Outsourcing Decryption of Multi-Authority ABE Ciphertexts

Keying Li and Hua Ma
(Corresponding author: Keying Li)

Department of mathematics, Xidian University
Taibai South Road, Xian 710071, China
(Email: likeying240818@msn.com)
(Received Oct. 24, 2012; revised and accepted Mar. 20, 2013)

Abstract

The notion of multi-authority attribute based encryption was introduced by Chase in TCC 2007. In this paper, we improve Chase's scheme to allow encryptors to determine how many attributes are required for each ciphertext from related attribute authorities. The proposed scheme can be seen as a multi-trapdoor construction. Furthermore, we apply the LMSSS to outsource the decryption of multi-authority attribute based encryption scheme for large universe. Also, the outsourcing scheme can be realized in the setting of multi-authority key-policy attribute based encryption. Both our schemes can be extended to RCCA secure ones.

Keywords: Interpolation, LMSSS, Multi-Authority ABE, outsourcing

1 Introduction

Shamir [27] put forward Identity based encryption(IBE), which is a variant of encryption allows users to choose any string as their public key. Knowing only the recipients identity (or email address), the sender can send messages. By this way, a separate infrastructure don't need to distribute public keys. The first IBE systems were presented by Boneh, Franklin [8] and Cocks [13], and since then, IBE has been widely researched in the literature [7, 9, 29].

This scenario, however, has some limitations. Each person don't necessarily have a unique string identifier. With another method, we can identify people according their attributes. Based on this idea, Sahai and Waters [26] proposed a fuzzy IBE scheme, which could be used for attribute based encryption. In their scheme, a sender can encrypt a message associated an set of attributes and a trapdoor d. Only a recipient who has at least d of the given attributes can decrypt the message. Though some dishonest users collude to gather d of the given attributes, they can't decrypt the ciphertext.

However, SW's scheme also has one major limitation. As in IBE scheme, in order to obtain a secret key, the user must go to a trusted party and prove his identity. In the same situation, each user must go to the trusted server, to prove that he has a certain set of attributes, e.g. student number, ages, and college department and then receive secret keys corresponding to each of those attributes. This means one trusted server who monitors all attributes, keeps records of SID, ages, and college department must be need.

In fact, we have three different entities responsible to manage their attributes (the department, Archives office, and the University office). So we can entrust each of these to different servers that perhaps honest-but-curious. Sahai and Waters raised the following opinion: constructing an ABE scheme, in which different authorities operate simultaneously to hand out secret keys for a different set of attributes. Melissa Chase [10] resolves this problem in the affirmative. They give an efficient scheme for multi-authority attribute based encryption. In their scheme, the sender specifies for each authority $\{j\}_{1\leq j\leq k}$ a set of attributes monitored by that authority and a trapdoor value d_k. A user who has at least d_k of the given attributes can decrypt the ciphertext. The central authority distribute the subkey s_j to each authority $\{j\}_{1\leq j\leq k}$, then compute $y_{j,u} = F_{s_j}(u)$. Secret key for user u is $D_{CA} = g^{y_0-\sum_{j=0}^{k} y_{j,u}}$. For each authority $\{j\}_{1\leq j\leq k}$, associates their secret key $y_{j,u} = F_{s_j}(u)$ with users attributes using (t, n) trapdoor secret sharing schemes. An encryptor can choose, for each authority, a number d_j and a set of attributes; he can then encrypt a message such that a user can only decrypt if he has at least d_j of the given attributes from each authority $\{j\}_{1\leq j\leq k}$. We can consider their Multi-authority ABE belong to Multi-trapdoor PKE. Their scheme can be extended to a Large Access Control Structure ABE scheme.

Their schemes also have one defect. The public key is $Y_0 = e(g, g)^{y_0}$. As the SK distributed by CA is $D_{CA} = g^{y_0-\sum_{j=0}^{k} y_{j,u}}$, when the date owner encrypt the message, He can't determine which authority's attributes

to be used as j is from 0 to k. That is to say, they can't specify a number D such that a user can decrypt if he has sufficient numbers of the given attributes from at least D authorities.

They also provided several extensions to their basic multi-authority scheme to solve this question. Central authority will now choose a random D_1 degree polynomial P with $P(0) = y_0$. For each authority k he will compute $P(k)$. The secret key from the central authority for user u will be $D_{CA} = g^{P(k)-F_{sk}(u)}$, $k = 1, \cdots, K$. But this method is not the traditional attribute-based encryption.

Our Contributions. We improve their techniques to also allow the encryptor to determine for each ciphertext how many attributes to require from concerned authority. He can also determine which authority's attributes to be used. We still use multi-times linear interpolation to resolve the problem. We remove the secret key generation algorithm from the central authority for user u, which greatly reduced the workload of CA. Our scheme could *reduce the number of ciphertext and user's secret key.*

In the real scenario, the department distribute student's numbers $\{Num_1, Num_2, Num_3\}$, Archives office manages the ages $\{Age_1, Age_2, Age_3\}$, and the University office determine the department {Communication engineering,Computer college, faculty of science}. The encryptor can choose $\{Num_1, Num_2\}$, $\{Age_1, Age_2\}$. In this example, we have 3 authorities, and the ciphertext will include 2 attributes from each. However, we only want to require that a user have satisfactory attributes from 2 out of the 3 authorities to decrypt.

We then construct the outsourcing decryption of Multi-authority CP-ABE scheme based their Multi-Authority scheme for Large Universe. We proposed the Multi-authority KP-ABE outsourcing decryption scheme in the discussion. Our scheme also can be extended to RCCA secure scheme.

Typical Usage Scenarios [16]. A client sends the transformation keys to the Cloud Proxy, who can potentially retrieve large Multi-Authority ABE ciphertexts that the user is interested in. Then, it forward to her small, constant size ElGamal type ciphertexts. The proxy could be the entity in a cloud environment, e.g. the client's mail server, or the ciphertext server. The local computation time for the client are immediate: than Compare with an Multi-Authority ABE ciphertext of [10] (for any policy size), a transformed ciphertext is always smaller and faster to be decrypted. Therefore, faster computations and smaller transmissions could be provided as the power consumption.

Related Work. Sahai and Waters [26] put forward Attribute-based encryption is first place. The first CP-ABE scheme was proposed by Bethencourt, Sahai, and Waters [3]. In their scheme, the security proof is in the generic group model, and it allows the ciphertext policies to be very expressive. Under the standard model, a provably secure CP-ABE scheme is presented by Cheung and Newport [12]. In their scheme, it supports AND-Gates policies which deals with negative attributes explicitly and uses wildcards in the ciphertext policies. In a novel way, Goyal et al. [15] use "universal" access tree to transform a KP-ABE system into a CP-ABE one, and a bounded ciphertext policy ABE was proposed. To support general access formulas, Waters [31] first come up with the secure CP-ABE scheme. Lewko et al. [19] proposed a fully secure CP-ABE scheme by using the dual system encryption techniques [18, 30].

Some other ABE schemes are researched in detail. For the purpose of having constant ciphertext length, Emura et al. [14] presented a novel scheme using AND-Gates policy. Li et al. [20] proposed An expressive decentralizing KP-ABE Scheme with Constant-Size Ciphertext. Nishide et al. [24] come up with a method to solve hiding access structure problem in ABE. In Bobba [5] scheme, same attributes in different sets. Dual-policy attribute-based encryption was put forward by Attrapadung et al. [1], which allows KP and CP act on encrypted data simultaneously. Green, Hohenberger and Waters [16] proposed *outsourcing* decryption ABE CT scheme, which can be traced back to the PRE [4]. Predicate encryption was presented by Katz, Sahai, and Waters [17] and was extended research by Okamoto et al. [25]. Tang and Ji [28] put forward Verifiable Attribute Based Encryption. Multiple authorities were introduced in [10] and [11]. Multiple authorities ABE schemes are of two kinds, those with CA [6, 10], and those without CA [11, 21]. Nali et al. [23] use threshold Attribute-Based Encryption for practical Biometric-based access Control.

Organization. The paper is organized as follows. We give necessary background information in Section 2. We present our Multi-authority ABE scheme and Outsourcing Multi-authority ABE in Section 3 and Section 4. We extent our scheme in Section 5 and give simulate results in Section 6. Finally, give the conclusion in Section 7.

2 Preliminaries

2.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map with the properties:

1) Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.

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

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are both efficiently computable.

2.2 Linear Multi-secret Sharing Schemes

Definition 1. *[22] Let $\boldsymbol{K}$ be a finite field. Let AS_1, AS_2, $\cdots$, AS_m be access structures over $\boldsymbol{P}$, $S^1 \times \cdots \times S^m$ be the secret-domain, $S_1, \cdots, S_n$ be the share-domain and*

$\boldsymbol{R}$ *be the set of random inputs. We may assume that* $S^1 = \cdots = S^m = \boldsymbol{K}$. *A* ***LMSSS*** *realizing the multi-access structure* $AS_1, AS_2, \cdots, AS_m$ *is composed of the distribution function*

$$\prod : \boldsymbol{K}^m \times \boldsymbol{R} \to S_1, \cdots, S_n$$

$\prod(s^1,\cdots,s^m,r)=(\prod_1(s^1,\cdots,s^m,r),\cdots,\prod_n(s^1,\cdots,s^m,r))$ *and the reconstruction function* $Re = Re^i_A:(S_1 \times \cdots \times S_n)|_A \to K|1 \le i \le m$, $A \in AS_i$ *such that the following conditions hold:*

1) $S_1, \cdots, S_n$ *and* $\boldsymbol{R}$ *are finitely dimensional linear spaces over* $\boldsymbol{K}$, *i.e., there exist positive integers* $d_i, 1 \le i \le n$, *and* l *such that* $S_i = K^{d_i}$ *and* $\boldsymbol{R} = \boldsymbol{K}^l$. *Usually* $d = \sum_{i=1}^n d_i$ *is called the size of the LMSSS.*

2) *The reconstruction function is linear, that is, for any set* $A \in AS_i, 1 \le i \le m$, *there exists a set of constants* $\{\alpha^i_{kj} \in \boldsymbol{K}|1 \le k \le n, P_k \in \boldsymbol{A},\ 1 \le j \le d_k\}$ *such that for any* $S^i \in \boldsymbol{K}$ *with* $A \in AS_i$ *and* $r \in \boldsymbol{R}$,

$$\begin{aligned} s^i &= Re^i_A(\prod(s^1, \cdots, s^m, r)|A) \\ &= \sum_{P_k \in A} \sum_{j=1}^{d_k} \alpha^i_{kj} \prod_{kj}(s^1, \cdots, s^m, r) \end{aligned}$$

There is corresponding relation between monotone span programs and linear multi-secret sharing schemes [32].

2.3 Access Structure

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

We take the attributes as the role of the parties. The access structure $\mathbb{A}$ contain the authorized sets of attributes. We only pay attention to the monotone access structures. It is also possible to realize general access structures using the techniques [16] by defining the "not" of an attribute as a separate attribute altogether. Thus, the number of attributes in the system will be doubled.

3 Multi-authority ABE

3.1 Multi-authority ABE

First of all, we discuss the proposed Multi-Authority ABE [10] algorithm. The scheme is sid-secure according to their definition. The public key is $Y_0 = e(g,g)^{y_0}$. As the SK distributed by CA is $D_{CA} = g^{y_0 - \sum_{j=0}^{k} y_{j,u}}$,

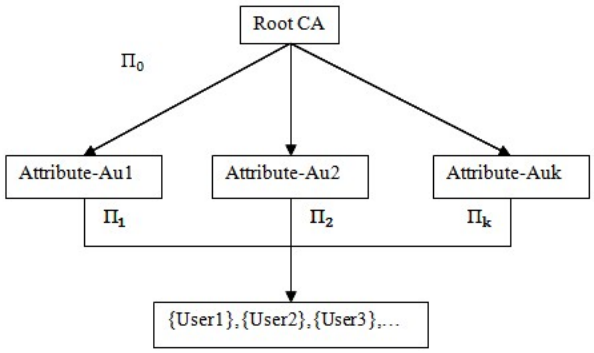

Figure 1: Multi-authority ABE system

when the DO encrypt the message, he cant control the trapdoor d, that is to say, he can't remove the attributes that distributed by one authority but not useful. If there is a mass of users, the CA's workload would increase.

3.2 Our Scheme

We describe techniques to allow the encryptor to determine for each ciphertext how many attributes to require from concerned authority and determine which authority is to choose. When the trapdoor d changes, the user need not get the new secret key from the central authority. This reduces the work of the central authority. Our scheme also can be used to single authority ABE scheme.

A Multi-Authority ABE system is composed of k attribute authorities and one central authority. Each attribute authority is also assigned a trapdoor value $trap_k$. The system uses the following algorithms:

Setup: The trusted party run algorithm, it takes as input the security parameter. Outputs a (PK_j,SK_j) pair for each of the attribute authorities, and also outputs a system public key and master secret key which will be used by the central authority.

Central Key Generation: The central authority runs the randomized algorithm. It takes as input the *MSK*, user's *GID* and a set of attributes of the authorities, outputs secret key for attribute Authorities.

Attribute Key Generation: Attribute authority runs this algorithm. It takes as input the authority's secret key, the authority's value $trap_k$, and a set of attributes in the authority's domain $A^j_{C,j=1,\cdots,k}$, (A_C denote the attribute set of a ciphertext).Output: Authority Public key, secret key for the user.

Encryption: A sender runs the algorithm. He takes input a set of attributes for each authority, a message, and the system public key and outputs the ciphertext.

Decryption: The user takes input the ciphertext, which was encrypted under attribute set A^j_C and decryption

keys for an attribute set A_u. He can get the message m if $A_u^k \bigcap A_C^k \geqslant trap_k$ for all authorities k.

Let κ be the security parameter. Require that the number of authorities K, and the number of attributes n_k monitored by each authority, be upper bounded by a number n which is polynomial in κ.

3.3 Security Model [10]

Consider the game as follows:

Setup.

- The adversary sends a list of attribute sets $A_C = A^0, A^1, \cdots, A^k, A^1, \cdots, A^k$ for each authority, A^0 for all authorities. He must also provide some corrupted authorities and the central authority is ruled out.
- The challenger generates parameters for the system and sends public keys to honest authorities and secret keys for all corrupt authorities.

Secret Key Queries.

- The adversary can make secret key queries to the authorities or to the central authority many times. However, the adversary cannot:
  1) requests enough attributes to decrypt the challenge ciphertext[1];
  2) queries the same authority twice with the same GID.

Challenge.

- The adversary submits two equal length messages M_0 and M_1 with attribute set A_C.
- The challenger chooses a bit b, computes the encryption of M_b for attribute set A_C, and sends it to the adversary.

More Secret Key Queries.

- The adversary can make more secret key queries subject to the requirements described above.

Guess.

- The adversary outputs a guess b' that message $M_{b'}$ has been encrypted.
- The adversary succeed if he can correctly identify the encrypted message, i.e. if $b = b'$.

Definition 3. *A Multi-authority attribute scheme is sid-secure if the adversarys advantage is negligible in the above game. The advantage is* $Pr[b' = b] - 1/2$.

[1] for each GID, there must be at least one honest authority $k$from which the adversary requests fewer than $trap_k$ of the attributes given in A_C^k.

3.4 The Proposed Scheme

We describe techniques to allow the encryptor to determine for each ciphertext how many attributes to require from each authority and which authority. Our scheme is as follows:

System Init. Choose the prime order groups $\mathbb{G}$, $\mathbb{G}_1$, bilinear map $\mathbb{G} \to \mathbb{G}_1$, and generate $g \leftarrow \mathbb{G}$. Choose seeds s_{Au} and s_j for all authorities. Also choose GID[6], PRF, $d-1$ degree polynomial $p_0(x)$ and $t_j - 1$ degree polynomial $p_j(x), j = 1, \cdots, k$. $\{t_{j,i}\}_{j=1,\cdots,k, i=1,\cdots,n} \leftarrow \mathbb{Z}_q$, $\{t_j\}_{j=1,\cdots,k} \leftarrow \mathbb{Z}_q$.

Central Authority. Central Authority compute $y_0 = F_{s_{Au}}(GID)$. Let $p_0(0) = y_0$. System Public Key $Y_0 = e(g,g)^{y_0}$. Central Authority Secret Key: s_{Au}, y_0. **MSK**: y_0. Secret Key for attribute Authorities: Let $y_j = p_{s_{Au}}(j), j = 1, \cdots, k$, SK for Attribute Authority j: y_j.

Attribute Authority j.

Authority Secret Key : y_j, s_j, $t_{j,1}, \cdots, t_{j,n}$, t_j.

Authority Public Key : $T_{j,1}, \cdots, T_{j,n}$
where $T_{j,i} = g^{t_j \cdot t_{j,i}}$; $Y_j = e(g,g)^{y_j}$.

Secret Key for User: Let $y_j = p_{s_j}(0)$. Secret Key: $\{D_{j,i} = g^{\frac{p_{s_j}(i)}{t_{j,i} \cdot t_j}}\}_{i \in A_u, j \in A_k}$.

Encryption for attribute set A_C^j.

1) If the attributes controlled by single one authority j, choose random $s \leftarrow \mathbb{Z}_q$. $E = Y_j^s m$,$\{E_{j,i} = T_{j,i}^s\}_{i \in A_C^j}$.
2) If the attributes controlled by more than one authorities, Choose random $s \leftarrow \mathbb{Z}_q$. $E = Y_0^s m$, $\{E_{j,i} = T_{j,i}^s\}_{i \in A_C^j}$.

Decryption. For each authority j, for the attributes $i \in A_C^j \bigcap A_u$, compute

Situation 1. $e(E_{j,i}, D_{j,i}) = e(g,g)^{p_{s_j}(i)s}$. Interpolate to find $Y_j^s = e(g,g)^{p_{s_j}(0)s} = e(g,g)^{y_j s}$ for each authority j, then $m = E/Y_j^s$.

Situation 2. $e(E_{j,i}, D_{j,i}) = e(g,g)^{p_{s_j}(i)s}$. Interpolate to find $Y_j^s = e(g,g)^{p_{s_j}(0)s} = e(g,g)^{y_j s}$ for each authority j, Interpolate to find $Y_0^s = e(g,g)^{p_0(0)s} = e(g,g)^{y_0 s}$, then $m = E/Y_0^s$.

Theorem 1. *This scheme is sid-secure according to the* ***Definition 3****.*

Proof. Now we will give a brief proof. The Situation 1 is the single authority case, which had been proved security already. So we only need to prove the Situation 2. The user doesn't need to query to the central authority. We can proof simulate the way in [10]. □

3.5 Proof

Suppose that an $\mathcal{A}$ plays the security game described in secure model and succeeds with non-negligible probability ϵ. Then it could build a simulator $\mathcal{B}$ that can attack the **BDH** [7, 26] assumption with the help of the $\mathcal{A}$ in the selective sid-security model with advantage ϵ.

First, assume that even when the $PRFF_{s_k}$ is replaced by a truly random function for each honest authority k, the $\mathcal{A}$ would still succeed with the same advantage.

- Given the tuple $[A = g^a, B = g^b, C = g^c, Z = e(g,g)^{abc}]$ and $e(g,g)^R$ for random $R \leftarrow \mathbb{Z}_q$.
- Receive A_C and list of corrupted authorities *Corr* from adversary.
- *Init*:
  - Authority j system PK : $Y_j = e(A,B)$, implicitly set $y_j = ab$.
  - Honest Authority j attributes PK is: Choose random $\alpha_j, \beta_{j,i}$. PK: $\{T_{j,i} = g^{\alpha_j \cdot \beta_{j,i}}\}_{i \in A_u \bigcap A_C^k}$, $\{T_{j,i} = B^{\alpha_j \cdot \beta_{j,i}}\}_{i \in A_u - A_C^k}$.
  - Corrupt Authority j SK is: Randomly choose $t_{j,i}, t_j \leftarrow \mathbb{Z}_q$, choose PRF key s_j. SK: s_j, $\{t_{j,i}\}$, t_j.
- SK queries: Let $\widehat{j}(u)$ be the first authority j queried such that $|A_u^j \bigcap A_C^j| < d$.
  - SK queries for user u to Honest Attribute Authorities $j \neq \widehat{j}(u)$: For these authorities we will implicitly set $p(0) = F_{s_j}(u) = z_{j,u} b$. Choose a random $z_{j,u}$ and randomly choose a polynomial ρ satisfy $\rho(0) = z_{j,u}$. Set $p(i) = b\rho(i)$. Now for $i \in A_C^k$, $t_j = \alpha_j$, $t_{j,i} = \beta_{j,i}$, so $D_{j,i} = g^{\frac{p(i)}{t_{j,i} * t_j}} = g^{b\rho(i)/\beta_{j,i} \cdot \alpha_j}$; for $i \notin A_C^k$, $t_j = b * \alpha_j$, $t_{j,i} = b * \beta_{j,i}$, so $D_{j,i} = g^{p(i)/t_{j,i} * t_j} = g^{\rho(i)/\beta_{j,i} * \alpha_j}$. SK: $\{D_{j,i} = B^{\rho(i)/\beta_{j,i} * \alpha_j}\}_{i \in (A_u^k \bigcap A_C^k)}$, $\{D_{j,i} = g^{\rho(i)/\beta_{j,i} * \alpha_j}\}_{i \in (A_u^k - A_C^k)}$.
  - SK queries for user u to Honest Attribute Authorities $j = \widehat{j}(u)$: For authority $\widehat{j}$ for user u, $\mathcal{B}$ choose random $r_{j,u}$ and set $p(0) = F_{s_j}(u) = ab + b * r_{j,u}$. Choose $t-1$ random points v_i. For $i \in A_C^j$, we will implicitly set $p(i) = v_i b$. For these attributes, $t_j = \alpha_j$, $t_{j,i} = \beta_{j,i}$, so that means $D_{j,i} = g^{p(i)/t_{j,i} * t_j} = B^{v_i/\beta_{j,i} * \alpha_j}$. Let $p(0) = F_{s_j}(u) = ab + b * r_{j,u}$, and we have now set $p(i) = v_i b$ for $t-1$ other points. Thus p is fully determined, and by interpolation, for any other attribute i, defined $p(i) = \Delta_0(i)(ab + r_{j,u} b) + \sum \Delta_j(i) v_j b$. For these attributes $t_j = b * \alpha_j$, $t_{j,i} = b * \beta_{j,i}$, so $D_{j,i} = g^{p(i)/t_{j,i} * t_j} = g^{\Delta_0(i) a} * g^{\frac{\Delta_0(i) r_{j,u} + \sum \Delta_j(i) v_j}{\beta_{j,i} * \alpha_j}} = A^{\Delta_0(i)} * g^{\frac{\Delta_0(i) r_{j,u} + \sum \Delta_j(i) v_j}{\beta_{j,i} * \alpha_j}}$. SK: $\{D_{j,i} = B^{\frac{v_i}{\beta_{j,i} * \alpha_j}}\}_{i \in A_u^k \bigcap A_C^k}$, $\{D_{j,i} = A^{\Delta_0(i)} * g^{\frac{\Delta_0(i) r_{j,u} + \sum \Delta_j(i) v_j}{\beta_{j,i} * \alpha_j}}\}_{i \in A_u^k - A_C^k}$.

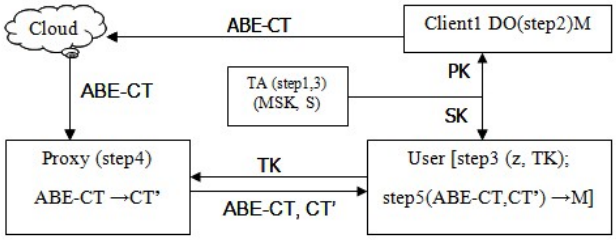

Figure 2: ABE with outsourcing

- **Challenge.** $\mathcal{B}$ receive M_0, M_1 from $\mathcal{A}$, and pick a random $b \in \{0,1\}$. it output the challenge Ciphertext: $Zm_{b'}$, $E = g^c = C$, $\{E_{j,i} = C^{\beta_{j,i} * \alpha_j}\}_{i \in A_C}$.
- **Guess.** for Z: Receive a guess b and If $b = b'$ guess $e(g,g)^{abc}$ otherwise guess $e(g,g)^R$.

An adversary which breaks this encryption scheme with advantage ϵ implies $\mathcal{B}$ can break the **BDH** Assumption with nonegligible advantage $\epsilon/2$. We can conclude that this encryption scheme is sid-secure. [26]

4 Outsourcing Decryption of Multi-Authority CP-ABE CT

4.1 ABE with Outsourcing [16]

Let S represent a set of attributes with an access structure $\mathbb{A}$. For generality, we will define $(I_{enc}; I_{key})$ as the inputs to the encryption and key generation function respectively. In a CP-ABE scheme $(I_{enc}; I_{key}) = (\mathbb{A}; S)$. A CP-ABE scheme with outsourcing functionality consists of five algorithms:

Setup(λ, U), Encrypt$(PK, (M, \rho))$, KeyGen$(MSK; S)$ Transform$(TK; CT)$, Decrypt$(SK; CT)$.

4.2 Outsourcing Decryption of Multi-Authority CP-ABE CT

In our scheme, we also use $D-1$ degree polynomials to split up the secret y_0. Our goal is to realize outsourcing decryption for Multi-Authority CP-ABE.

First method, choose K matrixes and $\rho_j, 1 \leq j \leq k$, using outsourcing algorithm [16], the user get $e(g,g)^{y_j s/z}$, and $e(g,g)^{y_j s}$. Interpolate to find

$$Y_0^s = e(g,g)^{p(0)s} = e(g,g)^{y_0 s}$$

Then $m = E/Y_0^s$.

But it can construct only one M in outsourcing algorithm.

Setup(λ, U). The algorithm takes as input a security parameter and a universe description U. Let $U = \{0,1\}^*$. It then chooses a group $\mathbb{G}$ of prime order p, a generator g and a hash function F that maps

$\{0,1\}^*$ to $\mathbb{G}$. And then, it chooses random exponents $y_j, a \in \mathbb{Z}_p^n$. The authority sets $\{g^{y_j}\}_{0\le j\le k}$ as his secret key. It publishes the public parameters as:

$$PK = g, \{e(g,g)^{y_j}\}_{0\le j\le k}, g^a, F$$

Encrypt$(PK, m, (M,\rho))$. The encryption algorithm takes as input PK and a message m to encrypt. In addition, it takes as input access structure $AS_0, \cdots, AS_k$ and $\mathcal{M}(M,\rho)$[2]. The function ρ associates rows of M to authorities and attributes that control by concerned authorities. The algorithm first chooses a random vector $\overrightarrow{v} = (s_0, s_1, \cdots, s_k, e_{k+1}, \cdots, e_n) \in \mathbb{Z}_p^n$. These values will be used to share the encryption exponent $s_0, s_1, \cdots, s_k$. For $i = 1$ to l, it calculates $\lambda_i = \overrightarrow{v} \cdot M_i$, where M_i is the vector corresponding to the ith row of M. In addition, the algorithm chooses random $\{r_{j,i} \in \mathbb{Z}_p\}_{0\le j\le k, 1\le i\le l}$. The cipher text is published as CT:

$$\begin{aligned} C &= m \cdot e(g,g)^{\sum_{0\le j\le k} y_j s_j}, \\ C'_j &= g^{s_j}, 0 \le j \le k \end{aligned}$$

$(C_{j,i} = g^{a\lambda_i} \cdot F(\rho(i))^{-r_{j,i}}, D_{j,i} = g^{r_{j,i}})_{0\le j\le k, 1\le i\le l}$, along with a description of (M,ρ).

KeyGen(SK, S_j). The key generation algorithm runs **KeyGen(SK,**S_j**)** to obtain $SK'_j = (K' = g^{y_j} g^{at'_j}$, $L' = g^{t'_j}$, $\{K'_x = F(x)^{t'_j}\}_{x\in S_0}$, $\{K'_{j,y} = F(y)^{t'_j}\}_{y\in S_j})$, $1 \le j \le k$. It chooses a random value $z \in \mathbb{Z}_p^*$. It sets the transformation key TK as: PK, $K_j = {K'_j}^{1/z} = g^{y_j/z} g^{at}$, $L_j = L'^{1/z} = g^t$, $\{K_x\}_{x\in S_0} = \{{K'_x}^{1/z}\}_{x\in S_0}$, $\{K_{j,y}\}_{y\in S_j} = \{{K'_{j,y}}^{1/z}\}_{y\in S_j}$, $1 \le j \le k$, and the private key SK as (z, TK).

Transform(TK,CT). The transformation algorithm inputs a transformation key. $TK = (PK, K_j, L_j, \{K_x\}_{x\in S_0}, \{K_{j,y}\}_{y\in S_j, 1\le j\le k})$ for $k+1$ sets $\{S_j\}_{0\le j\le k}$, and a ciphertext $CT = (C, C', (C_{j,i}, D_{j,i}))$, $0 \le j \le k$, $1 \le i \le l$ for access structure (M,ρ). If $\{S_j\}_{0\le j\le k}$ does not satisfy the access structure, it outputs $\perp$. Suppose that $\{S_j\}_{0\le j\le k}$ satisfies the access structure and let $I_j \subset \{1,2,\cdots,l\}_{0\le j\le k}$ be defined as $I_j = \{i : \rho(i) \in S_j\}$. Then, let $\{\omega_{j,i} \in \mathbb{Z}_p\}_{i\in I_j, 0\le j\le k}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of secrets $s_0, s_1, \cdots, s_k$ according to M, then $\sum_{i\in I_j} \omega_{j,i}\lambda_i = s_j$,$0 \le j \le k$. The transformation algorithm computes

$$\begin{aligned} &\prod_{0\le j\le k} \frac{e(C'_j, K_j)}{(e(\prod_{i\in I_j} C_{j,i}^{\omega_{j,i}}, L_j) \cdot \prod_{i\in I_j} e(D_{j,i}^{\omega_{j,i}}, K_{j,\rho(i)}))} \\ &= e(g,g)^{\sum_{0\le j\le k} y_j s_j / z} \end{aligned}$$

It outputs the partially decrypted ciphertext CT' as $(C, e(g,g)^{\sum_{0\le j\le k} y_j s_j/z})$, which can be viewed as the ElGamal ciphertext (mG^{zd}, G^d) where $G = e(g,g)^{1/z} \in G_t$ and $\sum_{0\le j\le k} y_j s_j \in \mathbb{Z}_p$.

Decryptout(SK,CT). The decryption algorithm inputs a private key z, and a ciphertext CT. If the ciphertext is not partially decrypted, then the algorithm first executes **Transform (TK,CT)**. If the output is $\perp$, then this algorithm outputs $\perp$ as well. Otherwise, it takes the ciphertext (T_0, T_1) and computes $T_0/T_1^z = m$.

Notice that for single authority we let

$$C = me(g,g)^{y_j s_j}{}_{j\in 1,\cdots,k}$$

Theorem 2. *Since the scheme of Waters in [7] is a selectively CPA-secure outsourcing scheme, our scheme is also CPA-secure outsourcing scheme.*

5 Extention

5.1 Outsourcing Decryption of Multi-Authority KP-ABE

Setup(λ, U). The setup algorithm takes as input a security parameter and a universe description U, let $U = \{0,1\}^*$. It then chooses a group $\mathbb{G}$ of prime order p, a generator g and a hash function F that maps $\{0,1\}^*$ to $\mathbb{G}$. In addition, it chooses random values $\alpha \in z_p$ and $h \in \mathbb{G}$. The authority sets SK: $y_j, 0 \le j \le k$ as the master secret key.The public key is published as

$$PK = g, g^{y_j}, 0 \le j \le k, F, h.$$

Encrypt$(PK, m, S_{j,0\le j\le k})$. The encryption algorithm takes as input the public parameters PK, a message m to encrypt, and $k+1$ sets $\{S_j\}_{0\le j\le k}$. Then the algorithm chooses random $s_j \in \mathbb{Z}_p$. The cipher text is published as CT:

$$\begin{aligned} C &= m \cdot e(g,g)^{\sum_{0\le j\le k} y_j s_j}, \\ C'_j &= g^{s_j}, 0 \le j \le k, \\ \{C_{j,x} &= F(x)^{s_j})\}_{x\in s_j}. \end{aligned}$$

KeyGen$(MSK, (M,\rho))$. For $0 \le j \le k, 1 \le i \le l$, The key generation algorithm runs **KeyGen**$(y_j, (M,\rho))$ to obtain

$$\begin{aligned} SK' &= (PK, D'_{j,i}, R'_{j,i}) \\ D'_{j,i} &= h^{\lambda_t} \cdot F(\rho(i))^{-r'_{j,i}}, \\ R'_{j,i} &= g^{r'_{j,i}}. \end{aligned}$$

It randomly chooses a value $z \in \mathbb{Z}_p^*$. Let $r'_{j,i}/z$ as $r_{j,i}$, it sets the transformation key TK as: $(D_{j,i} = {D'_{j,i}}^{1/z} = h^{\lambda_t/z} \cdot F(\rho(i))^{-r_{j,i}}$, ${R'_{j,i}}^{1/z} = g^{r_{j,i}})$, $0 \le j \le k, 1 \le i \le l$ and the private key SK as z.

[2] $\mathcal{M}(M,\rho)$ is the Monotone span programs of the monotone Boolean functions $f_0, \cdots, f_k$

Transform(TK,CT). The transformation algorithm takes as input a transformation key $TK = (PK, (D_{j,i}, R_{j,i})_{1\leq j\leq k})$ for access structure (M, ρ) and a ciphertext

$$CT = (C, C', \{C_{j,x}\}_{x\in S_j})$$

for $k+1$ sets $\{S_j\}_{0\leq j\leq k}$. If $\{S_j\}_{0\leq j\leq k}$ does not satisfy the access structure, it outputs $\perp$. Suppose that $\{S_j\}_{0\leq j\leq k}$ satisfies the access structure and let $I_j \subset \{1, 2, \cdots, l\}_{0\leq j\leq k}$ be defined as $I_j = \{i : \rho(i) \in S_j\}$. Then, let $\{\omega_{j,i} \in \mathbb{Z}_p\}_{i\in I_j, 0\leq j\leq k}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secrets $s_0, s_1, \cdots, s_k$ according to M, then $\sum_{i\in I_j} \omega_{j,i}\lambda_i = y_j$, $0 \leq j \leq k$. The transformation algorithm computes

$$\prod_{0\leq j\leq k} \frac{e(C'_j, \prod_{i\in I_j} D_{j,i}^{\omega_{j,i}})}{(\prod_{i\in I_j} e(C_{j,\rho(i)}^{\omega_{j,i}}, R_{j,i})} = e(g,g)^{\sum_{0\leq j\leq k} y_j s_j/z}$$

It outputs the partially decrypted ciphertext CT' as $(C, e(g,g)^{\sum_{0\leq j\leq k} y_j s_j/z})$.

Decrypt(SK;CT). The decryption algorithm takes as input a private key $SK = z$ and a ciphertext CT. If the ciphertext is not partially decrypted, then the algorithm first executes **Transform**(TK, CT). If the output is $\perp$, then this algorithm outputs $\perp$ as well. Otherwise, it takes the ciphertext (T_0, T_j) and computes $T_0/T_1^z = m$.

5.2 RCCA-Secure Multi-Authority ABE with Outsourcing

We can also change our outsourcing CP-ABE or KP-ABE scheme to RCCA-secure scheme (or secure against replay able chosen cipher text attacks) if all polynomial time adversaries have at most a negligible advantage in the RCCA game [16].

6 Simulation in Practice

Experimental setup. We have known that both decryption time and ciphertext size in the CP-ABE scheme have relationship with the complexity of the ciphertext's policy. Considering that, in our experiments, we generated a collection of 100 distinct ciphertext policies of the form $(A_1$ AND A_2 AND $\cdots$ AND $A_N)$, where each A_i is an attribute, for values of N increasing from 1 to 100. Suppose that these attributes are controlled by four Authorities. In each authority, we constructed a corresponding decryption key that contained the $N/4$ attributes necessary for decryption.

We also encapsulated a random 128-bit symmetric key under each of 25 different policies, then decrypted the resulting ABE ciphertext using the normal (non-outsourced) Decrypt algorithm. We repeated each of our experiments vast times on our PC device to smooth any experimental variability and averaged to obtain our decryption timings. Figure 3 shows the size of the resulting ciphertexts, and the measured decryption times on our PC test platforms.

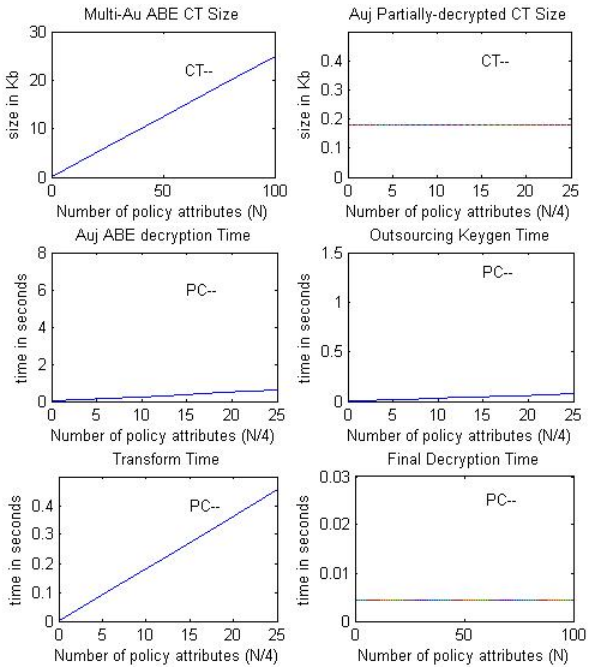

Figure 3: PC results for our CP-ABE scheme with outsourcing

Next, we generate a TK from the appropriate N-attribute Multi-Authority ABE decryption key and applying the Transform algorithm to the Multi-Authority ABE ciphertext using this key. As the attributes controlled by four Authorities, so it can use Parallel Computing to generate TK, which accelerates the computation speed. Finally we decrypted the resulting transformed ciphertext. Figure 3 shows the time required for each of those operations.

7 Conclusions

Green, Hohenberger and Waters [16] brought up outsourcing the decryption of ABE ciphertexts in cloud computing environment. Chase [10] put forward Multi-Authority ABE and also show how to apply the techniques to achieve a multiauthority version of the large universe fine grained access control ABE. We improved his scheme to allow the encryptor to determine how many attributes to require for each ciphertext from concerned authorities, and reduced the number of ciphertext and user's secret keys. The CA need not distribute secret key for user. Only improved his scheme, could We apply the LMSSS to construct the outsourcing decryption of Multi-authority ABE scheme, including CP-ABE and KP-ABE, which can also be extended to RCCA secure scheme. Our scheme can be used to the cloud computing environment. The security can be

guaranteed as in [16].

Acknowledgements

We would like to thank Xiaofeng Chen for the suggestions to improve this paper. Also, we are grateful to the anonymous referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (No. 61272455).

References

[1] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *ACNS '09*, LNCS 5536, pp. 168–185, Springer-Verlag, 2009.

[2] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

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

[4] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Eurocrypt '98*, pp. 127–144, 1998.

[5] R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in *ESORICS '09*, LNCS 5789, pp. 587–604, Springer-Verlag, 2009.

[6] V. Božović, D. Socek, R. Steinwan, and VI. Villinyi, "Multi-authority attribute based encryption with honest-but-curious central authority," *International Journal of Computer Mathematics*, vol. 89, pp. 268–283, 2012.

[7] D. Boneh and X. Boyen, "Efficient selective-id secure identity based encryption without random oracles," in *Eurocrypt '04*, LNCS 3027, pp. 54–73, Springer-Verlag, 2004.

[8] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Crypto '01*, LNCS 2139, pp. 213–229, Springer-Verlag, 2001.

[9] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in *Eurocrypt '03*, LNCS 2656, pp. 255–271, Springer-Verlag, 2003.

[10] M. Chase, "Multi-authority attribute based encryption," in *Proceedings of the Theory of Cryptography Conference(TCC)*, pp. 515–534, 2007.

[11] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pp. 121–130, 2009.

[12] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 456–465, 2009.

[13] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Proceedings of Cryptography and Coding*, LNCS 2260, pp. 360–363, Springer-Verlag, 2001.

[14] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *ISPEC '09*, LNCS 5451, pp. 13–23, Springer-Verlag, 2009.

[15] V. Goyal, O. Pandey A. Jain, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *ICALP, Part II*, pp. 579–591, 2008.

[16] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," *USENIX Security*, pp. 523–538, 2011.

[17] J. Katz, A. Sahai, , and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Eurocrypt '08*, LNCS 4965, pp. 146–162, Springer-Verlag, 2008.

[18] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure hibe with short ciphertexts," in *TCC '10*, LNCS 5978, pp. 455–479, Springer-Verlag, 2010.

[19] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption,".

[20] Q.Y. Li, H. Xiong, F.L. Zhang, and Sh.K. Zeng, "An expressive decentralizing kp-abe scheme with constant-size ciphertext," *International Journal of Network Security*, vol. 15, no. 3, pp. 131–140, 2013.

[21] H. Lin, Z.F. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proceedings of the Cryptology in India*, pp. 426–436, 2008.

[22] M.L. Liu, L.L. Xiao, and Zh. Zhang, "Linear multi-secret sharing schemes based on multi-party computation," *Finite Fields and Their Applications*, vol. 12, pp. 704–713, 2006.

[23] D. Nali, C. Adams, and A. Miri, "Using threshold attribute-based encryption for practical biometric-based access control," *International Journal of Network Security*, vol. 1, no. 3, pp. 173–182, 2005.

[24] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures,".

[25] T. Okamoto and K. Takashima, "Hierarchical predicate encryption for inner-products," in *Asiacrypt '09*, LNCS 5912, pp. 214–231, Springer-Verlag, 2009.

[26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Eurocrypt '05*, LNCS 3494, pp. 457–473, Springer-Verlag, 2005.

[27] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Crypto '84*, LNCS 196, pp. 47–53, Springer-Verlag, 1984.

[28] Q. Tang and D.Y. Ji, "Veriable attribute based encryption," *International Journal of Network Security*, vol. 10, no. 2, pp. 114–120, 2010.

[29] B. Waters, "Efficent identity based encryption without random oracles," in *Eurocrypt '05*, LNCS 3494, pp. 114–127, Springer-Verlag, 2005.
[30] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *Crypto '09*, LNCS 5677, pp. 619–636, Springer-Verlag, 2009.
[31] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient,and provably secure realization," in *Public Key Cryptography-PKC '11*, LNCS 6571, pp. 53–70, Springer-Verlag, 2011.
[32] L. L. Xiao and M.L. Liu, "Linear multi-secret sharing schemes," *Science in China Ser. F Informatiom Sciences*, vol. 48, pp. 125–136, 2005.

Keying Li is now pursuing Master and doctor of Faculty of science, Xidian University, Xi'an, China. His research interests cover the attributes based encryption, cloud computing, lossy trapdoor function, lossy encryption, e-cash payment.

Hua Ma, professor of Faculty of science, Xidian University, Xi'an, China. Her research directions including The Theory and technology in e-commerce security, Design and analysis of fast public key cryptography, Theory and technology of the network security.