Efficient CP-ABE Scheme Resistant to Key Leakage for Secure Cloud-Fog Computing

Fengwei Cheng[1], Sai Ji[1,2], Chin-Feng Lai[3*]

[1] School of Computer Science, Nanjing University of Information Science & Technology, China
[2] College of Information Engineering, Suqian University, China
[3] Department of Engineering Science, National Cheng Kung University, Taiwan
chengfw198@126.com, jisai@nuist.edu.cn, cinfon@ieee.org

Abstract

With the evolution of the Internet, people's lives are growing increasingly intelligent, and massive amounts of data are also generated. The combination of cloud computing and fog computing can store and process it efficiently. However, a new and urgent challenge has emerged about how to ensure that data in cloud and fog nodes are accessed securely. In order to address this issue, we propose an efficient anti-key leakage CP-ABE (EAL-CP-ABE) scheme based on cloud-fog computing. In our scheme, we achieve the identity tracing of the authorized user who leaked the private key by means of white-box tracing and the maximum number of user accesses. Meanwhile, we design a reward and punishment mechanism which rewards those with a good reputation and revokes authorized users who have exceeded the access limit or sold private keys to further reduce the possibility of key leakage. Due to limited local resources, we use fog nodes to achieve outsourced decryption of ciphertext. In addition, the detailed security analysis is provided to prove that the EAL-CP-ABE scheme is IND-CPA secure and traceable based on the hardness assumptions. Finally, we demonstrate that the EAL-CP-ABE scheme is efficient through the performance analysis.

Keywords: Access control, CP-ABE, Traceability, Privacy-preserving

1 Introduction

Along with the continuous innovation of technology for the Internet and IoT, cloud computing is constantly developing and maturing. Cloud services gradually permeate into healthcare, transportation and other fields from the Internet industry [1]. Cloud computing uses virtual technology to manage resources and achieves efficient processing of massive data through distributed data storage technology and parallel computing pattern, which liberates enterprises from the technical problems of handling huge data and effectively helps them reduce costs and optimize efficiency [2-3]. It is designed to offer users unrestricted real-time data storage whenever and wherever they need it [4-5]. However, the enormous quantity of data and the growing number of devices accessing the network led to overload and high latency in data transmission. Thus, fog computing emerges as the times require. In 2012, Cisco introduced the notion of fog computing. It is a distributed computing platform with a pool of resources that is comprised of one or more ubiquitous fog devices [6]. And it provides storage, computing and network services among intelligent devices and cloud servers, which are usually embedded at the network edge. Rather than substituting for cloud computing, fog computing is regarded as an extension of cloud computing, which can respond to the user's demand for operations in greater real-time.

Since cloud and fog nodes are not fully trusted, it is extremely hard to guarantee the confidentiality of the information stored in them [7-8]. There has been a surge in data privacy concerns, including data leakage and user privacy exposure in recent years [9-10]. So, we cannot only rely on cloud or fog service providers to offer data access control as they may not be able to enforce it fairly for profit. The urgent issue currently is how to ensure that the data information stored in the honest-and-curious cloud or fog nodes is securely accessed. The attribute-based encryption (ABE) achieves one-to-many communication and fine-grained access control systems through attributes set and access structure, which can solve the above-mentioned issue. The ABE mechanism can be classified into two models and one of them is key-policy ABE (KP-ABE) [11]. The set of attributes is used to encrypt the ciphertext, and the access policy is an element of the private key of this type. It is suitable for static scenes such as pay-TV channels. And the other is ciphertext-policy ABE (CP-ABE) [12]. Compared to the former, the ciphertext in the CP-ABE scheme is obtained by encrypting the plaintext with the access policy, and the private key involves the set of attributes. It has received more attention and research than KP-ABE because the data owner takes charge of formulating the access policy on this approach.

However, in the actual scenario, some malicious authorized users may sell their private keys for profit, which leads to the leakage of private keys. One possible solution to resist key leakage is to trace the hidden malicious users. So, Ning et al. [13] added the large space and white-box traceability features to the original CP-ABE scheme [12], which enables effective traceability of malicious users. Nevertheless, the scheme merely implements the tracing of malicious users without the revocation operation, which is essential to guarantee the security of the system. For this, Li et al. [14] brought up a CP-ABE scheme with the properties of traitor traceability and revocation. However, their scheme involves a relatively high computational cost in the decryption phase, which is not feasible for devices with limited

*Corresponding Author: Chin-Feng Lai; E-mail: cinfon@ieee.org

DOI: 10.53106/160792642022122307002

computing resources. To address the issue, Premkamal et al. [15] brought up a novel traceable CP-ABE through a proxy server that traces malicious users who access data exceptionally. In addition, a privacy-preserving CP-ABE scheme put forward by Han et al. [16] can simultaneously realize the tracing, revocation and ciphertext-update features. Although all of the above schemes tackle the issue of key leakage to some degree, these schemes are not sufficient regarding security or efficiency. There is still a common issue that they fail to consider to actively identify potential users who may compromise keys.

In this work, we first consider that if the authorized user's private key is in the hands of different people through sale, the frequency of accessing data in the cloud owing to the leaked private key will increase. We establish a maximum access mechanism that restricts the number of times a user can access data in the cloud during a period of time, and once the number of accesses exceeds the set value, we can consider that the user is at the risk of the private key leakage. In the meantime, we will trace the identity of the user. Then, we add additional protection based on passive tracing. When the private key is found to be sold in the market, we trace the key owner's identity. Furthermore, we set up a reward and punishment mechanism to further resist key leakage. In terms of rewards, we regularly give rewards to users with a good reputation, such as coupons, etc. In terms of punishment, users at the risk of key leakage cannot get any rewards. Also, if the times that users access data in the cloud is too frequent, we blacklist them, i.e., revoke the users' access permission so that they cannot access the resources. Finally, taking into account the limitations of traditional cloud, such as low bandwidth, network instability and the high computational overhead of decrypting locally [17], we realized higher processing efficiency and lower computational overhead through the characteristics of low latency and decentralized layout of the fog nodes.

1.1 Contributions

We put forward an efficient anti-key leakage CP-ABE (EAL-CP-ABE) scheme based on cloud-fog computing to address the issue of malicious authorized users frequently leaking their private keys. The main contributions of EAL-CP-ABE scheme are shown as follows:

- **We present an anti-key leakage scheme about two types of protection: active and passive protection.** In active protection, we set a limited access control mechanism, which means that authorized users can only be able to visit the cloud data for limited times in a period of time, so as to restrict the scope of key leakage, and once the threshold is surpassed, the trusted authority will be notified to trace that malicious user's identity. In passive protection, if we find a private key that is sold in the market, we can trace the key owner's identity through a trusted authority.
- **We propose the outsourced decryption mechanism through fog nodes.** If and only if that user's access permission has not been revoked and his or her attributes set fulfills the access policy, the fog node will partially decrypt ciphertext stored in the cloud and send it to the data user so as to lower the computation overhead. Furthermore, considering the risk of outsourced decryption, we integrate validation parameters into the scheme to ensure the integrity of the data.
- **We put forward a reward and penalty mechanism to reward users with a good reputation and penalize malicious users.** And we firstly set the initial values t_0, t_1 and $t_0 < t_1$. If the times that the data users access cloud data in a period of time are less than t_0, then the users can be given appropriate benefits, such as rewarding vouchers. Otherwise, we do not offer any rewards to the users. In addition, if the times that the users access cloud data in a period of time is more than t_1 or the user's private key is found for sale, then his or her access permission will be revoked. This can, to a certain extent, reduce the possibility that authorized users sell their private keys, and thus it can indirectly resist key leakage.

1.2 Related Work

The concept of attribute-based encryption (ABE) [18] was firstly proposed in 2005, which is the foundation of identity-based encryption. The next year, Goyal et al. [11] designed the key-policy ABE scheme on the basis of their scheme. Then, Bethencourt et al. [12] put forward the ciphertext-policy ABE scheme that enables encrypted data security though the cloud cannot be trustworthy. After that, scholars have carried out considerable research on CP-ABE because it achieves fine-grained access control and is well suited to real-world scenarios. Depending on the requirements of different applications, researchers have added other features of the CP-ABE schemes. Here we focus on the characteristics of malicious user traceability, outsourced decryption and user revocation relevant to our work.

Malicious user traceability. In [13], Ning et al. brought up two practical large universe CP-ABE schemes which have the feature of white-box traceability. The latter implements constant storage for traitor traceability based on the former. Their schemes are constructed by prime order bilinear groups and proved secure on the basis of the hardness assumption. In [19], Yan et al. presented a novel anti-key delegation abuse and traceable attribute-based encryption scheme, which achieves higher performance by taking advantage of a traceability method based on a short signature structure. Han et al. [16] put forward a hidden-policy CP-ABE scheme that implements the functions of revocation and white-box traceability. Their scheme is proved IND-CPA secure on the basis of the hardness assumptions in the standard model.

Outsourced decryption. Premkamal et al. [20] brought up a verifiable outsourced CP-ABE scheme in terms of data privacy in cloud computing. In addition, the scheme restricts users' data access to satisfy the demands of business applications. A fair outsourced decryption ABE scheme was put forward by Zheng et al. [21] that utilizes blockchain and sampling technology. The scheme used smart contracts and sampling technology to ensure fairness between agents outsourcing and users. Feng et al. [22] designed an edge-smart IoV parallel outsourced decryption of the ABE model. The scheme can improve decryption efficiency effectively based on Spark and MapReduce methods.

User revocation. To achieve user revocation, Ramu et al. [23] optimized the original CP-ABE scheme so as to realize user revocation. The scheme implements fine-grained user revocation by means of an immediate attribute modification method. Sethia et al. [24] proposed scalable revocation and constant ciphertext length for CP-ABE scheme that is secure on the basis of the hardness assumption against CCA attacks. Ge et al. [25] designed a specific revocable ABE scheme that enables cloud storage to revoke the recipient with the data integrity directly. And the scheme takes advantage of updating the access policy to achieve user revocation.

All of the above schemes have made great contributions to guaranteeing that data stored in the cloud or in fog nodes can be accessed securely. But there is still a lack of a more comprehensive scheme. Therefore, we proposed an efficient anti-key leakage CP-ABE (EAL-CP-ABE) scheme based on cloud-fog computing, which can trace and revoke malicious users who leak private keys as well as reduce local computational overhead.

1.3 Organization

Section 2 presents some important preliminaries about this paper, including access structure, linear secret sharing scheme and complexity assumptions. Section 3 shows the overview of system and security of the proposed EAL-CP-ABE scheme. Then, we present the construction of our EAL-CP-ABE scheme in section 4. Section 5 gives the correctness and the security analysis of our EAL-CP-ABE scheme. Furthermore, the performance analysis of our EAL-CP-ABE scheme is illustrated in section 6. At last, we draw the conclusion of this paper in section 7.

2 Preliminaries

2.1 Access Structure

Given that $A = \{A_1, A_2, ..., A_n\}$ is a set of groups and a collection $\mathbb{C} \subseteq 2^{\{A_1, A_2, ..., \mathrm{A_n}\}}$. An access structure [26] is a collection $\mathbb{C}$ with non-empty subsets, i.e., $\mathbb{C} \subseteq 2^{A} \setminus \{\phi\}$. Particularly, for $\forall E, F:$ in case that $E \in \mathbb{C}$ and $E \subseteq F$, then $F \in \mathbb{C}$ is satisfied, we define $\mathbb{C}$ as monotone. We call the sets which are in collection $\mathbb{C}$ as the authorized sets while the sets which are not in collection $\mathbb{C}$ are referred to as the unauthorized sets.

2.2 Linear Secret Sharing Scheme

We utilize LSSS [26] to build an access policy as (T, f), where the matrix T is m rows and n columns and f associates rows of the matrix T with attributes. A LSSS scheme contains the following two steps:

Share $((\mathrm{T}, f), s)$ **:** The share step is to split the secret value $s \in Z_p$ by means of the matrix T. Given a vector $\vec{\mu} = (s, \mu_2, \mu_3, ..., \mu_n)$, where $\mu_2, \mu_3, ..., \mu_n \in Z_p$ are randomly selected to share the secret value s. Let $\lambda_i = \mathrm{T}_i \cdot \vec{\mu}$ where T_i means matrix's i-th row vector.

Reconstruction $(\lambda_1, \lambda_2, ..., \lambda_m, (\mathrm{T}, f))$ **:** The reconstruction step is to recover the secret value s from the shares of the parties. Given the authorized set S and $I = \{i : f(i) \in S\} \subseteq \{1, 2, ..., m\}$. There remains constant $\{w_i \in Z_p\}_{i \in I}$ for which $\sum_{i \in I} w_i \mathrm{T}_i = (1, 0, 0, ..., 0)$ holds. In other words, we can recover the secret value s according to $\sum_{i \in I} w_i \lambda_i = s$.

2.3 Complexity Assumptions

We use decisional q-parallel BDHE assumption [27] and l-SDH assumption [13] to prove the proposed EAL-CP-ABE scheme IND-CPA secure and traceable. We give $\hat{G}, \hat{G}_T$ which are two multiplicative cyclic groups of prime order p. The generator element of $\hat{G}$ is $\hat{g}$ and e is a bilinear map as $e : \hat{G} \times \hat{G} \to \hat{G}_T$. The detailed definition of these hardness assumptions is as follows.

Assumption 1. (q-parallel BDHE): we randomly choose $s, m, t_1, t_2, ..., t_q \in Z_p$. Then, construct $\vec{Q} = \{$

$$\hat{g}, \hat{g}^{s}, \hat{g}^{m}, ..., \hat{g}^{m^q}; \hat{g}^{m^{q+2}}, ..., \hat{g}^{m^{2q}}$$

$$\forall_{1 \le j \le q} \ \hat{g}^{s \cdot t_j}, \hat{g}^{m/t_j}, ..., \hat{g}^{m^q/t_j}; \hat{g}^{m^{q+2}/t_j}, ..., \hat{g}^{m^{2q}/t_j}$$

$$\forall_{1 \le j, k \le q, k \ne j} \ \hat{g}^{m \cdot s \cdot t_k / t_j}, ..., \hat{g}^{m^q \cdot s \cdot t_k / t_j} \}.$$

This assumption means no PPT adversary is able to differentiate between $e(\hat{g}, \hat{g})^{m^{q+1}s}$ and a random value $T \in G_T$.

Assumption 2. (l-SDH): we randomly choose $x \in Z_p^*$. Then, construct $(t+1)$-tuple $(\hat{g}, \hat{g}^{x}, \hat{g}^{x^2}, ..., \hat{g}^{x^t})$. This assumption means no PPT adversary is able to compute the pair $(b, \hat{g}^{1/(x+b)})$.

3 The Overview of System and Security

3.1 System Model

Just as displayed in Figure 1, the system model is composed of five entities which are the Trusted Authority (TA), Cloud Service Provider (CSP), Data Owner (DO), Data User (DU) and Fog Node (FN). The details are listed as follows.

Trusted Authority (TA): TA publishes the system parameters. And it is a fully trusted entity to create a unique private key for each DU. Furthermore, it realizes the tracing of malicious DU and updates unrevoked DUs' private keys and the list of the partial private key. Then it transmits the updated list of partial private key to FN for achieving revocation of malicious DUs.

Cloud Service Provider (CSP): CSP supplies storage space for ciphertext and updates the list of times that DU accesses data from CSP through the download request, and if the times that DU accesses data from CSP exceeds the threshold value, CSP will send the blinded id of DU to TA for tracing the identity of DU. In addition, CSP regularly rewards DUs with good credit.

Data Owner (DO): DO firstly encrypt the file by the symmetric key. Then, DO designates an access policy and encrypts the symmetric key using attribute-based encryption.

Finally, DO uploads the encrypted ciphertext tuple, which contains symmetric encrypted file and ciphertext based on attribute encryption to the cloud.

Data User (DU): DU initiates a download request to FN. If DU's attributes set fulfills the access policy and his or her access permission is not revoked, DU can download the semi-decrypted ciphertext. Then DU utilizes his or her private key to decrypt the semi-decrypted ciphertext for the corresponding symmetric key. Finally, DU decrypts the encrypted file to obtain the plaintext file by the symmetric key.

Fog Node (FN): FN verifies DU's identity according to the information of the download request sent by DU. Meanwhile, FN checks whether it exists in the list of the partial private key. If not, the request is sent to the cloud. Otherwise, the download request is ignored.

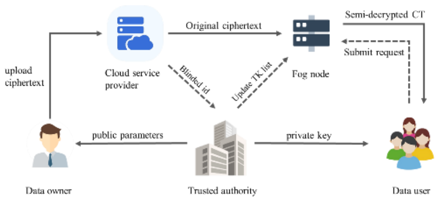

Figure 1. The proposed system model

3.2 The Brief Workflow of EAL-CP-ABE Scheme

The proposed EAL-CP-ABE scheme resistant to key leakage based on cloud-fog computing includes several algorithms as follows:

$Init(\lambda,\mathbb{N})\rightarrow(Params,MSK)$: The Init step is run by TA. Given the security parameter λ and attribute space $\mathbb{N}$, and it generates the public parameters $Params$ and the master secret key MSK.

$KeyGen(Params,MSK,id,S)\rightarrow(SK,TK)$: The KeyGen step is run by TA. Given $Params$, MSK, DU's identity id and the set S of DU's attributes, and it generates the private key SK and the partial private key PSK for each DU.

$Encrypt(Params,m,(\mathrm{T},f))\rightarrow CT$: The Encrypt step is run by DO. Given $Params$, the plaintext m and the access policy (T,f) which DO assigns, and it outputs the ciphertext CT. And DO uploads it to CSP.

$Decrypt(CT,TK,SK)\rightarrow m$: The Decrypt step is divided into two parts. First, FN decrypts the original ciphertext CT by the partial private key PSK to derive the semi-decrypted CT. After that, DU utilizes his or her private key SK to decrypt the semi-decrypted CT and obtain the plaintext m.

$Trace(Params,SK\,/\,count)\rightarrow id\,/\,\perp$: The Trace step run by TA is divided into two cases: internal tracing and external tracing. When it is an internal tracing, TA traces DU's identity id based on Z_{id} where Z_{id} is sent to TA from CSP. In other words, the times that DU accesses the data from CSP has exceeded the threshold value in a period of time. When it is an external tracing, TA firstly checks whether SK is in well formed to confirm whether the private key SK should be traced. If the private key SK can pass the key sanity check, TA will trace DU's identity id based on the private key and output DU's identity id. Otherwise, the step outputs $\perp$.

$KeyUpdate(id)\rightarrow(SK',TK')$: The KeyUpdate step is run by TA. Given the identity id of malicious DU who is traced, and it returns the updated private key SK' and partial private key PSK' for the other DUs except for the identity id.

3.3 Threat Model and Security Demands

The threat model and security demands of EAL-CP-ABE scheme are introduced in the segment. We first put forward a threat model definition of the proposed scheme. The adversary attempts to take data from CSP without a valid private key. In addition, CSP and FN are semi-honest entities that will execute their assigned tasks honestly, but they will also try to gain some information that they are not authorized to obtain for their benefit. Finally, the malicious DUs that have been revoked try to acquire the data from CSP.

On the basis of the proposed threat model, we put forward security demands for EAL-CP-ABE scheme. On the one hand, we claim that TA is a fully trusted entity and strictly performs its appointed tasks. On the other hand, we claim that there is no collusion among CSP, FN and DUs. In general, only if DU's attributes set satisfies the access policy and his or her access permission is not revoked, he or she can access the data in the cloud.

3.4 Security Model

We present the security model of EAL-CP-ABE scheme including IND-CPA security and traceability.

(1) IND-CPA security. The EAL-CP-ABE scheme is IND-CPA secure if the adversary has the negligible advantage in the following game.

Init. The adversary selects a challenge access policy (T^{*},f^{*}).

Setup. The challenger generates the public parameters and sends it to the adversary.

Query-I. The adversary performs key queries about the set S of the attributes where $S\not\in\mathrm{T}^{*}$.

Challenge. The adversary chooses two equal plaintexts m_0,m_1 and passes them to the challenger. The challenger randomly selects $\beta\in\{0,1\}$ to compute the ciphertext and returns it to the adversary.

Query-II. The adversary performs key queries as in Query-I.

Guess. The adversary puts out the guess β'. If $\beta'=\beta$, the adversary will win the game.

The advantage of the adversary winning the game is defined as $\varepsilon=|\Pr[\beta'=\beta]-1/2|$.

(2) Traceability. The EAL-CP-ABE scheme is traceable if the adversary has the negligible advantage in the following games.

Setup. The challenger generates the public parameters and sends it to the adversary.

Key Query. The adversary submits $(id_1,S_1),(id_2,S_2),\ldots,(id_q,S_q)$ to the challenger for querying private keys. The challenger generates $\{SK_i\}_{i=1,2\ldots,q}$ and returns them to the adversary.

Key Forgery. The adversary outputs a private key SK_A. If $Trace(Params, SK) \notin \{\perp, id_1, id_2, ..., id_q\}$, the adversary will win the game.

The advantage of the adversary winning the game is defined as

$$\varepsilon = \Pr[Trace(Params, SK) \notin \{\perp, id_1, id_2, ..., id_q\}].$$

4 Construction

In this segment, we summarize our used notations in Table 1 for the EAL-CP-ABE scheme. Furthermore, we depict the detailed construction of the proposed EAL-CP-ABE scheme.

Table 1. Notations

Notation	Content
p	A prime number
$\hat{G}, \hat{G}_T$	multiplication cyclic groups of prime order p
$\hat{g}$	A generator of $\hat{G}$
e	A bilinear map as $e: \hat{G} \times \hat{G} \to \hat{G}_T$
$Params$	The public parameters
MSK	The master secret key
SK_{id}	DU's private key
PSK_{id}	Partial DU's private key
$UList$	List of DUs' identities
$PSKList$	List of the partial private key
TL	List of times that DU accesses data from CSP

4.1 Init

TA executes the Init step to generate the system parameters. Given the security parameter λ and attribute space $\mathbb{N}$. Firstly, TA randomly selects $h_1, h_2, ..., h_n, R \in \hat{G}, \alpha, a \in Z_p$ where n is the size of the attribute space and computes the values $e(\hat{g}, \hat{g})^{\alpha}$ and $\hat{g}^{a}$. In addition, TA selects hash function: $H: Z_p \to \hat{G}$. Then TA outputs $Params = \{p, \hat{g}, e(\hat{g}, \hat{g})^{\alpha}, \hat{g}^{a}, h_1, h_2, ..., h_n, R, H\}$ and $MSK = \{a, \alpha\}$. Finally, TA sends $Params$ to other entities and keeps MSK.

4.2 KenGen

TA performs the KenGen step to create the private key for each DU. Every DU holds a unique identity id in the system. Let $Params$, MSK, DU's identity id and the set $S \in \mathbb{N}$ of DU's attributes be used as inputs, and it initializes $UList$ and $PSKList$ as $\varnothing$. Then, TA randomly chooses $u, y, t \in Z_p$ where t is a unique random value and calculates the following parameters:

$$Z_{id} = H(id \,||\, t), L = y, L_0 = \hat{g}^{\alpha/(a+y)} R^u, L_1 = \hat{g}^u,$$
$$L_2 = \hat{g}^{au}, \{K_x = h_x^{(a+y)u}\}_{x \in S}.$$

Set the private key $SK_{id} = \{t, Z_{id}, L, L_0, L_1, L_2, \{K_x\}_{x \in S}\}$ for DU. In addition, TA computes $L_0' = \hat{g}^{\alpha/(a+y)t} R^u$ and sets the partial private key $PSK_{id} = \{Z_{id}, L, L_0', L_1, L_2, \{K_x\}_{x \in S}\}$. After that, TA keeps the list $UList = UList \cup \{Z_{id}, id\}$ of DUs' identities and updates the list $PSKList = PSKList \cup \{Z_{id}, TK_{id}\}$ of the partial private key. Finally, TA sends $PSKList$ to the fog nodes in order to implement the outsourced decryption and keeps $UList$.

For the purpose of active protection, TA updates the list $TL = TL \cup \{Z_{id}, count_{id}\}$ of times that DU accesses data from CSP and sends it to CSP. The $count_{id}$ is initialized to zero and the range is set between t_0 and t_1. If $count_{id} < t_0$, it means that DU has a good reputation and he or she will receive a reward such as a coupon. If $t_0 < count_{id} < t_1$, it means that DU has the potential to compromise private key and he or she will not receive any reward. If $count_{id} > t_1$, it means that the times that DU accesses data from CSP has exceeded the threshold value over a period of time and he or she will be withdrawn from access permission.

4.3 Encrypt

DO implements the Encrypt step and inputs $Params$, the plaintext document $File$ and the access policy (T, f), where the matrix T is l rows and n columns and f associates each row of the matrix T with each set of DU's attributes. DO randomly selects a symmetric key $\widetilde{SK} \in Z_p$ to encrypt plaintext document $File$ as the symmetric ciphertext $\widetilde{CT}$.

DO selects a random vector $\vec{\mu} = (s, \mu_2, \mu_3, ..., \mu_n)$ where $\mu_2, \mu_3, ..., \mu_n \in \mathrm{Z}_p$ to share the secret value $s \in Z_p$. And each share λ_i is calculated by $\lambda_i = \mathrm{T}_i \cdot \vec{\mu}$ where T_i means matrix's i-th row vector. At last, DO selects $r_1, r_2, ..., r_l \in Z_p$ to generate the tuple $(\widetilde{CT}, CT) = \{\widetilde{CT}, (M, f), C_0, C_1, C_2, \{B_{1,j}, B_{2,j}\}_{j \in [1,l]}\}$ as

$$C_0 = \widetilde{SK} \cdot e(\hat{g}, \hat{g})^{\alpha s}, C_1 = \hat{g}^{s}, C_2 = \hat{g}^{as}$$
$$B_{1,j} = R^{\lambda_j} h_{f(j)}^{-r_j}, B_{2,j} = \hat{g}^{r_j}.$$

For guaranteeing the integrity of the ciphertext and obtain the correct symmetric key $\widetilde{SK}$. DO computes $\bar{C} = H(\widetilde{SK})$ which is used by DU to authenticate the integrity of the ciphertext. Finally, DO uploads the ciphertext tuple $(\widetilde{CT}, CT, \bar{C})$ to CSP.

4.4 Decrypt

To obtain the encrypted files in the cloud, DU firstly sends a download request $\{Z_{id}, ver = Z_{id} \oplus L_1\}$ to FN. FN receives the request and searches for whether a transformation key exists based on the value of Z_{id} and verify $Z_{id} \overset{?}{=} ver \oplus L_1$. If the above condition holds, FN sends the request to CSP. Otherwise, FN ignores it. When CSP receives the request, it queries the value of $count$ in the list TL of times that DU accesses data from CSP according to Z_{id}. If $count < t_0$, then it will send the ciphertext tuple to FN and calculate

$count = count + 1$. If $count > t_1$, then CSP will stop sending the ciphertext tuple and send Z_{id} to TA. When FN receives the encrypted file, it will perform the semi-decryption of the ciphertext CT according to the partial private key PSK_{id}. DU's attributes set fulfills the access policy. There exists constant $\{\mathrm{w}_i\}_{i\in[1,l]}$ for which $\sum_{i\in I} w_i \cdot \mathrm{T}_i = (1,0,0,..,0)$. That means $\sum_{i\in I} w_i \cdot \lambda_i = s$. Then, FN computes

$$\begin{aligned}
Q &= \prod_{i\in I}[e(B_{1,i},(L_1)^L \cdot L_2)e(K_{f(i)},B_{2,i})]^{w_i} \\
&= \prod_{i\in I}[e(R^{\lambda_i} h_{f(i)}^{-r_i},\hat{g}^{(a+y)u})e(h_{f(i)}^{(a+y)u},\hat{g}^{r_i})]^{w_i} \\
&= \prod_{i\in I}[e(R^{\lambda_i},\hat{g}^{(a+y)u})]^{w_i} \\
&= e(R,\hat{g})^{(a+y)us} \\
V &= e(L_0{}',(C_1)^L \cdot C_2) \\
&= e(\hat{g}^{\alpha/(a+y)t}R^u,\hat{g}^{(a+y)s}) \\
&= e(\hat{g},\hat{g})^{\alpha s/t}e(R,\hat{g})^{(a+y)us} \\
D &= V/Q = e(\hat{g},\hat{g})^{\alpha s/t}
\end{aligned}$$

Then, FN transmits the ciphertext tuple $(\widetilde{CT}, D, \bar{C})$ which is partially decrypted to DU. After DU receives the semi-decrypted ciphertext tuple $(\widetilde{CT}, D, \bar{C})$, he or she computes $C_0/(D)^t = \widetilde{SK}$ and verifies $\bar{C} \overset{?}{=} H(\widetilde{SK})$. If this equation holds, DU decrypts $\widetilde{CT}$ with the symmetric key $\widetilde{SK}$ to obtain the plaintext document file $File$. Otherwise, the decryption abort.

4.5 Trace

The Trace step is divided into two cases: internal tracing and external tracing.

Internal tracing: when the cloud finds that $count_{id} > t_1$ in a period of time, it sends Z_{id} to TA, and TA queries the list $UList$ of DUs' identities according to Z_{id} for tracing DU's identity id.

External tracing: when someone's private key SK_{id} is found to be sold in the market, TA firstly judges whether the private key SK_{id} is well-formed in order to decide whether it is a forgery key. If the private key is not a forgery key, we need to trace the identity of DU's private key. The KeySanityCheck algorithm to check whether the well-formed of the private key is as follows,

$$L \in \mathrm{Z_p},\ Z_{id}, L_0, L_1, L_2, K_x \in \widehat{G}. \tag{1}$$

$$e(L_1, \hat{g}^a) = e(L_2, \hat{g}). \tag{2}$$

$$e(L_0, \hat{g}^L \cdot \hat{g}^a) = e(\hat{g}, \hat{g})^\alpha e(R, L_2 \cdot L_1{}^L). \tag{3}$$

$$\exists\ x \in S, s.t.\ e(K_x, \hat{g}) = e(L_1{}^L L_2, h_x). \tag{4}$$

In case that the private key SK_{id} fulfills the above equations $(1)-(4)$, it will pass the KeySanityCheck algorithm. Then TA queries the list $UList$ of DUs' identities according to Z_{id} for tracing DU's identity id. Otherwise, the algorithm outputs $\perp$.

4.6 KeyUpdate

After querying the identity id of malicious DU which has been traced, TA updates the other unrevoked user keys and the list of the partial private key to achieve the revocation of the malicious user.

Firstly, TA chooses a random $t' \in Z_p$ to compute new $SK_j' = \{t', Z_j' = H(j \| t'), L, L_0, L_1, L_2, \{K_x\}_{x\in S}\}_{j\notin id}$. Then, TA updates the list $UList' = UList \not\subset \{Z_{id}, id\}$ of DUs' identities, the list $PSKList' = \{Z_j', PSK_j'\}$ of the partial private key where $PSK_j' = \{Z_j', L, L_0{}'' = (L_0)^{1/t'}, L_1,$ $L_2, \{K_x\}_{x\in S}\}$, $Z_j' = H(j \| t')$, $j \notin id$ and the list $TL' = \{Z_j', count_j\}_{j\notin id}$ of times that DU accesses data from CSP.

5 Correctness and Security Analysis

In this segment, we give the correctness and security analysis of the EAL-CP-ABE scheme.

5.1 Correctness

In the Decrypt step, suppose that PSK_{id} and SK_{id} are generated by TA and the set of DU's attributes fulfills the access policy, we have

$$\begin{aligned}
Q &= \prod_{i\in I}[e(B_{1,i},(L_1)^L \cdot L_2)e(K_{f(i)},B_{2,i})]^{w_i} \\
&= \prod_{i\in I}[e(R^{\lambda_i} h_{f(i)}^{-r_i},(\hat{g}^u)^y \cdot \hat{g}^{au})e(h_{f(i)}^{(a+y)u},\hat{g}^{r_i})]^{w_i} \\
&= \prod_{i\in I}[e(R^{\lambda_i} h_{f(i)}^{-r_i},\hat{g}^{(a+y)u})e(h_{f(i)}^{(a+y)u},\hat{g}^{r_i})]^{w_i} \\
&= \prod_{i\in I}[e(R^{\lambda_i},\hat{g}^{(a+y)u})e(h_{f(i)}^{-r_i},\hat{g}^{(a+y)u})e(h_{f(i)}^{(a+y)u},\hat{g}^{r_i})]^{w_i} \\
&= \prod_{i\in I}[e(R^{\lambda_i},\hat{g}^{(a+y)u})]^{w_i} \\
&= e(R,\hat{g}^{(a+y)u})^{\prod_{i\in I}\lambda_i \cdot w_i} \\
&= e(R,\hat{g})^{(a+y)us} \\
V &= e(L_0{}',(C_1)^L \cdot C_2) \\
&= e(\hat{g}^{\alpha/(a+y)t}R^u,(\hat{g}^s)^y\hat{g}^{as}) \\
&= e(\hat{g}^{\alpha/(a+y)t}R^u,\hat{g}^{(a+y)s}) \\
&= e(\hat{g}^{\alpha/(a+y)t},\hat{g}^{(a+y)s})e(R^u,\hat{g}^{(a+y)s}) \\
&= e(\hat{g},\hat{g})^{\alpha s/t}e(R,\hat{g})^{(a+y)us} \\
D &= \frac{V}{Q} = \frac{e(\hat{g},\hat{g})^{\alpha s/t}e(R,\hat{g})^{(a+y)us}}{e(R,\hat{g})^{(a+y)us}} = e(\hat{g},\hat{g})^{\alpha s/t}
\end{aligned}$$

and $C_0/(D)^t = \widetilde{SK} \cdot e(\hat{g},\hat{g})^{\alpha s}/(e(\hat{g},\hat{g})^{\alpha s/t})^t = \widetilde{SK}$ hold. Thus, the correctness of the Decrypt step holds.

5.2 Security Analysis

Theorem 1: Provided that the decisional q-parallel BDHE assumption holds, our EAL-CP-ABE scheme is IND-CPA secure under selective access policy ($q \geq l^*, n^*$ where l^*, n^* are the row and column of the challenge matrix).

Proof: Provided that there is a PPT adversary Adv_A with advantage ε who can break the EAL-CP-ABE scheme, we can build a simulator Sim_B with advantage $\varepsilon / 2$ to figure out the decisional q-parallel BDHE assumption.

Firstly, the challenger Cha_C randomly chooses $s, m, b_1, b_2, ..., b_q \in Z_p$ and gives the vector

$$\vec{y} = \{\hat{g}, \hat{g}^s, \hat{g}^m, \hat{g}^{m^2}, ..., \hat{g}^{m^q}, , \hat{g}^{m^{q+2}}, \hat{g}^{m^{2q}}$$

$$\forall_{1 \leq j \leq q} \ \hat{g}^{s \cdot b_j}, \hat{g}^{m/b_j}, ..., \hat{g}^{m^q/b_j}, , \hat{g}^{m^{q+2}/b_j}, ..., \hat{g}^{m^{2q}/b_j}$$

$$\forall_{1 \leq j \leq q, k \neq j} \ \hat{g}^{m \cdot s \cdot b_k / b_j}, ..., \hat{g}^{m^q \cdot s \cdot b_k / b_j}\}.$$ Then, Cha_C selects $\varpi \in \{0,1\}$ at random. If $\varpi = 0$, let $T = e(\hat{g}, \hat{g})^{m^{q+1}s}$. If $\varpi = 1$, Cha_C randomly selects $V \in \hat{G}_T$ and sets $T = V$. Furthermore, Cha_C sends $(\vec{y}, T)$ to Sim_B.

Init. Firstly, Adv_A selects (T^*, f^*) as challenge access policy and transmits it to Sim_B.

Setup. Sim_B sets $e(\hat{g}, \hat{g})^\alpha = e(\hat{g}, \hat{g})^{\alpha'} e(\hat{g}^m, \hat{g}^{m^q})$ so that $\alpha = \alpha' + m^{q+1}$ where $\alpha' \in Z_p$. As for every $x \in [1, u]$, let X be the collection of indices i in $f^*(i) = x$ and Sim_B chooses a random $z_x \in Z_p$. Then Sim_B computes $h_x = \hat{g}^{z_x} \prod_{i \in X} \hat{g}^{m\mathrm{T}^*_{i,1}/b_i} \hat{g}^{m^2\mathrm{T}^*_{i,2}/b_i} \cdots \hat{g}^{m^n\mathrm{T}^*_{i,n}/b_i}$. In particular, if $X \in \varnothing$, let $h_x = \hat{g}^{z_x}$. In addition, Sim_B randomly chooses $a \in Z_p$, a hash function H and computes $\hat{g}^a$. Let $R = \hat{g}^m$. Sim_B publishes $Params = \{\hat{g}, p, e(\hat{g}, \hat{g})^\alpha, \hat{g}^a, h_1, h_2, ..., h_u, R, H\}$.

Phase I. Adv_A performs key queries about the set S of the sequence attributes where $S \notin \mathrm{T}^*$. Sim_B finds the vector $\vec{w}' = (w_1, w_2, ..., w_n) \in Z_p^n$ so that $\vec{w}'\mathrm{T}^*_i = 0$ where $w_1 = -1$ for all i satisfying $f^*(i) \in S$. Then Sim_B randomly chooses $t, y, r \in Z_p$ and computes $L_1 = \hat{g}^{\frac{r}{a+y}} \prod_{i=1}^{n} (\hat{g}^{w_i m^{q+1-i}})^{\frac{1}{a+y}} = \hat{g}^u$ which implies $u = \frac{1}{a+y}(r + w_1 m^q + w_2 m^{q-1} + ... + w_n m^{q-n+1})$. And Sim_B computes the following parameters:

$$L_2 = \hat{g}^{\frac{ar}{a+y}} \prod_{i=1}^{n} (\hat{g}^{w_i m^{q+1-i}})^{\frac{a}{a+y}} = \hat{g}^{au},$$

$$L_0 = \hat{g}^{\frac{\alpha' + mr}{a+y}} \prod_{i=2}^{n} (\hat{g}^{w_i m^{q+2-i}})^{\frac{1}{a+y}} = \hat{g}^{\alpha/(a+y)} \hat{g}^{mu} = \hat{g}^{\alpha/(a+y)} R^u.$$

In addition, Sim_B calculates the value of K_x in the following two cases: 1) $x \in S, f^*(i) \neq x$, $K_x = \hat{g}^{z_x}$; 2) $x \in S, f^*(i) = x$, $K_x = \{L_1^{z_x} \prod_{i \in X} \prod_{j=1}^{n} (\hat{g}^{(m^j/b_i)r} \prod_{k=1, k \neq j}^{n} (\hat{g}^{m^{q+1+j-k}/b_i})^{w_k})^{\mathrm{T}^*_{i,j}}\}^{(a+y)}$. And Sim_B sends $SK = \{Z_{id} = H(t), L = y, L_0, L_1, L_2, \{K_x\}_{x \in S}\}$ to Adv_A

Challenge. When **Phase I** is finished, Adv_A outputs two plaintexts m_0, m_1 with equal length to Sim_B. Sim_B randomly chooses $\beta \in \{0,1\}$ to compute $C_0 = m_\beta \cdot T \cdot e(\hat{g}^{\alpha'}, \hat{g}^s), C_1 = \hat{g}^s, C_2 = (\hat{g}^s)^a$. Then Sim_B selects some random numbers $y_2', y_3', ..., y_n' \in Z_p^n$. Let $\vec{\lambda} = (s, sm + y_2', sm^2 + y_3', ..., sm^{n-1} + y_n')$ to share s. Define E_i as the set that all k satisfying $k \neq i$ and $f^*(k) = f^*(i)$. Then, Sim_B can compute $B_{2,i} = \hat{g}^{r_i} \hat{g}^{-sm^i}$ and

$$B_{1,i} = h_{f^*(i)}^{-am^i} (\prod_{j=2}^{n} (\hat{g}^m)^{\mathrm{T}^*_{i,j} y_j'})(\hat{g}^{s \cdot b_i})^{-z_{f^*(i)}} \cdot (\prod_{K \in E_i} \prod_{j=1}^{n} (\hat{g}^{m^j \cdot s \cdot (b_i / b_j)^{\mathrm{T}^*_{k,j}}}))$$

where $r_1, r_2, ..., r_l \in Z_p$. Then, Sim_B computes $\bar{C} = H(m_\beta)$ and transmits the challenge ciphertext $(CT, \bar{C}) = \{(\mathrm{T}^*, f^*), C_0, C_1, B_{1,j}, B_{2,j}, \bar{C}\}$ to Adv_A.

Phase II. Adv_A performs key queries as in **Phase I**. Sim_B similarly gives the results of the queries.

Guess. Adv_A puts out the guess β'. In case that $\beta' = \beta$, Sim_B puts out $\varpi' = 0$. Otherwise, Sim_B puts out $\varpi' = 1$. When $\varpi = 1$, any information about β cannot be derived by Adv_A. And we can get $Pr[\beta' \neq \beta \mid \varpi = 1] = \frac{1}{2} = Pr[\varpi' = \varpi \mid \varpi = 1]$. When $\varpi = 0$, Adv_A can obtain information about the ciphertext. So $Pr[\beta' = \beta \mid \varpi = 0] = \varepsilon + \frac{1}{2} = Pr[\varpi' = \varpi \mid \varpi = 0]$. Finally, Sim_B has the following advantage in breaking the decisional q-parallel BDHE assumption as

$$Pr[\varpi' = \varpi] = Pr[\varpi' = \varpi \mid \varpi = 0] \cdot Pr[\varpi = 0]$$

$$+ Pr[\varpi' = \varpi \mid \varpi = 1] \cdot Pr[\varpi = 1] - \frac{1}{2}$$

$$= (\varepsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{\varepsilon}{2}.$$

Theorem 2: In case that the l-SDH assumption holds, our EAL-CP-ABE scheme is traceable over $q \leq l$, where q is the number of key queries by the adversary.

Proof: In case that there is an adversary Adv_A with a non-negligible advantage that can win the traceability game after making q key queries, we can build a simulator Sim_B with a non-negligible advantage to solve the l-SDH problem.

Sim_B instantiates the l-SDH problem $INS_{l-SDH} = (p, \hat{G}, \hat{G}_T, e, \tilde{g}, \tilde{g}^a, \tilde{g}^{a^2}, ..., \tilde{g}^{a^l})$ where $a \in Z_p$, $\tilde{g} \in \hat{G}$. The goal of Sim_B is to compute the pair $(c, \tilde{g}^{1/(a+c)})$ so as to address the l-SDH problem. Adv_A plays the tracing game with Sim_B as follows.

Setup. Sim_B randomly chooses q different values $c_1, c_2, ..., c_q \in Z_p$. Let the polynomial function $f(y)=\prod_{i=1}^{q}(y+c_i)=\sum_{i=0}^{q}\alpha_i y^i$ where $\alpha_i \in Z_p, i=0,1,2,...,q$. Then Sim_B computes

$$\hat{g}=\prod_{i=0}^{q}(\tilde{g}^{a^i})^{\alpha_i}=\tilde{g}^{f(a)}, \hat{g}^a=\tilde{g}^{f(a)a}.$$

Sim_B randomly selects $h_1, h_2, ..., h_u, R \in \hat{G}, \alpha, \delta \in Z_p$, hash function: $H: Z_p \to \hat{G}_T$ and transmits $Params=\{\hat{g}, e(\hat{g},\hat{g})^\alpha, \hat{g}^a, h_1, h_2, ..., h_u, R=\hat{g}^\delta, H\}$ to Adv_A.

Key Query. Adv_A hands (id_i, S_i) over to Sim_B for querying private keys related to (id_i, S_i) where $i \le q$. Let the polynomial function

$$f_i(y)=f(y)/(y+c_i)=\prod_{j=1, j\ne i}^{q}(y+c_j)=\sum_{j=0}^{q-1}(\beta_j y^j).$$

Sim_B calculates $\sigma_i=\prod_{j=0}^{q-1}(\tilde{g}^{a^j})^{\beta_j}=\tilde{g}^{f_i(a)}=\tilde{g}^{f(a)/(a+c_i)}=\hat{g}^{1/(a+c_i)}$. And Sim_B randomly selects $u, t \in Z_p$ and generates $SK_{id}=\{t, Z_{id}=H(id\,||\,t), L=c_i, L_0=(\sigma_i)^\alpha R^u=\hat{g}^{\alpha/(a+c_i)}R^u, L_1=\hat{g}^u, L_2=\hat{g}^{au}, K_x=h_x^{(a+c_i)u}\}$ which is sent to Adv_A.

Key Forgery. Adv_A transmits a private key SK_A that is forged to Sim_B. Define ε_A as the event in which Adv_A earns the game, i.e., SK_A meets the equations $(1)-(4)$ and $L \notin (c_1, c_2, ..., c_q)$. In case that ε_A does not happen, Sim_B selects a random pair (c, w) as the l-SDH problem where $c \in Z_p$ and $w \in \hat{G}$. If ε_A happens, then we set the polynomial function $f(y)=\xi(y)(y+L)+\xi_{-1}$ where $\xi(y)=\sum_{i=0}^{q-1}(\xi_i y^i)$ and $\xi_{-1} \in Z_p$. It should be noted that $\xi_{-1} \ne 0$ and $(y+L)$ cannot divide $f(y)$ due to $L=c_i \notin (c_1, c_2, ..., c_q)$. Sim_B can compute $1/\xi_{-1} (\text{mod } p)$ due to $gcd(\xi_{-1}, p)=1$. Suppose that $L_1=\hat{g}^u$ where $u \in Z_p$ is unknown, we can get $L_2=\hat{g}^{au}$ and $L_0=\hat{g}^{\alpha/(a+L)}R^u$ according to the equations (2) (3). B computes (c, w) as follows,

$$\sigma=(L_0/L_1^\delta)^{\alpha^{-1}}=\hat{g}^{\frac{1}{a+L}}=\tilde{g}^{\frac{f(a)}{a+L}}=\tilde{g}^{\xi(a)}\tilde{g}^{\frac{\xi_{-1}}{a+L}},$$
$$w=(\sigma \cdot \prod_{i=0}^{q-1}(\tilde{g}^{a^i})^{-\xi_i})^{\frac{1}{\xi_{-1}}}=\tilde{g}^{\frac{1}{a+L}},$$
$$c=L \; mod \; p \in Z_p.$$

We can calculate $e(\tilde{g}^a \cdot \tilde{g}^c, w)=e(\tilde{g}^{a+L}, \tilde{g}^{\frac{1}{a+L}})=e(\tilde{g},\tilde{g})$, the pair (c, w) is the way to address the l-SDH problem. Let $\varepsilon_{SDH}(c, w)$ be the event that the pair (c, w) is the way to address the l-SDH problem. Then B has the probability in solving the l-SDH problem as

$$\begin{aligned}
Pr[\varepsilon_{SDH}] &= Pr[\varepsilon_{SDH} \mid \overline{Awin}] \cdot Pr[\overline{Awin}] \\
&+ Pr[\varepsilon_{SDH} \mid Awin \wedge gcd(\xi_{-1}, p) \ne 1] \\
&\cdot Pr[Awin \wedge gcd(\xi_{-1}, p) \ne 1] \\
&+ Pr[\varepsilon_{SDH} \mid Awin \wedge gcd(\xi_{-1}, p) = 1] \\
&\cdot Pr[Awin \wedge gcd(\xi_{-1}, p) = 1] \\
&= 0 + 0 + 1 \cdot Pr[Awin \wedge gcd(\xi_{-1}, p) = 1] \\
&= \varepsilon_A.
\end{aligned}$$

This contradicts the l-SDH hardness assumption.

6 Performance Evaluation

The theoretical analysis and experimental analysis of our EAL-CP-ABE scheme are shown in this segment.

6.1 Theoretical Analysis

In Table 2, we contrast the functionality of our EAL-CP-ABE scheme with other related schemes, including access structure, active protection, traceability, outsourced decryption, user revocation and integrity. We can see that the access structures of the mentioned schemes are designed on the basis of the linear secret sharing scheme. Furthermore, only our EAL-CP-ABE scheme supports active protection, while other solutions merely enable passive tracing of malicious users or even no traceability. Han et al.'s [16] scheme supports the traceability of malicious users and utilizes the revocation list to enable the revocation of malicious users. However, the scheme does not support the outsourced decryption of the ciphertext, which has a high local computation overhead on the client-side. And it lacks verification of ciphertext integrity. Ge et al.'s [25] scheme achieves the user revocation by updating the access policy and can ensure the integrity of the updated ciphertext. But their scheme does not have the features of tracing malicious users and ciphertext outsourced decryption. In addition, the CP-ABE schemes [28-30] all support the traceability of the malicious users and cannot ensure the integrity of the ciphertext. And these two schemes [28-29] support outsourced decryption. Among these schemes [28-30], only Wang et al.'s [29] scheme can revoke malicious users through the key updating and the ciphertext updating.

In Table 3, we perform a computational cost comparison of our EAL-CP-ABE scheme with other schemes. Our scheme is superior to other schemes [16, 28-30] in the KenGen algorithm and Encrypt algorithm. Because we reduce the computational cost by decreasing exponential and multiplicative operations. In the User Decrypt algorithm, it is obvious that our scheme and the schemes [28-29] have lower computational cost than these schemes [16, 30]. Because we all take advantage of the cloud or the fog nodes to perform semi-decryption of the ciphertext, which largely reduces the computational cost of the decryption on the client-side. In addition, the Trace algorithm of our scheme is better than these schemes [16, 28-30].

Table 2. Functional comparison

Scheme	Access structure	Active protection	Traceability	Outsourced decryption	User revocation	Integrity
[16]	LSSS	×	✓	×	✓	×
[25]	LSSS	×	×	×	✓	✓
[28]	LSSS	×	✓	✓	×	×
[29]	LSSS	×	✓	✓	✓	×
[30]	LSSS	×	✓	×	×	×
Ours	LSSS	✓	✓	✓	✓	✓

Note. Active protection means actively monitoring for the presence of potentially malicious users leaking keys rather than passively finding the key leakage.

Table 3. Comparison of computational cost

Scheme	KeyGen	Encrypt	User Decrypt	Trace
[16]	(n_s +6)E+(n_s +1)M	(4 n_l +3+r)E+(n_l +1)M	(2 n_r +4)P+(n_r +2)E +(n_r +3)M+D	(n_s +6)P+(n_s +2)E +(n_s +3)M
[28]	(4 n_s +4)E+(2 n_s +1)M	(5 n_l +3)E+(2 n_l +1)M	3E+D	(3 n_s +6)P+(n_s +3)E +(2 n_s +3)M
[29]	(4 n_s +4)E+M	(4 n_l + n_l r+3)E +(n_l +1)M	E+D	(2 n_s +4)P+4E+3M
[30]	(n_s +11)E+(n_s +10)M+H	(3 n_l +5)E+(n_l +1)M	(2 n_r +2)P+(n_r +3)E +(n_r +5)M+2D	(2 n_s +8)P+3E+7M+H
Ours	(n_s +4)E+H	(3 n_l +3)E+(n_l +1)M	E+D	(2 n_s +4)P+2E+4M

Note. H is specified as the operation of the hash function. P is specified as a bilinear pairing operation. E is specified as the exponent operation in the group G, G_T. M is specified as the multiplication operation in the group G, G_T. M is specified as the division operation in the group G, G_T. n_s is specified as the number of the user attributes. n_l is specified as the number of the rows in the access policy. n_r is specified as the number of the attributes satisfying the access policy in the user's private key. r is specified as the length of minimum set of nodes in the user tree.

6.2 Experimental Analysis

We perform experimental simulations to assess the performance of the proposed EAL-CP-ABE scheme. In the experiment, we use the Java Pairing-Based Cryptography Library (JPBC) [31] and the type A elliptic curve on a PC. The hardware and system of the PC we use are Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz with 8.00 GB RAM and Ubuntu 18.04.5 64-bit. We simulate KeyGen, Encrypt, User Decrypt and Trace algorithms where the evaluation time is measured in milliseconds. And we set the range of user attributes and the access policy rows from 10 and 50. we perform each experiment 100 times and take the mean value to obtain a more precise execution time.

In Figure 2 and Figure 3, we display the execution time of our scheme with two other schemes [28, 30] in the KeyGen algorithm and the Encrypt algorithm. From Figure 2, it is obvious that the execution time of the scheme [28] is much higher than our scheme and the scheme [30]. Since the scheme [28] has more parameters associated with attributes, it leads to more execution time of the KeyGen algorithm with the growing number of attributes. In Figure 3, the execution time of our scheme and scheme [30] is lower than that of scheme [28] as both our scheme and scheme [30] have fewer exponent operations. Figure 4 illustrates that our scheme and the scheme [28] are far better than the scheme [30] in the User Decrypt algorithm. Because our scheme and the scheme [28] both take advantage of outsourced decryption which makes the overhead of user-side decryption much lower. And the execution time of our EAL-CP-ABE scheme is slightly lower than the scheme [28] because the user of the scheme [28] needs to compute some parameters before the cloud performs a semi-decryption operation. Furthermore, in Figure 5, we see that the proposed EAL-CP-ABE scheme is superior to the schemes [28, 30] regarding the execution time of the Trace algorithm. In conclusion, our experimental consequences are in accordance with the consequences of the theoretical analysis.

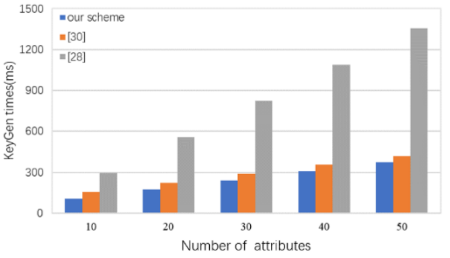

Figure 2. The operating time of KeyGen algorithm

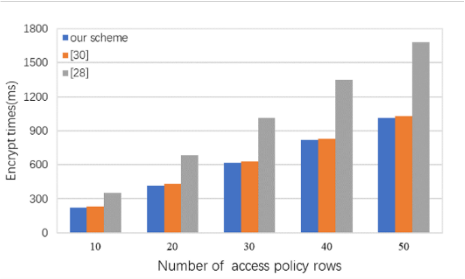

Figure 3. The operating time of Encrypt algorithm

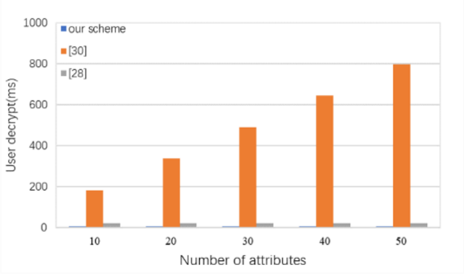

Figure 4. The operating time of User Decrypt algorithm

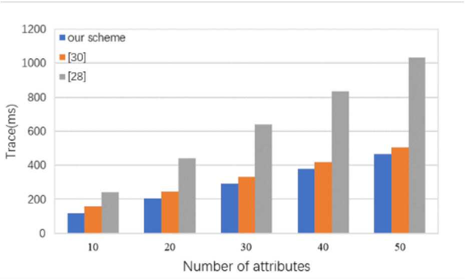

Figure 5. The operating time of Trace algorithm

7 Conclusion

In this paper, an anti-key leakage scheme by means of maximum access and white-box tracing has been presented. And we utilize the fog nodes to implement the outsourced decryption of ciphertext. In addition, we design a reward and penalty mechanism to further reduce the risk of key leakage. Through this mechanism, we reward users with a good reputation and penalize users at risk of leaking their private keys by either not granting rewards or revoking their access permissions. The EAL-CP-ABE scheme is proved IND-CPA secure and traceable on the basis of the hardness assumptions under the selected models. In addition, the theoretical and experimental analysis illustrates that our EAL-CP-ABE scheme is efficient and practical.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants No. 61922045, No. U21A20465, No. 62172292, No. 61877034.

References

[1] B. D. Deebak, F. H. Memon, S. A. Khowaja, K. Dev, W. Wang, N. M. F. Qureshi, C. Su, Lightweight Blockchain Based Remote Mutual Authentication for AI-Empowered IoT Sustainable Computing Systems, *IEEE Internet of Things Journal*, pp. 1-9, February, 2022.

[2] J. Shen, T. Zhou, X. Chen, J. Li, W. Susilo, Anonymous and Traceable Group Data Sharing in Cloud Computing, *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 4, pp. 912-925, April, 2018.

[3] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, Y. Xiang, Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing, *IEEE Transactions on Dependable and Secure Computing*, Vol. 16, No. 6, pp. 996-1010, November-December, 2019.

[4] D. Liu, Y. Zhang, W. Wang, K. Dev, S. A. Khowaja, Flexible Data Integrity Checking with Original Data Recovery in IOT-Enabled Maritime Transportation Systems, *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-12, November, 2021.

[5] T. Miao, J. Shen, X. Jin, J. F. Lai, Fine-grained and Efficient Access Control in E-health Environment, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2169-2176, December, 2019.

[6] P. Zhang, J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, X. Luo, A Survey on Access Control in Fog Computing, *IEEE Communications Magazine*, Vol. 56, No. 2, pp. 144-149, February, 2018.

[7] J. Shen, H. Yang, P. Vijayakumar, N. Kumar, A Privacy Preserving and Untraceable Group Data Sharing Scheme in Cloud Computing, *IEEE Transactions on Dependable and Secure Computing*, pp. 1-13, January, 2021.

[8] J. Shen, J. Shen, X. Chen, X. Huang, W. Susilo, An Efficient Public Auditing Protocol with Novel Dynamic Structure for Cloud Data, *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 10, pp. 2402-2415, October, 2017.

[9] J. Shen, T. Zhou, Z. Cao, Protection Methods for Cloud Data Security, *Journal of Computer Research and Development*, Vol. 58, No. 10, pp. 2079-2098, October, 2021.

[10] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, C. Su, Blockchain-Based Reliable and Efficient Certificateless Signature for IIoT Devices, *IEEE Transactions on Industrial Informatics*, pp. 1-9, May, 2021.

[11] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, *Proceedings of the 13th ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 2006, pp. 89-98.

[12] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-Policy Attribute-Based Encryption, *2007 IEEE Symposium on Security and Privacy (SP'07)*, Berkeley, CA, USA, 2007, pp. 321-334.

[13] J. Ning, X. Dong, Z. Cao, L. Wei, 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, June, 2015.

[14] X. Li, K. Liang, Z. Liu, D. S. Wong, Attribute Based Encryption: Traitor Tracing, Revocation and Fully Security on Prime Order Groups, *CLOSER 2017: Proceedings of the 7th International Conference on Cloud Computing and Services Science*, Porto, Portugal, 2017, pp. 281-292.

[15] P. K. Premkamal, S. K. Pasupuleti, P. Alphonse, Traceable CP-ABE for Outsourced Big Data in Cloud Storage, *International Conference on Computing and Information Technology*, Bangkok, Thailand, 2019, pp. 213-226.

[16] D. Han, N. Pan, K. C. Li, A Traceable and Revocable Ciphertext-Policy Attribute-Based Encryption Scheme Based on Privacy Protection, *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, No. 1, pp. 316-327, January-February, 2022.

[17] T. Khalid, M. A. K. Abbasi, M. Zuraiz, A. N. Khan, M. Ali, R. W. Ahmad, J. J. Rodrigues, M. Aslam, A Survey on Privacy and Access Control Schemes in Fog Computing, *International Journal of Communication Systems*, Vol. 34, No. 2, pp. e4181, January, 2021.

[18] A. Sahai, B. Waters, Fuzzy Identity-Based Encryption, *Annual International Conference on The Theory and Applications of Cryptographic Techniques*, Aarhus, Denmark, 2005, pp. 457-473.

[19] X. Yan, X. He, T. Liu, Q. Ye, J. Yu, Y. Tang, Traceable Attribute-Based Encryption Scheme with Key-Delegation Abuse Resistance, *Journal on Communications*, Vol. 41, No. 4, pp. 150-161, April, 2020.

[20] P. K. Premkamal, S. K. Pasupuleti, P. Alphonse, A New Verifiable Outsourced Ciphertext-Policy Attribute-Based Encryption for Big Data Privacy and Access Control in Cloud, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 7, pp. 2693-2707, July, 2019.

[21] H. Zheng, J. Shao, G. Wei, Attribute-Based Encryption with Outsourced Decryption in Blockchain, *Peer-to-Peer Networking and Applications*, Vol. 13, No. 5, pp. 1643-1655, September, 2020.

[22] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, S. Mumtaz, Attribute-Based Encryption with Parallel Outsourced Decryption for Edge Intelligent Iov, *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 11, pp. 13784-13795, November, 2020.

[23] G. Ramu, B. E. Reddy, A. Jayanthi, L. N. Prasad, Fine-Grained Access Control of EHRs in Cloud Using CP-ABE with User Revocation, *Health and Technology*, Vol. 9, No. 4, pp. 487-496, August, 2019.

[24] D. Sethia, A. Shakya, R. Aggarwal, S. Bhayana, Constant Size CP-ABE with Scalable Revocation for Resource-Constrained IOT Devices, *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2019, pp. 0951-0957.

[25] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, L. Fang, Revocable Attribute-Based Encryption with Data Integrity in Clouds, *IEEE Transactions on Dependable and Secure Computing*, pp. 1-10, March, 2021.

[26] A. Beimel, Secure Schemes for Secret Sharing and Key Distribution, Ph. D. Thesis, *Technion-Israel Institute of technology*, Haifa, Israel, 1996.

[27] B. Waters, Ciphertext-policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, *International Workshop on Public Key Cryptography*, Taormina, Italy, 2011, pp. 53-70.

[28] Q. Li, H. Zhu, Z. Ying, T. Zhang, Traceable Ciphertext-Policy Attribute-Based Encryption with Verifiable Outsourced Decryption in eHealth Cloud, *Wireless Communications and Mobile Computing*, Vol. 2018, Article No. 1701675, June, 2018.

[29] S. Wang, K. Guo, Y. Zhang, Traceable Ciphertext-Policy Attribute-Based Encryption Scheme with Attribute Level User Revocation for Cloud Storage, *Plos One*, Vol. 13, No. 10, Article No. e0206952, September, 2018.

[30] J. Ning, Z. Cao, X. Dong, L. Wei, White-Box Traceable CP-ABE for Cloud Storage Service: How to Catch People Leaking Their Access Credentials Effectively, *IEEE Transactions on Dependable and Secure Computing*, Vol. 15, No. 5, pp. 883-897, September-October, 2018.

[31] A. De Caro, V. Iovino, Jpbc: Java Pairing Based Cryptography, *2011 IEEE Symposium on Computers and Communications (ISCC)*, Kerkyra, Greece, 2011, pp. 850-855.

Biographies

Fengwei Cheng received the B.E. degree in 2020. He is currently working toward the M.E. degree at the Nanjing University of Information Science and Technology, Nanjing, China. His current research interests include information security, access control, and cryptography.

Sai Ji received his M.S. degree from the Nanjing Aeronautics and Astronautics University (NUAA), Nanjing, China, in 2006. He works as an Associate Professor at the NUIST. His research interests are in the areas of structural health monitoring, and WSNs. Ji has published more than 20 journal/conference papers.

Chin-Feng Lai received the Ph.D. degree in engineering science from National Cheng Kung University, Tainan,Taiwan, in 2008. Since 2016, he has been an Associate Professor of Engineering Science, National Cheng Kung University, Tainan. His research focuses on Internet of Things, body sensor networks, e-healthcare, mobile cloud computing, etc.