABE scheme with constant-size ciphertext and supports any non-monotone access structure.

In this paper, based on [8] and [1], we propose a decentralizing KP-ABE scheme with constant ciphertext size which supports any non-monotone access structure. In our scheme, the size of ciphertext is independent on the number of attributes and the access policy can be more expressive than former MA-ABE schemes. Our scheme does not rely on central authority and the security is guaranteed when at least one authorities are uncorrupted. Meanwhile, in our scheme users can receive the decryption key acquired through the joint calculation with authority without letting the authority know any useful information about this key. Therefore, user's privacy has been improved. Our scheme can be proved to be semantic secure in so-called Selective-Set Model based on the n-DBDHE assumption. We give a brief comparison between some well-known MA-ABE scheme and our's proposition in Table 1, where n denote the number of attributes used in ciphertext and ℓ is the size of an access formula. To the best of our knowledge, this is the first MA-ABE which realizes constant ciphertext size and such expressive policy.

In Section 2, we provide some necessary back ground knowledge about access structure and linear secret-sharing scheme. After that, we define decentralizing KP-ABE algorithm and its security notion formally. Our construction and its security argument will be proposed in Section 3 and Section 4 respectively. Some discussion will be given in Section 4, and finally, we conclude in Section 5.

2 Background

In this section, we give the definitions for access structure and relevant background on linear secret share scheme. Then, we present the formal definition of decentralizing KP-ABE scheme and its security model. At last, we offer the basic knowledge of bilinear map and the computational assumption which our scheme is based on.

2.1 Access Structure and Linear Secret Share Scheme

Definition 1. *(Access structure[2]) Let* $\{P_1, P_2, \ldots, P_n\}$ *be a set of parties. A collection* $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ *is monotone if* $\forall B, C$*:if* $B \in \mathbb{A}$ *and* $B \subseteq C$ *then* $C \in \mathbb{A}$*. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection)* $\mathbb{A}$ *of non-empty subsets of* $\{P_1, P_2, \ldots, P_n\}$*, i.e* $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \backslash \{\phi\}$*. The sets in* $\mathbb{A}$ *are called authorized sets, and the sets not in* $\mathbb{A}$ *are called unauthorized sets.*

In the context, all the parties can be seen as attributes. Thus, the access structure $\mathbb{A}$ will contain the authorized sets of attributes. [19] suggests a way to convert any monotone formula to non-monotone formula. For any monotone access structure $\mathbb{A}$, the underlying parties in set $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ can be named as positive or negative, which is denoted by x or x' respectively. Negative attributes are conceptually seen as the negation of positive attributes. In reality, if one person possesses a certain attribute, he is of its positive attribute value; if not, he is of its negative attribute value. Meanwhile, it is required that if $x \in \mathcal{P}$, then $x' \in \mathcal{P}$ and vice versa. For each monotone access structure $\mathbb{A}$ over a set $\mathcal{P}$, one can define a non-monotone access structure $\tilde{\mathbb{A}} = NM(\mathbb{A})$ over $\tilde{\mathcal{P}}$ of all the positive parties in $\mathcal{P}$. Firstly, an operator $N(\cdot)$ is defined as follow. For every set $\tilde{S} \in \tilde{\mathcal{P}}$, one imposes $\tilde{S} \subset N(\tilde{\mathcal{P}})$; and for each party $x \in \tilde{\mathcal{P}}$ such that $x \notin \tilde{S}$, $x \in N(\tilde{\mathcal{P}})$. Finally, $NM(\mathbb{A})$ is defined by saying that $\tilde{S}$ is the authorized set of $NM(\mathbb{A})$ if and only if S is the authorized set of $\mathbb{A}$.

Definition 2. *(Linear Secret-Sharing Schemes (LSSS)[24]) A secret-sharing scheme* Π *over a set of parties* $\mathcal{P}$ *is called linear (over* $\mathbb{Z}_p$*) if*

1) The shares for each party form a vector over $\mathbb{Z}_p$.

2) There exists a matrix an L with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, 2, \ldots, \ell$, the i'th row of L we let the function ρ defined the party labeling row i as $\rho(i)$.When we consider the column vector $\vec{v} = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are random chosen, then Lv is the vector of ℓ shares of the secret s according to Π. The share $(Lv)_i$ belongs to the party $\rho(i)$.

As it is pointed out in [24], every LSSS enjoys the *linear re-construction* property which is defined as follow: suppose Π is an LSSS for the access structure $\mathbb{A}$. If $\omega \in \mathbb{A}$ be any authorized set and let $I \subset \{1, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in \omega\}$. Then there exist constants $\{\mu_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \mu_i \lambda_i = s$. In [2], the author has proved the equivalence between monotone access structure and LSSS. Therefore, both in our main construction and proof we will use LSSS to express monotone access structure.

2.2 Bilinear Map

In our construction, bilinear map is the crucial component. The definition is as follow. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are three cyclic groups of prime order p with the multiplicative group action. g_1 and g_2 is the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Bilinear map $\hat{e}(\cdot, \cdot) : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a deterministic function that takes as input one element from $\mathbb{G}_1$, one element from $\mathbb{G}_2$, and outputs an element in target group $\mathbb{G}_T$ which satisfies following properties:

1) Bilinearity: $\hat{e}({g_1}^a, {g_2}^b) = \hat{e}(g_1, g_2)^{ab}$ for $\forall a, b \in \mathbb{Z}_p$;

2) Non-degeneracy: $\hat{e}(g_1, g_2) \neq 1$, where 1 is the identity element of $\mathbb{G}_T$;

3) Computability: There is an efficient algorithm to compute $\hat{e}(u, v)$ for all $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$.

In our paper, we assume there exist a computable isomorphic $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ from $\mathbb{G}_2$ to $\mathbb{G}_1$ with $\psi(g_2) = g_1$. Meanwhile we assume there isn't an efficient computable isomorphic ψ' from $\mathbb{G}_1$ to $\mathbb{G}_2$. We recommend the paper [6] for reference where the realizing of this kind of bilinear map is discussed.

2.3 n-Decisional Bilinear Diffie-Hellman Exponent Assumption

Security of our system is based on the intractable complexity assumption which is called the n-decisional bilinear Diffie-Hellman Exponent assumption (n-DBDHE). Let algorithm $\mathcal{G}$ takes as input a security parameter λ, and outputs the parameters of bilinear map $\Upsilon = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e})$ where p is a big prime with $|p| = \lambda$, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are three cyclic groups of prime order p, g_1 and g_2 are generator of $\mathbb{G}_1$ and $\mathbb{G}_2$, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map. Meanwhile, there exist an efficient computable isomorphic ψ from $\mathbb{G}_2$ to $\mathbb{G}_1$ and there isn't an efficient computable isomorphic ψ' from $\mathbb{G}_1$ to $\mathbb{G}_2$. We define n-DBDHE problem as follow: Giving the bilinear map parameters Υ and a vector $\vec{y} = (g_1, g_2, {g_2}^{\gamma}, {g_2}^{\gamma^2}, \ldots, {g_2}^{\gamma^n}, , {g_2}^{\gamma^{n+2}}, \ldots, {g_2}^{\gamma^{2n}}, {g_2}^s, T)$ for some unknown random numbers $\gamma, s \in \mathbb{Z}_p$. It is required to distinguish whether $T = \hat{e}(g_1, g_2)^{\gamma^{n+1} s}$ or T as a random element in $\mathbb{G}_T$.

An algorithm $\mathcal{B}$ that outputs $z \in \{0, 1\}$ has advantage ε in solving n-DBDHE if

$$|Pr[\mathcal{B}(\vec{y}, T = \hat{e}(g_1, g_2)^{\gamma^{n+1} s}) = 0] - Pr[\mathcal{B}(\vec{y}, T = R) = 0]| \geq \varepsilon$$

and the probability depends on the coin tosses used by $\mathcal{B}$.

We note with the existence of efficient computable isomorphic ψ, we can efficiently compute ${g_1}^{\gamma^i}$ and ${g_1}^s$, since we know ${g_2}^{\gamma^i}$ and ${g_2}^s$ for $\psi({g_2}^{\gamma^i}) = {g_1}^{\gamma^i}$ and $\psi({g_2}^s) = {g_1}^s$.

Definition 3. *We say that the* (t, ε)*-decisional n-DBDHE assumption holds if no t time algorithm has most* ε *negligible advantage in solving the n-DBDHE problem.*

2.4 Syntax and Secure Definition for Decentralizing KP-ABE

Without loss of generality, we assume that there are N attribute authorities in our decentralizing KP-ABE scheme and k'th authority denoted by AA_k. Our decentralizing KP-ABE scheme consists of four PPT algorithms and is defined as follow:

Setup$(1^\lambda, n)$**:** The setup algorithm takes as input the secure parameter λ and upper bound n of the attribute numbers used in ciphertext. Then, it outputs the public parameters MPK and N authorities' master secret key $\{MSK_k\}_{k\in\{1,2,\ldots,N\}}$.

KeyGen$(MPK, \{MSK_k\}_{k\in\{1,2,\ldots,N\}}, u, \{\tilde{\mathbb{A}}_k\})$**:** The key generation algorithm takes as input the public parameters MPK, the master secret key, user's GID u and an access structure set $\{\tilde{\mathbb{A}}_k\}$ for every authority. It then outputs the user's decryption key SK_u.

Encryption(MPK, M, ω)**:** Encryption algorithm takes as input the public parameters MPK, message M and attribute set ω which describes the ciphertext. It outputs the ciphertext CT.

Decryption(MPK, SK_u, CT)**:** The decryption algorithm takes as input the public parameters MPK, user's decryption key associated with the access structure $\{\tilde{\mathbb{A}}_k\}$ and a ciphertext encrypted under attribute ω. If $\{\mathbb{A}_k\}$ is satisfied by ω, then the decryption procedure will recover the message M. Otherwise, it will output a special symbol $\perp$.

We now describe the sematic secure definition of decentralizing KP-ABE scheme in the Selective-Set model which is analogous to Selective-ID model in identity based encryption [5]. In this model, adversary must declare the attribute set ω^* which it intends to challenge upon before it sees the public parameters, and it is also allowed to request decryption key for any access structure which is not satisfied by ω^*. Now, we give the formal secure game between adversary $\mathcal{A}$ and challenger $\mathcal{B}$ as follow:

*Init***:** Adversary $\mathcal{A}$ firstly declares the set of attribute $\omega^* = \{\omega_k^*\}$, which it wants to be challenged on, where ω_k^* denotes the partial attributes monitored by AA_k. Besides, $\mathcal{A}$ also submits a corrupted authorities list $K_{corr} \subset \{1, \ldots, N\}$.

*Setup***:** Challenger $\mathcal{B}$ runs setup algorithm of decentralizing KP-ABE and delivers the public parameters MPK and $\{MSK_k\}_{k\in K_{corr}}$ to $\mathcal{A}$.

*Phase*1**:** In this phase, $\mathcal{A}$ may provide any $\{\tilde{\mathbb{A}}_k\}$ and GID u to $\mathcal{B}$ to require decryption key with restriction that: 1) $\{\tilde{\mathbb{A}}_k\}$ is not satisfied by ω^* (i.e. at least one j such that $\omega_j^* \notin \tilde{\mathbb{A}}_j^*$); 2) For one GID u and one authority AA_k, $\mathcal{A}$ can require the secret decryption key component only once and 3) $\mathcal{A}$ cannot make key requirement for AA_k, where $k \in K_{corr}$. Then $\mathcal{B}$ runs $KeyGen$ algorithm and returns the corresponding decryption key to $\mathcal{A}$.

*Challenge***:** In this phase, $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. Challenger $\mathcal{B}$ then flips a fair coin $b \in \{0, 1\}$, and runs $Encrypt(MPK, M_b, \omega^*)$. The challenge ciphertext is given to $\mathcal{A}$.

*Phase*2**:** This phase is as same as *Phase*1.

*Guess***:** $\mathcal{A}$ output the guess bit b' for b and we say $\mathcal{A}$ success if $b' = b$.

We define the advantage of adversary A in above game by: $Adv_{\mathcal{A}}^{N-MA-ABE}(1^\lambda) = |Pr[b' = b] - 1/2|$.

Definition 4. *A decentralizing KP-ABE scheme is (t, l, ε)-sematic secure in the Selective-Set model if for any t-times adversary, who corrupts most l authorities ($l \leq N$), has most ε negligible advantage in above game.*

3 Our Construction

Now we describe our decentralizing KP-ABE construction with constant-size ciphertext. Firstly, we give some notations and states used in our scheme. We will treat a vector as a column vector. For group $\mathbb{G} = \langle g \rangle$ with prime order p and any vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top \in \mathbb{Z}_p^n$, $g^{\vec{\alpha}}$ stands for the vector group elements $(g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_n})^\top \in \mathbb{G}^n$. For $\vec{\alpha}$ and $\vec{z}$, we denote their inner product as $\langle \vec{\alpha}, \vec{z} \rangle = \vec{\alpha}^\top \vec{z} = \sum_{i=1}^n \alpha_i z_i$. We assume that when an user, with GID $u \in \mathbb{Z}_p$, applies for her secret decryption key, a non-monotone access structure set is input, which is denoted by $\{\tilde{\mathbb{A}}_k\}$. We have $\tilde{\mathbb{A}}_k = NM(\mathbb{A}_k)$ for some monotone access structure $\mathbb{A}_k$ associated with some LSSS $\Pi_k = (L, \rho)_k$ over the attribute parties $\mathcal{P}_k$, which is monitored by authority AA_k.

Setup$(1^\lambda, n)$**:** The algorithm inputs security parameter λ and upper bound n of the number of attributes used in the ciphertext. It calls algorithm $\mathcal{G}(1^\lambda)$ to get the parameters of bilinear map $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e})$. For every k, AA_k chooses $v_k \in \mathbb{Z}_p$ randomly and computes public value $Y_k = e(g_1, g_2)^{v_k}$. Then $Y = e(g_1, g_2)^{\sum_{k=1}^n v_k}$ is publicly accessible to everyone. Consequently, each pair of AA_k and AA_j selects a secret pseudo random function (PRF) seed $s_{kj} \in \mathbb{Z}_p$. AA_k chooses $x_k \in \mathbb{Z}_p$, computes $y_k = g_1^{x_k}$ and defines a PRF by $PRF_{kj}(u) = g_1^{x_k x_j/(s_{kj}+u)}$. It then picks two random vectors $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top \in \mathbb{Z}_p^n$ and $\vec{\beta} = (\beta_0, \beta_1, \ldots, \beta_n)^\top \in \mathbb{Z}_p^{n+1}$. It sets $\beta' = (\beta_1, \beta_2, \ldots, \beta_n)^\top$, then defines $\vec{H} = (h_1, \ldots, h_n)^\top = g_1^{\vec{\alpha}}$ and $\vec{U} = (u_0, \ldots, u_n)^\top = g_1^{\vec{\beta}}$. The public parameters is formed as:

$$MPK = (Y, y_k, \vec{H}, \vec{U}),$$

and every authority keeps

$$MSK_k = \left(x_k, \{s_{kj}\}_{j\in\{1,\ldots,N\}\backslash\{k\}}\right)$$

as secret key.

KeyGen$(MPK, \{MSK_k\}_{k\in\{1,\ldots,N\}}, u, \{\tilde{\mathbb{A}}_k\})$**:** A user submits an access structure set $\{\tilde{\mathbb{A}}_k\}$ and its GID u. Then it executes the following procedure with AA_k:

1) For $j \in \{1,\ldots,N\}\backslash\{k\}$, user u starts $N-1$ independent invocations of the anonymous key issuing protocol of [8] with input $g = y_j^{x_k}$, $h = g_1$, $\alpha_k = \delta_{jk}R_{kj}$, $\beta_k = \delta_{kj}$ and get $\{D_{kj}\}_{j\in\{1,\ldots,N\}\backslash\{k\}}$ from AA_k. We have $D_{kj} = g_1^{R_{kj}}/PRF_{kj}(u)$ if $k > j$ and $D_{kj} = g_1^{R_{kj}}/PRF_{kj}(u)$ otherwise.
2) AA_k sets its own secret value for user u as $y_{k,u} = v_k - \sum_{j\in\{1,\ldots,N\}\backslash\{k\}} R_{kj}$, then generates a vector $(y_{k,u}, s_2, \ldots, s_n)$, and uses $\Pi_k = (L,\rho)_k$ to get the shares of $y_{k,u}$, denoted by $\{\lambda_i\}$.
3) For party $\check{x}_i \in \mathcal{P}_k$ who has the positive attribute value $x_i \in \mathbb{Z}_p$, AA_k picks a random value $r_i \in \mathbb{Z}_p$, computes $D_{k,i} = \{D_{k,i,1}, D_{k,i,2}, K_{\vec{\rho_i},i}\}$ as follow: set vector $\vec{\rho_i} = (\rho_{i,1},\ldots,\rho_{i,n})^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$ and computes $D_{k,i,1} = g_1^{\lambda_i}u_0^{r_i}$, $D_{k,i,2} = g_2^{r_i}$, $K_{\vec{\rho_i},i} = (K_{i,2}, K_{i,3}, \ldots, K_{i,n}) = ((u_1^{-x_i}u_2)^{r_i}, \ldots, (u_1^{-x_i^{n-1}}u_n)^{r_i}) = g_1^{r_i M_{\vec{\rho_i}}^\top \vec{\beta'}}$, where $M_{\vec{\rho_i}} = \begin{pmatrix} -\frac{\rho_{i,2}}{\rho_{i,1}}, -\frac{\rho_{i,3}}{\rho_{i,1}}, \ldots, -\frac{\rho_{i,n}}{\rho_{i,1}} \\ I_{n-1} \end{pmatrix} = \begin{pmatrix} -x_i, -x_i^2, \ldots, -x_i^{n-1} \\ I_{n-1} \end{pmatrix}$, I_{n-1} denotes identity matrix with order $n-1$.
4) For party $\check{x}_i \in \mathcal{P}_k$ who has the negative attribute value $x_i \in \mathbb{Z}_p$, AA_k picks a random value $r_i \in \mathbb{Z}_p$, computes $D_{k,i} = \{D_{k,i,1}, D_{k,i,2}, K_{\vec{\rho_i},i}\}$ as follow: set vector $\vec{\rho_i} = (\rho_{i,1},\ldots,\rho_{i,n})^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$ and computes $D_{k,i,1} = g_1^{\lambda_i}h_1^{r_i}$, $D_{k,i,2} = g_2^{r_i}$, $K_{\vec{\rho_i},i} = (K_{i,2}, K_{i,3}, \ldots, K_{i,n}) = ((h_1^{-x_i}h_2)^{r_i}, \ldots, (h_1^{-x_i^{n-1}}h_n)^{r_i}) = g_1^{r_i M_{\vec{\rho_i}}^\top \vec{\alpha}}$.
5) When user u get R_{kj} from all authorities, it computes $D_u = \prod_{(k,j)\in\{1,\ldots,N\}\times(\{1,\ldots,N\}\backslash\{k\})} D_{kj} = g_1^{R_u}$ where $R_u = \sum_{(k,j)\in\{1,\ldots,N\}\times(\{1,\ldots,N\}\backslash\{k\})} R_{kj}$. At last, user u gets its secret key

$$SK_u = (\{D_{k,i}\}_{k\in\{1,\ldots,N\},\check{x}_i\in\mathcal{P}}, \{\tilde{\mathbb{A}}_k\}, D_u).$$

Note the Access Structure set $\{\tilde{\mathbb{A}}_k\}$ is included in SK_u.

Encryption(MPK, M, ω)**:** To create the ciphertext, the encryption algorithm chooses $s \in \mathbb{Z}_p$ randomly, uses $\omega = (\omega_1, \omega_2, \ldots, \omega_q)$ (where $q \le n$) to get a vector $\vec{Y} = (y_1, y_2, \ldots, y_n)^\top \in \mathbb{Z}_p^n$ as a coefficient vector from $P_\omega[Z] = \prod_{\sigma\in\omega}(Z-\sigma) = \sum_{i=1}^{q+1} y_i Z^{i-1}$, where if $i > q+1$, the coordinates $y_{q+2}, y_{q+3}, \ldots, y_n$ are set to 0. The ciphertext CT is computed as follow:

$$C_0 = MY^s, C_1 = g_2^s, C_2 = (u_0\prod_{i=1}^n u_i^{y_i})^s, C_3 = (\prod_{i=1}^n h_i^{y_i})^s$$

Decryption(MPK, SK_u, CT)**:** The decrypting algorithm first extracts $\{\tilde{\mathbb{A}}_k\}$ from SK_u and $\omega = (\omega_1, \omega_2, \ldots, \omega_q)$ from CT. Use the same manner to generate vector $\vec{Y} = (y_1, y_2, \ldots, y_n)^\top$ as above. For non-monotone $\tilde{\mathbb{A}}_k \in \{\tilde{\mathbb{A}}_k\}$, we must have $\tilde{\mathbb{A}}_k = NM(\mathbb{A}_k)$ for some monotone access structure $\mathbb{A}_k$ associated with LSSS $\Pi_k = (L,\rho)_k$ defined over the attribute parties $\mathcal{P}_k$. Therefore, If $I'_k = \{i : \check{x}_i \in \omega_k\}$ is the authority set of $\tilde{\mathbb{A}}_k$, $I_k = \{i : \check{x}_i \in \omega'_k\}$ is the authority set of $\mathbb{A}_k$. So We can get reconstructing coefficients $\{\mu_i\}_{i\in I_k}$ of $\Pi_k = (L,\rho)_k$ such that $\sum_{i\in I_k}\lambda_i\mu_i = y_{k,u}$. Then the algorithm computes as follow:

1) For the party $\check{x}_i \in \mathcal{P}_k$ who has the positive attribute value $x_i \in \omega$, the decryption procedure sets $\vec{\rho_i} = (\rho_{i,1},\ldots,\rho_{i,n})^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$ and computes $\check{D}_{k,i} = D_{k,i,1}\prod_{j=2}^n K_{i,j}^{y_j} = g_1^{\lambda_i}(u_0\prod_{i=1}^n u_i^{y_i})^{r_i}$, (since $P_\omega[x_i] = \sum_{j=1}^{q+1} y_i x_i^{j-1} = \langle\vec{Y}, \vec{\rho_i}\rangle = \prod_{\sigma\in\omega}(x_i - \sigma) = 0$) and then gets $\hat{e}(\check{D}_{k,i}, C_1)/\hat{e}(C_2, D_{k,i,2}) = \hat{e}(g_1,g_2)^{s\lambda_i}$.
2) For the party $\check{x}_i \in \mathcal{P}_k$ who has the positive attribute value $x_i \notin \omega$, the decryption procedure sets $\vec{\rho_i} = (\rho_{i,1},\ldots,\rho_{i,n})^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$ and computes $\check{D}_{k,i} = \prod_{j=2}^n K_{i,j}^{y_j} = (h_1^{-\langle\vec{\rho_i},\vec{Y}\rangle/\rho_{i,1}}\prod_{i=1}^n h_i^{y_i})^{r_i}$, Then we have $[\hat{e}(\check{D}_{k,i}, C_1)/\hat{e}(C_3, D_{k,i,2})]^{\rho_{i,1}/\langle\vec{\rho_i},\vec{Y}\rangle} = \hat{e}(h_1,g_2)^{sr_i}$. and $\hat{e}(D_{k,i,1}, C_1) = \hat{e}(g_1,g_2)^{s\lambda_i}\hat{e}(h_1,g_2)^{sr_i}$. Thus $\hat{e}(g_1,g_2)^{s\lambda_i}$ is recovered.
3) Finally, decryption algorithm uses $\{\mu_i\}_{i\in I_k}$ to compute $\prod_{k=1}^N\prod_{i\in I_k}(\hat{e}(g_1,g_2)^{s\lambda_i})^{\mu_i}\hat{e}(D_u, C_1) = \hat{e}(g_1,g_2)^{s\sum_k v_k} = Y^s$ and $C_0/Y^s = M$ to get the message.

If we split each I_k into $I_{k,0}\cup I_{k,1}$ where $I_{k,0}$ and $I_{k,1}$ correspond to positive and negative attributes respectively, decryption can more efficiently compute $\hat{e}(g_1,g_2)^{sy_{k,u}} =$

$$\frac{\hat{e}\left(\prod_{i\in I_{k,0}}\check{D}_{k,i}^{\mu_i}\prod_{i\in I_{k,1}}(D_{k,i,1}^{\mu_i}\cdot\check{D}_{k,i}^{\mu_i\cdot\rho_{i,1}/\langle\vec{\rho_i},\vec{Y}\rangle}), C_1\right)}{\hat{e}(C_2, \prod_{i\in I_{k,0}}D_{k,i,2}^{\mu_i})\cdot\hat{e}\left(C_3, \prod_{i\in I_{k,1}}\check{D}_{k,i,2}^{\mu_i\cdot\rho_{i,1}/\langle\vec{\rho_i},\vec{Y}\rangle}\right)}.$$

Then we get the blinding factor:

$$Y^s = \prod_{k=1}^N \hat{e}(g_1,g_2)^{sy_{k,u}}\hat{e}(D_u, C_1).$$

so as to recover the message. Therefor, decryption algorithm only needs to do 3 pairing calculations for each authority and constant (i.e. $3N+1$, where N is the constant number of authority) pairing calculations totally, which are uncorrelated with the number of attributes.

4 Security Proof

We prove our scheme is selective secure based on the following theorem:

Theorem 1. *Above decentralizing KP-ABE scheme with the maximal bound n for the number of attributes per ciphertext is* $(poly(t), N-1, \varepsilon)$*-semantic secure in* Selective-Set model*, if* (t, ε)*-n-DBDHE assumption hold.*

(Note the user's privacy relies on the security of key issuing protocol. However, it follows the assumptions and security proof of **Theorem 7.5** in Section **7.5.1** of [11].)

Proof. We now show how to construct a simulation algorithm $\mathcal{B}$ to solve n-DBDHE problem by using an adversary $\mathcal{A}$.

Without loss of generality, we assume the attribute universal set is $\mathcal{P} = \{\mathcal{P}_k\}$, where $k \in \{1, 2, \ldots, N\}$ and $\mathcal{P}_k$ is managed by authority AA_k.

Init $\mathcal{B}$ gets the instance of n-DBDHE problem $\vec{y} = (g_1, g_2, g_2{}^{\gamma}, g_2{}^{\gamma^2}, \ldots, g_2{}^{\gamma^n}, , g_2{}^{\gamma^{n+2}}, \ldots, g_2{}^{\gamma^{2n}}, g_2{}^{s}, T)$. It needs to decide whether $T = \hat{e}(g_1, g_2)^{\gamma^{n+1}s}$ or T is a random element in $\mathbb{G}_T$. At the same time, $\mathcal{B}$ receives the challenge attribute set $\omega^* = \{\omega_k^*\}$ and a corruption AA list $K_{corr} \subset \{1, 2, \ldots, N\}$. We write $\omega^* = \{\omega_1^*, \omega_2^*, \ldots, \omega_n^*\}$ in some order.

Setup $\mathcal{B}$ uses the received parameters to simulate the public parameters MPK. Firstly, it uses ω^* to compute vector $\vec{Y} = (y_1, y_2, \ldots, y_n)^\top$ as the coefficients of $P_\omega[Z] = \prod_{\sigma\in\omega}(Z-\sigma) = \sum_{i=1}^{q+1} y_i Z^{i-1}$ ($\mathcal{B}$ sets $y_i = 0$ if $i > n-1$). We let $\vec{\gamma} = (\gamma, \gamma^2, \ldots, \gamma^n)^\top$ which will be used later and $\mathcal{B}$ randomly picks $k^* \in \{1, 2, \ldots, N\} \backslash K_{corr}$.

1) For honest AA_k: $\mathcal{B}$ chooses a random value $x_k \in \mathbb{Z}_p$ and computes $y_k = g_1^{x_k}$. if $k \neq k^*$, it randomly picks $t_k \in \mathbb{Z}_p$ and sets $v_k = t_k$; otherwise, it implicitly sets $v_{k^*} = \gamma^{n+1}\delta_0 + t_{k^*}$ for some random t_{k^*} and $\delta_0 \in \mathbb{Z}_p$. For each pair of honest authorities k and j, $\mathcal{B}$ picks a random PRF seed $s_{kj} \in \mathbb{Z}_p$.

2) For corrupted AA_k: $\mathcal{B}$ chooses a random value $x_k \in \mathbb{Z}_p$ and a random PRF seed $s_{kj} \in \mathbb{Z}_p$ for each pair of corrupted authorities k and j. Then it gives these values to adversary $\mathcal{A}$.

Now, $\mathcal{B}$ simulates $Y = \hat{e}(g_1, g_2)^{\sum_k v_k}$, vector $\vec{U}$ and $\vec{H}$ as follow:

1) $\mathcal{B}$ computes $Y = \hat{e}(g_1, g_2)^{\gamma^{n+1}\delta_0}\ \hat{e}(g_1, g_2)^{\sum_{k\notin K_{corr}} t_k}\ \hat{e}(g_1, g_2)^{\sum_{k\in K_{corr}} v_k}$. We note $\hat{e}(g_1, g_2)^{\gamma^{n+1}\delta_0} = \hat{e}(g_1^{\gamma}, g_2^{\gamma^n})^{\delta_0}$ which is computable for $\mathcal{B}$.

2) For the vector $\vec{U}$ which is related to positive attributes: $\mathcal{B}$ picks $\theta_0 \in \mathbb{Z}_p$ and computes $u_0 = g_1^{\theta_0} g_1^{-\langle\vec{\gamma},\vec{Y}\rangle}$. For $\vec{u}' = (u_1, u_2, \ldots, u_n)^\top$, it chooses a random vector $\vec{\theta} \in \mathbb{Z}_p^n$, and then sets $\vec{u}' = g_1^{\vec{\gamma}} g_1^{\vec{\theta}}$ (i.e. $\vec{\beta}' = \vec{\gamma} + \vec{\theta}$). It is easy to see $\vec{U}$ is uniformly distributed.

3) For the vector $\vec{H}$ which is related to positive attributes: For $\omega^* = \{\omega_1, \omega_2, \ldots, \omega_n\}$, $\mathcal{B}$ defines their corresponding vectors $\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_q$ as $\vec{X}_k = (1, \omega_k, \ldots, \omega_k^{n-1})^\top \in \mathbb{Z}_p^n$. From above, we know $M_{\vec{x}_k} = \begin{pmatrix} -\omega_k, -\omega_k^2, \ldots, -\omega_k^{n-1} \\ I_{n-1} \end{pmatrix}$. Then, for each $k \in [1, q]$, it picks vector $\vec{b}_k$ which forms a $n\times n$ matrix $\mathbf{B} = (\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_q, \vec{0}, \ldots, \vec{0})$ such that $\vec{b}_k^\top M_{\vec{x}_k} = \vec{0}$. $\mathcal{B}$ chooses a random vector $\vec{\delta} \in \mathbb{Z}_p^n$ and set $\vec{H} = g_1^{\mathbf{B}\vec{a}} g_1^{\vec{\delta}}$, where $\vec{a} = (\gamma^n, \gamma^{n-1}, \ldots, \gamma)^\top \in \mathbb{Z}_p^n$. It is easy to see $\vec{H}$ is distributed uniformly.

Now $\mathcal{B}$ gives the public parameter MPK to $\mathcal{A}$. And for each authority AA_k:

1) If AA_k is a honest authority, then $k \notin K_{corr}$, and AA_k's private key $MSK_k = (x_k, \{s_{kj}\}_{j\in\{1,\ldots,N\}\backslash\{k\}})$ are kept by $\mathcal{B}$;

2) If AA_k has been corrupted, that is $k \in K_{corr}$. AA_k's private key $MSK_k = (x_k, \{s_{kj}\}_{j\in\{1,\ldots,N\}\backslash\{k\}})$ are given to adversary $\mathcal{A}$.

Phase 1 In this phase, $\mathcal{A}$ may query some access structure set $\{\tilde{\mathbb{A}}_k\}$ to honest authorities (especially, $\tilde{\mathbb{A}}_k$ for AA_k) with a GID u to get corresponding attribute key components including $\{D_{kj}\}_{j\in\{1,\ldots,N\}\backslash\{k\}}$ and $D_{k,i}$. Let $k = \hat{k}(u)$ be the first authority the adversary $\mathcal{A}$ queried such that $\omega_k \notin \tilde{\mathbb{A}}_k$. The attribute key will depend on k:

1) For $k \neq \hat{k}(u)$, $\mathcal{B}$ sets $y_{k,u} = t_k + z_{k,u}$ for random $z_{k,u} \in \mathbb{Z}_p$. It picks $s_2, s_3, \ldots, s_n \in \mathbb{Z}_p$, uses vector $(y_{k,u}, s_2, \ldots, s_n)^\top \in \mathbb{Z}_p^n$ and $\Pi_k = (L, \rho)_k$) to generate shares $\{\lambda_i\}$.

   - If $\check{x}_i \in \mathcal{P}_k$ who has the negative attribute value $x_i \in \mathbb{Z}_p$. $\mathcal{B}$ first picks a random value $r_i \in \mathbb{Z}_p$, computes $D_{k,i,1} = g_1^{\lambda_i} h_1^{r_i}$, $D_{k,i,2} = g_2^{r_i}$. Then it sets vector $\vec{\rho}_i = (\rho_{i,1}, \ldots, \rho_{i,n})^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$ and computes $K_{\vec{\rho}_i,i} = (K_{i,2}, K_{i,3}, \ldots, K_{i,n}) = ((h_1^{-x_i} h_2)^{r_i}, \ldots, (h_1^{-x_i^{n-1}} h_n)^{r_i})$. So $D_{k,i} = \{D_{k,i,1}, D_{k,i,2}, K_{\vec{\rho}_i,i}\}$.
   - If $\check{x}_i \in \mathcal{P}_k$ who has the positive attribute value x_i. $\mathcal{B}$ can compute $D_{k,i}$ as the same manner above. That is $D_{k,i,1} = g_1^{\lambda_i} u_0^{r_i}$, $D_{k,i,2} = g_2^{r_i}$, and $K_{\vec{\rho}_i,i} = ((u_1^{-x_i} u_2)^{r_i}, \ldots, (u_1^{-x_i^{n-1}} u_n)^{r_i})$.

   Then, $\mathcal{B}$ must simulate AA_k to generate $\{D_{kj}\}_{j\in\{1,\ldots,N\}\backslash\{k\}}$ which is computed from PRF. Since $y_{k,u} = t_k + z_{k,u} = v_k - \Sigma_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R_{kj}$,

we let $R = \Sigma_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R_{kj}$. Therefore, when $k = k^*$, $v_{k^*} = \gamma^{n+1}\delta_0 + t_{k^*}$ and $R = \gamma^{n+1}\delta_0 - z_{k,u}$; otherwise, $k \neq k*$, we have $R = -z_{k,u}$. Now let's calculate D_{kj} in two different situations:

- $k \neq k^*$: In this condition, $g_1^{v_k}$ and is computable and $g_1^R = g_1^{y_{k,n}}/g_1^{v_k} = g_1^{-z_{k,u}}$. Then $\mathcal{B}$ picks R'_{kj} randomly for $j \in \{1,\ldots,N\}\backslash\{k\}$ which satisfies $\Sigma_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R'_{kj} = -z_{k,u}$. So for $k > j$, $\mathcal{B}$ sets $D_{kj} = g_1^{R'_{kj}} PRF_{kj}(u)$ and $D_{kj} = g_1^{R'_{kj}}/PRF_{kj}(u)$ otherwise. (Note B implicitly sets $R_{kj} = R'_{kj}$);
- $k = k^*$: In this case, $g_1^{v_k} = g_1^{\gamma^{n+1}\delta_0 + t_k}$ is not computable for $\mathcal{B}$. At this time, $\mathcal{B}$ chooses R'_{kj} which satisfies $\Sigma_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R'_{kj} = -z_{k,u}$. Now, $\mathcal{B}$ implicitly sets $R_{k\hat{k}(u)} = \gamma^{n+1}\delta_0 + R'_{k\hat{k}(u)}$ and then uses $PRF_{k\hat{k}(u)} g_1^{\gamma^{n+1}\delta_0}$ instead of normal $PRF_{k\hat{k}(u)}$ (Note, we must also use this to compute $PRF_{\hat{k}(u)k}$ later for consistency. Without loss of generality, we assume some $k > \hat{k}(u)$ here. Therefore, $\mathcal{B}$ can compute D_{kj} as above.

2) For $k = \hat{k}(u)$, $\mathcal{B}$ implicitly sets $y_{k,u} = \gamma^{n+1}\theta_0 + t_k + z_{k,u}$ for some random $z_{k,u} \in \mathbb{Z}_p$. Since $\omega_k^* \notin \tilde{\mathbb{A}}_k$, so we have $\omega_k^{*\prime} = N(\omega_k^*)$ and $\omega_k^{*\prime} \notin \mathbb{A}_k$ for some monotone access structure $\mathbb{A}_k$ which is defined over $\mathcal{P}_k$ and satisfies $\tilde{\mathbb{A}}_k = NM(\mathbb{A}_k)$. Meanwhile, $\mathbb{A}_k$ is associated with a LSSS $\Pi_k = (L,\rho)_k$. Therefore, $\vec{1} = (1,0,\ldots,0)^\top \in \mathbb{Z}_p^n$ is not in the span of $L_{\omega_k^{*\prime}}$ which is the sub-matrix of L formed by rows corresponding to attributes in $\omega_k^{*\prime}$. Due to the **proposition 1** of **appendix A.2** in [13], we have that there must exist an efficiently computable vector $\vec{\omega} \in \mathbb{Z}_p^n$ such that $\langle L_{\omega_k^{*\prime}}, \vec{\omega}\rangle = 0$ and $\langle \vec{1}, \vec{\omega}\rangle = 1$. Now $\mathcal{B}$ implicitly sets $\vec{v} = \vec{\xi} + (\gamma^{n+1}\delta_0 + t_k + z_{k,u} - \xi_1)\vec{\omega}$ for some random vector $\vec{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)^\top \in \mathbb{Z}_p^n$ and it is apparent to see $\vec{v}$ is distributed uniformly. If $x_i \in \omega_k^{*\prime}$, then $\langle L_{\omega_k^{*\prime}}, \vec{v}\rangle = \langle L_{\omega_k^{*\prime}}, \vec{\xi}\rangle$ which is uncorrelated with $\gamma^{n+1}\delta_0 + t_k + z_{k,u}$; otherwise, if $x_i \notin \omega_k^{*\prime}$, there must exist computable values V_1 and $V_2 \in \mathbb{Z}_p$ so that $\lambda_i = \langle L_{\omega_k^{*\prime}}, \vec{v}\rangle = V_1(\gamma^{n+1}\delta_0 + t_k + z_{k,u}) + V_2$. Now $\mathcal{B}$ computes private attribute key components $D_{k,i}$ as follow:

- For party $\check{x}_i \in \mathcal{P}_k$ who has the negative attribute value x_i:
  - If $x_i \in \omega_k^*$(and thus $x_i \notin \omega_k^{*\prime}$), so $\langle L_{\omega_k^{*\prime}}, \vec{v}\rangle = V_1(\gamma^{n+1}\delta_0 + t_k + z_{k,u}) + V_2$ for some V_1 and $V_2 \in \mathbb{Z}_p$. Assume $x_i = \omega_i \in \omega_k^*$, $\vec{\rho_i} = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top \in \mathbb{Z}_p^n$. $\mathcal{B}$ chooses $r_i{}'$ randomly and implicitly sets $r_i = r_i{}' + \delta_0\gamma^i$. We argue that $D_1 = g_1^{\gamma^{n+1}\delta_0 + t_k + z_{k,u}} h_1^{r_i{}'}$ is computable for simulator $\mathcal{B}$. The reason is as follow: $D_1 = g_1^{\gamma^{n+1}\delta_0 + t_k + z_{k,u}} h_1^{r_i{}'} = g_1^{t_k + z_{k,u}} g_1^{\gamma^{n+1}\delta_0} h_1^{r_i{}'}$ and $h_1 = g_1^{\delta_1 + \sum_{i=1}^q a_i} = g_1^{\delta_1}\Pi_{i=1}^q g_1^{\gamma^{n+1-i}}$. So, $D_1 = g_1^{t_k + z_{k,u}} g_1^{\gamma^{n+1}\delta_0} h_1^{r_i}(g_1^{\delta_1}\Pi_{i=1}^q g_1^{\gamma^{n+1-i}})^{-\delta_0\gamma^{n+1}} = g_1^{t_k + z_{k,u}} h_1^{r_i}((g_1^{\delta_1})^{\gamma_i}\prod_{i=1,i\neq j} g_1^{\gamma^{n+1-i+j}})^{-\delta_0}$, which is independent on term γ^{n+1}. Then $\mathcal{B}$ picks a random value $r' \in \mathbb{Z}_p$ and computes $D_{k,i,1} = D_1^{V_1} g_1^{V_2} h_1^{r'} = g_1^{V_1(\gamma^{n+1}\delta_0 + t_k + z_{k,u}) + V_2} h_1^{r'_i V_1 + r'} = g_1^{\lambda_i} h_1^{r'_i V_1 + r'}$ and $D_{k,i,2} = D_2^{V_1} g_2^{r'} = g_2^{r_i{}' V_1 + r'}$, where $D_2 = g_2^{r_i}(g_2^{\gamma^i})^{-\delta_0}$. For $K_{\vec{\rho}_i}$, $\mathcal{B}$ first computes $K'_{\vec{\rho}_i,i} = (K'_{i,2}, K'_{i,3}, \ldots, K'_{i,n}) = \{(h_1^{-\frac{\rho_i}{\rho_1}} h_i)^{r'_i}\}_{i=2,\ldots,n} = g_1^{r'_i M_{\vec{\rho}_i}^\top \vec{\alpha}}$. We note that $M_{\vec{\rho}_i}^\top \vec{\alpha} = M_{\vec{\rho}_i}^\top \mathbf{B}\vec{a} + M_{\vec{\rho}_i}^\top \vec{\delta}$ is independent on γ^{n+1-i}, so $g_1^{r'_i M_{\vec{\rho}_i}^\top \vec{\alpha}}$ does not include term $g_1^{\gamma^{n+1}}$ (recall $r_i = r_i{}' + \delta_0\gamma^i$). Therefore $K'_{\vec{\rho}_i,i}$ is computable for $\mathcal{B}$. At last $\mathcal{B}$ computes $K_{\vec{\rho}_i,i}$ by $K_{i,j} = (K'_{i,j})^{V_1}(h_1^{-\frac{\rho_i}{\rho_1}} h_j)^{r'} = (h_1^{-\frac{\rho_i}{\rho_1}} h_j)^{r'_i V_1 + r'}$, and then forms $D_{k,i}$ properly.
  - If $x_i \notin \omega_k^*$(and thus $x_i \in \omega_k^{*\prime}$), as mentioned above, $\lambda_i = \langle L_{\omega'_k}, \vec{v}\rangle$ is independent on $\gamma^{n+1}\delta_0 + t_k + z_{k,u}$ and thus is computable for $\mathcal{B}$. So $\mathcal{B}$ can choose random exponent r_i to form $D_{k,i}$ normally.
- For party $\check{x}_i \in \mathcal{P}_k$ who has the positive attribute value x_i:
  - If $x_i \in \omega_k^*$(and thus $x_i \in \omega_k^{*\prime}$), $\lambda_i = \langle L_{\omega'_k}, \vec{v}\rangle$ is independent on $\gamma^{n+1}\delta_0 + t_k + z_{k,u}$ and $\mathcal{B}$ is able to compute $D_{k,i}$
  - If $x_i \notin \omega_k^*$(and thus $x_i \notin \omega_k^{*\prime}$), $\mathcal{B}$ implicitly sets $\lambda_i = \langle L_{\omega_k^{*\prime}}, \vec{v}\rangle = V_1(\gamma^{n+1}\delta_0 + t_k + z_{k,u}) + V_2$ and then considers $n \times (n-1)$ matrix: $\begin{pmatrix} -x_i, -x_i^2, \ldots, -x_i^{n-1} \\ I_{n-1} \end{pmatrix}$. Since $x_i \notin \omega^*$, we have $\vec{\zeta}^\top M_{\vec{\rho}_i} = \vec{0}$ where $\vec{\zeta} = (\zeta_1, \zeta_2, \ldots,)^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top \in \mathbb{Z}_p^n$. We now explain how to compute $(D_1, D_2, K'_{\vec{\rho}_i,i}) = (g_1^{\gamma^{n+1}\delta_0 + t_k + z_{k,u}} u_0^{r_i}, g_2^{r_i}, g_1^{r'_i M_{\vec{\rho}_i}^\top \vec{\beta}'})$ where $\vec{\beta}' = (\beta_1, \beta_2, \ldots, \beta_n)^\top$ having been defined above. $\mathcal{B}$ implicitly sets $r_i = R'_i + \delta_0(\zeta_1\gamma^n + \zeta_2\gamma^{n-1} + \ldots + \zeta_n\gamma)/\langle \vec{Y}, \vec{\zeta}\rangle$ (since $\langle \vec{Y}, \vec{\zeta}\rangle \neq 0$). Note, the coefficient of $\langle \vec{Y}, \vec{\gamma}\rangle$ is $\delta_0\langle \vec{Y}, \vec{\zeta}\rangle/\langle \vec{Y}, \vec{\zeta}\rangle = \delta_0$, so when computing $g_1^{\gamma^{n+1}\delta_0 + t_k + z_{k,u}} u_0^{r_i} = g_1^{\gamma^{n+1}\delta_0 + t_k + z_{k,u}}(g_1^{\theta_0} g_1^{-\langle \vec{Y}, \vec{\gamma}\rangle})^{r_i}$, term $g_1^{\gamma^{n+1}\delta_0}$, which $\mathcal{B}$ cannot compute, will be canceled out. Therefore $\mathcal{B}$ is able to compute D_1. Meanwhile, $D_2 = g_2^{r_i} = g_2^{r'_i} g_2^{\delta_0 \sum_{i=1}^n \gamma^{n+1-i}}$ is also computable from n-DBDHE pa-

rameters. Because in each coordinate of vector $(\zeta_1\gamma^n + \zeta_2\gamma^{n-1} + \ldots + \zeta_n\gamma)M_{\vec{\rho_i}}^{\top}\vec{\gamma}$, no term γ^{n+1} will appear. So we can compute $K'_{\vec{\rho}_i,i} = g_1^{r_i M_{\vec{\rho}_i}^{\top}\vec{\beta}'} = g_1^{r_i M_{\vec{\rho}_i}^{\top}(\vec{\gamma}+\vec{\theta})}$ without knowing of $g_1^{\gamma^{n+1}}$.

Again, let's calculate D_{kj} in two different situations:

- $k = k^*$: In this case, we have $v_{k^*} = \gamma^{n+1}\delta_0 + t_k$ and note $y_{k,u} = \gamma^{n+1}\delta_0 + t_k + z_{k,u}$, so $R = \sum_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R_{kj} = -z_{k,u}$. $\mathcal{B}$ randomly picks $R'_{kj} \in \mathbb{Z}_p$ for $j \in \{1,2,\ldots,N\}\backslash\{k\}$, which satisfies $\sum_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R'_{kj} = -z_{k,u}$, and gets $D_{kj} = g_1^{R'_{kj}} PRF_{kj}(u)$ or $D_{kj} = g_1^{R'_{kj}}/PRF_{kj}(u)$.
- $k \neq k^*$: In this case, we must have some $k' = k^* < \hat{k}(u)$ and $k' \neq \hat{k}(u) = k$. Remember that we use $PRF_{\hat{k}(u)k'} g_1^{\gamma^{n+1}\delta_0}$ instead of normal $PRF_{\hat{k}(u)k'}$. Then $\mathcal{B}$ chooses $R'_{kj} \in \mathbb{Z}_p$ which satisfies $\sum_{j\in\{1,2,\ldots,N\}\backslash\{k\}} R'_{kj} = z_{k,u}$ for all $j \neq k'$ and computes $D_{kj} = g_1^{R'_{kj}} PRF_{kj}(u)$ or $D_{kj} = g_1^{R'_{kj}}/PRF_{kj}(u)$. $\mathcal{B}$ implicitly sets $R'_{kk'} = R_{kk'} - \gamma^{n+1}\delta_0$ and gets $D_{kk'} = g_1^{R'_{kk'}}/(PRF_{kk'}(u))g_1^{\gamma^{n+1}\delta_0}$.

Above all, $\mathcal{B}$ gets all the attribute decryption key components and forms

$$SK_u = (\{D_{k,i}\}_{k\in\{1,\ldots,N\},\check{x}_i\in\mathcal{P}}, \{\tilde{\mathbb{A}}_k\})$$

and $\{D_{kj}\}_{j\in\{1,2,\ldots,N\}\backslash\{k\}}$ for adversary $\mathcal{A}$.

Challenge In this phase simulator $\mathcal{B}$ sets challenge ciphertext as follow:

$$C_0 = T^{\delta_0}\hat{e}(g_1, g_2^s)^{\sum_{k\notin K_{corr}} t_k}\hat{e}(g_1, g_2^s)^{\sum_{k\in K_{corr}} v_k},$$

$$C_1 = g_2^s,\ C_2 = h^{\theta_0+\langle\vec{\gamma},\vec{\theta}\rangle},\ C_3 = h^{\langle\vec{Y},\vec{\delta}\rangle}.$$

It is easy to see that C_0, C_1, C_2 are well-formed ciphertext and then We make it clear that the same is true for C_3.

Here we have $C_3 = (\prod_{i=1}^n h_i^{y_i})^s = g_1^{\langle\vec{Y},\mathbf{B}\vec{a}+\vec{\delta}\rangle s} = g_1^{(\langle\vec{Y},\mathbf{B}\vec{a}\rangle+\langle\vec{Y},\vec{\delta}\rangle)s}$ and argue $\langle\vec{Y},\mathbf{B}\vec{a}\rangle = 0$. Recall $\vec{Y}$ is computed from challenge attribute set $\omega^* = \{\omega_1^*, \omega_2^*, \ldots, \omega_n^*\}$ and $\mathcal{B}$ sets $\vec{X}_k = (1, \omega_k, \ldots, \omega_k^{n-1})^{\top}$. So we have a vector $\vec{\eta} = (y_2, y_3, \ldots, y_n)^{\top}$ such that $M_{\vec{X}_k}\vec{\eta} = \vec{Y}$. Therefore $\langle\vec{Y},\mathbf{B}\vec{a}\rangle = \vec{Y}^{\top}\cdot\mathbf{B}\vec{a} = \vec{Y}^{\top}\sum_{k=1}^q a_k\vec{b}_k = \sum_{k=1}^q a_k\vec{Y}^{\top}\vec{b}_k = \sum_{k=1}^q a_k\vec{\eta}^{\top}M_{\vec{X}_k}^{\top}\vec{b}_k = 0$ (recall $M_{\vec{X}_k}^{\top}\vec{b}_k = 0$).

Phase 2 In this phase, $\mathcal{B}$ acts as same as *Phase* 1.

Guess Finally, $\mathcal{A}$ outputs its guess b'. If $b' = b$, $\mathcal{B}$ output $T = \hat{e}(g_1, g_2)^{\gamma^{n+1}s}$ or T is a random element $\in \mathbb{G}_T$, otherwise.

we can see that if the adversary $\mathcal{A}$ has non-negligible advantage over above scheme, we will also have non-negligible advantage to find the solution of the instance of n-DBDHE problem. This completes the proof. □

5 Discussions

In this section, we briefly discuss the structure and efficiency of our scheme which is proposed above.

The main advantage in the scheme is that the number of ciphertext size is $\mathcal{O}(1)$. There are only 4 group elements in the ciphertext regardless of the number of attribute be used. The cost of this advantage is longer private key of size $\mathcal{O}(t \cdot n)$, where t denotes the number of attribute in access structure. Another advantage of our scheme is that any non-monotone access structure is supported. Therefore, more expressive access policy on encrypted data in multi-authority setting can be realized. What' more, by using the anonymous key issuing protocol, any user can use any alias to apply for its private key without letting authority know its real GID. This improves the privacy of users. We also note that our scheme supports large attribute universe. One drawback in construction is that we can only realize AND-Gate structure between the authorities, and we note Lewko *et al.*'s construction [15] can realize more flexible relation among authorities.

Now we briefly analysis the computational efficiency of our scheme. Decryption algorithm is the most consuming part of the ABE scheme since it requires large amount of pairing calculations. However, our scheme enjoys only constant pairing calculations with the modified decryption equations and uncorrelated with size of attributes (see Section 3). This is an useful property since decryption terminal has lower computing power in general. On the other hand, almost all MA-ABE schemes need pairing calculations linearly with size of attributes. Since pairing is the most expensive computation overhead than multiplication and modular exponentiation in group, thus our scheme enjoys more efficient decryption cost. In Table 2, we give the brief comparison in numbers of pairing calculation among the proposed MA-ABE schemes in decryption phase. n and ℓ denote number of ciphertext attributes and the size of access structure respectively the same as them in Table 1. N denotes the number of authorities which be seen as constant number. t denotes the threshold value of (t, N) threshold secret-sharing scheme. Another cost part of our scheme is key generation algorithm which requires secret key issuing protocol. However, this procedure only needs once for each user unless key redistribution is required.

6 Conclusion

In this paper, we have proposed an expressive decentralizing KP-ABE scheme. In our construction, the ciphertext size does not rely on the number of attribute used

Table 2: Pairing calculations in decryption algorithm among MA-ABE schemes

	[7]	[17]	[8]	[15]	[18]	Our Scheme
Pairing Cost	$N \cdot \mathcal{O}(n)$	$t \cdot \mathcal{O}(n)$	$N \cdot \mathcal{O}(n)$	$N \cdot \mathcal{O}(\ell)$	$N \cdot \mathcal{O}(\ell)$	$3N + 1$

in ciphertext. What's more, we can reduce it to non-interactive n-DBDHE assumption in Selective-Set model. To the best of our knowledge, this is the first multi-authority realizing such expressive access policy and constant ciphertext size. Further studies can be focused on how to construct decentralizing CP-ABE with constant ciphertext size and more efficient decentralizing ABE schemes.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61003230) and Open fund of State Key Laboratory of Information Security. The authors are grateful to the anonymous reviewers for valuable comments.

References

[1] N. Attrapadung and B. Libert, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proceedings of 14th International Conference on Practice and Theory in Public Key Cryptography (PKC 2011)*, pp. 90–108, Taormina, Italy, March 2011.

[2] A. Beimel, "Secure schemes for secret sharing and key distribution," Israel Institute of Technology, Haifa, Israel, 1996.

[3] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proceedings of Advances in Cryptology - 25th Annual International Cryptology Conference (CRYPTO 2005)*, pp. 258–275, Santa Barbara, California, USA, August 2005.

[4] V. Božović, D. Socek, R. Steinwandt, and V. Villanyi, "Multi-authority attribute based encryption with honest-but-curious central authority," *IACR Cryptology ePrint Archive*, 2009.

[5] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proceedings of Advances in Cryptology - 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2004)*, pp. 207–128, Interlaken, Switzerland, May 2004.

[6] D. Charles, "On the existence of distortion maps on ordinary elliptic curves," *IACR Cryptology ePrint Archive*, 2006.

[7] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography (TCC 2007)*, pp. 515–534, Amsterdam, The Netherlands, February 2007.

[8] M. Chase and S. Chow, "Improving privacy and security in multi-authority attribute based encryption," in *Proceddings of 16th ACM Conference on Computer and Communications Security (CCS'2009)*, pp. 121–130, Chicago, Illinois, USA, October 2010.

[9] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Proceedings of 5th International Conference, ProvSec 2011*, pp. 84–101, Xi'an, China, October 2011.

[10] L. Cheung and C. Newport, "Provably secure ciphertext policy attribute based encryption," in *Proceedings of 14th ACM Conference on Computer and Communication Security (CCS'2007)*, pp. 456–465, Alexandria, VA, USA, October 2007.

[11] S. S. M. Chow, "New privacy-preserving architectures for identity-/attribute-based encryption," New York University, 2010.

[12] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proceedings of Information Security Practice and Experience - 8th International Conference (ISPEC 2009)*, pp. 13–23, Xi'an, China, April 2009.

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

[14] J. Herranz, F. Laguillaumie, and C. Rafols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proceedings of 13th International Conference on Practice and Theory in Public Key Cryptography (PKC 2010)*, pp. 19–34, Paris, France, May 2010.

[15] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of Advances in Cryptology - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588, Tallinn, Estonia, May 2011.

[16] J. Li, K. Ren, and K. Kim, "A2be: Accountable attribute-based encryption for abuse free access control," *IACR Cryptology ePrint Archive*, 2009.

[17] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proceedings of 9th International Conference on Cryptology in India (INDOCRYPT 2008)*, pp. 426–436, Kharagpur, India, December 2008.

[18] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random

oracles," in *Proceedings of 16th European Symposium on Research in Computer Security (ESORICS 2011)*, pp. 278–297, Leuven, Belgium, September 2011.

[19] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of 14th ACM Conference on Computer and Communication Security (CCS'2007)*, pp. 195–203, Alexandria, VA, USA, October 2007.

[20] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 158–164, 1978.

[21] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of Advances in Cryptology - 25th Annual International Cryptology Conference (CRYPTO 2005)*, pp. 457–473, Santa Barbara, California, USA, August 2005.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of Advances in Cryptology - 4th Annual International Cryptology Conference (CRYPTO 1984)*, pp. 47–53, Santa Barbara, California, USA, August 1984.

[23] B. Waters, "Dual system encryption: realizing fully secure ibe and hibe under simple assumptions," in *Proceedings of Advances in Cryptology - 29th Annual International Cryptology Conference (CRYPTO 2009)*, pp. 619–636, Santa Barbara, CA, USA, August 2009.

[24] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proceedings of 14th International Conference on Practice and Theory in Public Key Cryptography (PKC 2011)*, pp. 90–107, Taormina, Italy, March 2011.

[25] Z. Zhou and D. Huang, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," in *Proceedings of the 17th ACM conference on Computer and communications security (CCS 2010)*, pp. 753–755, New York, NY, USA, October 2010.

Qinyi Li is a M.S. student in the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). He received his B.S. degree from Southwest University of Science and Technology, Mianyang, P.R China in 2010. His research interests include: information security and Cryptography.

Hu Xiong is a lecturer in the UESTC. He received his M.S. and PH.D degree from UESTC in 2004 and 2009, respectively. His research interests include: information security and cryptography.

Fengli Zhang is a professor in the UESTC. She received her M.S. and PH.D. degree UESTC in 1986 and 2007 respectively. His research interests include: information security and distributed mobile data management.

Shengke Zeng received her Master's degree from Xihua University, China in 2008. She is currently a PhD Candidate in the School of Computer Science and Engineering, UESTC. Her research interests are digital signature and anonymous authentication protocol.