Efficient CP-ABE Scheme with Shared Decryption in Cloud Storage

Ningyu Chen, Jiguo Li, Yichen Zhang, and Yuyan Guo

Abstract—Attribute based encryption (ABE) is a preferred technology used to access control the data stored in the cloud servers. However, in many cases, the authorized decryption user may be unable to decrypt the ciphertext in time for some reason. To be on the safe side, several alternate users are delegated to cooperate to decrypt the ciphertext, instead of one user doing that. We provide a ciphertext-policy ABE scheme with shared decryption in this paper. An authorized user can recover the messages independently. At the same time, these alternate users (semi-authorized users) can work together to get the messages. We also improve the basic scheme to ensure that the semi-authorized users perform the decryption tasks honestly. An integrated access tree is used to improve the efficiency for our scheme. The new scheme is proved CPA-secure in the standard model. The experimental result shows that our scheme is very efficient on both computational overhead and storage cost.

Index Terms—Cloud storage, ciphertext-policy, ABE, shared decryption, CPA-secure.

1 Introduction

CLOUD storage [1,2] is a new storage technology based on network and cloud computing, which provides "unlimited" storage resources for data users. Users can easily access the data stored in the cloud from anywhere in the world. More personal and corporate data are being stored on cloud storage servers. These businesses and individuals can significantly reduce the cost of data storage and management by storing their data on the remote cloud storage servers. However, the cloud service provider, such as Google Cloud, IBM Cloud, and Microsoft Cloud, may be curious or profit-driven to leak users' sensitive data. In addition, these data stored on remote cloud storage servers may be attacked, modified, and disclosed by hackers. Therefore, users tend to encrypt their files before storing these files on an untrusted cloud storage server. In order to ensure the correctness of the files, some remote data integrity checking schemes [3-7] were proposed. However, the data for cloud storage still faces some challenges [8].

Attribute based encryption (ABE) is a hot research topic in cryptography in recent years, which can realize the privacy protection of data stored on the cloud servers. Sahai et al. [9] extended the previous identity based encryption and proposed the concept of ABE. In the presented ABE, an attribute set is used to replace the identity of a user. Existing ABE schemes are divided into two categories: key-policy ABE (KP-ABE) schemes and ciphertext-policy ABE (CP-ABE) schemes. Goyal et al. [10] gave a KP-ABE scheme in 2006. In this scheme, an access structure is related to the private key of a user. At the same time, an attribute set is related to the ciphertext. In 2007, Bethencourt et al. [11] provided a CP-ABE scheme. His scheme is more practical and more flexible than KP-ABE scheme. In a CP-ABE scheme, an attribute set is related to the private key of the user, while an access structure is related to the ciphertext. A user is able to decrypt the ciphertext only if his/her attribute set satisfies the access policy.

Later, various ABE schemes [12-26] were proposed. To protect the privacy of the users, ABE schemes with hidden access structure are proposed [12-15]. Because the attributes belonging to a user are usually managed by different authorities, multi-authority ABE [16-21] schemes are presented. ABE schemes with outsourced decryption (ABE-OD) [22-25] are proposed to improve the efficiency of decryption for the users. Searchable ABE schemes [25,26] are proposed to improve the efficiency of searching on the data encrypted with ABE. ABE is becoming increasingly popular in data access control systems [27-32].

From the viewpoint of the users, security and efficiency are two main requirements for any ABE scheme. It is very important to improve the efficiency of these existing secure ABE schemes. In 2016, Wang et al. [33] provided a file hierarchical ciphertext-policy attribute based encryption (FH-CP-ABE) scheme, which greatly improved the efficiency of scheme [11] without sacrificing the security. In many cases, these stored data files in public cloud are characterized by multi-level hierarchy. Scheme [33] combines multiple different hierarchical access policy trees into a single one. As shown in the Fig. 1, file m_1 and file m_2 have a hierarchical relationship on access. Access tree $\mathcal{T}_1$ can be integrated with access tree $\mathcal{T}_2$ into a new access tree $\mathcal{T}$. Finally, file m_1 and file m_2 can be encrypted simultaneously by using the

- N. Chen and J. Li are with the College of Computer and Information, Hohai University, Nanjing211100, Jiangsu, China. E-mail: cny314@163.com
- J. Li and Y. Zhang are with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou350117, Fujian, China and Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou350117, China, and J. Li is also State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China. E-mail: ljg1688@163.com, zyc_718@163.com
- Y. Guo is with School of Computer Science and Technology, Huaibei Normal University, 235000, Anhui, China. E-mail: guoyuyan428@163.com

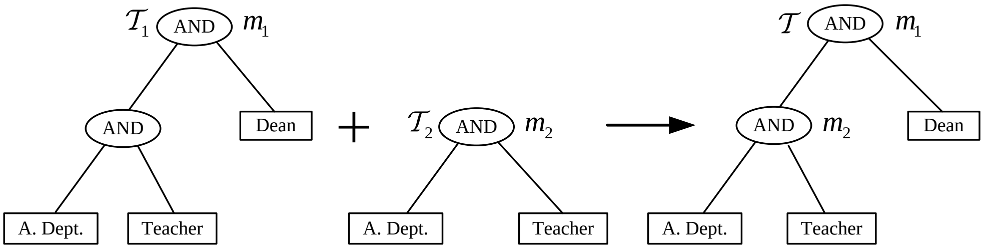


Fig. 1. An example of the construction of an integrated access tree.

access tree $\mathcal{T}$, instead of encrypting it twice with two different access trees. Scheme [33] is efficient on both encryption and decryption, and the scheme also saves the storage overhead of the ciphertext greatly. However, scheme [33] is not suitable for the case of multi-authority system, where the attributes belonging to a user are managed by different authorities. To solve the issue, Zhang et al. [34] improved above FH-CP-ABE scheme and provided a multi-authority hierarchical ABE scheme. There are multiple authorities in scheme [34] and an integrated access tree is used to encrypt these hierarchical files. However, there is a central authority in scheme [34], which is not sufficient for distributed systems. To overcome this issue, based on the hierarchical structure of personal health record (PHR) files, Guo et al. [35] provided a unique ABE scheme with multiple authorities. In addition, Li et al. [36] gave a more practical file hierarchical CP-ABE scheme to overcome the disadvantage that FH-CP-ABE scheme [33] cannot encrypt more than one file in the same level. For those big companies or institutions, scheme [36] can securely and efficiently store their data in the cloud, which is more flexible and practical than scheme [33]. However, if the stored files have not the characteristic of multiple hierarchical structures, above schemes [33-36] will fail due to the lack of the integrated access trees. In order to solve this issue, Fu et al. [37] proposed an attribute based hierarchy encryption scheme (ABHE) to encrypt a collection of documents. Based on the access attributes of these files in the document collection, an integrated access tree is constructed with a greedy algorithm in scheme [37]. Finally, the document collection is encrypted with the integrated access tree, just like the above scheme [33-36].

1.1 Motivation and Contribution

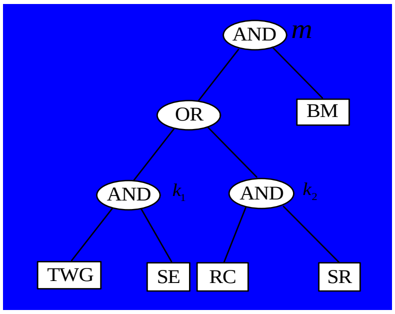


Fig. 2. An example for application.

In the cloud storage, it is very important for the authorized decryption user to decrypt the ciphertext quickly and correctly. In many cases, the authorized decryption user may encounter some urgent or dangerous events and be unable to recover the message online in time. At this point, there should be another user to decrypt the ciphertext instead of the authorized decryption user. However, it is not secure if the ciphertext is decrypted by only one designated user. So, several users are delegated to cooperate to decrypt the ciphertext, instead of one user to do that. In other words, this message is actually shared by these delegated users.

We put forward a CP-ABE with shared decryption (CP-ABE-SD) scheme to address the above problem. Besides the authorized user, multiple delegated users can also collaborate to recover the message in our solution. At the same time, we can verify the correctness of the decrypted results. To reduce the computation cost for encryption and decryption and save storage costs, an integrated access tree is used in our scheme as that in the scheme [37]. Finally, the plaintext is encrypted with the integrated access tree. Frequent data encryption and decryption operations make cloud storage very inefficient. In our scheme, the shared message is encrypted only once and size of the ciphertext is small. Our solution improves the efficiency of cloud storage.

Let's imagine the following scenario. We assume that department A of one company consists of a technical work group (TWG) and a research center (RC). There are fewer senior engineers (SE) in the technical work group and fewer senior researchers (SR) in the research center, and they are especially important to the department. For some reason, the business manager (BM) of the department does not decrypt the ciphertext stored in the cloud in time. The department will have an emergency meeting. And then a senior engineer from the technical work group and a senior researcher from the research center will work together to decrypt the ciphertext downloaded from the cloud servers. We assume that the business manager of the department is also a senior engineer or a senior researcher of the department. As shown in the Fig. 2, we create an integrated access tree. The business manager can decrypt the ciphertext independently. At the same time, a senior engineer of the technical work group and a senior researcher of the research center can "decrypt" the ciphertext and obtain the vital data k_1 and k_2 respectively, and they can cooperate to recover the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2020.3043950, IEEE Transactions on Computers

message m.

To achieve more flexible access control, the threshold gate in our integrated access tree include AND gate, OR gate and $n-\text{of}-m(n<m)$ gate. The main contributions of our paper are as below.

Firstly, based on an integrated access tree, we put forward an efficient and practical ABE scheme with shared decryption (CP-ABE-SD). An authorized decryption user can decrypt the ciphertext. At the same time, multiple delegated users can collaborate to decrypt the ciphertext.

Secondly, based on the decisional bilinear Diffie–Hellman (DBDH) assumption, the security of our scheme is proved. The experimental result shows that our scheme is efficient.

Thirdly, an improved scheme is proposed to ensure that each delegated user is honest in the shared decryption phase.

1.2 Organization of the Paper

The structure of our paper is arranged as below. In Section 2, some basic background knowledge for the proposed scheme is demonstrated. The formal definition and the detailed construction of CP-ABE-SD scheme and the improved scheme are introduced respectively in Section 3 and Section 4. The security model of proposed scheme is given, the security of the scheme is proved under the standard model in Section 5. The theoretical analysis and the experimental result are given in Section 6. The conclusion is given in Section 7.

TABLE 1
NOTATIONS.

Symbol	Description
1^K	A security parameter
$\mathbb{A}$	An access structure
$\mathcal{T}$	An access tree
CS	Cloud server
TA	Trusted authority
DO	Data owner
DU	Data user
V_i	A level node
H_i	A collusion-resistant hash funtion
S	An attribute set of the data user
∇	Lagrange coefficient

2 Preliminaries

The basic background knowledge is given in this section, such as bilinear maps [10,11,38,39], security assumption, access structure. The notations used in this paper are described in TABLE 1.

2.1 Bilinear Maps

$\mathbb{G}_1$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order p. $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_T$ is bilinear map with generator g of $\mathbb{G}_1$ taking the below properties.

Bilinearity: For any $a_1,a_2\in\mathbb{Z}_p$ and $b_1,b_2\in\mathbb{G}_1$, $e(b_1^{a_1},b_2^{a_2})=e(b_1,b_2)^{a_1a_2}$.

Non-degeneracy: $e(g,g)\neq 1$.

Computability: Both the group operations of $\mathbb{G}_1$ and the bilinear map $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_T$ are efficiently computable.

2.2 Security Assumption

The decisional bilinear Diffie-Hellman (DBDH) problem [10,11,40] is described as below. g is generator in $\mathbb{G}_1$ and $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_T$ is bilinear map. Suppose that x_1,x_2,x_3,x_4 are four randomly elements in $\mathbb{Z}_p$. Let $X_1=g^{x_1}$, $X_2=g^{x_2}$ and $X_3=g^{x_3}$. The adversary wants to distinguish between $e(g,g)^{x_1x_2x_3}$ and $e(g,g)^{x_4}$. $\mathcal{B}$ is a probabilistic polynomial time algorithm [41,42] with advantage ε in solving above problem if:

$$Adv_{\mathcal{B}}=\left|\begin{matrix}\Pr[\mathcal{B}(X_1,X_2,X_3,e(g,g)^{x_1x_2x_3})=0]\\-\Pr[\mathcal{B}(X_1,X_2,X_3,e(g,g)^{x_4})=0]\end{matrix}\right|\geq\varepsilon$$

2.3 Access Structure

Let $\{attr_1,attr_2,\cdots,attr_n\}$ be an attribute set. There is a monotone collection $\mathbb{A}\subseteq 2^{\{attr_1,attr_2,\cdots,attr_n\}}$: For $\forall X_1,X_2$: if $X_1\in\mathbb{A}$ and $X_1\subseteq X_2$, then $X_2\in\mathbb{A}$. A monotone access structure $\mathbb{A}$ is a monotone collection for non-empty subsets in $\{attr_1,attr_2,\cdots,attr_n\}$, i.e., $\mathbb{A}\subseteq 2^{\{attr_1,attr_2,\cdots,attr_n\}}\setminus\{\varnothing\}$. The sets belonged to $\mathbb{A}$ are authorized sets; Otherwise, these sets are called unauthorized sets [17,43].

3 Formal Definition of CP-ABE-SD Scheme

3.1 Formal Definition of CP-ABE-SD Scheme

Let $\mathcal{T}$ be an integrated tree and it denotes an access structure. There are multiple level nodes in an integrated tree, which means that there are multiple decryption users in our scheme. Let $V=\{V_R,V_1,V_2,\cdots,V_k\}$ be the set of the level nodes in $\mathcal{T}$. A level node V_i is a non-leaf node or leaf node in $\mathcal{T}$, which is exactly the same as non-level nodes except for its level properties. $\{V_R,V_1,V_2,\cdots,V_k\}$ are arranged in order from top to bottom in $\mathcal{T}$. The root node of $\mathcal{T}$ is level node V_R. As shown in the Fig. 3, there are three level nodes V_R,V_1 and V_2 in $\mathcal{T}$, and V_R is also the root node in $\mathcal{T}$.

There are multiple leaf nodes and non-leaf nodes in the integrated tree $\mathcal{T}$. Each non-leaf node and leaf node represents a threshold gate and an attribute respectively. Each node in $\mathcal{T}$ is denoted with (x,y), where x represents the row in the node and y represents the column in the node. Let $h_{(x,y)}$ denote the threshold value in the node (x,y) and $num_{(x,y)}$ denote the number of children nodes in the non-leaf node (x,y). Obviously, if (x,y) is a non-leaf node, then $num_{(x,y)} \geq h_{(x,y)} \geq 0$. When (x,y) is a non-leaf node and $num_{(x,y)} = h_{(x,y)}$, (x,y) represents an *AND* gate. When (x,y) is a non-leaf node and $h_{(x,y)} = 1$, (x,y) represents an *OR* gate. The threshold value of the leaf node (x,y) is $h_{(x,y)} = 1$.

Other functions are described as follows. $attr(x,y)$ is defined as the attribute value in the leaf node (x,y). $pare(x,y)$ returns the parent node of node (x,y). $inde(x,y)$ outputs a unique number associated with node (x,y), where the number is assigned to each node in an arbitrary manner.

In addition, $\mathcal{T}_{(x,y)}$ is represented as a subtree in $\mathcal{T}$ rooted at (x,y). If an attribute set S satisfies the subtree, we set $\mathcal{T}_{(x,y)}(S) = 1$. $\mathcal{T}_{(x,y)}(S)$ is recursively calculated as below. If (x,y) is a leaf node in $\mathcal{T}$, $\mathcal{T}_{(x,y)}(S) = 1$ if and only if $attr(x,y) \in S$. If (x,y) is a non-leaf node in $\mathcal{T}$, $\mathcal{T}_{(x,y)}(S) = 1$ if and only if at least $h_{(x,y)}$ children return 1.

3.2 System Model

In our model, the file is encrypted with a symmetric session key. At the same time, CP-ABE-SD scheme is utilized to encrypt the session key. An attribute set is associated with the private key of an authorized user or a semi-authorized user. Fig.4 is the framework of the system. As shown in Fig.4, our model includes four entities, Cloud Server, Trusted Authority, Data User and Data Owner.

Cloud Server (CS): It honestly executes the data storage and transmission service. It is a storage service provider.

Trusted Authority (TA): It generates master secret key and public key of the system. Moreover, it issues the private key for each user.

Data Owner (DO): It wants to store and share the message on the cloud servers. It first encrypts the message with a symmetric session key. An access tree with multiple nodes $\{V_R, V_1, V_2, \cdots, V_k\}$ is defined and utilized to encrypt the session key. Finally, the encrypted message and session key are stored on the storage servers.

Data User (DU): There are two kinds of data users in our system. For an authorized user, his attributes satisfy the whole access tree $\mathcal{T}$ rooted at V_R. He can successfully recover the message. For a semi-authorized user, his attributes satisfy a subtree of $\mathcal{T}$ rooted at V_i ($i \neq R$). These semi-authorized users can work together to recover the message.

Definition 1. A CP-ABE-SD scheme possesses the four algorithms: **Setup, KeyGen, Encrypt** and **Decrypt**.

Setup $(1^K) \to (MK, PK)$. This algorithm is executed by TA. A security parameter 1^K is inputted, it returns the public key PK and the master secret key MK of the system.

Encrypt $(PK, ck, \mathbb{A}) \to CT$. DO performs the encryption algorithm, which inputs a session key ck, PK and an access structure $\mathbb{A}$ and returns the ciphertext CT.

KeyGen $(S, PK, MK) \to SK$. The operation is performed by TA. The algorithm inputs attribute set S, PK and MK and outputs the secret key SK related to the attribute set S.

Decrypt $(CT, PK, SK) \to ck$. The algorithm inputs CT, SK and PK, it outputs session key ck or $\perp$. This operation is run by DU.

4 Concrete Construction

In this section, we present the detailed construction of CP-ABE-SD scheme. Based on the scheme [11], we add an integrated access structure [33-37], which allows multiple semi-authorized users to work together to decrypt the correct message. In some special application scenarios, some semi-authorized user may be dishonest. We make a slight modification to CP-ABE-SD scheme. The modified scheme allows any user to verify the correctness of the share submitted by each semi-authorized user. In our schemes, the Lagrange coefficient ∇ is computed as below. Given a set of attributes S and any $\xi \in \mathbb{Z}_p$, $\nabla_{\xi,S} = \prod_{\upsilon \neq \xi, \upsilon \in S} (x-\upsilon)/(\xi-\upsilon)$.

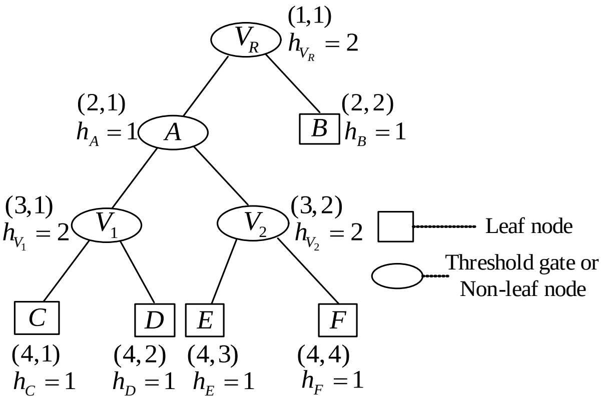


Fig. 3. An integrated access tree.

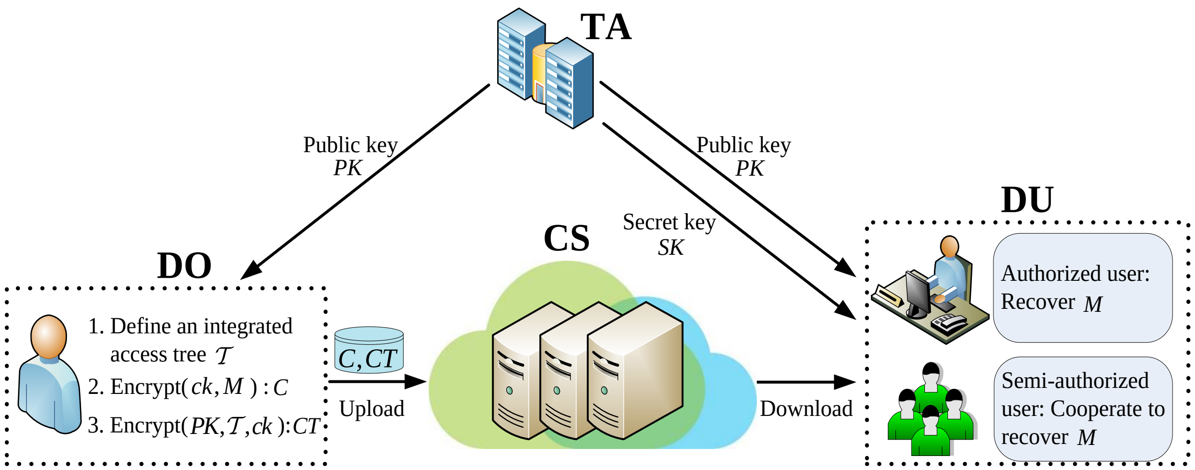

TA
Public key
PK
Public key
PK
Secret key
SK
DO
CS
DU
1. Define an integrated access tree $\mathcal{T}$
2. Encrypt(ck, M) : C
3. Encrypt($PK, \mathcal{T}, ck$):CT
C, CT
Upload
Download
Authorized user: Recover M
Semi-authorized user: Cooperate to recover M

Fig. 4. The system model of CP-ABE-SD scheme.

4.1Proposed CP-ABE-SD Scheme

Let 1^K be a security parameter and $\mathbb{G}_1$ be bilinear multiplicative group with prime order p. Let $e:\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ be bilinear map. g is a random generator in $\mathbb{G}_1$. An attribute set $S=\{S_1,S_2,\cdots,S_i\}\subset \mathbb{Z}_p$ is a subset of the universe attribute set. The session key ck is an element in group $\mathbb{G}_T$. $H_1:\mathbb{Z}_p \to \mathbb{G}_1$, $H_2:\mathbb{G}_T \to \{0,1\}^*$ and $H_3:\{0,1\}^* \to \mathbb{Z}_p$ are three random hash functions.

Setup $(1^K)\to(MK,PK)$. The trusted authority initializes the system and performs the algorithm. The algorithm randomly selects $\alpha,\beta\in\mathbb{Z}_p$ and computes $e(g,g)^\beta$ and $f=g^\alpha$. The public key is $PK=\{g,\mathbb{G}_1,\mathbb{G}_T,H_1,H_2,H_3,e(g,g)^\beta, f=g^\alpha\}$. $MK=\{\alpha,\beta\}$ is the master secret key of the system.

Encrypt $(PK,ck,\mathcal{T})\to CT$. An integrated access tree $\mathcal{T}$ is inputted as an access structure in this algorithm. Data owner encrypts a session key ck under the access structure.

(1) Data owner creates an integrated access tree $\mathcal{T}$ and sets the root node V_R and the other k nodes $V_1,V_2,\cdots,V_k$ as level nodes in $\mathcal{T}$. Next, it randomly selects $t,t_1,t_2,\cdots,t_{k\text{-}1}\in\mathbb{Z}_p$ to compute the following operations.

$$t_k=t-\sum_{i=1}^{k-1}t_i$$

$$C_V=ck\cdot e(g,g)^{\beta t}$$

$$C_{V_i}=g^{t_i}\quad (i=1,\ 2,\ \cdots,\ k)$$

$$tag=H_3(H_2(ck)\,\|\,H_2(C_V))$$

where $\|$ denotes the connection of two bit strings.

(2) Next, for each node (x,y) in $\mathcal{T}$, the algorithm constructs a polynomial $q_{(x,y)}$. The degree of $q_{(x,y)}$ is $d_{(x,y)}=h_{(x,y)}-1$, where $h_{(x,y)}$ is the threshold value of (x,y). The construction of these polynomials begins with the root node V_R. The algorithm constructs them in a top-down manner till to all leaf nodes. For the root node V_R, the algorithm sets $q_{V_R}(0)=t$. For other nodes (x,y) in $\mathcal{T}$, the algorithm sets $q_{(x,y)}(0)=q_{V_i}(0)=t_i$ if this node is the level node V_i; Otherwise, $q_{(x,y)}(0)=q_{pare(x,y)}(inde(x,y))$. The algorithm continues to find out the remaining $d_{(x,y)}$ points to construct $q_{(x,y)}$. Let $k_{(x,y)}$ denote the number of level nodes among the children of the node (x,y). Here we assume $k_{(x,y)}\le d_{(x,y)}$. For each level node V_j that is (x,y)'s child, the algorithm sets $q_{(x,y)}(inde(V_j))=q_{V_j}(0)=t_j$. The algorithm sets the rest of $d_{(x,y)}$-$k_{(x,y)}$ points randomly to complete $q_{(x,y)}$. Finally, all polynomials are constructed.

(3) The data owner calculates $C_{(x,y)}=f^{q_{(x,y)}(0)}$ and $C'_{(x,y)}=H_1(attr(x,y))^{q_{(x,y)}(0)}$ for every leaf node (x,y).

(4) Finally, the algorithm outputs the ciphertext CT. $CT=\{\mathcal{T},tag,C_V,\{C_{V_i}\},\{C_{(x,y)},C'_{(x,y)}\}\}\ (i=1,2,\cdots,k)$.

KeyGen $(S,PK,MK)\to SK$. In this phase, the trusted authority randomly selects $r_j\in\mathbb{Z}_p$ for each attribute $j\in S$ and $r\in\mathbb{Z}_p$. The secret key SK of the attribute set S is generated as below.

$$SK=(D=g^\beta g^r,\forall j\in S: D_j=g^{\frac{r}{\alpha}}H_1(j)^{\frac{r_j}{\alpha}},D'_j=g^{r_j})$$

Decrypt $(CT,PK,SK)\to ck$. The user can obtain the decryption key that matches his/her attribute set S. In this phase, he/she begins to decrypt the ciphertext using his/her decryption secret key. For an authorized user, his/her attributes match the whole access tree, so he/she can decrypt the ciphertext independently. Once the authorized user cannot decrypt the ciphertext for some

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2020.3043950, IEEE Transactions on Computers

reason, then the specified multiple semi-authorized users can work together to decrypt the ciphertext.

(1) First, the algorithm creates a recursive function $DecryNode(CT,SK,(x,y))$.

For a leaf node (x,y), if $attr(x,y)\notin S$, then $DecryNode(CT,SK,(x,y))=\perp$; Otherwise, let $i=attr(x,y)$, and $DecryNode(CT,SK,(x,y))$ is computed as:

$$\begin{aligned} &DecryNode(CT,SK,(x,y)) \\ &=\frac{e(C_{(x,y)},D_i)}{e(C'_{(x,y)},D'_i)} \\ &=\frac{e(f^{q_{(x,y)}(0)},g^{\frac{r}{\alpha}}H_1(i)^{\frac{r_i}{\alpha}})}{e(H_1(attr(x,y))^{q_{(x,y)}(0)},g^{r_i})} \\ &=e(g,g)^{rq_{(x,y)}(0)} \end{aligned}$$

For a non-leaf node (x,y), we recursively compute $DecryNode(CT,SK,(x,y))$ as follow: For each child node Λ of (x,y), we compute $F_\Lambda=DecryNode(CT,SK,\Lambda)$. Let $M_{(x,y)}$ denote a node set composed of arbitrary $h_{(x,y)}$ child nodes Λ where $F_\Lambda\neq\perp$. If there is no such set, $F_{(x,y)}=\perp$; Otherwise, function $F_{(x,y)}$ is computed as below.

$$\begin{aligned} F_{(x,y)} &= \prod_{\Lambda\in M_{(x,y)}} F_\Lambda^{\nabla_{i,M'_{(x,y)}}(0)} \\ &= \prod_{\Lambda\in M_{(x,y)}} (e(g,g)^{rq_\Lambda(0)})^{\nabla_{i,M'_{(x,y)}}(0)} \\ &= \prod_{\Lambda\in M_{(x,y)}} (e(g,g)^{rq_{(x,y)}(i)})^{\nabla_{i,M'_{(x,y)}}(0)} \\ &= e(g,g)^{rq_{(x,y)}(0)} \end{aligned}$$

where $i=index(\Lambda)$, $M'_{(x,y)}=\{index(\Lambda):\Lambda\in M_{(x,y)}\}$.

(2) Next, decryption algorithm continues to be run to get the session key ck.

For an authorized user, his/her attributes match the whole access tree $\mathcal{T}$, the user gets $e(g,g)^{rt}$ after the previous decryption operation step. He/She computes $e(g,g)^{\beta t}$ as below.

$$\begin{aligned} &\frac{e(\prod_{i=1}^{k}C_{V_i},D)}{DecryNode(CT,SK,V_R)} \\ &=\frac{e(g^{\sum_{i=1}^{k}t_i},g^\beta g^r)}{e(g,g)^{rq_{V_R}(0)}} \\ &=\frac{e(g^t,g^\beta g^r)}{e(g,g)^{rt}} \\ &=e(g,g)^{\beta t} \end{aligned}$$

Finally, the authorized user computes $C_V/e(g,g)^{\beta t}=(ck\cdot e(g,g)^{\beta t})/e(g,g)^{\beta t}=ck$.

For a semi-authorized user, his/her attributes satisfy a subtree in $\mathcal{T}$ rooted at the level node V_i. The user gets $e(g,g)^{rt_i}$ after the previous decryption operation step. He/She computes $e(g,g)^{\beta t_i}$ as his share as follows.

$$\begin{aligned} &\frac{e(C_{V_i},D)}{DecryNode(CT,SK,V_i)} \\ &=\frac{e(g^{t_i},g^\beta g^r)}{e(g,g)^{rq_{V_i}(0)}} \\ &=\frac{e(g^{t_i},g^\beta g^r)}{e(g,g)^{rt_i}} \\ &=e(g,g)^{\beta t_i} \end{aligned}$$

Then, the semi-authorized users work together to decrypt the ciphertext. The decryption is calculated as follows.

$$\prod_{i=1}^{k}e(g,g)^{\beta t_i}=e(g,g)^{\sum_{i=1}^{k}\beta t_i}=e(g,g)^{\beta\sum_{i=1}^{k}t_i}=e(g,g)^{\beta t}$$

$$C_V/e(g,g)^{\beta t}=(ck\cdot e(g,g)^{\beta t})/e(g,g)^{\beta t}=ck$$

The algorithm computes $H_2(ck)$ and verifies the correctness of the following equation.

$$H_3(H_2(ck)\,\|\,H_2(C_V))\overset{?}{=}tag$$

If the equation holds, it means that all semi-authorized users are trustworthy and the algorithm outputs ck. Otherwise, the algorithm outputs $\perp$.

4.2 Improved CP-ABE-SD Scheme

We notice that there is a small inadequacy in CP-ABE-SD scheme: If some semi-authorized user is dishonest, he will show a forged $e(g,g)^{\beta t_i}$ during the shared decryption phase. The algorithm simply outputs $\perp$ without knowing who the dishonest user is. Now let us make a simple fix for the problem. We slightly modify our CP-ABE-SD scheme. We add a public verification to check the validity of the shares submitted by each semi-authorized user. A collusion-resistant hash function is used in the verification process of the scheme. Furthermore, the improved scheme is proved CPA secure in the standard model. We ignore the security proof of the improved scheme, which is almost identical to that of CP-ABE-SD scheme. The improved scheme is described as below.

1^K is security parameter, and $\mathbb{G}_1$ is bilinear multiplicative group with prime order p. $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_T$ is an efficient bilinear map. g is generator of $\mathbb{G}_1$. Let $S=\{S_1,S_2,\cdots,S_i\}\subset\mathbb{Z}_p$ denote a set of attributes, and it is also a subset of the universe set of the attributes. ck denotes a session key, which is a random element in group $\mathbb{G}_T$. $H_1:\mathbb{Z}_p\to\mathbb{G}_1$, $H_2:\mathbb{G}_T\to\mathbb{Z}_p$ and $H_3:\mathbb{Z}_p\to\mathbb{Z}_p$ are three random hash functions.

Setup $(1^K)\to(MK,PK)$. This phase is similar to that in CP-ABE-SD scheme.

Encrypt $(PK,ck,\mathcal{T})\to CT$**.** An integrated access tree $\mathcal{T}$ is inputted as an access structure in this algorithm. Data owner encrypts a session key ck under the access structure.

(1) Data owner creates an integrated access tree $\mathcal{T}$ and sets the root node V_R and the other k nodes $V_1,V_2,\cdots,V_k$ as level nodes in $\mathcal{T}$. Next, it selects $t,t_1,t_2,\cdots,t_{k\text{-}1} \in \mathbb{Z}_p$ randomly to compute the following operations.

$$t_k = t - \sum_{i=1}^{k-1} t_i$$

$$C_V = ck \cdot e(g,g)^{\beta t}$$

$$C_{V_i} = g^{t_i} \ (i=1,\ 2,\ \cdots,\ k)$$

$$\pi_i = H_3(H_2(e(g,g)^{\beta t_i}))\ (i=1,\ 2,\ \cdots,\ k)$$

(2) For every leaf node (x,y) of $\mathcal{T}$, the calculation of $C'_{(x,y)} = H_1(attr(x,y))^{q_{(x,y)}(0)}$ and $C_{(x,y)} = f^{q_{(x,y)}(0)}$ is similar to that in CP-ABE-SD scheme.

(3) The encryption algorithm outputs the ciphertext $CT = \{\mathcal{T}, C_V, \{C_{V_i}\}, \{C_{(x,y)}, C'_{(x,y)}\}, \{\pi_i\}\}\ (i=1,2,\cdots,k)$.

KeyGen $(S,PK,MK) \rightarrow SK$. The phase is similar to that in CP-ABE-SD scheme.

Decrypt $(CT,PK,SK) \rightarrow ck$.

(1) The calculation of $e(g,g)^{rq_{(x,y)}(0)}$ is similar to that in CP-ABE-SD scheme.

(2) Next, decryption algorithm continues to be run to get the session key ck.

For an authorized user, the decryption operation is similar to that in CP-ABE-SD scheme.

For a semi-authorized user, his/her attributes satisfy a subtree in $\mathcal{T}$ rooted at the level node V_i. The user gets $e(g,g)^{rt_i}$ after the previous decryption operation step. He/She can compute his/her share $e(g,g)^{\beta t_i}$. The calculation of $e(g,g)^{\beta t_i}$ is similar to that in CP-ABE-SD scheme.

Then, all semi-authorized users cooperate to decrypt together. Each semi-authorized user submits a value $e(g,g)^{\beta t_i^*}$ as his/her share. These semi-authorized users cooperate to compute $ck^* = C_V / \prod_{i=1}^{k} e(g,g)^{\beta t_i^*}$ and a set $\{\varsigma_i^* = H_2(e(g,g)^{\beta t_i^*})\}\ (i=1,\ 2,\ \cdots,\ k)$. All ς_i^* are published and any user can publicly verify the validity of the shares shown by these semi-authorized users. If $H_3(\varsigma_i^*)=\pi_i$, it is the real share $e(g,g)^{\alpha t_i}$. Otherwise, it is a forged share.

If all the semi-authorized users are honest, ck^* is the real session key ck.

5 Security Model and Security Proof

We give the security model of the CP-ABE-SD scheme and reduce the security of the scheme to the DBDH assumption in this section.

5.1 Security Model

We suppose that a secure symmetric encryption algorithm is utilized to encrypt the file. So, we only need to analyze the security of CP-ABE-SD. We assume that the root node is the only level node in the integrated access tree in CP-ABE-SD scheme. The security model for our scheme is similar to scheme [11]. The CPA security game is defined between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.

Init: $\mathcal{A}$ selects an access policy $\mathbb{A}^*$ that he/she wants to challenge upon. Then he/she delivers $\mathbb{A}^*$ to the challenger $\mathcal{C}$.

Setup: The challenger $\mathcal{C}$ executes Setup algorithm of CP-ABE-SD. The operation returns the public key PK and $\mathcal{C}$ sends PK to $\mathcal{A}$.

Query phase I: $\mathcal{A}$ repeatedly launches private key queries for a series of attribute sets $S_1,S_2,\cdots,S_{q_1}$, where no any set S_i satisfies $\mathbb{A}^*$**.** $\mathcal{C}$ answers these queries by executing the KeyGen operation.

Challenge: Adversary $\mathcal{A}$ picks two messages m_0 and m_1 of equal length and submits them to $\mathcal{C}$. $\mathcal{C}$ picks a message m_ξ randomly, then encrypts the message m_ξ under $\mathbb{A}^*$. Finally, $\mathcal{C}$ obtains the ciphertext CT^*. $\mathcal{C}$ returns CT^* to the adversary.

Query phase II: Same as the above Query phase I.

Guess: Adversary $\mathcal{A}$ outputs the guess $\xi' \in \{0,1\}$.

If $\xi' = \xi$, $\mathcal{A}$ wins the security game. The advantage of $\mathcal{A}$ in winning the security game is $Adv_{\mathcal{A}}^{CPA}(1^K) = \left|P_r[\xi' = \xi] - 1/2\right|$

***Definition* 2.** If no probabilistic polynomial time adversary wins the above game at non-negligible advantage, CP-ABE-SD scheme is CPA secure.

5.2 Security Proof

The security of the proposed CP-ABE-SD scheme is reduced to the DBDH assumption.

Theorem. CP-ABE-SD scheme is CPA secure in the standard model if the DBDH assumption holds. Concretely, if the adversary $\mathcal{A}$ breaks CP-ABE-SD scheme at the non-negligible advantage $\varepsilon = Adv_{\mathcal{A}}^{CPA}(1^K)$ in polynomial time, we can construct a simulator $\mathcal{B}$ to solve the DBDH problem with non-negligible advantage $\varepsilon/2$ in polynomial-time.

Proof: At first, the challenger $\mathcal{C}$ generates two multiplicative groups $\mathbb{G}_1, \mathbb{G}_T$. At the same time, a bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is also created. Then $\mathcal{C}$ selects a generator g for $\mathbb{G}_1$ and randomly chooses group element $W \in \mathbb{G}_T$. $\mathcal{C}$ randomly chooses $x_1,x_2,x_3 \in \mathbb{Z}_p$ and gets three group elements $X_1 = g^{x_1}, X_2 = g^{x_2}, X_3 = g^{x_3}$. Y is an element in group $\mathbb{G}_T$. $\mathcal{C}$ flips a fair coin $\rho \in \{0,1\}$. If $\rho = 1$, $\mathcal{C}$ sets $Y = W$, else it sets $Y = e(g,g)^{x_1 x_2 x_3}$. $\mathcal{C}$ sends $\langle X_1, X_2, X_3, Y \rangle$ to $\mathcal{B}$. $\mathcal{B}$ outputs his/her guess ρ' on ρ.

Init: Adversary $\mathcal{A}$ chooses an access structure $\mathbb{A}^*$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2020.3043950, IEEE Transactions on Computers

TABLE 2
COMPARISON WITH SCHEME [11,33,36]

	Scheme [11]	Scheme [33]	Scheme [36]	CP-ABE-SD scheme	Improved CP-ABE-SD scheme
Encryption time	$2(\lvert\mathbb{A}_{Id_1}\rvert+\cdots+\lvert\mathbb{A}_{Id_k}\rvert)E_{\mathbb{G}_1}$ $+kE_{\mathbb{G}_1}+kE_{\mathbb{G}_T}+(\lvert\mathbb{A}_{Id_1}\rvert$ $+\cdots+\lvert\mathbb{A}_{Id_k}\rvert)E_H$	$(k+2\lvert\mathbb{A}_{Id_R}\rvert)E_{\mathbb{G}_1}+$ $(k+2j\lvert\mathbb{A}_{tn}\rvert)E_{\mathbb{G}_T}+$ $(\lvert\mathbb{A}_{Id_R}\rvert+j\lvert\mathbb{A}_{tn}\rvert)E_H$	$(k+2\lvert\mathbb{A}_{Id_R}\rvert+j\lvert\mathbb{A}_{tn}\rvert)E_{\mathbb{G}_1}$ $+(k+2j\lvert\mathbb{A}_{tn}\rvert)E_{\mathbb{G}_T}+(k$ $+\lvert\mathbb{A}_{Id_R}\rvert+j\lvert\mathbb{A}_{tn}\rvert)E_H$	$(k+2\lvert\mathbb{A}_{Id_R}\rvert)E_{\mathbb{G}_1}$ $+E_{\mathbb{G}_T}+(\lvert\mathbb{A}_{Id_R}\rvert$ $+3)E_H$	$(k+2\lvert\mathbb{A}_{Id_R}\rvert)E_{\mathbb{G}_1}$ $+(k+1)E_{\mathbb{G}_T}+$ $(2k+\lvert\mathbb{A}_{Id_R}\rvert)E_H$
Decryption time	$2(\lvert\mathbb{A}_{Id_1}\rvert+\cdots+\lvert\mathbb{A}_{Id_k}\rvert)E_e$ $+kE_e+(\lvert\mathbb{A}_{Id_1}\rvert+\cdots+$ $\lvert\mathbb{A}_{Id_k}\rvert)E_{\mathbb{G}_T}$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)E_e+$ $\lvert\mathbb{A}_{Id_R}\rvert E_{\mathbb{G}_T}+(\lvert\mathbb{A}_{Id_R}\rvert$ $+j\lvert\mathbb{A}_{tn}\rvert)E_H$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)E_e+(\lvert\mathbb{A}_{Id_R}\rvert$ $+k)E_{\mathbb{G}_T}+(\lvert\mathbb{A}_{Id_R}\rvert+j\lvert\mathbb{A}_{tn}\rvert$ $+k)E_H$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)E_e$ $+\lvert\mathbb{A}_{Id_R}\rvert E_{\mathbb{G}_T}+$ $3E_H$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)E_e$ $+\lvert\mathbb{A}_{Id_R}\rvert E_{\mathbb{G}_T}+$ $2kE_H$
The size of SK	$(1+2\lvert S\rvert)L_{\mathbb{G}_1}$	$(1+2\lvert S\rvert)L_{\mathbb{G}_1}$	$(1+2\lvert S\rvert)L_{\mathbb{G}_1}$	$(1+2\lvert S\rvert)L_{\mathbb{G}_1}$	$(1+2\lvert S\rvert)L_{\mathbb{G}_1}$
The size of CT	$2(\lvert\mathbb{A}_{Id_1}\rvert+\cdots+\lvert\mathbb{A}_{Id_k}\rvert)L_{\mathbb{G}_1}$ $kL_{\mathbb{G}_1}+L_{\mathbb{G}_T}+L_{\mathbb{Z}_p}$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)L_{\mathbb{G}_1}+$ $(k+2j\lvert\mathbb{A}_{tn}\rvert)L_{\mathbb{G}_T}$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k+j\lvert\mathbb{A}_{tn}\rvert)L_{\mathbb{G}_1}$ $+j\lvert\mathbb{A}_{tn}\rvert L_{\mathbb{G}_T}+kl$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)L_{\mathbb{G}_1}$ $+L_{\mathbb{G}_T}+L_{\mathbb{Z}_p}$	$(2\lvert\mathbb{A}_{Id_R}\rvert+k)L_{\mathbb{G}_1}$ $+L_{\mathbb{G}_T}+kL_{\mathbb{Z}_p}$

that he/she wishes to challenge upon. Then $\mathcal{A}$ sends $\mathbb{A}^*$ to $\mathcal{B}$.

Setup: $\mathcal{B}$ randomly selects $\beta' \in \mathbb{Z}_p$. Let $f = g^{\alpha} = g^{x_2} = X_2$. $\mathcal{B}$ implicitly sets $\beta = x_2(\beta' + x_1)$ and computes $e(g,g)^{\beta}$ = $e(g,g)^{x_2(\beta'+x_1)}$ = $e(g,g)^{x_2\beta'}e(g,g)^{x_1x_2}$ = $e(g,X_2)^{\beta'}e(X_1,X_2)$. $\mathcal{B}$ responses $PK = \{g, \mathbb{G}_1, \mathbb{G}_T, H_1, H_2, H_3, e(g,X_2)^{\beta'}e(X_1,X_2), X_2\}$ as public key of $\mathcal{A}$.

Query phase I: $\mathcal{A}$ adaptively chooses multiple attribute sets $S_1, S_2, \cdots, S_q$ to obtain the secret keys of these attribute sets, where no any set S_i satisfies $\mathbb{A}^*$. $\mathcal{B}$ generates the corresponding private attribute key for each S_i as follows. Firstly, $\mathcal{B}$ randomly chooses $r' \in \mathbb{Z}_p$. He/She implicitly sets $r = x_2(r' - x_1)$ and computes $D = g^{\beta}g^{r} = g^{x_2(\beta'+x_1)}g^{x_2(r'-x_1)} = g^{x_2\beta'+x_2r'} = X_2{}^{\beta'+r'}$. $\mathcal{B}$ randomly picks $r_j' \in \mathbb{Z}_p$ and implicitly sets $r_j = x_2r_j'$ for each attribute $j \in S_i$. He/She computes $D_j = g^{\frac{r}{\alpha}}H_1(j)^{\frac{r_j}{\alpha}} = g^{\frac{x_2(r'-x_1)}{\alpha}}H_1(j)^{\frac{x_2r_j'}{\alpha}} = g^{r'-x_1}H_1(j)^{r_j'} = \frac{g^{r'}H_1(j)^{r_j'}}{X_1}$, $D_j' = g^{r_j} = g^{x_2r_j'} = X_2{}^{r_j'}$. Finally, He/She delivers the secret key $SK = (D, \forall j \in S_i : D_j, D_j')$ to $\mathcal{A}$.

Challenge: Adversary $\mathcal{A}$ sends two messages m_0 and m_1 with equal length to $\mathcal{B}$. $\mathcal{B}$ flips a fair coin $\xi \in \{0,1\}$. Next, he/she selects the message m_{ξ} to encrypt under $\mathbb{A}^*$. $\mathcal{B}$ sets $g^t = g^{x_3} = X_3$ and randomly chooses $\Omega_i \in \mathbb{G}_1(i = 1,2,\cdots,k-1)$. Finally, he/she computes the ciphertext CT^* as follows.

$$C_V = m_{\xi}(e(g,g)^{\beta t}) = m_{\xi}e(g,g)^{x_2(\beta'+x_1)x_3} = m_{\xi}e(X_2,X_3)^{\beta'}Y$$

$$C_{V_i} = \Omega_i(i = 1,\ 2,\ \cdots,\ k-1)$$

$$C_{V_k} = \frac{X_3}{\prod_{i=1}^{k-1}\Omega_i}$$

$\mathcal{B}$ delivers $CT^* = \{C_V, \{C_{V_i}\}\}$ $(i = 1,\ 2,\ \cdots,\ k)$ to $\mathcal{A}$.

Query phase II: This phase is similar to phase I.

Guess: $\mathcal{A}$ gives its guess ξ' about ξ.

If $\xi \neq \xi'$, $\mathcal{B}$ returns 1 to indicate $Y = W$. In this case, the adversary $\mathcal{A}$ cannot predict the correct ciphertext.

$$Pr[\mathcal{B}(g^{x_1}, g^{x_2}, g^{x_3}, Y = W) = 0] = 1/2$$

If $\xi = \xi'$, $\mathcal{B}$ returns 0 to indicate $Y = e(g,g)^{x_1x_2x_3}$. In this case, the adversary $\mathcal{A}$ can predict the correct ciphertext. Simulator $\mathcal{B}$ has an advantage $1/2 + \varepsilon$ to break the DBDH assumption. Here ε is the advantage of $\mathcal{A}$ giving the right answer.

$$Pr[\mathcal{B}(g^{x_1}, g^{x_2}, g^{x_3}, Y = e(g,g)^{x_1x_2x_3}) = 0] = 1/2 + \varepsilon$$

Thus, the overall advantage of simulator $\mathcal{B}$ in winning this security game is

$$Adv_{\mathcal{B}} = (1/2)\times(1/2) + (1/2)\times(1/2+\varepsilon) - 1/2 = \varepsilon/2 .$$

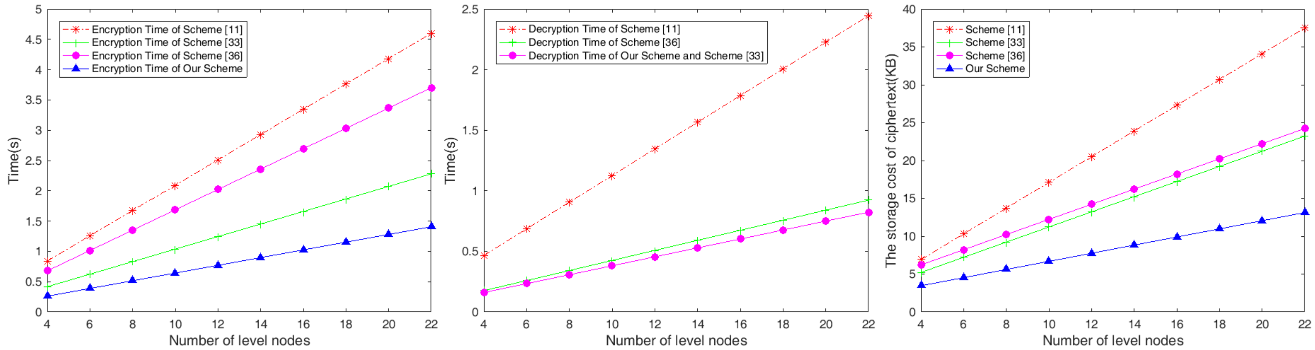
6 PERFORMANCE ANALYSIS

We present a basic attribute based encryption scheme with shared decryption and an improved scheme. Our scheme is constructed from scheme [11]. Different from scheme [11], an integrated access tree is used as the access structure in our schemes. In addition, an integrated access tree is also used in scheme [33,36]. We compare our schemes with them [11,33,36] in terms of the cost of the storage and computation.

6.1 Theoretical Analysis

Suppose the integrated access structure $\mathbb{A}$ includes k level nodes. Let $\lvert\mathbb{A}_{Id_R}\rvert$ and $\lvert\mathbb{A}_{Id_i}\rvert$ denote the total number of the leaf nodes in $\mathbb{A}$ and in the subtree rooted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2020.3043950, IEEE Transactions on Computers



(a) Encryption time

(b) Decryption time

(c) Ciphertext storage space

Fig. 5. Experimental results.

at level node V_i in $\mathbb{A}$ respectively. Let $E_{\mathbb{G}_1}$ and $E_{\mathbb{G}_T}$ represent the running time of an exponentiation calculation in group $\mathbb{G}_1$ and $\mathbb{G}_T$ respectively, and here we ignore the running time of the multiplication operation in the two groups. We use E_e to represent the running time of a pairing computation over elliptic curves. $|S|$ denotes the number of the attributes of a user. The running time of the hash computation on H is denoted as E_H. The length of the element in $\mathbb{G}_1, \mathbb{G}_T, \mathbb{Z}_p$ is denoted as $L_{\mathbb{G}_1}, L_{\mathbb{G}_T}, L_{\mathbb{Z}_p}$ respectively. In addition, $|\mathbb{A}_{tn}|$ denotes the number of the transport nodes of the access tree in scheme [33,36]. j denotes the maximum number of the child nodes in all transport nodes in the scheme [33,36]. l denotes the length of the session key in scheme [36]. We compare the proposed schemes with CP-ABE scheme [11,33,36] with respect to storage cost and computational overhead, without considering the cost of construction and storage of the access structure.

TABLE 2 is the result of the comparison between our schemes and scheme [11,33,36]. It shows that the cost of the encryption, decryption and storage is mainly impacted by the number of level nodes in these schemes. The encryption time of our schemes and scheme [11,33,36] all increases linearly with the number k of the level nodes in the access tree. Compared with scheme [11,33,36], the encryption cost of both the CP-ABE-SD scheme and the improved CP-ABE-SD scheme has a competitive advantage. Similarly, the decryption cost of all schemes also increases linearly with k. The advantage of our CP-ABE-SD scheme is obvious compared with scheme [11,33,36]. The length of the private key in our two schemes is the same as scheme [11,33,36], and it increases linearly with the number of the attributes of the user. Furthermore, the length of the ciphertext CT in these schemes also increases linearly with k. Obviously, our CP-ABE-SD scheme have less storage cost than scheme [11,33,36].

6.2 Experimental Analysis

For better performance analysis, we implement scheme [11,33,36] and our CP-ABE-SD scheme according to the Stanford Pairing-Based Cryptography Library [44,45] and the CP-ABE toolkit. The experiment is performed by a computer configured as 64-bit Windows 7 OS, 4 GB RAM, and Intel(R) Core (TM) i7 CPU @2.3GHz. We use a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over the field $\mathbb{F}_q$, where q is 512 bits. We implement two schemes on the group. Our algorithm is implemented in C language, and the collusion-resistant hash functions in two schemes come from SHA-256. These experiments are carried out 100 times independently and we pick their averages.

In the simulation, we assume that all threshold gates used in the access structure are AND gates, which is the worst case of the algorithm. We construct the integrated access tree as scheme [37]. For comparison purposes, we assume the number of the level nodes is $k = \{4, 6, 8, 10, 12, 14, 16, 18, 20, 22\}$ respectively. We set the size of the session key in scheme [36] is $l = 160$ bits. Each simulation is executed completely independent to each other. Fig. 5 gives the computational overhead of encryption and decryption stages under different number of the level nodes in two schemes. Obviously, compared with scheme [11,33,36], our scheme has lower costs in these two stages. Fig. 5 also shows that our scheme has an advantage over that of scheme [11,33,36] in terms of storage costs.

7 Conclusions

We provide two ciphertext-policy attribute based encryption schemes with shared decryption. There are two kinds of data users in our schemes. For an authorized user, he/she can recover the message independently. When the authorized user cannot decrypt the ciphertext in time for some reason, these semi-authorized users can cooperate to decrypt the ciphertext to replace the authorized user. An integrated access tree is used in proposed schemes to improve the efficiency of the schemes. The security for our schemes is proved under the DBDH assumption. The experimental result shows that CP-ABE-SD scheme is better than scheme [11,33,36] in terms of storage cost and computational overhead.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (61972095, U1736112, 61772009, 61902140, 61822202, 61872089), the Anhui Provincial Natural Science Foundation under Grant No. 1908085QF288.

REFERENCES

[1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, “Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 337–368, 2014.

[2] J. Aikat et al., “Rethinking security in the era of cloud computing,” *IEEE Security Privacy*, vol. 15, no. 3, pp. 60-69, Jun. 2017.

[3] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Transactions on Cloud Computing*, DOI: 10.1109/TCC.2019.2929045.

[4] J. Li, H. Yan, and Y. Zhang, “Certificateless public integrity checking of group shared data on cloud storage,” *IEEE Transactions on Services Computing*, to be published. DOI 10.1109/TSC.2018.2789893.

[5] H. Yan, J. Li, and J. Han, “A novel efficient remote data possession checking protocol in cloud storage,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 78-88, Jan. 2017.

[6] H. Yan, J. Li, and Y Zhang, “Remote data checking with designated verifier in cloud storage,” *IEEE Systems Journal*, vol. 14, no. 2, pp. 1788-1797, 2020.

[7] J. Li, H. Yan, and Y. Zhang, “Identity-based privacy preserving remote data integrity checking for cloud storage,” *IEEE Systems Journal*. DOI:10.1109/JSYST.2020.2978146.

[8] L. Zhang, H. Xiong, Q. Huang, J. Li, K. K. Raymond Choo, and J. Li, “Cryptographic solutions for cloud storage: challenges and research opportunities,” *IEEE Transactions on Services Computing*, DOI: 10.1109/TSC.2019.2937764.

[9] A. Sahai and B. Waters, "Fuzzy identity based encryption," *Advances in Cryptology-Eurocrypt 2005, Lecture Notes in Computer Science, vol. 3494, Springer, 2005*, pp. 457-473.

[10] V. Goyal, O. Pandey, A. Sahai, and Brent Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc. 13th ACM Conference on Computer and Communications Security*, 2006, pp. 89–98.

[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *IEEE Symposium on Security and Privacy*, vol. 2008, pp. 321-334, Jun. 2007.

[12] J. Lai, R. H. Deng, and Y. Li, “Expressive CP-ABE with partially hidden access structures,” *Proc. 7th ACM Symposium on Information, Computer and Communications Security*, pp. 18-19, 2012.

[13] S. Yu, K. Ren, and W. Lou, “Attribute-based content distribution with hidden policy,” *Proc. IEEE 4th Workshop on Secure Network Protocols*, pp. 39-44, 2008.

[14] N. Doshi and D. Jinwala, “Hidden access structure ciphertext policy attribute based encryption with constant length ciphertext,” *IEEE international Conference on Computer and Communication Technology*, Nov. 2011, pp. 515–523.

[15] J. Li, Y. Zhang, J. Ning, X. Huang, G. S.Poh, and D. Wang, “Attribute based encryption with privacy protection and accountability for cloudIoT,” *IEEE Transactions on Cloud Computing*, 2019, DOI:10.1109/TCC.2020.2975184.

[16] M. Chase and S. S. M. Chow, “Improving privacy and security in multi-authority attribute-based encryption,” *Proc. 14th ACM conference on Computer and Communications Security*, pp. 121-130, 2009.

[17] 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, pp. 487–497, Nov. 2015.

[18] J. Li, W. Yao, J. Han, et al, “User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage,” *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767-1777, Jun. 2018.

[19] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: effective data access control for multi-authority cloud storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1790-1801, Nov. 2013.

[20] Y. Miao, R. Deng, X. Liu, K. R. Choo, H. Wu, and H. Li, "Multi-authority attribute-based keyword search over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, DOI: 10.1109/TDSC. 2019.2935044.

[21] K. Yang and X. Jia, "Expressive efficient and revocable data access control for multi-authority cloud storage", *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1735-1744, Jul. 2014.

[22] J. Li, Y. Wang, Y. Zhang, and J. Han, “Full verifiability for outsourced decryption in attribute based encryption,” *IEEE Transactions on Services Computing*, to be published. DOI: 10.1109/TSC.2017.2710190.

[23] J. Li, J, F. Sha, Y. Zhang, X. Huang, and J. Shen, “Verifiable outsourced Decryption of Attribute-Based Encryption with Constant Ciphertext Length,” *Secur. Commun. Netw*, to be published. DOI: 10.1155/2017/ 3596205.

[24] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the decryption of ABE ciphertexts,” *Proc. 20th USENIX Security Symposium*, pp. 34-34, 2011.

[25] J. Li, X. Lin, Y. Zhang, and J. Han, “KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage,” *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715-725, 2017.

[26] Y. Lu, J. Li, and Y. Zhang, “Privacy-preserving and pairing-free multi-recipient certificateless encryption with keyword search for cloud-assisted IIoTs,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2553-2562, 2020.

[27] J. Li, Q. Yu, and Y. Zhang, “Key-policy attribute-based encryption against continual auxiliary input leakage,” *Information Sciences*, vol. 470, pp. 175–188, 2019.

[28] J. Ning, Z. Cao, X. Dong, H. Ma, L. Wei, and K. Liang, “Auditable σ-times outsourced attribute-based encryption for access control in cloud computing,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94-105, Aug. 2017.

[29] Y. Miao, X. Liu, K. K. Raymond Choo, R. H. Deng, J. Li, H. Li, and J. Ma, “Privacy-preserving attribute-based keyword search in shared multi-owner setting,” *IEEE Transactions on Dependable and Secure Computing*, 2019. DOI: 10.1109/TDSC.2019.2897675.

[30] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, “White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1274-1288, Jun. 2015.

[31] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, “Flexible and fine-grained attribute-based data storage in cloud computing,” *IEEE Trans. Service Comput.*, vol. 10, no. 5, pp. 785-796, 2017.

[32] Y. Guo, J. Li, Y. Zhang, and J. Shen, “Hierarchical attribute-based encryption with continuous auxiliary inputs leakage,” *Security and Communication Networks*, vol. 9, no.18, pp. 4852-4862, Dec. 2016.

[33] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265-1277, Jun. 2016.

[34] Z. Zhang, C. Li, B. Gupta, and D. Niu, “Efficient compressed ciphertext length scheme using multi-authority CP-ABE for hierarchical attributes,” *IEEE Access*, vol. 6, pp. 38273-38284, 2018. DOI: 10.1109/ACCESS.2018. 2854600.

[35] R. Guo, X. Li, D. Zheng, et al, “An attribute-based encryption scheme with multiple authorities on hierarchical personal health record in cloud,” *The Journal of Supercomputing*, pp. 1-20, 2018.

[36] J. Li, N. Chen, and Y. Zhang, "Extended file hierarchy access control scheme with attribute based encryption in cloud computing," *IEEE Transactions on Emerging Topics in Computing*. DOI: 10.1109/TETC.2019. 2904637.

[37] J. Fu and N. Wang, "A practical attribute-based document collection hierarchical encryption scheme in cloud computing," *IEEE Access*, vol. 7, pp. 36218-36232, 2019. DOI: 10.1109/ACCESS.2019.2905346.

[38] Y. Lu, J. Li, and Y. Zhang, “Secure channel free certificate-based searchable encryption withstanding outside and inside keyword

guessing attacks," *IEEE Transactions on Services Computing*, DOI: 10.1109/ TSC. 2019.2910113.

[39] J. Coron, T. Lepoint, and M. Tibouchi, "Practical multilinear maps over the integers," *in Advances in Cryptology–Crypto 2013, Springer, Aug. 2013*, pp. 476–493.

[40] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *Society for Industrial and Applied Mathematics*, vol. 33, no. 1, pp. 167-226, Jan. 2004.

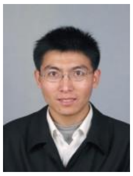
[41] J. Song, Y. Liu, J. Shao, and C. Tang, "A dynamic membership data aggregation (DMDA) protocol for smart grid," *IEEE Systems Journal*, DOI: 10.1109/JSYST.2019.2912415.

[42] C. Wang, C. Wang, Z. Wang, X. Ye, J. X. Yu and B. Wang, "Deep direct: learning directions of social ties with edge-based network embedding (Extended Abstract)," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, DOI: 10.1109/TKDE.2018.2877748.

[43] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Information Sciences*, vol. 484, pp. 113–134, 2019.

[44] B. Lynn, "The stanford pairing based crypto library," May 7, 2014, [Online]. Availiable: http://crypto.stanford.edu/pbc

[45] V. Shoup, "A proposal for an iso standard for public key encryption (version 2.1),"2001, [Online]. Availiable: http://eprint.iacr.org/ 2001/112



Ningyu Chen received the B.S. degrees in mathematics & applied mathematics form Jiangxi Normal university, China, in 2001 and M.S. degrees in computer science & technology from Shanghai University, China, in 2005. He is currently a Ph.D student of Hohai University, China. His research interests include cloud computing security and applied cryptography.

Jiguo Li received his B.S. degree in mathematics from Heilongjiang University, Harbin, China in 1996, M.S. degree in mathematics and Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China in 2000 and 2003, respectively. During 2006.9-2007.3, he was a visiting scholar at Centre for Computer and Information Security Research, School of Computer Science & Software Engineering, University of Wollongong, Australia. During 2013.2-2014.1, he was a visiting scholar in Institute for Cyber Security in the University of Texas at San Antonio. He is currently a professor with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China and College of Computer and Information, Hohai University, Nanjing, China. His research interests include cryptography and information security, cloud computing, wireless security and trusted computing etc. He has published over 150 research papers in refereed international conferences and journals. His work has been cited more than 3000 times at Google Scholar. He has served as program committee member in over 30 international conferences and served as the reviewers in over 90 international journals and conferences.

Yichen Zhang received the Ph.D. degree in the College of Computer and Information, Hohai University, Nanjing, China in 2015. She is currently an associate professor with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China. Her research interests include cryptography, network security. She has published over 30 research papers in refereed international conferences and journals.

Yuyan Guo received her B.S. and M.S. degrees from Huaibei Normal University, Huaibei, China in 2005 and 2008, respectively, and Ph.D. degree in computer science from Hohai University, Nanjing, China in 2016. She is currently a lecturer with the School of Computer Science and Technology, Huaibei Normal University, Huaibei, China. Her research interests include cryptography and information security, cloud computing and trusted computing etc. She has published over 10 research papers in refereed international conferences and journals.