Improving security and efficiency for multi-authority access control system in cloud storage

Qi Li Jianfeng Ma Rui Li Ximeng Liu Jinbo Xiong

November 15, 2013

Abstract

Multi-Authority Attribute-Based Encryption (MA-ABE) is an emerging cryptographic primitive for enforcing fine-grained attribute-based access control on the outsourced data in cloud storage. However, most of the previous multi-authority attribute-based systems are either proven security in a weak model or lack of efficiency in user revocation. In this paper, we propose a novel multi-authority attribute-based data access control system for cloud storage. We construct a new multi-authority CP-ABE scheme with decryption outsourcing. We largely eliminate the decryption overhead for users by outsourcing the undesirable bilinear pairing operations to the cloud servers. The proposed scheme is proven adaptively secure in the standard model and supports any monotone access policy. We also design an efficient attribute-level user revocation approach with less computation cost. The security analysis, numeral comparisons indicate that the proposed system is secure, efficient and scalable.

Keywords: Cloud storage; Multi-authority; CP-ABE; Decryption outsourcing; Attribute-level revocation.

1 Introduction

Cloud storage is an promising application paradigm of cloud computing [25], which enables data owners to conveniently share their data files via the cloud. Since a large amount of individual data are hosted to the cloud servers, the concern about data confidentiality arises. To alleviate this problem, one common method is to encrypt the data before uploading it to the servers. Such approach also introduces an challenge to the access control over the encrypted data, since the cloud servers cannot be fully trusted and may attempted to access and analyze the personal data for illegal or financial purposes.

Attribute-based encryption [28] is an applicable cryptographic technique for clouding storage, which simultaneously attains data confidentiality and fine-grained access control. In an ABE scheme, the access policy is defined over various attributes. More precisely, ABE schemes can be divided into two types: Key-policy ABE (KP-ABE) and Ciphertext-policy ABE (CP-ABE). In KP-ABE, an access policy is associated with user's private keys, while a set of attributes is associated with the ciphertext. In CP-ABE, the circumstance is conversed, the ciphertext is labeled with an access policy, while user's private keys are labeled with a set of attributes. Especially, CP-ABE is more suitable for the data provider to define the access policy.

Recently, there are several attribute-based access control schemes in the clouds. Basing on KP-ABE [12] and proxy re-encryption [4], Yu *et. al* proposed a fine-grained data access control system in cloud computing [34]. Applying CP-ABE [3], Hur [14] proposed an attribute-based data sharing scheme. However, in [12, 34, 3, 14], the attribute universe is assumed to be managed by a single authority. This premise may not capture the practical requirement in clouds, where user's attributes may issued by different authorities. For instance, Alice wants to encrypt a message under access policy (" UNIVERSITY. MIT. GRADUATE" and "IBM. ENGINEER "). In this way, only the recipient who is the graduate of university MIT and now employed as an engineer by IBM can recover the message. University MIT is responsible to issue attributes to students, while IBM is responsible to distribute attributes to its employees.

Towards addressing this problem, several multi-authority attribute-based access control schemes [7, 27, 31, 8, 18, 32, 22, 24] have been proposed. Nevertheless, these schemes are either proven to be secure in the selective model [6] or lack of efficient revocation approach. For example, in [32], Yang *et. al* proposed an multi-authority data access control system for cloud storage. Once an attribute is revoked from some users, the AA has to compute a new update key for each unrevoked user. In the worst case, when an attribute is revoked from only one user, the AA still has to calculate $n-1$ update keys, where n is the number of users who possess the attribute. Such revocation approach makes their scheme less practical. Thus, how to construct an efficient and secure multi-authority access control system for cloud storage remains an challenge problem.

In this paper, we present a new fine-grained attribute-based access control scheme for multi-authority cloud storage applications. Moreover, we alleviate the decryption overhead for users by outsourcing the complicated bilinear pairing computation to the clouds. In the proposed scheme, a data provider can define flexible access policies over descriptive attributes and encrypt the sensitive data before uploading it to the cloud severs. A user is authorized only if he possesses proper attributes which satisfy the access policy deployed in the data. To resisting collusion attacks from unauthorized users, a unique global identifier (gid) is issued to each user in the system. We also provide an efficient attribute-level revocation method for our scheme. That is, when an attribute is revoked from some users, they will not lose all the access privileges. Our revocation approach also achieves two security requirements. On one hand, after some attributes have been revoked from a user, he cannot decrypt the new encrypted data if his remaining attributes don't satisfy the access policy (Forward Secrecy). On the other hand, when a new user join in the system, he is not able to decrypted the prior encrypted data even if he has the corresponding attributes (Back Secrecy). In summary, this work makes the following contributions:

1. We present a novel fully secure multi-authority CP-ABE scheme. Our scheme is constructed on composite order groups, and supports any monotone access policy which can be expressed by a linear secret sharing scheme (LSSS). We prove the adaptive security of our scheme in the standard model. Compared with the fully secure multi-authority scheme [24], the single CA in our scheme can not decrypt any ciphertext. Thus, our scheme does not require multiple CAs and is more efficient and acceptable for real applications.

2. We design an outsourcing paradigm to alleviate the undesirable computation cost for users. By extending the decryption outsourcing approach [13] to the multi-authority settings, we delegate the bilinear pairing operations to the cloud servers without leaking data contents. As a result, each user only needs to calculate one exponent operation. Thus, decryption overhead for the users are saved significantly.

3. We propose an efficient revocation approach for the proposed multi-authority CP-ABE scheme. Basing on the revocation method [16], we realize efficiently immediate attribute-level revocation while achieving both the backward and forward secrecy. Different from [16], we let the AAs be in charge of executing key updating for users, and the cloud server is responsible to

re-encrypt the ciphertext. Thus, our revocation paradigm is more appropriate and acceptable in practice.

The remaining of this paper is organized as follows. Section 2 introduces the related works on attribute-based access control systems. In Section 3, we give the overview of the system model, the threat model and the security requirements. We discuss the definitions of access structure, LSSS scheme and our system model in Section 4. In Section 5, we propose the detailed construction of our scheme. In Section 6 and 7, we analyze our scheme in terms of the security and performance, respectively. We conclude in Section 8. The Appendix describes the assumptions, the security game and proof.

2 Related works

Attribute-based Encryption is regarded as a useful and promising cryptographical technique which realizes secure and flexible fine-grained access control in cloud scenarios. Since Sahai and Waters [28] first presented the notion of Attribute-based Encryption (ABE), various KP-ABE schemes [12, 26, 7, 8] and CP-ABE schemes [3, 11, 9, 30, 21, 22, 24, 20, 29]have been proposed. However, due to lack of efficient revocation mechanisms, these ABE schemes can not be directly employed in cloud storage systems.

Several attribute-based access control system dealing with revocation problems have been proposed. Basing on the KP-ABE scheme [12], Yu *et al.* [34] introduced a fine-grained data access control system in cloud computing. In this scheme, the data provider first encrypts the sensitive data with a data encryption key (DEK). He then encrypts the DEK by applying a KP-ABE technique. Nevertheless, the data provider is required to be online all the time, which is not suitable in real applications. Their scheme also achieves the attribute-level revocation. That is, when a user loses one or more attributes, he can still access the data as long as his remain attributes can satisfy the access policy. In contrast, user-level revocation [26, 1] is a Coarse-grained approach. Once the user is revoked, it loses all the access privileges. In [15] and [16], the authors presented two attribute-based access control scheme in data outsourcing systems by employing the CP-ABE technique [3]. These schemes put the operations of key updating and ciphertext updating on the cloud server simultaneously, which may not be appropriate for protecting the users' privacy. In addition, these schemes [34, 15, 16] only work in such systems where the attributes are governed by only one authority and can not serve for multi-authority systems.

In [27], Ruj *et al.* proposed a distributed access control scheme is the clouds by employing the multi-authority CP-ABE technique [22]. When an attribute is revoke from some users, the data provider updates the ciphertext and transmits them to the unrevoked users. The communication overhead will be a performance bottleneck in large scale systems. In [31], Yang and Jia presented a multi-authority attribute-based access control system in cloud storage. For each data provider, a user has to obtain a set of private keys from the AAs. It is inflexible and inefficient in practice applications. Furthermore, in the user revocation phase, this scheme also requires the data provider to participate in the ciphertext updating. It implicitly means that the data provider has to always be online. Basing on [8] and [34], Li *et al.* [23] proposed a scalable and secure sharing system for personal health records. They also presented an efficient attribute-level user revocation mechanism. However, this scheme works in the key-policy settings. Jung *et al.* [18] proposed a privacy preserving CP-ABE schemes for multi-authority clouds system. In the revocation phase, the user with re-encryption privilege will re-define the access policy and re-encrypt the data. Such user-level revocation approach is not suitable and efficient for clouds applications. In [32], Yang *et*

Table 1: A Comparison between current attribute-based access control schemes and ours

Schemes	CP/KP	Multi -authority	Security	Standard Model	Decryption Outsourcing	Key update by	ciphertext update by
[22]	CP	YES	Adaptive	NO	NO	\	\
[24]	CP	YES	Adaptive	Yes	NO	\	\
[34]	KP	NO	Selective	YES	NO	Provider	Server
[35]	CP	NO	Selective	YES	NO	AA	Server
[15][16]	CP	NO	Selective	NO	NO	Server	Server
[23]	KP	YES	Selective	YES	NO	AA	Server
[27]	CP	YES	Selective	NO	NO	Provider	\
[31]	CP	YES	Selective	NO	NO	AA	Provider and Server
[18]	CP	YES	Selective	NO	NO	Users with Privilege	\
[32]	CP	YES	Selective	NO	YES	AA	Server
Ours	CP	YES	Adaptive	YES	YES	AA	Server

al. proposed a DAC-MACS system by employing the decryption outsourcing technique [13]. The heavy bilinear pairing operations are outsourced to the clouds. However, their scheme caused heavy computation in revocation and was proved selectively secure in the random oracle model. Table 1 describes some characteristics of current attribute-based access control schemes in the clouds and ours.

3 System model and Threat model

In this section, we introduce the system model, threats model and security requirements of our multi-authority access control scheme for cloud storage.

3.1 System model

Fig 1 describes the architecture of the proposed access control system which is composed of the following 5 entities:

1. Central Authority (CA): The CA sets up its public parameters. It is in charge of issuing an *gid*-related key to the user. It will not participate in any attribute-related operations.

2. Attribute Authorities (AAs): Each AA is responsible to administer a distinct attribute domain, which is a subset of the system attribute universe. In our scheme, every attribute is managed by a single AA, but each AA can govern an arbitrary scale of attribute domain. While receiving the private key request from a user, it responds the attribute-related keys. Additionally, once one or more attributes are revoked from one or more users, it also executes the key updating process for unrevoked users.

3. Cloud Server: It is an entity which provides data storage service and decryption outsourcing service. Moreover, it also gives service to ciphertext re-encryption.

4. Data Providers: Before transmitting it to the cloud server, a data provider has to encrypt the data file under a DEk and then defines an access policy and enforces the policy on the DEK.

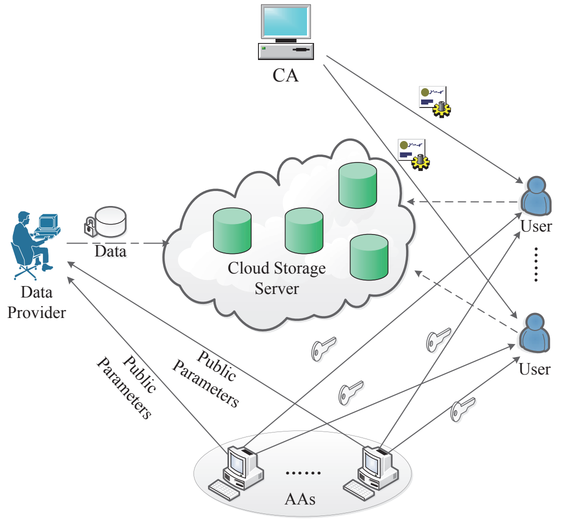

Figure 1: Architecture of Multi-authority Access Control System

It can also call for the cloud server to delete the data file.

5. Users: Each user with a *gid* is labeled by a set of attributes. He has to request the attribute-related keys from corresponding AAs. A user can download the encrypted data and call for decryption outsourcing service from the cloud server. But only the user who owns proper attributes can successfully decrypt the encrypted data.

3.2 Threats model

In this work, the CA is the only one which can be fully trusted. The AAs honestly distribute the keys and send the key updating message, but some of them may be corrupted by the adversary with attempts to find out information of the data file as much as possible. We assume that the AAs will never collude with any user.

As similar with the assumption in [10, 35, 23], the cloud server is assumed to be honest but curious. That is, the cloud server will follow the presented protocol in general, but may collude with malicious users or data providers to get illegal access privileges. However, it will not collude with the revoked users. We assume that the cloud server mostly focuses on information of data contents.

The users are assumed to be malicious all the time. They may collude with others and even the cloud server, and try to access the data that they are not authorized.

3.3 Security Requirement

3.3.1 Fine-Grained Data Access Control

In practical application scenarios, an increasing number of people host their sensitive data to the cloud and different users may own different access privileges over various types of data. In this case, it is acceptable for the data provider himself to define a flexible access policy on his data. Such access policy must identify that who has the access privilege. It must also be ensured that the unauthorized user can not access the data. Moreover, the unauthorized access from the cloud server must also be prevented.

3.3.2 Collusion Resistance

Multiple unauthorized users may cooperate to decrypt a ciphertext that none of them can decrypt alone. This requires our access system to secure against such collusion attacks. We do not consider such attack that the cloud server colludes with the revoked users.

3.3.3 Back Secrecy and Forward Secrecy

Similar as defined in [19], back secrecy means that a new joint user can not decrypt the ciphertext which is created before he joins in the system. Forward secrecy means that the revoked user can not decrypt the ciphertext which is created after he is revoked.

4 Preliminaries and Definition

4.1 Access Structure

Definition 1. *Access Structure [2]: Let $\mathbb{P} = \{P_1, P_2, \ldots, P_T\}$ denote a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_T\}}$ is monotonic if $\forall A_1, A_2$: if $A_1 \in \mathbb{A}$ and $A_1 \subseteq A_2$ then we have $A_2 \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\mathbb{P}$. That is, $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_T\}} \backslash \{\emptyset\}$. We say that the sets in $\mathbb{A}$ are the authorized sets, and the sets outside $\mathbb{A}$ are the unauthorized sets.*

Among ABE systems, the role of the parties is replaced by the descriptive attributes. In this way, the authorized set of attributes will be contained in the access structure $\mathbb{A}$. We focus on the monotonic access structure in this paper. To realize common access structures, one can simply consider the negation of an attribute as a separate attribute.

4.2 Linear Secret Sharing Schemes

Here we adopt the definition of linear secret sharing schemes (LSSS) from [2, 30]:

Definition 2. *Linear Secret Sharing Schemes: Let $\mathbb{P}$ be a set of parties, p be a prime. A secret sharing scheme Π over $\mathbb{P}$ is linear (over $\mathbb{Z}_p$) if it has the following properties*

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

2. There is a matrix $A \in \mathbb{Z}_p^{\ell \times n}$ which is called the share-generating matrix for Π. For all $i = 1, \ldots, \ell$, there exists a function ρ that labels the i-th row of A with a party. (i.e. $\rho \in \mathcal{F}([\ell] \to \mathbb{P})$). During generating the shares, we consider the column vector $\overrightarrow{v} = (s, r_2, \ldots, r_n)^\top$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n$ are randomly picked from $\mathbb{Z}_p$, then $A\overrightarrow{v}$ is the vector of ℓ shares of s according to Π. The shares $(A\overrightarrow{v})_i$ belongs to the party $\rho(i)$.

As shown in [2], each linear secret sharing scheme mentioned before must satisfy the linear reconstruction requirement, defined as follows: Assume that an access structure $\mathbb{A}$ is denoted by (A, ρ). Π is an LSSS for $\mathbb{A}$. Let S denote an authorized set. Then let $I = \{i : \rho(i) \in S\}$ be the index set of rows whose labels are in S. There exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that: if $\{\lambda_i = (A\overrightarrow{v})_i\}$ are valid shares of a secret s according to Π, then we have $\sum_{i \in I} \omega_i \lambda_i = s$. Moreover, such constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ can be found in time polynomial in the size of the share-generating matrix A. Nevertheless, if the set S is unauthorized, no such constants exist.

4.3 Revocable and Outsourced Multi-authority CP-ABE

A revocable and outsourced multi-authority access control scheme consists of the following algorithms:

GlobalSetup $(\lambda)\rightarrow(GPK)$: This algorithm takes in the security parameter λ, it then outputs the global parameters GPK for the system.

CASetup $(GPK) \rightarrow (CPK, CMK)$: The CA runs this algorithm with GPK as input to produce its public parameter CPK and the corresponding master secret key CMK. CPK will be used by AAs only.

AASetup $(GPK, f, U_f) \rightarrow (APK_f,\ AMK_f)$: Each AA_f runs this algorithm with GPK and its attribute domain U_f as input to produce the AA's public parameter APK_f and the corresponding master secret key AMK_f, where $U_f \subset U$, and for $i \neq j$, we have $U_i \bigcap U_j = \varnothing$.

Encrypt $(M, \mathbb{A}, GPK, \bigcup APK_f) \rightarrow (CT)$: This algorithm takes in GPK, a message M, an access structure $\mathbb{A}$ and the set of public parameters for relevant AAs. It produces a ciphertext CT. We assume the access structure $\mathbb{A}$ is implicitly included in CT.

CAKeyGen $(GPK) \rightarrow (gid, DSK_{gid},\ CASK_{gid}, CAPK_{gid})$: This algorithm takes in GPK. It then assigns a special gid to a user, and outputs a decryption key DSK_{gid} a gid-related private key $CASK_{gid}$ and a gid-related public key $CAPK_{gid}$, where DSK_{gid} will be used by the user, $CASK_{gid}$ will be used in pre-decrypt the ciphertext and $CAPK_{gid}$ will be used to generate the attribute-related keys by the AAs.

AAKeyGen $(S_{gid,f}, GPK, CPK,\ CAPK_{gid}, AMK_f) \rightarrow (ASK_{S,gid,f})$: When a user submits a set of attributes $S_{gid,f}$ belongs to AA_f to request the attribute-related key $ASK_{gid,f}$, AA_f runs this algorithm with $S_{gid,f}$ GPK, CPK,$CAPK_{gid}$ and AMK_f as input. If $CAPK_{gid}$ is invalid, it outputs $\perp$. Otherwise, it outputs $ASK_{S,gid,f} = \{ASK_{ATT,gid} | ATT \in S_{gid,f}\}$. We let $ASK_{S,gid} = \bigcup UASK_{S,gid,f}$ denotes the attribute-related key of S_{gid}, where $S_{gid} = \bigcup S_{gid,f}$. We assume the set S_{gid} is implicitly included in $ASK_{S,gid}$.

Pre-Decrypt $(CT, GPK, CASK_{gid},\ ASK_{S,gid}) \rightarrow (PDCT)$: This algorithm takes in CT, GPK, $CASK_{gid}$ and $ASK_{S,gid}$. It outputs the pre-decryption key $PDKEY$ of CT.

Decrypt $(CT, PDKEY, DSK_{gid}) \rightarrow (M)$: This algorithm takes inCT, $PDKEY$ and DSK_{gid}. It outputs the plaintext message M.

Re-KeyGen $(gid, RL_{ATT},\ ASK_{ATT,gid}) \rightarrow (ASK'_{ATT,gid})$: This algorithm takes in a gid, a revocation list of an attribute RL_{ATT} and the original key $ASK_{ATT,gid}$. If the $gid \notin RL_{ATT}$, it outputs a new $(ASK'_{ATT,gid})$.

Re-Encrypt $(CT, ATT_{RC}) \rightarrow (ASK'_{ATT,gid})$: If a revocation operation on attribute ATT_{RC} occurs, this algorithm takes in the original ciphertext CT and ATT_{RC}, it outputs a new ciphertext CT' which can only be decrypt by those who have appropriate attributes and are not in RL.

5 Multi-authority Access Control Scheme in Cloud Storage

We now present the detailed construction of our scheme as follows:

5.1 System Initialization

The system parameters are set by the following three algorithms.

GlobalSetup: Let $\mathbb{G}$ and $\mathbb{G}_1$ be two bilinear maps with order $N = p_1p_2p_3$, where p_1, p_2 and p_3 are 3 distinct primes. Let $\mathbb{G}_{p_i}$ be the subgroup of order p_i in $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ denote a bilinear map. g is a random chosen element from $\mathbb{G}_{p_1}$. X_3 is a generator of $\mathbb{G}_{p_3}$. Additionally, choose an UF-CMA (unforgeable under adaptive chosen message attacks) secure signature system $\sum_{sign} = (KeyGen, Sign, Verify)$. The GPK is published as: $GPK = (N, e, g, X_3, \sum_{sign})$.

CASetup: The CA runs the $KenGen$ algorithm and obtains the sign-key (CMK) and verify-key (CPK). The CPK will be used by the AAs only.

AASetup: Each AA_f sets its attribute universe U_f. For each $i \in U_f$, it chooses a random exponent $t_{f,i} \in \mathbb{Z}_N$ and computes $T_{f,i} = g^{t_{f,i}}$. It also chooses two random exponents $\alpha_f, a_f \in \mathbb{Z}_N$. Finally, the public parameter of AA_f is published as: $APK_f = (g^{a_f}, e(g,g)^{\alpha_f}, T_{f,i}\forall i)$. the master secret key of AA_f is $AMK_f = (\alpha_f, a_f, t_{f,i}\forall i)$.

Additionally, each AA_f also founds a binary tree $TREE_f$ as in Fig 2. In $TREE_f$, each node j is associated with a different key encryption key $KEK_{f,j}$. In addition, each leaf node is labeled by a user (gid). Moreover, there is a path P_j from j to the root node. That is, such a tree with hight h can accommodate at most 2^h users. When a new user comes to AA_f for requesting the attribute-related keys, besides generating the requested keys, AA_f adds the user to the leftmost leaf node and gives the user the path keys in the path from its leaf node to the root node. That is, for each attribute $i \in U_f$, AA_f establishes an attribute-user group $G_{f,i}$, which is a set composed of the users who own this attribute. We let $Gn_{f,i}$ denote the minimum set of nodes whose descendant nodes cover all the users in $G_{f,i}$. For example, when the user is added to node 11, he will obtain the path keys $\{KEK_{n_{11}}, KEK_{n_5}, KEK_{n_2}, KEK_{n_1}\}_f$. If $G_{f,i} = \{(n_8, gid_1), (n_9, gid_2), (n_{10}, gid_3), (n_{11}, gid_4), (n_{12}, gid_5), (n_{13}, gid_6)\}_f$, then $Gn_{f,i} = \{n_2, n_6\}_f$.

At the beginning of system initialization, each AA_f shares a unique attribute group key $AK_{f,i} \in \mathbb{Z}_N$ with the cloud server for each $i \in U_f$.

The system public parameters are published as (GPK, CPK, $\bigcup_{f=1}^{F} APK_f$), where F denotes the total number of the AAs in the system.

5.2 Data outsourcing

Before hosting the data file to the cloud server, the data provider has to enforce attribute-based access control on the outsourced data file. It first encrypts the data file by a data encryption key (DEK). e.g., a symmetric key. It then runs the **Encrypt** algorithm to encrypt the DEK under the defined access policy as follows:

Encrypt: The access policy $\mathbb{A}$ is expressed by (A, ρ), where A is a LSSS matrix with ℓ rows and n rows, and ρ associates each row A_x of A to attribute $\rho(x)$. This encryption algorithm picks a random vector $\overrightarrow{\upsilon} = (s, \upsilon_2, \ldots, \upsilon_n)^\top \in \mathbb{Z}_N^n$. For each $x \in [\ell]$, it selects a random exponent $r_x \in \mathbb{Z}_N$. It then computes:

$C = M \cdot (\prod_{f\in\mathbb{F}_E} e(g,g)^{\alpha_f})^s$, where $\mathbb{F}_E$ denotes the index set of AAs which possess the attributes in the access policy.

$C_0 = g^s$

$C_x = g^{\sum_{f\in\mathbb{F}_E} a_f \cdot A_x \overrightarrow{\upsilon}} T_{\rho(x)}^{-r_x}$

$D_x = g^{r_x}$

The ciphertext is denoted by $CT = (\mathbb{A}, C, C_0, \{C_x, D_x\}_{x\in[\ell]})$.

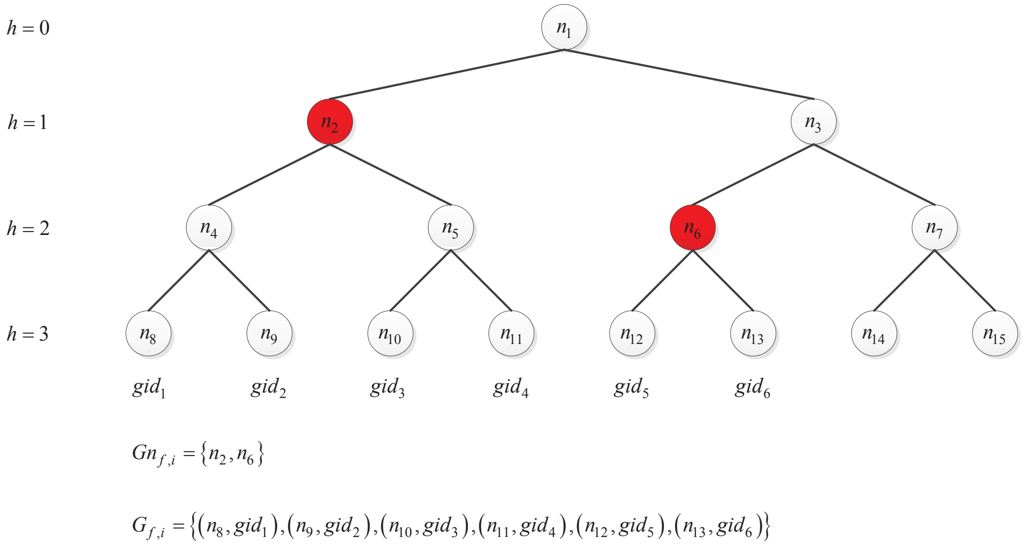

Figure 2: Overview of a binary tree with KEKs

Table 2: Data format on the cloud server

DATA ID	CT	$En(DATA)_{DEK}$

Finally, the data provider selects an unique ID for the data file and uploads the data file onto the cloud server in the format as shown in Table 2.

Once receiving the ciphertext CT uploaded by the data provider, the cloud server computes:

$D'_x = g^{r_x AK_x}$ for each attribute. It then replaces CT by $CT' = (\mathbb{A}, C, C_0, \{C_x, D'_x\}_{x \in [\ell]})$ and stores it.

5.2.1 User registration and Key generation

When a new user joins in the system, he has to register himself and will be distributed a unique *gid*. By running the **CAKeyGen** algorithm, the CA issues the *gid*-related keys to the users. Then, each AA runs the **AAKeyGen** algorithm and gives the attribute-related keys to the users.

CAKeyGen: For each user, the CA first chooses two random exponent $b_{gid}, c_{gid} \in \mathbb{Z}_N$, two random element $R_{gid}, R_{gid,0} \in G_{p_3}$ and computes $CASK_{gid} = L_{gid} = g^{b_{gid}/c_{gid}} R_{gid}$, $L_{gid,0} = g^{1/c_{gid}} R_{gid,0}$. After that, it uses CMK to sign on the string $(CMK,\ gid \parallel CASK_{gid} \parallel L_{gid,0})$ and gets a signature σ_{gid}. Let $CAPK_{gid} = (gid, CASK_{gid}, L_{gid,0}, \sigma_{gid})$. Finally, it sends the $DSK_{gid} = c_{gid}$, $CASK_{gid}$ and $CAPK_{gid}$ to the user *gid*.

AAKeyGen: After receiving the submitted key $CAPK_{gid}$, the AA_f first uses the CPK to verify whether the $CAPK_{gid}$ is valid. If not, it aborts. Otherwise, it issues a set of attributes $S_{gid,f}$ to the user in the $TREE_f$. It randomly selects $R_{gid,f,0} \in G_{p_3}$ and computes

$K_{gid,f} = L_{gid,0}^{\alpha_f} L_{gid}^{a_f} R_{gid,f,0} = g^{\alpha_f/c_{gid}} g^{a_f b_{gid}/c_{gid}} R_{gid,f}$. For each attribute $i \in S_{gid,f}$, it randomly picks $R'_{gid,f,i} \in G_{p_3}$ and computes $K_{gid,f,i} = L_{gid}^{t_{f,i}} R'_{gid,f,i} = T_{f,i}^{b_{gid}/c_{gid}} R_{gid,f,i}$. We write $R_{gid,f} = R_{gid,f,0} R_{gid,0}^{\alpha_f} R_{gid}^{a_f}$, $R_{gid,f,i} = R_{gid}^{t_{f,i}} R'_{gid,f,i}$. In addition, for each attribute $i \in S_{gid,f}$, the AA_f sets the set $G_{f,i}$ and $Gn_{f,i}$. It then encrypts $AK_{f,i}$ by $KEK_{f,j}$ if $j \in Gn_{f,i}$ is the ancestor node of the leaf node which the user is associated with. It finally sends $ASK_{S,gid,f} = (K_{gid,f}, \{K_{gid,f,i}\}_{i \in S_{gid,f}})$ and the encrypted $\{AK_{f,i} | i \in S_{gid,f}\}$ to the user.

5.2.2 Pre-decryption and user decryption

If the user's attribute set S_{gid} satisfies the access policy in the ciphertext, The message M can be recovered by the following 2 algorithms.

Pre-Decrypt: The user first computes $K'_{gid,f,i} = K_{gid,f,i}^{1/AK_{f,i}}$ for each attribute in the set S_{gid}. Let $ASK'_{S,gid,f} = (K_{gid,f}, \{K'_{gid,f,i}\}_{i \in S_{gid,f}})$ be the current keys. The user then sends the $ASK'_{S,gid,f}$s to the cloud server and asks it to pre-decrypt the CT. After receiving these keys, the cloud server runs the **Pre-Decrypt** algorithm and computes $K = \prod_{f \in \mathbb{F}_E} K_{gid,f}$ and constants $\omega_x \in \mathbb{Z}_N$, such that $\sum_{\rho(x) \in S_{gid}} \omega_x A_x = (1, 0, \cdots, 0)$. It then computes:

$$\begin{aligned} PDKEY &= \frac{e(K, C_0)}{\prod_{\rho(x) \in S_{gid}} (e(C_x, L_{gid}) e(D_x, K_{\rho(x)}))^{\omega_x}} \\ &= \frac{e(\prod_{f \in \mathbb{F}_E} g^{\alpha_f/c_{gid}} g^{a_f b_{gid}/c_{gid}} R_{gid,f}, g^s)}{\prod_{\rho(x) \in S_{gid}} (e(g^{\sum_{f \in \mathbb{F}_E} a_f \cdot A_x \vec{v}} T_{\rho(x)}^{-r_x}, g^{b_{gid}/c_{gid}} R_{gid}) e(g^{r_x}, T_{f,i}^{b_{gid}/c_{gid}} R_{gid,f,i}))^{\omega_x}} \\ &= e(g,g)^{\sum_{f \in \mathbb{F}_E} \alpha_f s / c_{gid}} \end{aligned}$$

Finally, it transmits $PDKEY$ to the user.
Decrypt: Each user can recover the message M by computing $M = \frac{C}{PDKEY^{c_{gid}}}$.

5.2.3 User revocation

To achieve fine-grained and on-demand user revocation, we employ the idea from [5][17] and propose an efficient and on-demand attribute-level user revocation method. To distinguish from the revocation approach [17], we assign the process of key updating for the unrevoked users to the AAs rather than the cloud server. The AAs update the user's keys by executing the following algorithm:

Re-KeyGen: Suppose the attribute $i' \in U_{f'}$ is revoked from the user gid', $AA_{f'}$ randomly picks $AK_{f',i'} \in \mathbb{Z}_N$. Whenever a user is about to holding or losing an attribute $i' \in U_{f'}$, $AA_{f'}$ transmits a new attribute group key $AK'_{f',i'}$ to the cloud server via a secure channel. Meanwhile, it also defines a new set $Gu_{f',i'}$ which denotes the minimum set of nodes whose descendant nodes cover all the unrevoked users. It then encrypts $AK'_{f',i'}$ by $KEK_{f',j}$ for each $j \in Gu_{f,i}$. Finally, it transmits the encrypted $\{AK'_{f',i'}\}_{KEK_{f,j}}$ to the unrevoked user which is labeled by the node that is the descendant of j. Fig 3 describes an example of user revocation. When the attribute $i' \in U_{f'}$ is revoked from the user gid_4, the $AA_{f'}$ sets $Gu_{f',i'} = \{n_4, n_{10}, n_6\}$. The user gid_2 will obtain the new $AK'_{f',i'}$ encrypted by KEK_{f',n_4}.

After receiving $\{AK'_{f',i'}\}_{KEK_{f,j}}$, the user recovers the new attribute group key $AK'_{f',i'}$ and updates his key $K_{gid,f',i'}$ by computing

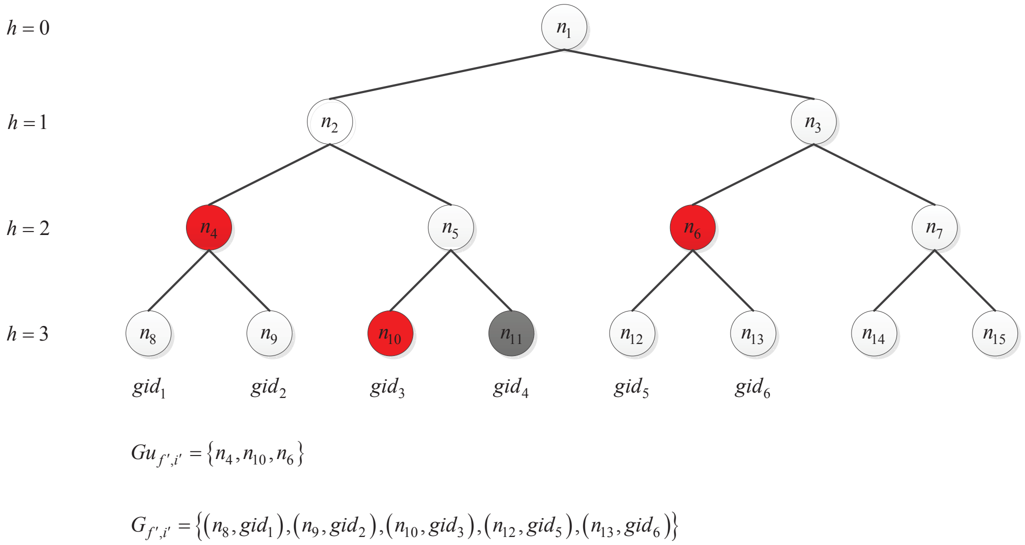

Figure 3: An example of attribute revocation

$K_{gid,f',i'} = T_{f,i}^{b_{gid}/AK'_{f',i'}c_{gid}} R_{gid,f,i}$

The other key elements in $ASK_{S,gid,f}$ will not change if these attributes are not revoked.

Re-Encrypt: When one or more users are revoked from attribute i''s $G_{f,i'}$. The AA_f transmits a re-encryption key $RK_{i'}$ of i' to the cloud server, which is in charge of re-encrypting the corresponding ciphertexts by executing the following two algorithm:

When a user is added into or deleted from an attribute group, the ciphertext element associated with this attribute has to be updated to ensure the forward secrecy and backward secrecy. However, if we demand the data provider to update his ciphertext, it will cause heavy computation overhead. Simultaneously, it will also enforce the data provider to keep online all time. Thus, it is inefficient and unpractical for the data provider to update his ciphertext. In this work, we assign the ciphertext update operation to the cloud server. By employing proxy encryption approach, there is no requirement of the cloud server to decrypt the ciphertext prior to updating.

When the cloud server receives $AK_{f',i'}$, it computes $D''_{x'} = g^{r_x AK_{f',i'}}$, where $\rho(x') = i' \in U_{f'}$. It then sets

$CT'' = (\mathbb{A}, C, C_0, \{C_x, D'_x\}_{x\in[\ell]\setminus\{x'\}}, \{C_{x'}, D''_{x'}\})$

It replaces CT' by CT''. It will store the latest version of the ciphertext only.

6 Security analysis

We now analyze the security of the proposed system by proving that it fulfills the security requirements represented in Section 3. We first prove our MA-CP-ABE scheme is fully secure against chosen plaintext attack in the standard model. Security in this model means that the adversary can not decrypt the ciphertext unless it possesses proper attributes which satisfy the intended access structure. In addition, we show that our revocation approach achieves both the forward secrecy and backward secrecy.

6.1 Fine-Grained Data Access Control

We demonstrate that the proposed scheme is proven fully secure in the standard model by proving Theorem 1. In addition, each user is distributed a random exponent c_{gid}. Therefore, neither the cloud server nor the unauthorized users can learn about the contents of the encrypted sensitive data. Notably, we achieve more strong notion of security while these multi-authority systems [31, 33] were proven secure in the selective model. The detailed formal security proof of our system can be found in **Appendix B**.

Theorem 1. *Suppose that the signature system is existentially unforgeable against adaptive chosen message attack (UF-CMA) and Assumption 1,2,3 [21] holds. Then no polynomial-time adversary $\mathcal{A}$ can break our MA-CP-ABE scheme with a non-negligible advantage.*

6.2 Collusion Resistance

We sketch the collusion resistance of the proposed system as follows: Like the previous multi-authority ABE schemes [7, 8, 22, 24], every user is issued a unique *gid* with which all of his private keys are linked. In our system, the private keys of each user are associated with the randomly chosen exponents b_{gid} and c_gid. Recall that the EDK is encrypted by the key $\prod_{f \in \mathbb{F}_E} e(g,g)^{\alpha_f s}$. To recover such key, the colluding users have to cancel the redundant element $e(\prod_{f \in \mathbb{F}_E} g^{a_f b_{gid}/c_{gid}} R_{gid,f}, g^s)$ in executing $e(K, C_0)$. Unfortunately, when they combine their key to recover $e(\prod_{f \in \mathbb{F}_E} g^{a_f b_{gid}/c_{gid}} R_{gid,f}, g^s)$ by computing $\prod_{\rho(x) \in S_{gid}} (e(C_x, L_{gid}) e(D_x, K_{\rho(x)}))^{\omega_x}$. they will fall due to the fact that b_{gid} and c_gid are independently and random picked for each user. Actually, the collusion resistance is implicitly proved in the security proof of the proposed system.

6.3 Back Secrecy and Forward Secrecy

Back secrecy: When a user is issued a new attributes at some point, the corresponding $AK_{f,i}$ will be updated and transmitted to the valid users (including the user). Moreover, the relevant ciphertext components are also re-encrypted under the new $AK_{f,i}$ by the cloud server. In this way, the ciphertext associated with old attribute group key can not be decrypted by the private keys labeled by new $AK_{f,i}$. Therefore, back secrecy is guaranteed.

Forward secrecy: When an attribute is revoked from a user at some time instance, the corresponding $AK_{f,i}$ will be updated and transmitted to the unrevoked users. meanwhile, the relevant ciphertext components are also re-encrypted under the new $AK_{f,i}$ by the cloud server. In this way, the ciphertext associated with new attribute group key can not be decrypted by the private keys labeled by old $AK_{f,i}$. Thus, forward secrecy is guaranteed.

Table 3: Notions used in numeric comparison

S_U	the sets of attributes held by a user
S_C	the sets of attributes included in the ciphertext
S_D	the sets of attributes used in decryption
ℓ	the number of rows of the access matrix A
$\mathbb{F}_U$	the index set of the AAs related to a user
$\|\vartheta\|$	the bit length of the element in $\mathbb{Z}_\vartheta$
$\|\mathbb{G}\|$	the bit length of the element in $\mathbb{G}$
$\|\mathbb{G}_1\|$	the bit length of the element in $\mathbb{G}_1$
E_1	one exponent operation in $\mathbb{G}$
E_2	one exponent operation in $\mathbb{G}_1$
E_3	one symmetry encryption on the $AK_{f,i}$
P	one bilinear map operation
m	the total number of users in the system
n	the number of users who possess the revoked attribute $ATT_{f,i}$
r	the number of users who will be revoked from $G_{f,i}$

7 Performance analysis

In this section, we compare the performance of our scheme with that of adaptively secure decentralizing CP-ABE scheme [22] and selectively secure multi-authority access control system [32], in terms of communication overhead and computation cost. The numeric comparison results are summarized in Table 4,5,6. The notations employed in these tables are described in Table 3:

7.1 Communication cost

In Table 4, we discuss the theoretical results of communication overhead which are mainly incurred by key generation and ciphertext transmission. Especially, We calculate the communication overhead in terms of the size of keys that a user can obtain from the AAs or CA and the size of ciphertext that a data provider has to host to the cloud servers. From the table, we can see that the proposed scheme and YJRZ's scheme incur less communication overhead from the cloud server to users, since only the elements C and $e(g,g)^{\sum_{f\in\mathbb{F}_E}\alpha_f s/c_{gid}}$ are sent. Our scheme require less number of elements which are transmitted from the data provider to the the cloud server. LW's scheme does not cause any communication overhead from the user to the cloud server, but incur heavy computation cost during decryption which will be discussed later.

In Table 5, We further compare the communication cost caused by key updating and ciphertext updating. For each revocation execution, we assume that only one attribute is revoked from some users. We note that LW's scheme does not provide any user revocation approach. If r users are revoked from $G_{f,i}$, YJRZ's scheme needs to send $(n-r)|\mathbb{G}|$ bits, while our scheme requires to pass $(n-r)|\mathbb{Z}|$ bits. Moreover, The AAs in YJRZ's scheme has to send additional updating message to notify the data providers that the public keys of the revoked attributes have been updated. This will cause heavy communication overhead incurred by frequently attribute revocation. Such process will increase the security risk of the system. However, our scheme realizes the user revocation by updating the key $AK_{f,i}$. There is no requirement to update the public keys. Thus, our scheme

Table 4: Comparison of Communication Overhead in Ordinary Executions

Communication Overhead from	LW's [22]	YJRZ's [32]	Ours	
CA to User	\	$2\lvert\mathbb{G}\rvert + 2\lvert\vartheta\rvert$	$2\lvert\mathbb{G}\rvert + 1\lvert\vartheta\rvert$	
AAs to User	$\lvert S_U\rvert\lvert\mathbb{G}\rvert$	$(3\lvert\mathbb{F}_U\rvert + \lvert S_U\rvert)\lvert\mathbb{G}\rvert$	$(\lvert\mathbb{F}_U\rvert + \lvert S_U\rvert)\lvert\mathbb{G}\rvert$	
Data Provider to Server	$2\ell\lvert\mathbb{G}\rvert + (1+\ell)\lvert\mathbb{G}_1\rvert$	$(3\ell+2)\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_1\rvert$	$(2\ell+1)\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_1\rvert$	
User to Server	\	$(3\lvert\mathbb{F}_U\rvert + \lvert S_U\rvert + 1)\lvert\mathbb{G}\rvert$	$(\lvert\mathbb{F}_U\rvert + \lvert S_U\rvert + 1)\lvert\mathbb{G}\rvert$	
Server to User	$2\ell\lvert\mathbb{G}\rvert + (1+\ell)\lvert\mathbb{G}_1\rvert$	$1\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_1\rvert$	$1\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_1\rvert$	

Table 5: Comparison of Communication Overhead in each revocation execution

Communication Overhead from	LW's [22]	YJRZ's [32]	Our scheme
AA_f to data providers	\	Notification	\
AA_f to unrevoked User	\	$(n-r)\lvert\mathbb{G}\rvert$	$(n-r)\lvert\vartheta\rvert$
AA_f to Server	\	$1\lvert\vartheta\rvert$	$1\lvert\vartheta\rvert$

requires less communication times and is more acceptable.

7.2 Computation cost

Table 6 shows the comparison results of computation cost between LW's scheme, YJRZ's scheme and ours. The computation overhead is calculated in terms of the times of exponent and bilinear pairing operation which are executed during encryption, decryption, ciphertext updating and key updating, respectively. We calculate the computation cost of key updating per attribute which is caused by AA_f to compute the key update message for the unrevoked users. As shown in Table 6, we can find that our scheme require less exponent operations during encrypting the DEK. In the decryption phase, YJRZ's scheme and ours require each user to compute the modular exponentiation only one time. It is unrelated to the number of attributes used in decryption. Nevertheless, The computation cost per a user in LW's scheme is $2|S_D|P + |S_D|E_2$. The cloud server in YJRZ's scheme and ours has to execute one time of exponent operation per attribute. During key updating, the AA_f in YJRZ's scheme has to generate a new unique update key for each unrevoked user. It is required to compute $n-r$ times of Modular Exponentiation. However, for all unrevoked users in our scheme, the AA_f chooses only one element in $\mathbb{G}_N$ and encrypts it under KEK_j, where $j \in Gu_{f,i}$. It needs to execute about $log\frac{m}{n-r}$ times of symmetrical encryption. Although the bilinear pairing operations on composite order groups is more complicated than that on prime order groups, our scheme does not add more computation cost on the user side since that the heavy computation cost is outsourced to the cloud server. Moreover, when some attributes are revoked, there is no requirement of the AAs in our scheme to generate different key updating subassembly for each unrevoked user. Therefore, our scheme is more efficient and scalable.

8 Conclusion

In this work, we proposed an attribute-based data access control system with multiple authorities in cloud storage. We first constructed a novel multi-authority CP-ABE scheme, which enables the data

Table 6: Comparison of Computation cost

Schemes	YJ's [22]	YJRZ's [32]	Ours
Encryption	$3\ell E_1 + (1+2\ell)E_2$	$(2+4\ell)E_1 + 1E_2$	$(1+3\ell)E_1 + 1E_2$
Decryption	$2\|S_D\|P + \|S_D\|E_2$	$1E_2$	$1E_2$
ciphertext-update	\	$1E_1$	$1E_1$
Key-update	\	$(n-r)E_1$	$log\frac{m}{n-r}E_3$

provider to define and enforce suitable access policy. The proposed scheme was proved adaptive security in the standard model and supports any monotone LSSS access policy. The decryption overhead of users is alleviated by outsourcing the complicated bilinear pairing operations to the clouds. We also presented an attribute-level user revocation method basing on the proxy encryption technique. The security analysis, numeric comparisons indicated that our data access control system is secure, scalable and efficient for cloud storage.

Acknowledgments

This research is supported by Changjiang Scholars and Innovative Research Team in University under grant No. IRT1078; The Key Program of NSFC- Guangdong Union Foundation under grant No. U1135002; Major national S&T program under grant No. 2011ZX03005-002; the Fundamental Research Funds for the Central Universities under grant No. JY10000903001. We thank the sponsors for their support and the reviewers for helpful comments.

References

[1] Nuttapong Attrapadung and Hideki Imai. Conjunctive broadcast and attribute-based encryption. In Hovav Shacham and Brent Waters, editors, *Pairing-Based Cryptography Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*, pages 248–265. Springer Berlin Heidelberg, 2009.

[2] A. Beimel. Secure schemes for secret sharing and key distribution. *DSc dissertation*, 1996.

[3] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP '07. IEEE Symposium on*, pages 321 –334, may 2007.

[4] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144. Springer Berlin Heidelberg, 1998.

[5] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 417–426, New York, NY, USA, 2008. ACM.

[6] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *Advances in Cryptology EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer Berlin Heidelberg, 2003.

[7] Melissa Chase. Multi-authority attribute based encryption. In SalilP. Vadhan, editor, *Theory of Cryptography*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534. Springer Berlin Heidelberg, 2007.

[8] Melissa Chase and Sherman S.M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 121–130, New York, NY, USA, 2009. ACM.

[9] Ling Cheung and Calvin Newport. Provably secure ciphertext policy abe. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 456–465, New York, NY, USA, 2007. ACM.

[10] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Over-encryption: management of access control evolution on outsourced data. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 123–134. VLDB Endowment, 2007.

[11] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded ciphertext policy attribute based encryption. In Luca Aceto, Ivan Damgrd, LeslieAnn Goldberg, MagnsM. Halldrsson, Anna Inglfsdttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 579–591. Springer Berlin Heidelberg, 2008.

[12] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 89–98, New York, NY, USA, 2006. ACM.

[13] Matthew Green, Susan Hohenberger, and Brent Waters. Outsourcing the decryption of abe ciphertexts. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 34–34, Berkeley, CA, USA, 2011. USENIX Association.

[14] J. Hur. Improving security and efficiency in attribute-based data sharing, 2011.

[15] Junbeom Hur. Improving security and efficiency in attribute-based data sharing. *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[16] Junbeom Hur and Dong Kun Noh. Attribute-based access control with efficient revocation in data outsourcing systems. *Parallel and Distributed Systems, IEEE Transactions on*, 22(7):1214–1221, 2011.

[17] Junbeom Hur and Dong Kun Noh. Attribute-based access control with efficient revocation in data outsourcing systems. *Parallel and Distributed Systems, IEEE Transactions on*, 22(7):1214–1221, 2011.

[18] Taeho Jung, Xiang yang Li, Zhiguo Wan, and Meng Wan. Privacy preserving cloud data access with multi-authorities. In *INFOCOM, 2013 Proceedings IEEE*, pages 2625–2633, 2013.

[19] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *Proceedings of the 7th ACM conference on Computer and communications security*, CCS '00, pages 235–244, New York, NY, USA, 2000. ACM.

[20] J. Lai, R.H. Deng, C. Guan, and J. Weng. Attribute-based encryption with verifiable outsourced decryption. *Information Forensics and Security, IEEE Transactions on*, 8(8):1343–1354, 2013.

[21] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer Berlin Heidelberg, 2010.

[22] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In KennethG. Paterson, editor, *Advances in Cryptology EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 568–588. Springer Berlin Heidelberg, 2011.

[23] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *Parallel and Distributed Systems, IEEE Transactions on*, 24(1):131–143, 2013.

[24] Zhen Liu, Zhenfu Cao, Qiong Huang, DuncanS. Wong, and TszHon Yuen. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In Vijay Atluri and Claudia Diaz, editors, *Computer Security ESORICS 2011*, volume 6879 of *Lecture Notes in Computer Science*, pages 278–297. Springer Berlin Heidelberg, 2011.

[25] Peter Mell and Timothy Grance. The nist definition of cloud computing (draft). *NIST special publication*, 800(145):7, 2011.

[26] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 195–203, New York, NY, USA, 2007. ACM.

[27] S. Ruj, A. Nayak, and I. Stojmenovic. Dacc: Distributed access control in clouds. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 91–98, 2011.

[28] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer Berlin Heidelberg, 2005.

[29] Zhiguo Wan, Jun'e Liu, and R.-H. Deng. Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *Information Forensics and Security, IEEE Transactions on*, 7(2):743–754, 2012.

[30] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography PKC 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer Berlin Heidelberg, 2011.

[31] Kan Yang and Xiaohua Jia. Attributed-based access control for multi-authority systems in cloud storage. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 536–545, 2012.

[32] Kan Yang, Xiaohua Jia, Kui Ren, and Bo Zhang. Dac-macs: Effective data access control for multi-authority cloud storage systems. In *INFOCOM, 2013 Proceedings IEEE*, pages 2895–2903, 2013.

[33] Shucheng Yu, Kui Ren, and Wenjing Lou. Fdac: Toward fine-grained distributed data access control in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(4):673–686, 2011.

[34] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[35] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 261–270, New York, NY, USA, 2010. ACM.

Appendix A: Assumptions and Security Game

Assumption 1. *Subgroup decision problem for 3 primes [21]. Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}&\mathbb{G} = (N = p_1p_2p_3, G, G_1, e) \overset{R}{\longleftarrow} \mathcal{G},\\&g \overset{R}{\longleftarrow} G_{p_1}, X_3 \overset{R}{\longleftarrow} G_{p_3},\\&D = (\mathbb{G}, g, X_3),\\&T_1 \overset{R}{\longleftarrow} G_{p_1p_2}, T_2 \overset{R}{\longleftarrow} G_{p_1}.\end{aligned}$$

The advantage with which an algorithm $\mathcal{A}$ can break Assumption 1 is defined as:

$$Adv1_{\mathcal{G},\mathcal{A}} = |Pr[\mathcal{A}(D,T_1) = 1] - Pr[\mathcal{A}(D,T_2) = 1]|$$

Definition 3. *We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{A}}$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.*

Assumption 2. *[21]. Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}&\mathbb{G} = (N = p_1p_2p_3, G, G_1, e) \overset{R}{\longleftarrow} \mathcal{G},\\&g, X_1 \overset{R}{\longleftarrow} G_{p_1}, X_2, Y_2 \overset{R}{\longleftarrow} G_{p_2}, X_3, Y_3 \overset{R}{\longleftarrow} G_{p_3},\\&D = (\mathbb{G}, g, X_1X_2, X_3, Y_2Y_3),\\&T_1 \overset{R}{\longleftarrow} G, T_2 \overset{R}{\longleftarrow} G_{p_1p_3}.\end{aligned}$$

The advantage with which an algorithm $\mathcal{A}$ can break Assumption 2 is defined as:

$$Adv2_{\mathcal{G},\mathcal{A}} = |Pr[\mathcal{A}(D,T_1) = 1] - Pr[\mathcal{A}(D,T_2) = 1]|$$

Definition 4. *We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{A}}$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.*

Assumption 3. *[21]. Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}
&\mathbb{G} = (N = p_1p_2p_3, G, G_1, e) \stackrel{R}{\longleftarrow} \mathcal{G}, \alpha, s \stackrel{R}{\longleftarrow} \mathbb{Z}_N\\
&g \stackrel{R}{\longleftarrow} G_{p_1}, X_2, Y_2, Z_2 \stackrel{R}{\longleftarrow} G_{p_2}, X_3 \stackrel{R}{\longleftarrow} G_{p_3},\\
&D = (\mathbb{G}, g, g^{\alpha}X_2, X_3, g^sY_2, Z_2),\\
&T_1 = e(g,g)^{\alpha s}, T_2 \stackrel{R}{\longleftarrow} G_1.
\end{aligned}$$

The advantage with which an algorithm $\mathcal{A}$ can break Assumption 2 is defined as:

$$Adv3_{\mathcal{G},\mathcal{A}} = |Pr[\mathcal{A}(D,T_1) = 1] - Pr[\mathcal{A}(D,T_2) = 1]|$$

Definition 5. *We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},\mathcal{A}}$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.*

We now introduce the security game run between an adversary $\mathcal{A}$ an a simulator $\mathcal{B}$. $\mathcal{A}$ is assumed to corrupt at most $F-1$ AAs. We let $\mathbb{F}_c$, $\mathbb{F}_{uc} = \mathbb{F}\backslash\mathbb{F}_c$ denote the index set of corrupted, uncorrupted AAs, respectively.

Setup: The simulator $\mathcal{B}$ runs the **GlobalSetup**, **CASetup** and **AASetup** algorithms. It then transmits the system public parameters GPK, CPK, and $\bigcup_{f=1}^{F} APK_f$ to the adversary $\mathcal{A}$. $\mathcal{A}$ appoints an index set of the AAs $\mathbb{F}_c$ which it wants to corrupt, where $\mathbb{F}\backslash\mathbb{F}_c \neq \varnothing$. For $f \in \mathbb{F}_c$, $\mathcal{B}$ sends the master key $\{AMK_f | f \in \mathbb{F}_c\}$ to $\mathcal{A}$.

Phase 1: The adversary can make adaptive secret key queries as follows:

CAkey queries: To answer these queries, $\mathcal{B}$ responds by gid, DSK_{gid}, $CASK_{gid}$ and $CAPK_{gid}$.

AAkey queries: The adversary makes AAkey queries by submitting $\bigcup S_{gid,f}$ and $CAPK_{gid}$ to $\mathcal{B}$, where $f \in \mathbb{F}_{uc}$. $\mathcal{B}$ returns the $\{ASK_{S,gid,f}\}_{f\in\mathbb{F}_{uc}}$.

Challenge: The adversary declares two equal-length message M_0, M_1 and an challenge access structure $\mathbb{A}^*$. $\mathcal{B}$ first flips a random coin $b \in \{0,1\}$. It then encrypts M_b under $\mathbb{A}^*$ and get the challenge ciphertext CT^*. It gives CT^* to $\mathcal{A}$.

Phase 2: The adversary can make adaptive secret key queries as in **Phase 1**.

Guess: $\mathcal{A}$ outputs its guess b' of b.

We note that the adversary can not make AAkey queries on the attribute set $S_{gid,f}$ such that $(\bigcup_{f\in\mathbb{F}_{uc}} S_{gid,f})\bigcup(\bigcup_{f\in\mathbb{F}_c} U_f)$ can satisfy the challenge access structure $\mathbb{A}^*$. The advantage of $\mathcal{A}$ is defined as $Pr[b' = b] - \frac{1}{2}$.

Appendix B: Security Proof

Before giving out our proof, we have to introduce the definitions of two additional structures: semi-functional ciphertexts and keys. These structures will not be employed in the real constructions, but are necessary in the proof. For each attribute $i \in U_f$, we picks a random exponent $z_{f,i} \in \mathbb{Z}_N$.

Semi-functional ciphertext: A semi-functional ciphertext is formed in the following way. We let g_2 be a generator of $\mathbb{G}_{p_2}$, c be a random chosen exponent from $\mathbb{Z}_N$. For each row $x \in [\ell]$, we randomly selects $\gamma_x \in \mathbb{Z}_N$. In addition, we chooses a random vector $\overrightarrow{y} \in \mathbb{Z}_N^n$. Then, we set:

$C_0 = g^s g_2^c$

For each $x \in [\ell]$:

$C_x = g^{\sum_{f \in \mathbb{F}_E} a_f \cdot A_x \overrightarrow{v}} T_{\rho(x)}^{-r_x} g_2^{A_x \overrightarrow{y} + \gamma_x z_{\rho(x)}}$

$D_x = g^{r_x} g_2^{-\gamma_x}$

Semi-functional key: For a gid, a semi-functional key will be one of the two following forms:

A semi-functional key of type 1: We picks random exponents $b, d_f \in \mathbb{Z}_N$ and set:

$L_{gid} = g^{b_{gid}/c_{gid}} R_{gid} g_2^b$

$L_{gid,0} = g^{1/c_{gid}} R_{gid,0}$

$CAPK_{gid} = (gid, CASK_{gid}, L_{gid,0}, \sigma_{gid})$

$DSK_{gid} = c_{gid}$

$K_{gid,f} = g^{\alpha_f/c_{gid}} g^{a_f b_{gid}/c_{gid}} R_{gid,f} g_2^{d_f}$

$K_{gid,f,i} = T_{f,i}^{b_{gid}/c_{gid}} R_{gid,f,i} g_2^{bz_{f,i}}$

A semi-functional key of type 2:

$L_{gid} = g^{b_{gid}/c_{gid}} R_{gid}$

$L_{gid,0} = g^{1/c_{gid}} R_{gid,0}$

$CAPK_{gid} = (gid, CASK_{gid}, L_{gid,0}, \sigma_{gid})$

$DSK_{gid} = c_{gid}$

$K_{gid,f} = g^{\alpha_f/c_{gid}} g^{a_f b_{gid}/c_{gid}} R_{gid,f} g_2^{d_f}$

$K_{gid,f,i} = T_{f,i}^{b_{gid}/c_{gid}} R_{gid,f,i}$

Remark that, if we use a semi-functional key to decrypt a normal ciphertext or a normal key to decrypt a semi-ciphertext, $\prod_{f \in \mathbb{F}_E} e(g,g)^{\alpha_f s}$ can be correctly computed. However, when a semi-functional key is used to decrypt a semi-functional ciphertext, we will get an additional term: $e(g_2, g_2)^{c \sum_{f \in \mathbb{F}_E} d_f - by_1}$, where y_1 is the first coordinate of $\overrightarrow{y}$. A semi-functional key of type 1 is said to be nominal if $c \sum_{f \in \mathbb{F}_E} d_f - by_1 = 0$. In this case, such a semi-functional key can decrypt the correspond semi-functional ciphertext.

To prove the adaptive security of our scheme from Assumptions 1,2,3, a sequence of games are used. The detailed definitions are given in the following:

Game$_{Real}$: The first game **Game**$_{Real}$ denotes the real security game. i.e., all users' keys and the challenge ciphertext are normal.

Game$_0$: In this game, all users' keys are normal, but the challenge ciphertext is semi-functional.

We let q be the number of key queries made by the adversary $\mathcal{A}$. For k from 1 to q, we consider:

Game$_{k,1}$: In this game, the first $k-1$ keys are semi-functional form of **type 2**. The k-th key is semi-functional form of **type 1**. The remaining keys are normal.

Game$_{k,2}$: In this game, the first k keys are semi-functional form of **type 2**. The remaining keys are normal.

Game$_{Final}$: In this game, all the keys are semi-functional form of **type 2**. The challenge ciphertext is a semi-functional encryption of a random message.

We will prove that these games are indistinguishable in the following 4 lemmas. Without loss of generality, we assume that the adversary $\mathcal{A}$ can corrupt all AAs but $AA_{f'}$. $\mathbb{B}$ will answer the key queries relevant $AA_{f'}$.

Lemma 1. *Given a UF-CMA secure signature scheme, suppose that there is a PPT adversary $\mathcal{A}$ such that **Game**$_{Real}Adv_{\mathcal{A}}$ − **Game**$_0 Adv_{\mathcal{A}} = \epsilon$. Then we can construct a PPT simulator $\mathcal{B}$ to break Assumption 1 with advantage ϵ.*

Proof. $\mathcal{B}$ is given the terms $\{g, X_3, T\}$. It then simulates $\mathbf{Game}_{Real}$ or $\mathbf{Game}_0$ with $\mathcal{A}$. $\mathcal{B}$ sets GPK, CPK and $\bigcup_{f=1}^{F} APK_f$ as in the real system. It then transmits them to the adversary along with $\{AMK_f\}_{f\in\mathbb{F}_c}$. $\mathcal{B}$ will respond the key queries from $\mathcal{A}$ by normal keys.

In the challenge phase, $\mathcal{A}$ gives $\mathcal{B}$ two equal-length message M_0, M_1 and the challenge access structure $\mathbb{A}^* = (A, \rho)$. To create the challenge ciphertext, $\mathcal{B}$ will set g^s as the G_{p_1} part of T. It implicitly means that T denotes the product of g^s and possibly an element $g_2^c \in G_{p_2}$. $\mathcal{B}$ then randomly picks $b \in \{0, 1\}$ and compute:

$C = M_b \cdot e(g^{\sum_{f\in\mathbb{F}_E} \alpha_f}, T)$

$C_0 = T$

$\mathcal{B}$ randomly selects $\overrightarrow{\delta} = (1, \delta_2, \ldots, \delta_n)^\top \in \mathbb{Z}_N^n$. For each $x \in [\ell]$, it also chooses random exponent $r'_x \in \mathbb{Z}_N$. It sets:

$C_x = T^{\sum_{f\in\mathbb{F}_E} a_f \cdot A_x \overrightarrow{\delta}} T^{-r'_x t_{\rho(x)}}$

$D_x = T^{r'_x}$

In this case, $\mathcal{B}$ implicitly sets $\overrightarrow{v} = s\overrightarrow{\delta}$ and $r_x = sr'_x$. If $T \in G_{p_1}$, CT^* is a correctly distributed normal ciphertext. However, if $T \in G_{p_1p_2}$, we implicitly set $\overrightarrow{y} = c\sum_{f\in\mathbb{F}_E} a_f \overrightarrow{\delta}$, $\gamma_x = -cr'_x$ and $z_{\rho(x)} = t_{\rho(x)}$. According to the Chinese Remainder Theorem, the values of $\sum_{f\in\mathbb{F}_E} a_f$, δ_2, …, δ_n, r'_x and $t_{\rho(x)}$ modulo p_1 are uncorrelated from those of modulo p_2. Thus, CT^* is a correctly distributed semi-functional ciphertext. Therefore, $\mathcal{B}$ can obtain advantage ϵ in breaking Assumption 1 by using the output of $\mathcal{A}$.

Lemma 2. *Given a UF-CMA secure signature scheme, suppose that there is a PPT adversary $\mathcal{A}$ such that $\boldsymbol{Game}_{k-1,2}Adv_{\mathcal{A}} - \boldsymbol{Game}_{k,1}Adv_{\mathcal{A}} = \epsilon$. Then we can employ $\mathcal{A}$ to construct a PPT simulator $\mathcal{B}$ to break Assumption 2 with advantage negligibly approximate to ϵ.*

Proof. $\mathcal{B}$ is given the terms $\{g, X_1X_2, X_3, Y_2Y_3, T\}$. It will play $\mathbf{Game}_{k-1,2}$ or $\mathbf{Game}_{k,1}$ with $\mathcal{A}$. Same as the games mentioned before, $\mathcal{B}$ sets GPK, CPK and $\bigcup_{f=1}^{F} APK_f$ and passes them to the adversary along with $\{AMK_f\}_{f\in\mathbb{F}_c}$.

$\mathcal{B}$ creates the first $k-1$ semi-functional keys of type 2 as follows:

For each CAkey query, $\mathcal{B}$ acts as the real system.

For each AAkey query, $\mathcal{B}$ first sets $K_{gid,f'} = g^{\alpha_{f'}/c_{gid}} g^{a_{f'}b_{gid}/c_{gid}} (Y_2Y_3)^{b_{gid}}$. For each attribute $i \in S_{gid,f'}$, it sets $K_{gid,f',i}$ as the real system. We remark that $K_{gid,f'}$ is properly distributed since that the value of b_{gid} modulo p_1 is uncorrelated to its values modulo p_2 and p_3.

$\mathcal{B}$ responds the k-th key query by a semi-functional key of type 1 in the following way:

To answer the k-th CAkey query, $\mathcal{B}$ chooses a gid and sets: $DSK_{gid} = c_{gid}$, $CASK_{gid} = L_{gid} = T^{1/c_{gid}} R_{gid}$, $L_{gid,0} = g^{1/c_{gid}} R_{gid,0}$ and $CAPK_{gid} = (gid, CASK_{gid}, L_{gid,0}, \sigma_{gid})$, where $c_{gid} \in \mathbb{Z}_N$ and $R_{gid}, R_{gid,0} \in G_{p_3}$. It means that $\mathcal{B}$ implicitly sets $g^{b_{gid}}$ identical to the G_{p_1} part of T.

To answer the k-th AAkey query, $\mathcal{B}$ randomly selects $R_{gid,f'} \in G_{p_3}$ and sets: $K_{gid,f'} = g^{\alpha_{f'}/c_{gid}} T^{a_{f'}/c_{gid}} R_{gid,f'}$. For each $i \in S_{gid,f'}$, $\mathcal{B}$ randomly picks $R_{gid,f',i} \in G_{p_3}$ and sets $K_{gid,f',i} = T^{t_{f',i}/c_{gid}} R_{gid,f',i}$.

We note that if $T \in G_{p_1p_3}$, this is a correctly distributed normal key. Otherwise, this is a correctly distributed semi-functional key of type 1.

For the key requests $> k$, $\mathcal{B}$ acts same as in the real construction and responds by the normal keys.

In the challenge phase, $\mathcal{A}$ submits two equal-length message M_0, M_1 and the challenge access structure $\mathbb{A}^* = (A, \rho)$ to $\mathcal{B}$. To create the semi-functional ciphertext, $\mathcal{B}$ implicitly sets $g^s = X_1$ and $g_2^c = X_2$. It also defines the vector $\overrightarrow{\zeta} = (\sum_{f\in\mathbb{F}_E} a_f, \zeta_2, \ldots, \zeta_n)^\top$, where $\zeta_2, \ldots, \zeta_n$ are randomly

chosen from $\mathbb{Z}_N$. For each $x \in [\ell]$, it selects a random exponent $r'_x \in \mathbb{Z}_N$. Finally, it sets the ciphertext as follows:
$C = M_b \cdot e(g^{\sum_{f \in \mathbb{F}_E} \alpha_f}, X_1 X_2)$
$C_0 = X_1 X_2$
$C_x = (X_1 X_2)^{A_x \overrightarrow{\zeta}} (X_1 X_2)^{-r'_x t_{\rho(x)}}$
$C_x = (X_1 X_2)^{r'_x}$

In this case, $\mathcal{B}$ implicitly sets $\overrightarrow{v} = s(\sum_{f \in \mathbb{F}_E} \alpha_f)^{-1} \overrightarrow{\delta}$ and $\overrightarrow{y} = c\overrightarrow{\delta}$. It also means that $\mathcal{B}$ sets $r_x = sr'_x$, $\gamma_x = -cr'_x$. Due to the argument in [21], we have: the k-th key and the challenge ciphertext are correctly distributed in the adversary's view with probability approximate to 1. Therefore, if $T \in G_{p_1 p_3}$, $\mathcal{B}$ has correctly simulated $\mathbf{Game}_{k-1,2}$. If $T \in G$ and the values of all γ_x modulo p_2 are non-zero, $\mathcal{B}$ has successfully simulated $\mathbf{Game}_{k,1}$. Hence, $\mathcal{B}$ can get advantage negligibly approximate to ϵ in breaking Assumption 2 by using the guess of $\mathcal{A}$.

Lemma 3. *Given a UF-CMA secure signature scheme, suppose that there is a PPT adversary $\mathcal{A}$ such that $\boldsymbol{Game}_{k,1} Adv_{\mathcal{A}} - \boldsymbol{Game}_{k,2} Adv_{\mathcal{A}} = \epsilon$. Then we can employ $\mathcal{A}$ to construct a PPT simulator $\mathcal{B}$ to break Assumption 2 with advantage ϵ.*

Proof. $\mathcal{B}$ is given the terms $\{g, X_1 X_2, X_3, Y_2 Y_3, T\}$. It will play $\mathbf{Game}_{k-1,2}$ or $\mathbf{Game}_{k,1}$ with $\mathcal{A}$. Same as the games mentioned before, $\mathcal{B}$ sets GPK, CPK and $\bigcup_{f=1}^{F} APK_f$ and passes them to the adversary along with $\{AMK_f\}_{f \in \mathbb{F}_c}$.

$\mathcal{B}$ constructs the $k-1$ semi-functional keys of type 2, the $> k$ normal keys and the challenge ciphertext same as in **lemma 2**. However, in **lemma 3**, we add an extra term in $K_{gid,f'}$. Hence, the k-th key is no longer nominally semi-functional and is uncorrelated to the value in the G_{p_2} part of the challenge ciphertext. The detailed construction of the k-th key is as follows:

To answer the k-th AAkey query, $\mathcal{B}$ randomly selects $R_{gid,f'} \in G_{p_3}$ and $h \in \mathbb{Z}_N$. It then sets: $K_{gid,f'} = g^{\alpha_{f'}/c_{gid}} T^{a_{f'}/c_{gid}} R_{gid,f'} (Y_2 Y_3)^h$. For each $i \in S_{gid,f'}$, $\mathcal{B}$ randomly picks $R_{gid,f',i} \in G_{p_3}$ and sets $K_{gid,f',i} = T^{t_{f',i}/c_{gid}} R_{gid,f',i}$.

In a word, the k-th key is properly distributed and is either semi-functional of type 2 ($T \in G_{p_1 p_3}$) or semi-functional of type 1 ($T \in G$). Thus, $\mathcal{B}$ will get advantage ϵ in breaking Assumption 2 by employing the guess of $\mathcal{A}$.

Lemma 4. *Given a UF-CMA secure signature scheme, suppose that there is a PPT adversary $\mathcal{A}$ such that $\boldsymbol{Game}_{q,2} Adv_{\mathcal{A}} - \boldsymbol{Game}_{Final} Adv_{\mathcal{A}} = \epsilon$. Then we can employ $\mathcal{A}$ to construct a PPT simulator $\mathcal{B}$ to break Assumption 3 with advantage ϵ.*

Proof. Given the terms $\{g, g^{\alpha} X_2, X_3, g^s Y_2, Z_2, T\}$, $\mathcal{B}$ will simulate $\mathbf{Game}_{q,2}$ or $\mathbf{Game}_{Final}$ with $\mathcal{A}$. It sets the system public parameters same as in the proofs mentioned before other than the $e(g,g)^{\alpha_{f'}}$ term, which is computed by $e(g,g)^{\alpha_{f'}} = e(g, g^{\alpha} X_2)$. The system public parameters are send to the adversary along with $\{AMK_f\}_{f \in \mathbb{F}_c}$.

For each CAkey query, $\mathcal{B}$ acts as the real system.

For each AAkey query, $\mathcal{B}$ responds by the semi-functional key of type 2. $\mathcal{B}$ sets $K_{gid,f'} = g^{\alpha_{f'}/c_{gid}} g^{a_{f'} b_{gid}/c_{gid}} (Z_2)^{b_{gid}} R_{gid,f'}$, where $R_{gid,f'}$ is set same as the real construction. For each $i \in S_{gid,f'}$, $\mathcal{B}$ randomly chooses $R_{gid,f',i} \in G_{p_3}$ and sets $K_{gid,f',i} = T^{t_{f',i}/c_{gid}} R_{gid,f',i}$.

After receiving two message M_0, M_1 and the challenge access structure $\mathbb{A}^* = (A, \rho)$, $\mathcal{B}$ makes the challenge ciphertext as follows:

It first randomly selects $\zeta_2, \ldots, \zeta_n$ from $\mathbb{Z}_N$ and defines the vector $\overrightarrow{\zeta} = (\sum_{f \in \mathbb{F}_E} a_f, \zeta_2, \ldots, \zeta_n)^{\top}$. For each $x \in [\ell]$, it also picks a random value $r'_x \in \mathbb{Z}_N$. It then sets:

$C = M_b \cdot \prod_{f \in \mathbb{F}_E \setminus \{f'\}} e(g,g)^{\alpha_f} \cdot T$
$C_0 = g^s Y_2$
$C_x = (g^s Y_2)^{A_x \overrightarrow{\zeta}} (g^s Y_2)^{-r'_x t_{\rho(x)}}$
$C_x = (g^s Y_2)^{r'_x}$
It implicitly means that $\mathcal{B}$ sets $\overrightarrow{v} = s(\sum_{f \in \mathbb{F}_E} \alpha_f)^{-1} \overrightarrow{\delta}$, $\overrightarrow{y} = c\overrightarrow{\delta}$, $r_x = sr'_x$ and $\gamma_x = -cr'_x$.

We note that if $T = e(g,g)^{s\alpha}$, this is a correctly distributed semi-functional ciphertext of M_b. Otherwise, this yields a correctly distributed encryption of a random message in G. Hence, $\mathcal{B}$ will obtain advantage ϵ in breaking Assumption 3 by employing the guess of $\mathcal{A}$.

Proof. If Assumptions 1,2 and 3 hold and the signature system $\sum_{sign}$ is UF-CMA secure, we have demonstrated via the previous 4 lemmas that the real security game is indistinguishable from $\mathbf{Game}_{Final}$, in which the value of b is information-theoretically hidden from the adversary. Thus, the adversary can not gain a non-negligible advantage in breaking our scheme.