A Privacy-Preserving Data Sharing System with Decentralized Attribute-based Encryption Scheme

Li Kang and Leyou Zhang
(Corresponding author: Li Kang)

School of Mathematics and Statistics, Xidian University
Xi'an, Shaanxi 710071, China
(Email: li_kkang@126.com)

(Received Mar. 26, 2019; Revised and Accepted Nov. 16, 2019; First Online Jan. 29, 2020)

Abstract

The huge storage space and powerful computing power make cloud servers be the best place for people to store data. However, convenience comes with the risk of data leakage as the cloud servers are not completely reliable. To ensure data confidentiality and user privacy, decentralized attribute-based encryption (ABE) can provide a solution. Nevertheless, most of existing works cannot provide a complete method since there are some vulnerabilities in users' collusion resilience and privacy protection. In this paper, a decentralized ciphertext-policy (CP) ABE scheme is proposed for the secure sharing of data. In the proposed scheme, without requiring any cooperation and knowing users' global identifiers (GID), authorities can issue private keys for users independently through a privacy-preserving key generation protocol. Additionally, this scheme supports policy anonymity. The security of the proposed scheme is reduced to the decisional bilinear Diffie-Hellman (DBDH) assumption. Theoretical analysis and performance evaluations illustrate the efficiency of the scheme.

Keywords: Cloud Storage; Data Sharing; Decentralized CP-ABE; Privacy-preserving

1 Introduction

1.1 Background

The emergence of the big data era makes cloud storage technology become the one of the hottest technologies. The cloud becomes the best choice for data users to store data as its storage space is much larger than the local one. The so-called cloud storage is a new technology that puts data on the cloud for human access, by which users can easily access data at any time and anywhere through any connected device. However, it also brings some challenges, such as data confidentiality.

In many cases, the data owners are reluctant to expose their data stored in the cloud server to all users as the data may be sensitive, such as the personal health record (PHR). PHR is a new summarized electronic record of an individual's medical data and information, which are often exchanged through cloud servers. However, the record may contain sensitive information about consumers, such as allergies, diseases, etc. Placing the raw PHR data directly on an unreliable third-party server will threaten the owner's privacy. Thus, to protect the data confidentiality, it is an effective way for the owner to execute encryption operation before uploading it to the cloud server.

In 2005, Sahai and Waters [30] proposed the concept of attribute-based encryption (ABE) to support fine-grained access control, in which only when the attributes of ciphertext match the decryption keys can the user successfully recover the received ciphertext. Since then, many studies [4,22,25] have been proposed. The multi-authority ABE (MA-ABE) [5] was introduced to reduce the burden on central authority (CA) and to divide its powers. In 2011, the decentralized ABE scheme [18] as a new MA-ABE scheme was proposed by Lewko and Waters. Neither CA nor authority cooperation exists in this construction, thus it is more in accord with the actual requirements for the independence of authorities. The basic properties [17,21] that all ABE schemes should satisfy are: fine-grained access control, scalability, data confidentiality, and collusion resistance.

In addition, the privacy protection of users cannot be ignored. Although the rapid development of cloud computing technology enables users to deal with a large amount of digital information, the privacy of personal data is also facing unprecedented challenges. After witnessing some information leakage incidents, people gradually realize the importance of privacy protection. They want to keep their data private while gaining legal access. Therefore, there is an urgent need for an encryption

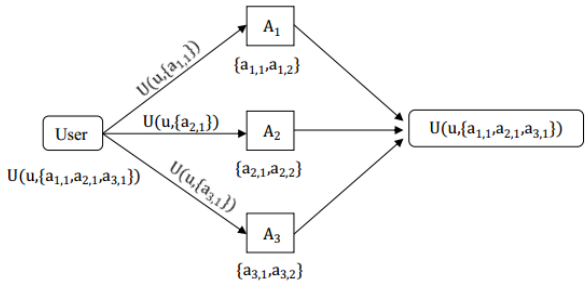

Figure 1: Colluding authorities gather information about the user

scheme that can protect users' privacy.

In the MA-ABE schemes, in order to protect data confidentiality against collusion attacks, the global identifier (GID) is usually tied to the decryption keys. Although the introduction of global identifier in secret keys enhances their uniqueness and can resist the collusion attack of unauthorized users to a certain extent, the user's privacy will inevitably be compromised if he/she directly sends the undisguised identifier to each authority for private key request. As shown in Figure 1, suppose that the authority $A_i (i = 1, 2, 3)$ manages the attribute set $\tilde{A}_i = \{a_{i,j}\}_{j=\{1,2\}}$, the user U who owns attributes $\{a_{1,1}, a_{2,1}, a_{3,1}\}$ and identifier u directly submits $(u, a_{i,1})$ to A_i for secret key request. By jointly tracking the same u [6], the colluding authorities can easily gather all the attributes attached to it. As a result, users' personal information $U(u, \{a_{1,1}, a_{2,1}, a_{3,1}\})$ is thoroughly exposed to them. Therefore, the hiding of GID is the primary task of constructing a privacy-preserving encryption scheme. Besides, the access policy also needs to be hidden because it indicates the legal recipient. Malicious users can infer the attributes of the recipient based on the access policy. This also compromises user privacy. Based on this, a secure decentralized scheme dedicated to privacy protection is proposed.

1.2 Related Work

Along with the development, many expansion schemes [1, 29, 35] were derived from the original ABE scheme [30]. In general, ABE comes in two categories: key-policy ABE (KP-ABE) [11] and ciphertext-policy ABE (CP-ABE) [2]. The CP-ABE scheme enables data owners to achieve absolute control over their data and decide who can and cannot access the data, since the access policy is determined by them. In the data encryption system, considering the computational and storage costs of the authorized organization, single-authority ABE scheme like [36] is no longer applicable. Since there is only a trusted central authority (CA) to manage all users, thus the computational and communication costs of CA have undoubtedly increased. More notably, the excessive power of CA will be a potential risk because it has all users' information and decryption keys, so that it can access all encrypted files. Once the CA is corrupted, the confidentiality of the data and the user privacy will be compromised. In addition, there are many categories of attributes in the real world, so it is impractical to have only one authority to manage them.

Chase [5] put forward a multi-authority ABE (MA-ABE) scheme to weaken the power of central authority (CA), in 2007. Instead of a single authority in the scheme, there are many authorities here to issue private keys for users. While sharing the pressure of single-authority, multiple authorities have differentiated its rights and provided a more secure and effective management mode. However, there is still a CA to manage other attribute authorities, and the multiple authorities need to cooperate to initialize the system. Chase and Chow [6] first introduced the concept of privacy protection and proposed a privacy-preserving MA-ABE scheme, which employed an anonymous key issuing protocol to achieve the goal of hiding the user's GID and employed a distributed pseudorandom functions (PRF) to get rid of the CA. Later, some MA-ABE schemes with privacy-preserving in PHR system have been proposed [19,27,32]. These schemes employed the distributed PRF technique in [6] to protect the GID information. In these schemes, there are multiple authorities that manage a set of disjoint attributes, though CA is not required, there is still a partnership among the multiple authorities namely every two authorities (A_k, A_j) must perform a 2-party key exchange to share the PRF seed $s_{k,j} = s_{j,k}$, which increases communication and computing costs. In 2011, Lewko and Waters [18] introduced a novel MA-ABE, namely decentralized attribute-based encryption scheme. Unlike the previous schemes, there is no CA in this scheme, and the authorities are independent of each other without any cooperation.

The first decentralized KP-ABE scheme considering user GID privacy was proposed by Han *et al.* [12], in 2012. Unfortunately, it turned out to be unsafe [10]. Then a series of improvement schemes [28, 34] were proposed. In scheme [34], Zhang *et al.* modified the original scheme and came up with a more secure scheme against user collusion. All the schemes mentioned above exploited the technique of hiding GID in [6] to produce the secret keys for users.

In 2015, Huang *et al.* [15] put forward a revocable MA-ABE scheme. In their scheme, a CA is needed to send the seed s_k to each authority A_k and generate the secret key component to users. To protect user GID and attributes from being leaked, Han *et al.* [13] proposed a PPDCP-ABE scheme. However, Wang *et al.* [31] found that it could neither resist the collusion attack nor achieve attributes hiding. In 2018, the scheme [14] gave a new idea. In their proposal, there exists an identity management (IDM) that actually acts as a fully trusted CA and multiple attribute authorities. IDM is responsible for generating a pseudonym $Pid_{U,j}$ for the user U corresponding to the authority A_j, and generating an anonymous identity certificate $(AID_{Cred,j})$ for the user by signing the pseudonym and attributes information. The $AID_{Cred,j}$ is

Table 1: Comparisons of the proposed scheme with other MA-ABE schemes

Schemes	Central Authority	*AA* Cooperation	GID Anonymous	Policy Anonymous	Decryption Outsourced	Collusion Resistance	
						User	Authority
Huang [15]	Yes	No	No	No	No	Yes	N-1
Feng [9]	Yes	Yes	Yes	Yes	No	Yes	N-2
Fan [7]	Yes	No	No	Yes	No	Yes	N-1
Hu [14]	Yes	No	Yes	No	No	Yes	N-1
Chase [6]	No	Yes	Yes	No	No	–	N-2
Qian [27]	No	Yes	Yes	No	No	–	N-1
Zhang [34]	No	No	Yes	No	No	Yes	N-1
Qian [26]	No	No	Yes	No	No	No	N-1
Feng [8]	No	No	No	Yes	No	Yes	–
Han [13]	No	No	Yes	No	No	No	N-1
Lyu [23]	No	No	Yes	No	Yes	No	N-1
Ours	No	No	Yes	Yes	Yes	Yes	N-1

sent to A_j for verification, and if the verification passes, the authority will generate partial secret keys for the user. During the whole process, the authority cannot know the user's GID and attributes information, so the user's privacy is protected. This scheme is resistant to users collusion and authorities collusion, however the existence of IDM and collaboration between IDM and multiple authorities suggests that further improvement is needed. In 2017, Lyu *et al.* [23] proposed a decentralized scheme to achieve GID anonymity and improve the efficiency by using the online/offline encryption and the verifiable outsource decryption. This scheme can tolerant $N-1$ compromised authorities. However, the authors did not mention the privacy of the access policy.

In consideration of GID and policy privacy, Qian *et al.* [26] employed the AND,OR gates on multi-valued attributes and put forward a PPDCP-ABE with fully hidden policy scheme in 2013. But a malicious user can guess the access structure with a simple test: $e(C_{i,j,1}, g) \stackrel{?}{=} e(\Pi_{k\in I_c} T_{i,j}^k, C_{i,j,2})$. Fan *et al.* [7] came up with a MA-ABE scheme for a hidden policy under the q-BDHE assumption. In 2018, a new access control system [8] was proposed by Feng *et al.*, where they divided attributes into attribute names and attribute values, and realized policy hiding by hiding attribute values. In this scheme, GID is bounded with the time period, namely $H_1(GID||Time)$, to resist the collusion attack of illegal users. In 2019, Feng *et al.* [9] improved the scheme [16] and proposed a privacy-preserving searchable CP-ABE scheme, which achieved attributes anonymity and collusion resistance. However, the CA is needed in this scheme to issue the random identity (RID) and user key (UK) for each user, where RID is used to replace the user's real identity to achieve identity anonymity.

1.3 Contributions

As shown in Table 1, some existing solutions either have weaknesses in privacy protection and collusion resistance, or require the CA or cooperation among multiple authorities to generate private keys for users. To improve the user privacy and data confidentiality, a privacy-preserving decentralized CP-ABE scheme is proposed, in which the CA is no longer needed and each authority can independently issue partial secret key for the user. The proposed scheme supports both user anonymity and collusion resistant, and the proxy server is introduced here to undertake partial decryption computation. The main contributions of the proposed paper are listed below.

- A decentralized CP-ABE scheme is presented to cater to the requirements of a distributed system where there is no central authority and multiple attribute authority can independently manage users' attributes and generate secret keys for them.
- By using the privacy-preserving key generation protocol (PPKeyGen) and composite order bilinear groups, we hide both GID and access policy, thus providing a higher privacy security.
- The proposed construction can resist collusion attacks of the unauthorized users and tolerant $N-1$ (N is the number of attribute authorities involved in encryption) corrupted authorities, which provides a guarantee for data confidentiality.
- Theoretical analysis and experimental simulations are given to enhance the credibility of the proposed scheme in terms of security and efficiency.

1.4 Organization

The organization of the rest paper is as follows. In Section 2, some preliminaries involved in this paper are listed. The system model and security model are described in Section 3. In Section 4, the proposed scheme is presented in detail. Sections 5 and 6 show the security analysis and performance analysis, respectively. A brief conclusion is given in Section 7.

2 Preliminaries

2.1 Composite Order Bilinear Groups

Assume that $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups with same order $N = pp_1$, where p and p_1 are two different primes, $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ is a bilinear map. $\mathbb{G}_p$ and $\mathbb{G}_{p_1}$ are subgroups of the group $\mathbb{G}$ having order p and p_1 respectively. The map e satisfies the following features:

1) Bilinearity: $\forall f, h \in \mathbb{G}, \forall u, v \in \mathbb{Z}_N$, here is the equation $e(f^u, h^v) = e(f, h)^{uv}$.

2) Non-degenerate: $\exists f \in \mathbb{G}$ such that $e(f,h)$ in group $\mathbb{G}_T$ has order N.

3) Orthogonality: Let g_p and g_{p_1} be the generator of group $\mathbb{G}_p$ and $\mathbb{G}_{p_1}$, respectively. Notice that e is efficiently computable, and it actually satisfies another property: $e(g_p, g_{p_1}) = 1$.

2.2 Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Suppose g is the generator of group $\mathbb{G}$, and a, b, c, z are random numbers selected from $\mathbb{Z}_p$. The DBDH assumption holds if given a tuple $(A, B, C, Z) = (g^a, g^b, g^c, Z)$, there is no algorithm $\mathcal{B}$ can distinguish $Z = e(g,g)^{abc}$ or $Z = e(g,g)^z$ in polynomial-time with non-negligible advantage. $\mathcal{B}$'s advantage is defined as:

$$Adv_{\mathcal{B}}^{DBDH} = |\Pr[\mathcal{B}(A, B, C, e(g,g)^{abc}) = 1] - \Pr[\mathcal{B}(A, B, C, e(g,g)^z) = 1]|.$$

2.3 Access Structure

The AND-gate on multi-valued attributes is applied in this scheme as an access policy.

Suppose $\mathcal{U} = [a_1, a_2, \cdots, a_n]$ is an attribute set, each attribute $a_i \in \mathcal{U}$ has n_i possible values $V_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,n_i}\}$. Let $S = [S_1, S_2, \cdots, S_n]$ be a user's attribute set where $S_i \in V_i$, and $W = [W_1, W_2, \cdots, W_n]$ be an access structure where $W_i \in V_i$. The symbol $S \models W$ indicates that the attribute set S satisfies the access structure W (namely, $S_i = W_i$), and $S \not\models W$ indicates the opposite.

2.4 Commitment

The commitment scheme adopted in this construction is the Pedersen commitment scheme [24], which is a perfectly hiding scheme. This scheme is stated as follows. Assume $\mathbb{G}$ is a prime-order group with generators $g_0, g_1, \cdots, g_k$. To commit messages $(m_1, m_2, \cdots, m_k)$, the user picks $r \in_R \mathbb{Z}_p$, and calculates $T = g_0^r \prod_{j=1}^{k} g_j^{m_k}$. The number r is used by the user to decommit the commitment T when needed.

2.5 Zero-Knowledge Proof

Zero knowledge proof (ZKP) is a kind of interactive proof that the prover proves some knowledge to the verifier without leaking them. The ZKP scheme proposed by Camenisch and Stadler [3] is briefly described as follows. Take the following formula as an example,

$$PoK\{(\alpha, \beta, \gamma) : y = g^\alpha h^\beta \wedge y_0 = g_0^\alpha h_0^\gamma\},$$

where $\{g, h\}$ and $\{g_0, h_0\}$ are generators of group $\mathbb{G}$ and $\mathbb{G}_0$, respectively. The values α, β and γ are knowledge that need to be proven, the remaining values are used by the checker to verify the equations.

3 Definition and Security Model

3.1 System Model

The system architecture is shown in Figure 2, where there are 5 entities: Data Owner, N Attribute Authorities, Cloud Storage Server, Proxy Server and Data User.

Data Owner: The owner encrypts data file under his/her own specified access policy, then upload the ciphertext to the cloud server.

Attribute Authorities: Authorities are responsible for generating secret keys for users, which are independent of each other and manage disjoint attribute sets. The authorities are not entirely credible as they will try to collude with other entities to gather useful information in order to obtain illegal profits.

Cloud Storage Server: The server is assumed to be an honest-but-curious entity that stores the encrypted data file uploaded by data owners and provides access channel for the users. It acts normally in most of time but may attempt to collect as much information as possible.

Proxy Server: The proxy server owns strong computing power and it only provides partial decryption computing service for users.

Data User: The user can issue secret key queries to the authorities and download any ciphertext on the cloud server. However, only when the user's attributes satisfy the access structure can the data be recovered. Users are assumed to be distrusted, and they have a tendency to conspire with other entities to illegally access the data file.

3.2 Outline of Decentralized ABE Scheme

Let $A_1, A_2, \cdots, A_N$ represent N attribute authorities, and each authority A_k monitors a disjoint attributes set $\tilde{A}_k$, namely, $\tilde{A}_k \cap \tilde{A}_j = \emptyset$ $(k, j \in [1, N] \wedge k \neq j)$. $\tilde{L}$ represents a list of attributes for the user U.

A decentralized ABE scheme has five algorithms as follows:

Global Setup(1^λ): Input λ as a security parameter, the algorithm outputs the system parameters PP.

Authority Setup(PP): Taking system parameters PP as input, each authority A_k executes this algorithm to generate its public-secret key pair (PK_k, SK_k).

KeyGen($PP, SK_k, GID, \tilde{A}_u^k$): Input the system parameters PP, secret keys SK_k, user's GID and attributes set $\tilde{A}_u^k$, where $\tilde{A}_u^k = \tilde{A}_k \bigcap \tilde{L}$, authority A_k performs this algorithm and returns secret keys SK_U^k for the user.

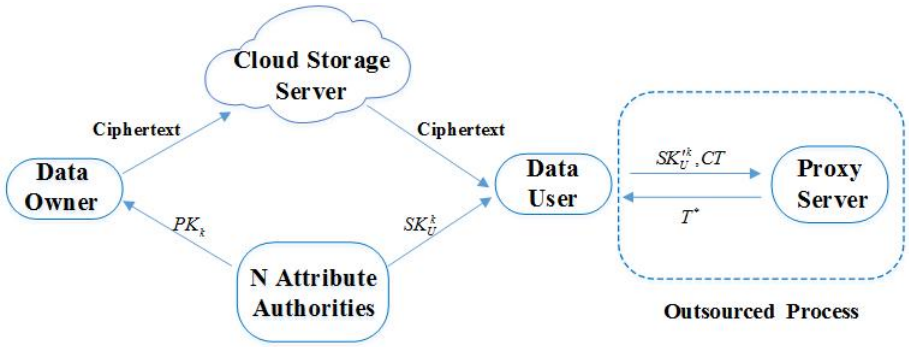

Figure 2: System model of the proposed scheme

Encrypt$(PP, PK_k, \mathcal{M}, W)$**:** With the system parameters PP, public keys PK_k, message $\mathcal{M}$ and access policy W as input, the encryption algorithm outputs CT as the corresponding ciphertext.

Decrypt(PP, GID, SK_U^k, CT)**:** This algorithm consists of two phases: *Pre-Decrypt* and *User-Decrypt*. The first phase is performed by the proxy server and the second is executed by the data user. The user first modifies the obtained secret keys SK_U^k to construct new keys $SK_U'^k$, and then sends the $(SK_U'^k, CT)$ to the proxy server for partial decryption calculations. Finally, the user performs the remaining decryption calculations with the results T^* returned by the proxy server.

3.3 Security Model

The security model is defined by a security game between adversary $\mathcal{A}$ and challenger $\mathcal{B}$.

Initialization: $\mathcal{A}$ provides an access policy $\mathcal{W}^*$ that he/she wants to challenge and a list of corrupted authorities $C_{\mathcal{A}}$ ($|C_{\mathcal{A}}| < N$) to $\mathcal{B}$.

Global Setup: $\mathcal{B}$ executes this algorithm and returns system parameters PP to $\mathcal{A}$.

Authority Setup: Two cases are discussed here:

1) For the corrupted authorities, namely $A_k \in C_{\mathcal{A}}$, $\mathcal{B}$ performs the authority setup algorithm and returns the public-secret key pair (PK_k, SK_k) to $\mathcal{A}$.
2) For the uncorrupted authorities, namely $A_k \notin C_{\mathcal{A}}$, $\mathcal{B}$ performs the authority setup algorithm and sends public keys PK_k to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ initiates a private key query to $\mathcal{B}$ by submitting a list of attributes $\tilde{L}$, and the only limitation is that the list of attributes submitted does not meet the access policy to be challenged, namely, $\tilde{L} \not\models \mathcal{W}^*$. Then $\mathcal{B}$ executes the algorithm KeyGen and outputs the secret keys to $\mathcal{A}$ correspondingly.

Challenge: The adversary $\mathcal{A}$ provides $\mathcal{M}_0$ and $\mathcal{M}_1$ to challenger $\mathcal{B}$, where $|\mathcal{M}_0| = |\mathcal{M}_1|$. Then $\mathcal{B}$ picks a random number $\xi \in \{0, 1\}$. The encryption algorithm is executed to encrypt the message $\mathcal{M}_\xi$ under the access policy $\mathcal{W}^*$ and outputs the ciphertext CT^* accordingly. Then $\mathcal{B}$ returns ciphertext CT^* to adversary $\mathcal{A}$.

Phase 2: Same as Phase 1.

Guess: $\mathcal{A}$ returns ξ' as a guess on ξ. If $\xi' = \xi$, $\mathcal{A}$ wins the game.

Definition 1. *If there is no t-time adversary who can make q secret key queries to break through the above game with a non-negligible advantage, then the scheme is* (t, q, ϵ) *secure in the selective model.*

4 Scheme Construction

4.1 Decentralized ABE Scheme

Global Setup(1^λ)**:** Taking the security parameter λ as input, this algorithm produces two cyclic groups $\mathbb{G}, \mathbb{G}_T$ with same order $N = pp_1$, where p and p_1 are two different primes, and a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Suppose that $\mathbb{G}_p$ and $\mathbb{G}_{p_1}$ are subgroups of $\mathbb{G}$ with order p and p_1, respectively. Let g_{p_1} be a generator of $\mathbb{G}_{p_1}$ and g_p, g_0, g_1 be generators of $\mathbb{G}_p$. $H : \{0,1\}^* \to \mathbb{Z}_N$ is a hash function. Public parameters are $PP = (e, p, p_1, g_p, g_0, g_1, g_{p_1}, H, \mathbb{G}, \mathbb{G}_T)$.

Authority Setup(PP)**:** Let $\tilde{A}_k$ be an attribute set monitored by authority A_k. Each A_k randomly selects $\alpha_k, \beta_k, a_{k,i,j} \in_R \mathbb{Z}_N$ and $R_k, R_{k,i,j} \in_R \mathbb{G}_{p_1}$ ($j = [1, n_i]$). Then A_k computes $Y_k = e(g_p, g_p)^{\alpha_k}, Z_k = g_p^{\beta_k} \cdot R_k$ and $A_{k,i,j} = g_p^{a_{k,i,j}} \cdot R_{k,i,j}$ as its public keys, namely $PK_k = (Y_k, Z_k, \{A_{k,i,j}\}_{j=[1,n_i]})$. The secret keys are $SK_k = (\alpha_k, \beta_k, \{a_{k,i,j}\}_{j=[1,n_i]})$.

KeyGen$(PP, SK_k, GID, \tilde{A}_u^k)$**:** Let $u = H(GID)$. Input PP, SK_k and user information $(u, \tilde{L})$, A_k performs the PPKeyGen algorithm to produce the decryption keys as follows. For $v_{k,i,j} \in \tilde{A}_u^k$, A_k picks $t_{k,i,j}^u \in_R \mathbb{Z}_N^*$ and sets $t_{k,u} = \sum t_{k,i,j}^u$. Then, A_k calculates:

$$D_{k,u} = g_p^{\alpha_k} g_0^{\frac{t_{k,u}}{\beta_k + u}} g_1^{u\beta_k}, D_{k,i,j} = g_0^{\frac{t_{k,i,j}^u}{(\beta_k + u)a_{k,i,j}}}$$

where $\tilde{A}_u^k = \tilde{A}_k \bigcap \tilde{L}$, $\tilde{L}$ represents an attribute list of the user.

Then, A_k outputs the secret keys

$$SK_U^k = (D_{k,u}, \{D_{k,i,j}\}_{v_{k,i,j} \in \tilde{A}_u^k}).$$

Encrypt$(PP, PK_k, \mathcal{M}, W)$**:** The owner employs the existing symmetric encryption algorithm to encrypt the data file M, and gets $CT' = Enc_{\mathcal{K}}(M)$, where $\mathcal{K}$ is the symmetric key. The decentralized ABE scheme is then applied to encrypt $\mathcal{K}$ under the access structure W defined by the encryptor. Finally, the owner uploads the ciphertext $\{CT, CT', CT''\}$ to the cloud server. Algorithm 1 gives the specific encryption steps. I_c represents an index set of relevant authorities A_k.

Algorithm 1 Encrypt

1: Begin
2: **Input** $\{PP, PK_k, M, \mathcal{K}, W\}$.
3: Select symmetric encryption algorithm Enc and key $\mathcal{K}$.
4: **for** data file M **do**
5: $CT' = Enc_{\mathcal{K}}(M)$.
6: **end for**
7: **for** symmetric key $\mathcal{K}$ **do**
8: $CT'' = g^{H(\mathcal{K})}$.
9: select $s \in_R \mathbb{Z}_N^*, R_1, R_2 \in_R \mathbb{G}_{p_1}$.
10: $C = \mathcal{K} \prod_{k \in I_c} Y_k^s, C_1 = g_p^s \cdot R_1, C_2 = (\prod_{k \in I_c} Z_k^s) \cdot R_2$.
11: **for** each attribute $v_{k,x,y} \in W$ **do**
12: select $R'_{k,x,y} \in_R \mathbb{G}_{p_1}$.
13: $C_{k,x,y} = A_{k,x,y}^s \cdot R'_{k,x,y}$.
14: **end for**
15: $CT = (C, C_1, C_2, \{C_{k.x,y}\}_{v_{k,x,y} \in W})$.
16: **end for**
17: **Output** $\{CT, CT', CT''\}$.
18: End

Decrypt(PP, GID, SK_U^k, CT)**:** Any user can download the ciphertext $\{CT, CT', CT''\}$ from the cloud server for decryption. To reduce the computational burden, the user can outsource some decryption operations to the proxy server. The decryption algorithm has two phases: *Pre-Decrypt* and *User-Decrypt*. The user first chooses $z \in_R \mathbb{Z}_N^*$, and then calculates the new keys NK. Then, the user sends (CT, NK) to the proxy server and keeps z secret.

Pre-Decrypt: This phase is executed by the proxy server. Input (CT, NK), the server performs partial decryption calculation and sends the result T^* to the user.

User-Decrypt: The user decrypts $\mathcal{K}$ through the secret value z and the result T^* returned by the server. Then, user uses the symmetric key $\mathcal{K}$ to get M.

The detail steps are shown in algorithm 2. In the check phase, the equation $g^{H(\mathcal{K})} \stackrel{?}{=} CT''$ is used to verify the correctness of the symmetric key $\mathcal{K}$. Only the user whose attributes satisfy the access policy can get the correct $\mathcal{K}$ and further run the symmetric decryption algorithm Dec to obtain the correct data file M. It can be known from the algorithm that user only needs to perform one exponential operation and one pairing operation to obtain the symmetric secret key $\mathcal{K}$, thus greatly reducing the computing cost of user.

Algorithm 2 Decrypt

1: Begin
2: **Input** $\{PP, SK_U^k, \{CT, CT', CT''\}\}$.
3: User selects a random number $z \in \mathbb{Z}_N^*$.
4: Calculate $SK_U'^k = (D'_{k,u}, D'_{k,i,j}) = (D_{k,u}^{1/z}, D_{k,i,j}^{1/z})$.
5: Set $NK = (SK_U'^k)$.
6: User sends (CT, NK) to the proxy server.
7: **Pre-Decrypt:**
8: Calculate $T^* = \frac{\prod_{k \in I_c} e(D'_{k,u}, C_1)}{\prod_{v_{k,x,y} \in W} e(D'_{k,i,j}, C_{k,x,y})}$.
9: Send T^* to the user.
10: **User-Decrypt:**
11: Calculate $B = e(C_2, g_1^u), \mathcal{K} = CB/(T^*)^z$.
12: Check $g^{H(\mathcal{K})} \stackrel{?}{=} CT''$
13: **if** $g^{H(\mathcal{K})} = CT''$ **then**
14: $M = Dec_{\mathcal{K}}(CT')$.
15: **else**
16: return $\perp$.
17: **end if**
18: **Output** M or $\perp$.
19: End

Correctness.

$$\begin{aligned} T^* &= \frac{\prod_{k \in I_c} e(D'_{k,u}, C_1)}{\prod_{v_{k,x,y} \in W} e(D'_{k,i,j}, C_{k,x,y})} \\ &= \frac{\prod_{k \in I_c} e((g_p^{\alpha_k} g_0^{\frac{t_{k,u}}{\beta_k + u}} g_1^{u\beta_k})^{1/z}, g_p^s \cdot R_1)}{\prod_{v_{k,x,y} \in W} e((g_0^{\frac{t_{k,i,j}^u}{(\beta_k+u)a_{k,i,j}}})^{1/z}, g_p^{sa_{k,x,y}} R_{k,x,y}^s R'_{k,x,y})} \\ &= \frac{\prod_{k \in I_c} (e(g_p, g_p)^{\alpha_k s} e(g_p, g_0)^{s \frac{t_{k,u}}{\beta_k + u}} e(g_p, g_1)^{su\beta_k})^{1/z}}{\prod_{k \in I_c} (e(g_p, g_0)^{s \frac{t_{k,u}}{\beta_k + u}})^{1/z}} \\ &= \prod_{k \in I_c} e(g_p, g_p)^{\alpha_k s/z} e(g_p, g_1)^{su\beta_k/z} \end{aligned}$$

$$B = e(C_2, g_1^u) = e(\prod_{k \in I_c} g_p^{\beta_k s} R_k^s \cdot R_2, g_1^u) = \prod_{k \in I_c} e(g_p, g_1)^{su\beta_k}$$

$$\mathcal{K} = BC/(T^*)^z = \frac{\mathcal{K} \cdot \prod_{k \in I_c} e(g_p, g_p)^{\alpha_k s} \cdot e(g_p, g_1)^{su\beta_k}}{(\prod_{k \in I_c} e(g_p, g_p)^{\alpha_k s/z} e(g_p, g_1)^{su\beta_k/z})^z}$$

Algorithm 3 PPKeyGen Protocol

Begin
U selects $u, \rho_0 \in_R \mathbb{Z}_N$, A_k selects $\rho_1, \beta_k \in_R \mathbb{Z}_N$
$U(u, \rho_0) \xleftrightarrow{2PC} A_k(\rho_1, \beta_k) : \eta = (u + \beta_k)\rho_0\rho_1 \bmod N$
U selects $z^*, z_1, z_2, z_3 \in_R \mathbb{Z}_N$, and computes $T = g_p^{z^*} g_1^u$, $P_0 = g_0^{\rho_0}, T' = g_p^{z_1} g_1^{z_2}$, $P_0' = g_0^{z_3}$
U returns (T, P_0, T', P_0') to A_k
A_k picks $c \in_R \mathbb{Z}_N$ and sends c to U
U sets $a_1 = z_1 - cz^*, a_2 = z_2 - cu, a_3 = z_3 - c\rho_0$ and sends (a_1, a_2, a_3) to A_k
A_k checks the zero-knowledge proof
if $T' = g_p^{a_1} g_1^{a_2} T^c$ and $P_0' = g_0^{a_3} P_0^c$ **then**
    $\forall v_{k,i,j} \in \tilde{A}_u^k$, A_k selects $t_{k,i,j}^u \in_R \mathbb{Z}_N$ and sets $t_{k,u} = \sum t_{k,i,j}^u$
    A_k selects $y_1, y_2, y_3, y_4, y_5 \in_R \mathbb{Z}_N$ and computes $P_1 = g_0^{\rho_1}$, $P_1' = g_0^{y_4}$, $Z_k' = g_p^{y_2}$, $\tilde{D}_{k,i,j} = P_1^{\frac{t_{k,i,j}^u}{\eta a_{k,i,j}}}$, $\tilde{D}_{k,u} = g_p^{\alpha_k} T^{\beta_k} P_0^{\frac{\rho_1 t_{k,u}}{\eta}}$, $\tilde{D}_{k,i,j}' = P_1^{y_5}$, $\tilde{D}_{k,u}' = g_p^{y_1} T^{y_2} P_0^{y_3}$
    A_k sends $(P_1, \tilde{D}_{k,u}, \tilde{D}_{k,i,j}, P_1', \tilde{D}_{k,u}', \tilde{D}_{k,i,j}', Z_k')$ to U
else
    Abort
end if
U chooses $c' \in_R \mathbb{Z}_N$, and sends c' to A_k
A_k sets $y_1' = y_1 - c'\alpha_k, y_2' = y_2 - c'\beta_k, y_3' = y_3 - c'\frac{t_{k,u}\rho_1}{\eta}, y_4' = y_4 - c'\rho_1, y_5' = y_5 - c'\frac{t_{k,i,j}^u}{\eta a_{k,i,j}}$
A_k sends $(y_1', y_2', y_3', y_4', y_5')$ to U.
U checks the proof
if $P_1' = g_0^{y_4'} P_1^{c'}$, $\tilde{D}_{k,u}' = g_p^{y_1'} T^{y_2'} P_0^{y_3'} (\tilde{D}_{k,u})^{c'}$, $\tilde{D}_{k,i,j}' = \tilde{D}_{k,i,j}^{c'} P_1^{y_5'}$ and $Z_k' = g_p^{y_2'} (Z_k)^{c'}$ **then**
    U computes $D_{k,u} = \frac{\tilde{D}_{k,u}}{Z_k^{z^*}}, D_{k,i,j} = \tilde{D}_{k,i,j}^{\rho_0}$, where $Z_k = g_p^{\beta_k}$
else
    Abort
end if
End

4.2 Privacy-Preserving Key Generation Protocol

Each user has a unique global identifier that distinguishes them from each other, if the GID is exposed to a malicious party, this will directly endanger the privacy. To achieve GID hidden, the knowledge of ZKP and commitment scheme can be utilized to conduct a privacy-preserving key generation protocol between the relevant authority and the user. Without revealing GID, user can prove his/her legitimacy to the authority and get the secret keys generated by the authority. The PPKeyGen algorithm is presented as follows.

1) The user U selects $u, \rho_0 \in \mathbb{Z}_N$, and the authority A_k selects $\rho_1, \beta_k \in \mathbb{Z}_N$. Then A_k executes the 2-party secure computing (2PC) protocol with U and gets $\eta = (u + \beta_k)\rho_0\rho_1 \bmod N$.

2) U picks $z^* \in_R \mathbb{Z}_N$ and runs the *Commit* algorithm on the GID u. Let T be the commitment value. U computes (T, P_0, T', P_0') and sends them to A_k. Moreover, U sends $PoK\{(u, \rho_0, z^*) : T = g_p^{z^*} g_1^u \wedge P_0 = g_0^{\rho_0}\}$ to A_k to anonymously prove that he/she has knowledge (u, ρ_0, z^*).

3) A_k checks the proof first. If the proof is correct, then A_k computes $(P_1, \tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ and sends them to U. Otherwise, A_k will terminate the algorithm. Then, A_k sends $PoK\{(\alpha_k, \beta_k, \rho_1, t_{k,u}, a_{k,i,j}) : \tilde{D}_{k,u} \wedge \tilde{D}_{k,i,j} \wedge P_1\}$ to U to anonymously prove that it has knowledge $(\alpha_k, \beta_k, \rho_1, t_{k,u}, a_{k,i,j})$.

4) Similarly, A_k checks the zero-knowledge proof. If this proof works correctly, U can compute $D_{k,u}$ and $D_{k,i,j}$. Otherwise, U stops this algorithm.

Algorithm 3 illustrates the specific steps.

The PPKeyGen algorithm needs to meet two features: *leak-freeness* and *selective-failure blindness*. In the first one, a malicious user running the PPKeyGen algorithm with trusted authorities would not gain more information than executing the KeyGen algorithm. The second means that a malicious authority cannot get any information about the user's GID, nor can it fail the PPKeyGen algorithm based on the user's selection of GID. The definitions of the two features are shown below.

Definition 2. *(leak-freeness). A PPKeyGen algorithm is leak-free if for all efficient adversaries $\mathfrak{U}$, there exists a simulator $\mathfrak{U}'$ such that no efficient distinguisher $\mathfrak{D}$ can distinguish whether $\mathfrak{U}$ is executing in the real game or in the ideal game with non-negligible advantage. The two games are defined as follows:*

Real Game: The distinguisher $\mathfrak{D}$ runs the setup algorithm as many times as it wants, the malicious user $\mathfrak{U}$ selects a GID u and executes the PPKeyGen algorithm with the honest authority.

Ideal Game: The distinguisher $\mathfrak{D}$ runs the setup algorithm as many times as it wants, and the simulator $\mathfrak{U}'$ chooses a GID u' and executes the KeyGen algorithm with a trusted authority.

Definition 3. *(selective-failure blindness). A PPKeyGen algorithm is selective-failure blind if no probably polynomial time adversary $\mathcal{A}_k$ can win the following game with non-negligible advantage.*

1) The malicious authority $\mathcal{A}_k$ submits public key PK_k and two global identifiers u_0, u_1.

2) A random bit $b \in \{0, 1\}$ is selected.

3) $\mathcal{A}_k$ is given two comments com_b and com_{1-b}, then it can black-box access oracles $U(PP, u_b, PK_k, com_b)$ and $U(PP, u_{1-b}, PK_k, com_{1-b})$.

4) The algorithm U returns the secret keys $SK_{U_b}^k$ and $SK_{U_{1-b}}^k$, separately.

5) If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} \neq \perp$, $\mathcal{A}_k$ is given $(SK^k_{U_b}, SK^k_{U_{1-b}})$; If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} = \perp$, $\mathcal{A}_k$ is given $(\epsilon, \perp)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} \neq \perp$, $\mathcal{A}_k$ is given $(\perp, \epsilon)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} = \perp$, $\mathcal{A}_k$ is given $(\perp, \perp)$.

6) Finally, $\mathcal{A}_k$ returns its guess b' on b. $\mathcal{A}_k$ wins the game if $b' = b$.

5 Security Analysis

5.1 Scheme Analysis

Identity Privacy: The PPKeyGen algorithm introduced in this paper is an interactive process between the user and each authority, which ensures that the user can obtain the correct secret key from the authority without directly providing the unencapsulated u. In this case, the identifier u is confidential to all authorities, so they can no longer track it to aggregate user's information. **Theorem 2** in Section 5.2 shows that this protocol satisfies *leak-freeness* and *selective-failure blindness*.

Collusion Resistance: In this construction, the authors bind all the user's key components to u, making the user's secret key unique. The power settings of g_0 and g_1 (namely, $\frac{t_{k,u}}{\beta_k + u}$ and $u\beta_k$, respectively) make it impossible for different users to achieve effective attacks by combining their secret keys. What's more, the scheme can prevent the collusion attack of multiple authorities. By observing the ciphertext component C in CT (namely, $\mathcal{K}\prod_{k\in I_c} e(g_p, g_p)^{\alpha_k s}$), it can be known that only when all relevant α_k is obtained can the message $\mathcal{K}$ be restored. Therefore, as long as one authority is honest in the set of authority involved, the attack will not succeed.

Hidden Policy: The authors implement attributes anonymity in the access policy by applying some random elements $(R_1, R_2, R'_{k,x,y})$ in group G_{p_1} on some ciphertext components $(C'_1, C_2, C_{k,x,y})$ [20,33]. Such a setting is very necessary. The existence of these random elements makes it difficult for an attacker to easily guess the value of the attribute embedded by the encryptor in the access policy. Without the introduction of these random elements, the original ciphertext would be $C'_1 = g_p^s, C'_2 = (\prod_{k\in I_c} {Z'_k}^s), C'_{k,x,y} = A'^s_{k,x,y}$, where $Z'_k = g_p^{\beta_k}, A'_{k,i,j} = g_p^{a_{k,i,j}}$. An attacker needs only a simple test $e(A'_{k,i,j}, C'_1) \stackrel{?}{=} e(g_p, C'_{k,x,y})$ to be able to guess whether the encryptor has embedded the attribute value $v_{k,i,j}$ in the access policy or not. In fact,

$$X' = e(A'_{k,i,j}, C'_1) = e(g_p^{a_{k,i,j}}, g_p^s) = e(g_p, g_p)^{sa_{k,i,j}}$$

$$Y' = e(g_p, C'_{k,x,y}) = e(g_p, g_p^{sa_{k,x,y}}) = e(g_p, g_p)^{sa_{k,x,y}}$$

If $X' = Y'$, namely, $a_{k,i,j} = a_{k,x,y}$, means that $v_{k,i,j} \in W$; If $X' \neq Y'$, namely, $a_{k,i,j} \neq a_{k,x,y}$, means that $v_{k,i,j} \notin W$.

Do the same operations on the proposed scheme, then

$$\begin{aligned} X = e(A_{k,i,j}, C_1) &= e(g_p^{a_{k,i,j}} \cdot R_{k,i,j}, g_p^s \cdot R_1) \\ &= e(g_p, g_p)^{sa_{k,i,j}} e(R_{k,i,j}, R_1) \\ Y = e(g_p, C_{k,x,y}) &= e(g_p, g_p^{sa_{k,x,y}} \cdot R^s_{k,x,y} \cdot R'_{k,x,y}) \\ &= e(g_p, g_p)^{sa_{k,x,y}} \end{aligned}$$

An attacker cannot determine whether the attribute value $v_{k,i,j}$ belongs to W by comparing the values of X and Y. Therefore, the privacy of the recipient is protected to a certain extent.

5.2 Security Proof

Theorem 1. *The proposed scheme is secure in the selective access policy model, assuming that the DBDH assumption holds.*

Proof. Assuming that the proposed scheme can be broken by an adversary $\mathcal{A}$ with a non-negligible advantage ϵ, then, using the ability of $\mathcal{A}$, a simulator $\mathcal{B}$ can be constructed to solve the DBDH problem with the advantage $\epsilon/2$. □

Firstly, the challenger generates a bilinear group $(e, p, p_1, \mathbb{G}, \mathbb{G}_T)$, where $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_{p_1}$. Then he/she picks a random number $\varphi \in \{0, 1\}$, and sets:

$$\begin{cases} Z = e(g_p, g_p)^{abc}, \varphi = 0 \\ Z = e(g_p, g_p)^z, \varphi = 1 \end{cases} \tag{1}$$

where, z is a random number in group $\mathbb{G}$. The simulator gets the challenge tuple $(g_p, g_{p_1}, A, B, C, Z) = (g_p, g_{p_1}, g_p^a, g_p^b, g_p^c, Z)$ from the challenger and finally returns a guess φ' on φ.

Initialization: The adversary $\mathcal{A}$ provides a set of corrupted authorities C_A and an access structure $\mathcal{W}^*$ which he/she wants to challenge. Suppose there is at least one completely honest authority A^* in the security game.

Global Setup: $\mathcal{B}$ randomly chooses $\gamma, \theta \in_R \mathbb{Z}_N$ and sets $g_0 = A \cdot g_p^\gamma = g_p^{a+\gamma}, g_1 = g_p^\theta$. Then, $\mathcal{B}$ sends $PP = (e, p, p_1, g_p, g_0, g_1, g_{p_1}, H, \mathbb{G}, \mathbb{G}_T)$ to adversary $\mathcal{A}$.

Authority Setup: Three cases are discussed here:

1) For the corrupted authorities $A_k \in C_A$, $\mathcal{B}$ randomly selects $\alpha_k, \beta_k, a_{k,i,j} \in_R \mathbb{Z}_N$ and $R_k, R_{k,i,j} \in_R \mathbb{G}_{p_1}$, then calculates $Y_k = e(g_p, g_p)^{\alpha_k}, Z_k = g_p^{\beta_k} \cdot R_k$ and $A_{k,i,j} = g_p^{a_{k,i,j}} \cdot R_{k,i,j}$. Then, $\mathcal{B}$ sends the secret keys $SK_k = (\alpha_k, \beta_k, \{a_{k,i,j}\}_{j=[1,n_i]})$ and the public keys $PK_k = (Y_k, Z_k, \{A_{k,i,j}\}_{j=[1,n_i]})$ to adversary $\mathcal{A}$.

2) For the authority A_k not corrupted and not A^*, $\mathcal{B}$ randomly chooses $\alpha_k, \beta_k, a_{k,i,j} \in_R \mathbb{Z}_N$ and $R_k, R_{k,i,j} \in_R \mathbb{G}_{p_1}$, computes $Y_k = e(g_p, g_p)^{b\alpha_k}, Z_k = g_p^{\beta_k} \cdot R_k, A_{k,i,j} = g_p^{a_{k,i,j}} \cdot R_{k,i,j}$, when $v_{k,i,j} \in \mathcal{W}^*$, or $A_{k,i,j} = g_p^{ba_{k,i,j}} \cdot R_{k,i,j}$, when $v_{k,i,j} \notin \mathcal{W}^*$. $\mathcal{B}$ sends the public keys $PK_k = (Y_k, Z_k, \{A_{k,i,j}\}_{j=[1,n_i]})$ to adversary $\mathcal{A}$.

3) For the authority A^*, $\mathcal{B}$ randomly selects β^*, $a^*_{i,j} \in_R \mathbb{Z}_N$ and $R^*, R^*_{i,j} \in_R \mathbb{G}_{p_1}$, then computes $Y^* = e(g_p^a, g_p^b) \cdot \prod_{A_k \in C_A} e(g_p, g_p)^{-\alpha_k} \cdot \prod_{A_k \notin C_A \cup A^*} e(g_p, g_p)^{-b\alpha_k}$ and $Z^* = g_p^{\beta^*} \cdot R^*$. $A^*_{i,j} = g_p^{a^*_{i,j}} \cdot R^*_{i,j}$, if $v_{i,j} \in \mathcal{W}^*$; $A^*_{i,j} = g_p^{ba^*_{i,j}} \cdot R^*_{i,j}$, if $v_{i,j} \notin \mathcal{W}^*$. Then, $\mathcal{B}$ sends the public keys $PK^* = (Y^*, Z^*, \{A^*_{i,j}\}_{j=[1,n_i]})$ to adversary $\mathcal{A}$.

Phase 1: The authority $\mathcal{A}$ issues a secret keys query for global identifier u^* with a list of attributes L^*, where $L^* \not\models \mathcal{W}^*$.

1) For $A_k \in C_A$, $\mathcal{A}$ can generate the user secret keys directly by himself.

2) For $A_k \notin C_A \cup A^*$, $\forall v_{k,i,j} \in \tilde{A}^k_{u^*}$, $\mathcal{B}$ picks $t^{u^*}_{k,i,j} \in_R \mathbb{Z}_N$ and sets $t_{k,u^*} = \sum t^{u^*}_{k,i,j}$. Then, $\mathcal{B}$ computes $D_{k,u^*} = B^{\alpha_k} g_0^{\frac{t_{k,u^*}}{\beta_k + u^*}} g_1^{u^*\beta_k}, D_{k,i,j} = g_0^{\frac{t^{u^*}_{k,i,j}}{(\beta_k + u^*)a_{k,i,j}}}$.

3) For $A_k = A^*$, $\forall v_{i,j} \in \tilde{A}^*_{u^*}$, $\mathcal{B}$ chooses $t^{u^*}_{i,j} \in_R \mathbb{Z}_N$, sets $t_{u^*} = \sum t^{u^*}_{i,j}$, and computes $D^*_{u^*} = B^{-\gamma} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} \prod_{A_k \in C_A} g_p^{-\alpha_k} \prod_{A_k \notin C_A \cup A^*} B^{-\alpha_k}$ and $D^*_{i,j} = g_0^{\frac{t^{u^*}_{i,j}}{(\beta^* + u^*)a_{i,j}}}$. Where,

$$\begin{aligned} D^*_{u^*} &= B^{-\gamma} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} \prod_{A_k \in C_A} g_p^{-\alpha_k} \prod_{A_k \notin C_A \cup A^*} B^{-\alpha_k} \\ &= g_p^{-b\gamma} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} g_p^{-(\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k)} \\ &= g_p^{ab - (\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k) - b(a+\gamma)} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} \\ &= g_p^{ab - (\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k)} g_0^{\frac{t_{u^*}}{\beta^* + u^*} - b} g_1^{u^*\beta^*} \end{aligned}$$

Let $t'_{u^*} = t_{u^*} - b(\beta^* + u^*)$, then

$$D^*_{u^*} = g_p^{ab - (\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k)} g_0^{\frac{t'_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*}$$

Therefore, $D^*_{u^*}$ is a valid secret key.

Challenge: The adversary $\mathcal{A}$ provides two messages of the same length $\mathcal{K}_0$ and $\mathcal{K}_1$. $\mathcal{B}$ chooses a random bit $\xi \in \{0,1\}$ and then encrypts the message $\mathcal{K}_\xi$: $CT^* = \{C^* = \mathcal{K}_\xi \cdot Z, C^*_1 = g^c \cdot R_1, C^*_2 = (\prod_{k \in I_c} g_p^{\beta_k c} \cdot R_k^c) \cdot R_2, \forall v_{k,x,y} \in \mathcal{W}^* : C^*_{k,x,y} = g_p^{ca_{k,x,y}} \cdot R^c_{k,x,y} \cdot R'_{k,x,y}\}$.

Phase 2: Same as Phase 1.

Guess: Adversary $\mathcal{A}$ returns the guess ξ' on ξ. If $\xi' = \xi$, simulator $\mathcal{B}$ returns $\varphi' = 0$ to the challenger; otherwise, simulator $\mathcal{B}$ returns $\varphi' = 1$.

If $\varphi = 1$, $Z = e(g_p, g_p)^z$ is a random value, adversary $\mathcal{A}$ cannot get any information about ξ, so $Pr[\xi' \neq \xi \mid \varphi = 1] = \frac{1}{2}$. Since $\mathcal{B}$ returns $\varphi' = 1$ when $\xi' \neq \xi$, thus $Pr[\varphi' = \varphi \mid \varphi = 1] = \frac{1}{2}$.

If $\varphi = 0$, then $Z = e(g_p, g_p)^{abc}$, the adversary $\mathcal{A}$ will get a valid ciphertext of message $\mathcal{K}_\xi$. The advantage of $\mathcal{A}$ in breaking the proposed scheme is ϵ (non-negligible) by definition, so $Pr[\xi' = \xi \mid \varphi = 0] = \frac{1}{2} + \epsilon$. Since $\mathcal{B}$ returns $\varphi' = 0$ when $\xi' = \xi$, thus $Pr[\varphi' = \varphi \mid \varphi = 0] = \frac{1}{2} + \epsilon$.

Therefore, the advantage of $\mathcal{B}$ to break the DBDH assumption is $| \frac{1}{2} Pr[\varphi' = \varphi \mid \varphi = 0] + \frac{1}{2} Pr[\varphi' = \varphi \mid \varphi = 1] - \frac{1}{2} | = \epsilon/2$ (non-negligible).

Theorem 2. *The proposed PPKeyGen algorithm is leak-free and selective-failure blind.*

Proof. (*Leak-freeness*) Suppose that there is a malicious user $\mathfrak{U}$ runs the PPKeyGen algorithm with an honest A_k in the real game. There should also exist a simulator $\tilde{\mathfrak{U}}$ runs the KeyGen algorithm with a trusted authority in the ideal game such that no distinguisher $\mathfrak{D}$ can effectively distinguish the two games. The simulator $\tilde{\mathfrak{U}}$ can simulate the communication between $\mathfrak{D}$ and $\mathfrak{U}$. $\tilde{\mathfrak{U}}$ works as follows:

1) $\tilde{\mathfrak{U}}$ sends PP and the public-key PK_k of authority A_k to the malicious user $\mathfrak{U}$.

2) The $\mathfrak{U}$ needs to prove to $\tilde{\mathfrak{U}}$ that he/she owns u in zero-knowledge by submitting two values (T, P_0). If the proof succeeds, then $\tilde{\mathfrak{U}}$ will gets (u, ρ_0, z^*) using rewind technique.

3) $\tilde{\mathfrak{U}}$ sends u to the trusted party and gets secret keys $(D_{k,u}, D_{k,i,j})$.

4) $\tilde{\mathfrak{U}}$ chooses $\rho \in_R \mathbb{Z}_p$, and calculates $\rho_1 = \rho/\rho_0, P_1 = g_0^{\rho_1}, \tilde{D}_{k,u} = D_{k,u} \cdot Z_k^{z^*}, \tilde{D}_{k,i,j} = D_{k,i,j}^{1/\rho_0}$. Then $\tilde{\mathfrak{U}}$ returns $(P_1, \tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ to $\mathfrak{U}$.

□

If $(D_{k,u}, D_{k,i,j})$ are correct keys from the trusted AA in the ideal game, then $(\tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ are correct keys from A_k in the real game. The distinguisher $\mathfrak{D}$ cannot distinguish the real game with the ideal game.

Proof. (*Selective-failure blindness*) The malicious authority A_k provides PK_k and two global identifiers (u_0, u_1). A random bit $b \in \{0,1\}$ is picked. A_k can

Table 2: The comparison of computing cost

Scheme	Setup	KeyGen	Encryption	Decryption
Huang [15]	$P+(N+2)E$	$(2\|A_U\|+N+1)E$	$(2\|A_c\|+2)E$	$(2\|A_c\|+N+1)P+\|A_c\|E$
Qian [26]	$P+(2N+m\cdot\|U\|)E$	$(2\|A_U\|+4N)E$	$(2\|A_c\|+2)E$	$(\|A_c\|+3N+1)P$
Qian [27]	$P+(3N+\|U\|)E$	$(\|A_U\|+3N(N-1))E$	$(\|A_c\|+2)E$	$(N\|A_c\|+1)P+(\|A_c\|+1)E$
Ours	$P+(2N+m\cdot\|U\|)E$	$(\|A_U\|+3N)E$	$(\|A_c\|+N+2)E$	$(\|A_c\|+N+1)P+E$

[1] P: the bilinear pairing operation. E: the exponential operation. $|*|$: the number of elements in $*$.
[2] A_U: the attribute set of U. A_c: the attribute set in ciphertext. m: the number of values for each attribute.

have a black box access to $U(u_b, com_b, PK_k, PP)$ and $U(u_{1-b}, com_{1-b}, PK_k, PP)$. Then, U runs PPKeyGen algorithm with A_k and returns secret keys $SK^k_{U_b}$ and $SK^k_{U_{1-b}}$: If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} \neq \perp$, A_k is given $(SK^k_{U_b}, SK^k_{U_{1-b}})$; If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} = \perp$, A_k is given $(\epsilon, \perp)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} \neq \perp$, A_k is given $(\perp, \epsilon)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} = \perp$, A_k is given $(\perp, \perp)$. Finally, A_k outputs a guess b' on b. □

In the PPKeyGen algorithm, U first computes T, P_0, and proves $PoK\{(u, \rho_0, z^*) : T = g_p^{z^*} g_1^u \wedge P_0 = g_0^{\rho_0}\}$. So far, the two oracles should be computationally indistinguishable to A_k. Otherwise, it will violate the commitment scheme's hiding property and the witness undistinguishable of the zero-knowledge proof. Assume that A_k outputs secret keys for the first oracle with some computing strategies. The next thing to prove is that A_k can predict secret keys for user without interaction with the two oracles:

1) A_k checks $PoK\{(\alpha_k, \beta_k, r_{k,u}, \rho_1, \eta) : \tilde{D}_{k,u} = g_p^{\alpha_k} T^{\beta_k} P_0^{\frac{\rho_1 t_{k,u}}{\eta}} \wedge P_1 = g_0^{\rho_1} \wedge \tilde{D}_{k,i,j} = P_1^{\frac{t^u_{k,i,j}}{\eta a_{k,i,j}}}\}$. If the proof fails, A_k outputs $SK^k_{U_0} = \perp$.

2) A_k generates different $(\tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ for the second oracle and proves $PoK\{(\alpha_k, \beta_k, r_{k,u}, \rho_1, \eta) : \tilde{D}_{k,u} = g_p^{\alpha_k} T^{\beta_k} P_0^{\frac{\rho_1 t_{k,u}}{\eta}} \wedge P_1 = g_0^{\rho_1} \wedge \tilde{D}_{k,i,j} = P_1^{\frac{t^u_{k,i,j}}{\eta a_{k,i,j}}}\}$. If the proof fails, A_k outputs $SK^k_{U_1} = \perp$.

3) Finally, A_k returns its prediction on (u_0, u_1). If $SK^k_{U_0} \neq \perp$ and $SK^k_{U_1} \neq \perp$, the prediction is $(SK^k_{U_0}, SK^k_{U_1})$; If $SK^k_{U_0} \neq \perp$ and $SK^k_{U_1} = \perp$, the prediction is $(\epsilon, \perp)$; If $SK^k_{U_0} = \perp$ and $SK^k_{U_1} \neq \perp$, the prediction is $(\perp, \epsilon)$; If $SK^k_{U_0} = \perp$ and $SK^k_{U_1} = \perp$, the prediction is $(\perp, \perp)$.

The predication has the same distribution with the oracles as A_k performs the same check as the honest user. Therefore, if A_k can predict the user secret keys, then A_k, with or without the final outputs, has the same advantage. It means that A_k can distinguish the two oracles before the prediction. However, based on the security of commitment scheme and zero-knowledge proof, the advantage of A_k in distinguishing the two oracles is negligible.

6 Performance Analysis

In this section, Table 2 and Figure 3 show the comparisons of computation costs and efficiency, respectively. The simulation is executed on a Windows machine with 2.70 GHz Intel(R) Core (TM) i5-4210U CPU and 4 GB RAM. The implementation is based on Java Pairing-Based Cryptography (PBC) Library (version 0.5.14). Random elements in group G_{p_1} are introduced into our scheme to implement policy hiding. Apart from this function, the authors compare the scheme with schemes [15,26,27]. Figure 3(a) and Figure 3(b) show the comparisons of Setup time and KeyGen time with different number of attribute authorities. In this simulation, the number of attribute authorities is increased from 2 to 14, each authority monitors 5 attributes, and each attribute has 3 values. Scheme [15] uses a hash function to map the user's attributes to a random group element. Attribute authorities do not need to calculate the public keys related to the attributes, so the setup stage of the scheme takes less time. Figure 3(b) shows that the proposed scheme takes less time to generate the secret keys than the other three schemes. The reason why the secret key generation time of scheme [27] increases rapidly with the number of authorities is that every two authorities have to interact with each other. The KeyGen time is a quadratic function of the independent variable N. Figure 3(c) and Figure 3(d) show the comparisons of encryption time and decryption time with different number of attributes in the access policy, and the number of attribute authorities in the system is fixed at 5. This proposed scheme encryption time is longer than scheme [27], but the difference is very small. Figure 3(d) shows that the proposed scheme is superior to the other three schemes in the decryption phase.

7 Conclusion

To implement data sharing, a privacy-preserving decentralized ABE scheme is proposed in this paper. All attribute authorities are independent of each other and do not require any collaboration. All users can get the secret keys from authorities by using the PPKeyGen protocol without revealing his/her GID, which meets users' requirement to protect their private information. Besides, this scheme supports policy anonymity, so one cannot get any information about the user attributes. The security of the scheme is proved under a standard model and the simulation results verify the efficiency of the scheme. The disadvantage of the proposed scheme is that it only supports

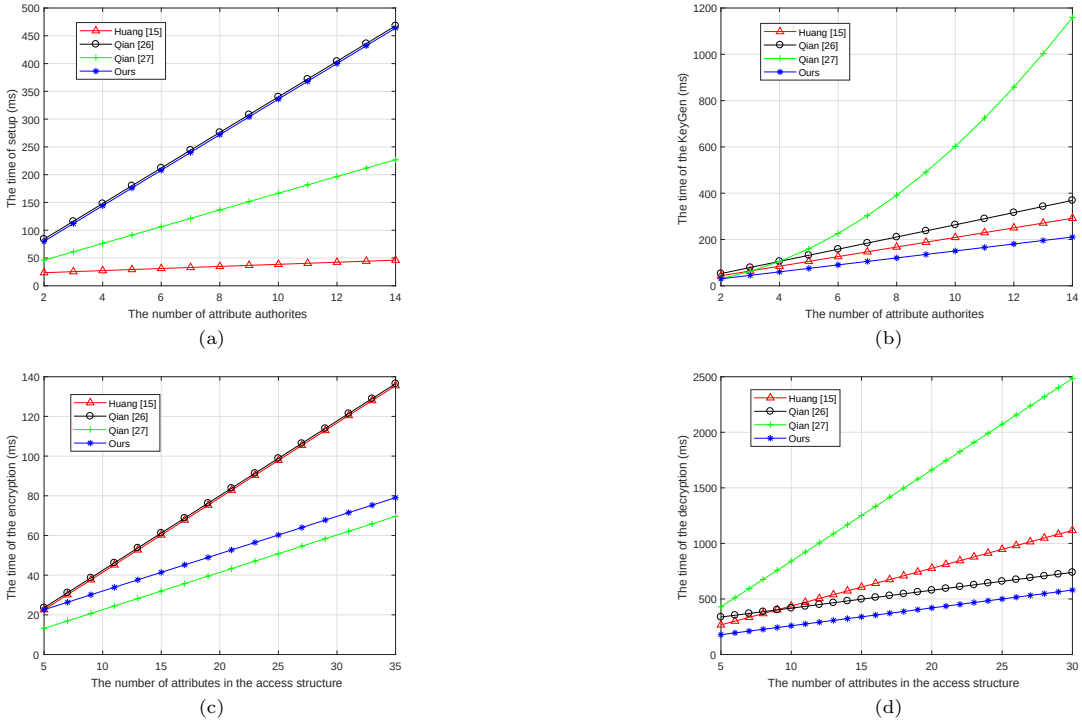

Figure 3: Efficiency comparisons of the proposed scheme with others

the AND-gate access structure. Constructing encryption schemes to support more flexible access structures are left as the future works.

Acknowledgments

This work was supported in part by the National Cryptography Development Fund under grant (MMJJ20180209).

References

[1] Y. Baseri, A. Hafid, and S. Cherkaoui, "Privacy preserving fine-grained location-based access control for mobile cloud," *Computers and Security*, vol. 73, pp. 249–265, 2018.

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

[3] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology (CRYPTO'97)*, pp. 410–424, 1997.

[4] Z. Cao, L. Liu, and Z. Guo, "Ruminations on attribute-based encryption," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, 2018.

[5] M. Chase, "Multi-authority attribute based encryption," in *Conference on Theory of Cryptography*, pp. 515–534, 2007.

[6] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Acm Conference on Computer and Communications Security*, pp. 121–130, 2009.

[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] T. Feng and J. Guo, "A new access control system based on CP-ABE in named data networking," *International Journal of Network Security*, vol. 20, no. 4, 2018.

[9] T. Feng, X. Yin, Y. Lu, J. Fang, and F.Li, "A searchable CP-ABE privacy preserving scheme," *International Journal of Network Security*, vol. 21, no. 4, 2019.

[10] A. Ge, J. Zhang, R. Zhang, C. Ma, and Z. Zhang, "Security analysis of a privacy-preserving decentralized key-policy attribute-based encryption scheme," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, 2013.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[12] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Dis-*

tributed Systems, vol. 23, no. 11, pp. 2150–2162, 2012.

[13] J. Han, W. Susilo, and Y. Mu, *et al.*, "PPDCP-ABE: Privacy-preserving decentralized ciphertext-policy attribute-based encryption," in *Computer Security*, pp. 73–90, 2014.

[14] S. Hu, J. Li, and Y. Zhang, "Improving security and privacy-preserving in multi-authorities ciphertext-policy attribute-based encryption," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 10, 2018.

[15] X. Huang, Q. Tao, B.Qin, and Z. Liu, "Multi-authority attribute based encryption scheme with revocation," in *International Conference on Computer Communication and Networks*, pp. 1–5, 2015.

[16] T. Jung, X. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, 2015.

[17] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, 2013.

[18] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology-eurocrypt -international Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588, 2011.

[19] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, 2013.

[20] X. Li, D. Gu, Y. Ren, N. Ding, and K. Yuan, "Efficient ciphertext-policy attribute based encryption with hidden policy," in *Internet and Distributed Computing Systems*, pp. 146–159, 2012.

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

[22] L. Liu, Z. Cao, and C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, 2018.

[23] M. Lyu, X. Li, and H. Li, "Efficient, verifiable and privacy preserving decentralized attribute-based encryption for mobile cloud computing," in *IEEE Second International Conference on Data Science in Cyberspace*, 2017. DOI: 10.1109/DSC.2017.8.

[24] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *International Cryptology Conference on Advances in Cryptology*, pp. 129–140, 1991.

[25] H. S. G. Pussewalage and V. A. Oleshchuk, "A distributed multi-authority attribute based encryption scheme for secure sharing of personal health records," in *Acm on Symposium on Access Control Models and Technologies*, pp. 255–262, 2017.

[26] 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*, pp. 363–372, 2013.

[27] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, 2015.

[28] Y. Rahulamathavn, 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, 2016.

[29] S. Rezaei, M. A. Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, 2018.

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

[31] 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, 2015.

[32] F. Xhafa, J. Feng, Y. Zhang, X. Chen, and J. Li, "Privacy-aware attribute-based phr sharing with user accountability in cloud computing," *Journal of Supercomputing*, vol. 71, no. 5, 2015.

[33] R. Xu, Y. Wang, and B. Lang, "A tree-based CP-ABE scheme with hidden policy supporting secure data sharing in cloud computing," in *International Conference on Advanced Cloud and Big Data*, pp. 51–57, 2013.

[34] L. Zhang, P. Liang, and Y. Mu, "Improving privacy-preserving and security for decentralized key-policy attributed-based encryption," *IEEE Access*, vol. 6, pp. 12736–12745, 2018.

[35] L. Zhang and H. Yin, "Recipient anonymous ciphertext-policy attribute-based broadcast encryption," *International Journal of Network Security*, vol. 20, no. 1, 2018.

[36] Y. Zhang, D. Zheng, and R.H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, pp. 2130–2145, 2018.

Biography

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.

Leyou Zhang is a professor in the school of mathemat-

ics and statistics at Xidian University, Xi'an China. He received his PhD from Xidian University in 2009. From Dec. 2013 to Dec. 2014, he is a research fellow in the school of computer science and software engineering at the University of Wollongong. His current research interests include network security, computer security, and cryptography.