# New Constructions of Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Computing

**Leyou Zhang[1], Yupu Hu[2]**
[1] Department of Mathematics, Xidian University, Xi'an, 710071, China
[e-mail: xidianzhangly@126.com, lyzhang@mail.xidian.edu.cn ]
[2] Key Laboratory of Computer Networks and Information Security, Xidian University, Xi'an, 710071, China;
*Corresponding author: Leyou Zhang

*Received December 23, 2012; revised April 21, 2013; accepted May 6, 2013; published May 31, 2013*

## Abstract

Cloud computing has emerged as perhaps the hottest development in information technology at present. This new computing technology requires that the users ensure that their infrastructure is safety and that their data and applications are protected. In addition, the customer must ensure that the provider has taken the proper security measures to protect their information. In order to achieve fine-grained and flexible access control for cloud computing, a new construction of hierarchical attribute-based encryption(HABE) with Ciphertext-Policy is proposed in this paper. The proposed scheme inherits flexibility and delegation of hierarchical identity-based cryptography, and achieves scalability due to the hierarchical structure. The new scheme has constant size ciphertexts since it consists of two group elements. In addition, the security of the new construction is achieved in the standard model which avoids the potential defects in the existing works. Under the decision bilinear Diffie-Hellman exponent assumption, the proposed scheme is provable security against Chosen-plaintext Attack(CPA). Furthermore, we also show the proposed scheme can be transferred to a CCA(Chosen-ciphertext Attack) secure scheme.

***Keywords:*** Cloud computing; Hierarchical Attribute-Based Encryption; Fine-Grained Access Control; Standard model

This work is supported in part by the Nature Science Foundation of China under grant (61100231, 61100165, 60970119), the Natural Science Basic Research Plan in Shaanxi Province of China (2012JQ8044, 2011JM8003), the Foundation of Education Department of Shanxi Province(2013JK1096) and the Fundamental Research Funds for the Central Universities of China.

**http://dx.doi.org/10.3837/tiis.2013.05.023**

## 1. Introduction

Cloud computing [1] is a general term for anything that involves delivering hosted services over the Internet. It provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. Now it has attracted extensive attention from both academia and industry.

In a cloud computing plateform, users no longer need to buy any hardware/software infrastructures or hire IT professionals to maintain IT systems, which make them to save a lot cost on devices and human resource. On the other hand, with the development of this technology, the cost of computation, application hosting, content storage and delivery are reduced significantly which offers at a relatively low price in a pay-as-you-use style. Although the great benefits brought by cloud computing paradigm are exciting, allowing a cloud service provider, operated for making a profit, to take care of confidential corporate data, raises underlying security and privacy issues. So security problems in cloud computing become serious obstacles which prevent its wide applications.

For security requirements, we only consider the followings in this paper [2].

gData confidentiality: The cloud users will first make sure that their data are confidential to restricted parties, including the cloud provider and their potential competitors etc.

gFine-grained access control mechanism: When enterprise users outsource confidential data for sharing on cloud servers, fine-grained, flexible and scalable access control is also strongly desired.

Attribute-based encryption can support the above security requirements. Attribute Based Encryption (ABE) was first proposed as a fuzzy version of identity-based encryption in [3], where an identity is viewed as a set of descriptive attributes. The private key for an identity *ID* can decrypt the message encrypted by the identity *ID'* if and only if *ID* and *ID'* are closer to each other than a pre-defined threshold *k* in terms of set overlap distance. There are two main variants of ABE proposed so far, namely Key Policy Attribute-based Encryption (KP-ABE) and Ciphertext Policy Attribute-based Encryption (CP-ABE) [4]. In CP-ABE, the user secret key is associated with a set of attributes and ciphertexts are embedded with an access structure. A user is able to decrypt the ciphertext only if the attributes associated with his secret keys satisfy the access structure of the ciphertext. KP-ABE is defined in the reverse way than CP-ABE. User secret keys in KP-ABE are embedded with an access structure and ciphertexts are associated with a set of attributes. Successful decryption of the ciphertext requires a match between the user's access structure and the ciphertext attribute set. Two forms are complementary to each other. To achieve flexible and fine-grained access control in cloud computing, many schemes [5-8] are proposed recently.

However, they may not work well in the cloud computing. At first, users may access data anytime and anywhere using any device. Second, the enterprise user may come from a large-scale enterprise. Hence a single Trusted Center (TA) would become a bottleneck for the following reasons:

g the private key generation is computationally expensive;

g the TA must verify proofs of identities singly;

g the TA must establish secure channels to transmit private keys.

Hierarchical identity-based encryption(HIBE) [9-12] can support the full delegation which reduces the workload on TA. It also can be used in the cloud computing. Recently, some schemes [13,14] have been proposed based on this delegation technique. These schemes achieve fine-grained access control in cloud computing. However, these schemes are based on the random oracles. It has been shown that when the random oracles are instantiated with concrete hash functions, the resulting scheme may not be secure [15]. So the schemes based on the random oracles have a potential security defect. In addition, the ciphertexts size relies on the depth of the user's attributes. In [16], authors also proposed a scheme based on HIBE. But it is lack of a concrete construction and security proof.

*Our Contributions.* A natural extension of the efforts is to provide a more efficient and secure scheme. We propose a new scheme in this paper. Our scheme is based on hierarchical identity-based encryption and CP-ABE. It has the advantages of scalability, flexibility and fine-grained access control over existing schemes. In addition, its security is achieved in the standard model. The ciphertext achieves O(1)-size which is more efficient than the existing schemes. Finally, we transfer it to a CCA secure scheme.

*Organization.* In Section 2, we will review some related definitions and security basis. Section 3 will show our works and efficiency analysis. We describe the security analysis in section 4. Finally, in Section 5, we present the conclusions of this paper.

## 2. Preliminaries

### 2.1 Bilinear Groups

$G$ and $G_1$ are cyclic groups of order *p*. $g$ is a random generator of $G$. A bilinear map *e* is a map $e: G \times G \to G_1$ with the following properties:

(i) Bilinearity: for all $u, v \in G$, $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$;

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

(iii) Computability: there is an efficient algorithm to compute $e(u, v)$ for all $u, v \in G$.

### 2.2 Hierarchical Identity-based Encryption

Hierarchical identity-based encryption (HIBE) [9-12] is a generalization of IBE. It allows a root PKG to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs. In a HIBE scheme, a root PKG needs only to generate private keys for domain level PKGs, who in turn generate private keys for users in their domains in the next level. Authentication and private key transmission can be done locally. To encrypt a message to Bob, Alice needs only to obtain the public parameters of Bob's root PKG (and Bob's identifying information); there are no lower level parameters. Another advantage of HIBE schemes is damage control: disclosure of a domain PKGs secret does not compromise the secrets of higher level PKGs. It is especially useful in large companies or e-government systems where there are hierarchical administrative issues needed to be taken care. HIBE provides one of the most direct and practical solutions to the key exposure problem for PKI applications that occur in daily life.

HIBE scheme consists of four probabilistic algorithms: Setup, Key Generation, Encryption and Decryption. Note that, for a HIBE of height *l* (hence forth denoted as *l*-HIBE) any identity *ID* is a tuple $(v_1, \ldots, v_k)$ where $1 \leq k \leq l$. The algorithms are specified as follows:

***Setup*:** The algorithm takes as input a security parameter and outputs the system

parameters (*PK*, *sk*), where *PK* is the public parameter of the *PKG* and *sk* denotes the master key of the PKG. It also gives a description of the message space, the ciphertext space and the identity space.

***Key Generation*:** The algorithm takes as input an identity $ID=(v_1, v_2, \cdots, v_k)$ at depth *k*, the public parameters of *PKG* and the private key $d_{ID|k-1}$ corresponding to the parent identity $(v_1, v_2, \cdots, v_{k-1})$ at depth *k*-1. It returns a private key $d_{ID}$ for *ID*.

If *k* = 1, then the private key is generated by the root PKG. It is not difficult to see that any entity which possesses a private key for a prefix of *ID* can generate a private key for *ID*.

***Encryption*:** This algorithm takes as input the identity *ID*, the public parameters *PK* of *PKG* and a message. Then it produces a ciphertext under *ID* and *PK*.

***Decryption*:** It takes as input the ciphertext and the private keys corresponding identity *ID* . It returns either the message or $\perp$ if the ciphertext is not valid.

### 2.3 CP-ABE

CP-ABE schemes consist of four algorithms [4]:

***Setup(k)***: The setup algorithm takes as input a security parameter *k* and outputs the public parameters *PK* and a master key *mk*.

***Keygen*($\omega$,*mk*)**: The algorithm takes as input the master key *mk* and a set of attributes $\omega$. The algorithm outputs a secret key $sk_\omega$ associated with $\omega$.

***Encrypt*(*m*, $\tau$ , *pk*)**: The encryption algorithm takes as input the public key *PK*, a message *m*, and an access tree $\tau$ representing an access structure. The algorithm will return the ciphertext $C_\tau$ such that only users who have the secret key generated from the attributes that satisfy the access tree will be able to decrypt the message.

***Decrypt*($C_\tau$ , $sk_\omega$)**: The decryption algorithm takes as input a secret key $sk_\omega$, a ciphertext $C_\tau$ associated with $\tau$, and it outputs a message *m* or an error symbol $\perp$.

### 2.4 Hierarchical CP-ABE-based Encryption (HCABE)

Following [13, 14, 16], a HCABE consists of six parties: Root Master, Domain Authorities, Data Owners, Data Consumers(enterprise users), a Cloud Service Provider, and a Trusted Center(TA). The Cloud Service Provider manages a Cloud to provide data storage service and other parties are organized in a hierarchical style as Fig. 1.[13,14] shows. The Root Master calls TA for generation and distribution of system parameters and domain keys. Each Domain Authority is responsible for managing the Domain Authorities at the next level or the Data Owners/Consumers in his domain. Data Owners encrypt their data files and store them in the Cloud for sharing. To access the shared data files, Data Consumers download data files of their interest from Cloud and then decrypt them. It consists of four algorithms as follows:

***Setup***: On input a security parameter, TA returns the system parameters together with the master key. The system parameters are publicly known while the master key is known only to the TA.

***Key Generation-Delegation***($DA_i$): When a new user wants to join the system, $DA_i$ will verify that it is a Domain Authority or a Data Owner. If it is a Domain Authority, then it is denoted $DA_{i+1}$. $DA_i$ will generate the master keys for $DA_{i+1}$. Otherwise, it is a user, $DA_i$ calls TA to generate the secrete keys for it.

***Encryption***: The encryption algorithm takes as input the public key *PK*, a message *m*, and

an access tree $\tau$ representing an access structure. The algorithm will return the ciphertext $C_\tau$ such that only users who have the secret key generated from the attributes that satisfy the access tree will be able to decrypt the message.

***Decryption***: The decryption algorithm takes as input a secret key $sk_\omega$, a ciphertext $C_\tau$ associated with $\tau$, and it outputs a message *m* or an error symbol $\perp$.

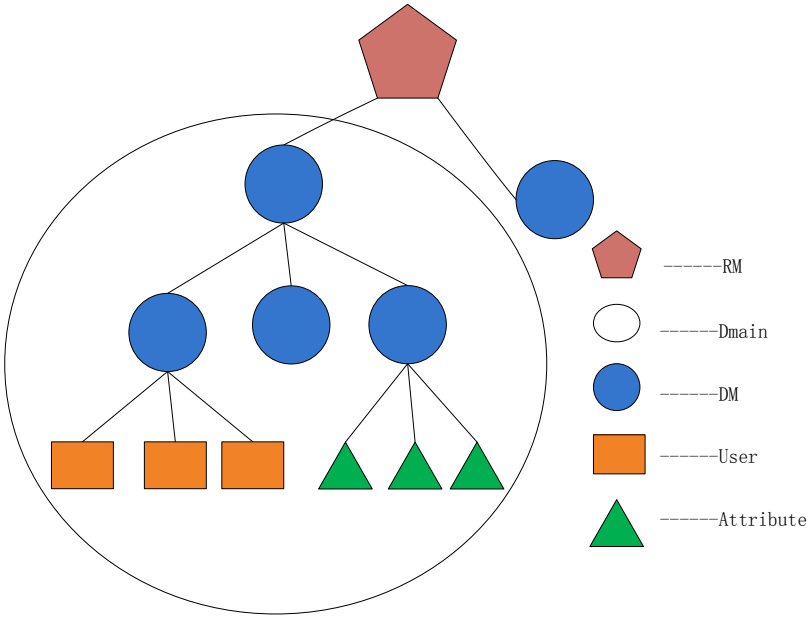


**Fig. 1.** HCABE

In this paper, we consider a hybrid encryption model. With this model, instead of a ciphertext, a session key is produced and encrypted, which is required to be indistinguishable from random. The session key will be used by a symmetric algorithm to encrypt message. We also call it key-encapsulation mechanism(KEM).

## 2.5. Security Model for HCABE

The security model is given as follows.

***Init*:** The adversary chooses the challenge access policy $W^*$ and gives it to challenger.

***Setup*:** The challenger runs the Setup algorithm and gives adversary the PK.

***Phase* 1(**A query any user key for his choice) **:** When A queries user for user attribute secret key on attribute *a*, which is administered by $DM_i$, the challenger first runs the algorithm to generate keys for $DM_i$, and then runs algorithm to generate an attribute secret key. These queries can be asked adaptively.

***Challenge*:** The challenger runs Encrypt algorithm to obtain $((C_0^*, C_1^*), K)$ under access policy $W^*$. Next, the challenger picks a random $b \in \{0, 1\}$. It sets $K_b = K$ and a random element to $K_{1-b}$. It then gives $((C_0^*, C_1^*), K_0, K_1)$ to the adversary.

***Phase* 2**: It runs as Phase 1.

***Guess***: The adversary gives its guess $b' \in \{0, 1\}$ and wins the game if $b'=b$.

The above indistinguishability game allows no decryption oracle query, then the HCABE scheme achieves only CPA security. If the decryption oracle query is allowed in phase1 and phase 2, we call it CCA security model.

### 2.6 Hardness Assumption

**Definition 1***:* Bilinear Diffie- Hellman Exponent(BDHE) problem is defined as follows: Given $g, h, g^{\alpha}, \ldots, g^{\alpha^n}, g^{\alpha^{n+2}}, \ldots, g^{\alpha^{2n}}$, output $e(g,h)^{\alpha^{l+1}}$ (If $h = g^c$, it will output $e(g,g)^{\alpha^{l+1}c}$). An algorithm $C$ has advantage $\varepsilon$ in solving BDHI in $G$ if

$$\Pr\{C(g, h, g^{\alpha}, \cdots, g^{\alpha^n}, g^{\alpha^{n+2}}, \cdots, g^{\alpha^{2n}}) = e(g,h)^{\alpha^{l+1}}\} \geq \varepsilon .$$

Let $y_i = g^{\alpha^i}$. The decision version of it is given in the following.

Given a tuple $(g, h, y_1, \ldots, y_n, y_{n+2}, \ldots, y_{2n}, T)$, the task of an algorithm $\Lambda$ is to decide whether $T = e(g,h)^{\alpha^{n+1}}$ or $T$ is random. $\Lambda$ outputs $b \in \{0, 1\}$ has advantage $\varepsilon$ in solving the decision BDHE problem in $G$ if

$$\Pr\{\Lambda(g, h, y_1, \cdots, y_n, y_{n+2}, \cdots, y_{2n}, e(g,h)^{\alpha^{l+1}}) = 0\} - \Pr\{\Lambda(g, h, y_1, \cdots, y_n, y_{n+2}, \cdots, y_{2n}, T) = 0\} \geq \varepsilon$$

**Definition 2**: The $(t, \varepsilon)$-decisional BDHE assumption holds if no $t$-time adversary has a non-negligible advantage $\varepsilon$ in solving the above game.

## 3. Our Works

### 3.1 New Scheme

Let $l$ denote the maximum size of the attribute set. Without loss of generality, we also map the attribute ($A_i$) to the following value:

$$A_i \rightarrow i, \ 1 \leq i \leq l.$$

Based on [13, 14, 16, 17], we give our construction as follows.

***Setup*** Let $G$ be the cyclic group with prime order $p$. TA first picks randomly a generator $g$ of group $G$ and two elements $\gamma, \alpha \in Z_p$. Then it computes $g_i = g^{\alpha^i}$ for $i = 1, \cdots, n, n+2, \cdots, 2n$, where $n = l$ and sets $v = g^{\gamma}$.

The public parameters are

$$PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v).$$

The master keys are $(\gamma, \alpha)$.

***Key Generation***($D_i$, $D_{i+1}$, $u$) Let $(D_1, \cdots, D_i)$ denote the $i$ level domain authorities. When a user wants to join this system, it is denoted as $u$ or $D_{i+1}$. The private keys are generated as follows: The key generalization algorithm chooses randomly $\gamma_1, \gamma_2, \ldots, \gamma_i \in Z_p$ subject to the constraint that $\gamma_1 + \gamma_2 + \ldots + \gamma_i = \gamma$. The master key for each level $D_i$ is $g^{\gamma_i}$.

If the user is a domain authority which is denoted as $D_{i+1}$, then delegation algorithm is run as follows:

Pick randomly $\lambda_1, \lambda_2, \ldots, \lambda_{i+1} \in Z_p$ subject to the constraint that $\lambda_1 + \lambda_2 + \ldots + \lambda_{i+1} = 0$. And take as input $g^{\gamma_j}$ with $1 \leq j \leq i$ and output the master key for $D_1, \ldots, D_i, D_{i+1}$ as follows:

$$g^{\gamma_1} g^{\lambda_1}, \ldots, g^{\gamma_i} g^{\lambda_i}, K_{i+1} g^{\lambda_{i+1}}$$

where $K_{i+1}$ is the identity element in $G$.

If it is a data owner user with the attribute list $\omega = \{\omega(j) \, j \in [1, k]\}$ where $1 \leq \omega(j) \leq l$, $D_i$

calls TA which runs the key generation algorithm as follows:

Pick $r_1, r_2, \mathrm{L}\ , r_k \in Z_p$ randomly and compute

$$d_{0j} = (g^{\gamma_i})^r\ ,\ \ d_j = (g^{\gamma})^{\alpha^{\omega(j)}+r_j} = g_{\omega(j)}^{\gamma} g^{r_j\gamma}\ ,$$

where $r = \sum_{i=1}^{k} r_i\ , 1 \le j \le k,\ \ k \le l$.

***Encryption*** To protect the data stored on the Cloud, the data owner encrypts the data files first and stores the encrypted data files on the Cloud. Each file is encrypted with a symmetric encryption key *K*, which is in turn encrypted with HCABE. The algorithm is run as follows:

Suppose AND-gate access policy is *W* with *k* attributes.

Pick a random $t \in Z_p$ and set $K = e(g_n, g_1)^{kt}$. Then construct the ciphertexts as

$$C = (C_0, C_1) = (g^t, (v \prod_{\omega(j)\in W} g_{n+1-\omega(j)})^t).$$

***Decryption*** Data consumer user verifies the attribute $\omega$ satisfies *W*. If it does not hold, then Data consumer user outputs $\perp$. Otherwise, Data consumer user runs the following algorithm:

For $1 \le j \le k$, compute

$$\begin{aligned} K_1 &= e(g_{\omega(j)}, C_1) = e(g^{\alpha^{\omega(j)}}, (v \prod_{\omega(j)\in W} g_{n+1-w(j)})^t) \\ &= e(g^{\alpha^{\omega(j)}}, (g^{\gamma} \prod_{\omega(j)\in W} g_{n+1-w(j)})^t) \\ &= e(g^{\alpha^{\omega(j)}}, (g^{\gamma} \prod_{j\in W} g^{\alpha^{n+1-j}})^t) \\ &= e(g^{\alpha^{\omega(j)}}, g^{t(\gamma+\sum_{j\in W}\alpha^{(n+1-j)})}) = e(g,g)^{t\gamma\alpha^{\omega(j)}+t\sum_{j\in W}\alpha^{n+1-j+\omega(j)}} \end{aligned}$$

$$\begin{aligned} K_2 &= e(C_0, d_j \prod_{j\in W, j\neq\omega(j)} g_{n+1-j+\omega(j)}) \\ &= e(g^t, (g^{\gamma_i})^{a^{w(j)}+r_j} \prod_{j\in W, j\neq\omega(j)} g^{a^{(n+1-j+w(j))}}) \\ &= e(g^t, g^{\gamma\alpha^{w(j)}+\gamma r_j} g^{\sum_{j\in W, j\neq w(j)}\alpha^{n+1-j+w(j)}}) \\ &= e(g,g)^{t(\gamma\alpha^{w(j)}+\gamma r_j)+t\sum_{j\in W, j\neq w(j)}\alpha^{n+1-j+w(j)}} \end{aligned}.$$

Then it computes $K_3$ as follows.

$$K_3 = \prod_{j=1}^{k} \frac{K_1}{K_2} = \prod_{j=1}^{k} \frac{e(g,g)^{t\gamma\alpha^{\omega(j)}+t\sum_{j\in W}\alpha^{n+1-j+\omega(j)}}}{e(g,g)^{t(\gamma\alpha^{w(j)}+\gamma r_j)+t\sum_{j\in W, j\neq w(j)}\alpha^{n+1-j+w(j)}}}$$

$$= \prod_{j=1}^{k} e(g,g)^{-t\gamma r_j + t\alpha^{n+1}} = e(g,g)^{kt\alpha^{n+1} - t\sum_{j=1}^{k}\gamma r_j}$$
$$= e(g,g)^{kta^{n+1} - t\gamma r},$$

where $r = \sum_{i=1}^{k} r_i$ . Then user can obtain

$$K = K_3 e(\prod_{j=1}^{k} d_{0j}, C_0)$$
$$= e(g,g)^{tk\alpha^{n+1} - t\gamma r} e(g^{\gamma r}, g^t)$$
$$= e(g,g)^{tk\alpha^{n+1}} = e(g^{\alpha^n}, g^{\alpha})^{tk} = e(g_n, g_1)^{tk}.$$

### 3.2. Efficiency

The ciphertext of the new work achieves O(1) −size since it consists of two group elements. So its storage achieves O(1) instead of O(*NT*) in the [13,14]. It reduces the storage cost of Data owner in the cloud. Furthermore, our scheme is based on HIBE, which can help us easily achieve the fine-grained access control. Our new scheme is constructed in the standard model. It avoids the potential security defects in [13, 14]. The private key in our scheme achieves O(*k*)-size(consists of 2*k* elements) where the elements of the private key for each attribute need 2 exponentiations(Exp) computations. The encryption algorithm needs 2 exponentiations(Exp) computations and *k*/2 multiplications(Multi), which is more efficient than existing works. The pairings in decryption needs to compute about 2*k* times.

Table 1 gives the comparisons of efficiency with other schemes. Table 2 gives the comparisons of computation complexity in different schemes.

Note: In Table 1, *l* and *T* denote the hierarchy depth and the minmum depth of all *DMs* administering respectively. *k* denotes the number of user(including domain authority). *SM* and *RO* are security model and random oracles respectively. PK and pk denote the public keys and private keys respectively. *N* is the number of conjunctive clauses in an access structure.

In Table 2, we use *k* and $t_i$ denotes the size of identity of *DM* and the number of user(including domain authority) attributes respectively.

**Table 1.** Comparison of the Efficiency in Different Schemes

| Schemes | SM | pk size | Ciphertext size |
|---|---|---|---|
| [13] | RO | $O(k)$ | $\sum_{i=1}^{N} it_i + 3$ |
| [14] | RO | $O(k)$ | $\sum_{i=1}^{N} it_i + 3$ |
| Ours | Standard | $O(k)$ | 2 |

**Table 2.** Comparison of computation complexity in differentschemes

| Schemes | Encryption(Exp) | Encryption(Multi) | Decryption(Exp) | Pairing |
|---|---|---|---|---|
| [13] | $O(NT)$ | $O(NT)$ | $O(k)$ | $O(t_i)$ |
| [14] | $O(NT)$ | $O(NT)$ | $O(k)$ | $O(t_i)$ |
| Ours | $O(1)$ | $O(k)$ | $O(k)$ | $O(k)$ |

From the table 1 and 2, one can obtain we have a much more efficient encryption algorithm than the existing works.

### 3.3. User Revocation

When there is a user to be revoked, the system must make sure the revoked user can not access the associated data files any more. It can be achieved by re-encrypting all the associated data files that used to be accessed by the revoked user [2].

### 3.4 CCA-Secure Scheme

Our initial scheme only achieves CPA security(Shown in next section ). The authors in the previous scheme claimed their scheme could be transformed to the CCA scheme. But the transformation methods in their paper were the general methods, such as one-time signature and MAC. These folklore construction methods lead to schemes that are somewhat inefficient in the real life.

In this paper, a direct technique is introduced to construct CCA secure scheme. It works as follows.

Let $l$ denote the maximum size of the attribute set. Without loss of generality, we also map the attribute ($A_i$) to the following value:

$$A_i \rightarrow i, \ 1 \le i \le l.$$

Based on [13, 14, 16, 17], we give our construction as follows.

***Setup*** Let $G$ be the cyclic group of prime order $p$. TA first picks randomly a generator $g$ of group $G$ and two elements $\gamma, \alpha \in Z_p$. Then it compute $g_i = g^{\alpha^i}$ for $i = 1, \cdots, n, n+2, \cdots, 2n$, where $n = l$ and sets $v = g^{\gamma}$. It also selects two elements $h_1, h_2$ from group $G$. Let $TCR$ denote a target collision resistant hash function[18].

The public parameters are

$$PK = (h_1, h_2, g, g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n}, v, TCR).$$

The master keys are $(\gamma, \alpha)$.

***Key Generation*** is the same as the previous scheme.

***Encryption*** To protect the data stored on the Cloud, the data owner encrypts the data files first and stores the encrypted data files on the Cloud. Each file is encrypted with a symmetric encryption key $K$, which is in turn encrypted with HCABE. The algorithm is run as follows:

Suppose AND-gate access policy is $W$ with $k$ attributes.

Pick a random $t \in Z_p$ and set $K = e(g_n, g_1)^{kt}$. Then construct the ciphertexts as

$$C = (C_0, C_1, C_2) = (g^t, (h_1^s h_2)^t, (v \prod_{\omega(j) \in W} g_{n+1-\omega(j)})^t),$$

where $s = TCR(C_0)$.

***Decryption*** Data consumer user first makes the following verifications:

(1) He/she verifies the attribute $\omega$ satisfies $W$. If it does not hold, then Data consumer user outputs $\perp$.

(2) He/she first computes $s = TCR(C_0)$, then verifies the following equation holds or not.

$$e(C_1, g) = e(h_1, C_0)^s e(h_2, C_0).$$

If it does not hold, then Data consumer user outputs $\perp$. Otherwise, he/she commits the following computation.

For $1 \le j \le k$, he/she compute

$$K_1 = e(g_{\omega(j)}, C_2) = e(g^{\alpha^{\omega(j)}}, (v \prod_{\omega(j)\in W} g_{n+1-w(j)})^t)$$
$$= e(g^{\alpha^{\omega(j)}}, g^{t(\gamma + \sum_{j\in W} \alpha^{(n+1-j)})}) = e(g,g)^{t\gamma\alpha^{\omega(j)} + t\sum_{j\in W} \alpha^{n+1-j+\omega(j)}}$$

$$K_2 = e(C_0, d_j \prod_{j\in W, j\neq\omega(j)} g_{n+1-j+\omega(j)})$$
$$= e(g^t, (g^{\gamma_i})^{a^{w(j)}+r_j} \prod_{j\in W, j\neq\omega(j)} g^{a^{(n+1-j+w(j))}})$$
$$= e(g^t, g^{\gamma\alpha^{w(j)}+\gamma r_j} g^{\sum_{j\in W, j\neq w(j)} \alpha^{n+1-j+w(j)}})$$
$$= e(g,g)^{t(\gamma\alpha^{w(j)}+\gamma r_j)+t\sum_{j\in W, j\neq w(j)} \alpha^{n+1-j+w(j)}}.$$

Then it computes $K_3$ as follows.

$$K_3 = \prod_{j=1}^{k} \frac{K_1}{K_2} = \prod_{j=1}^{k} \frac{e(g,g)^{t\gamma\alpha^{\omega(j)} + t\sum_{j\in W} \alpha^{n+1-j+\omega(j)}}}{e(g,g)^{t(\gamma\alpha^{w(j)}+\gamma r_j)+t\sum_{j\in W, j\neq w(j)} \alpha^{n+1-j+w(j)}}}$$

$$= \prod_{j=1}^{k} e(g,g)^{-t\gamma r_j + t\alpha^{n+1}} = e(g,g)^{kt\alpha^{n+1} - t\sum_{j=1}^{k} \gamma r_j}$$
$$= e(g,g)^{kta^{n+1} - t\gamma r},$$

where $r = \sum_{i=1}^{k} r_i$. Then user can obtain

$$K = K_3 e(\prod_{j=1}^{k} d_{0j}, C_0)$$
$$= e(g,g)^{tk\alpha^{n+1} - t\gamma r} e(g^{\gamma r}, g^t)$$
$$= e(g,g)^{tk\alpha^{n+1}} = e(g^{\alpha^n}, g^{\alpha})^{tk} = e(g_n, g_1)^{tk}.$$

*Efficiency*: This new scheme has the same distinct feature with the first scheme. The ciphertexts only have 3 elements. But we will show it achieves CCA security in the next section.

## 4. Security Analysis

**Theorem1** Suppose the *n*-DBDHE assumption holds. Then the proposed first scheme is CPA secure.

**Proof** Suppose an adversary *A* has advantage $\varepsilon$ in attacking our scheme. Using *A*, we

build an algorithm *B* that solves the decision *n*-DBDHE problem in *G* with the advantage $\varepsilon$. For a generator $g \in G$ and $\alpha \in Z_p$, set $y_i = g^{\alpha^i}$. Algorithm *B* is given as input a random tuple

$$(g, h, y_1, \ldots, y_n, y_{n+2}, \ldots, y_{2n}, T),$$

Algorithm *B*'s goal is to output 1 when $T = e(g,h)^{\alpha^{n+1}}$ and 0 otherwise. Algorithm *B* works by interacting with *A* in a game as follows:

***Init*:** The adversary *A* chooses the challenge access policy $W^*$ and gives it to *B*.

***Setup*:** *B* has received the tuple $(g, h, y_1, \ldots, y_n, y_{n+2}, \ldots, y_{2n}, T)$. In order to generate the PK for *A*, *B* chooses randomly $a \in Z_p$ and constructs

$$v = g^a (\prod_{j \in W^*} g_{n+1-j})^{-1} = g^{a - \sum_{j \in W^*} \alpha^{n+1-j}} = g^{\gamma}.$$

Then B sends

$$PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v)$$

to *A*, where $g_i = y_i$.

***Phase* 1** The adversary *A* issues a private key query for attribute $\omega \neq W^*$, where $|\omega| = k$. *B* first generate master keys for $D_1, \ldots, D_i$ as follows. Pick randomly $\gamma_1, \ldots, \gamma_{i-1}, \gamma_i' \in Z_p$ subject to the constraint that $\gamma_1 + \ldots + \gamma_{i-1} + \gamma_i' = 0$. It will implicitly set $\gamma_i = \gamma + \gamma_i'$. Then set the master key for each level as $g^{\gamma_1}, \ldots, g^{\gamma_{i-1}}, g^{\gamma_i'} v$.

Next, *B* generates the private keys for $\omega$ as follows:

Choose randomly $r_1, r_2, \ldots, r_k \in Z_p$, set

$$r = \sum_{i=1}^{k} r_i$$

and construct

$$d_{0j} = (g^{\gamma_i'} v)^r,$$

$$d_j = g_{\omega(j)}^a (\prod_{j \in W^*} g_{n+1-j+\omega(j)})^{-1} g_{\omega(j)}^{ar_j} (\prod_{j \in W^*} g_{n+1-j})^{-r_j},$$

where $1 \leq j \leq k$, The private keys are valid simulations.

In fact,

$$d_{0j} = (g^{\gamma_i'} v)^r = (g^{\gamma_i})^r,$$

$$d_j = g_{\omega(j)}^a (\prod_{j \in W^*} g_{n+1-j+\omega(j)})^{-1} g^{ar_j} (\prod_{j \in W^*} g_{n+1-j})^{-r_j},$$

$$= (g^{a - \sum_{j \in W^*} \alpha^{n+1-j}})^{\alpha^{\omega(j)} + r_j}$$

$$= (g^{\gamma})^{\alpha^{\omega(j)} + r_j}.$$

Finally, *B* sends the private keys to *A*.

***Challenge*** Now *B* will construct the challenge ciphertexts for $W^*$. *B* runs the encryption algorithm and computes the ciphertexts as follows.

$$C_0^* = h, C_1 = h^a, K^* = T^k.$$

If *B* is given a real D-BDHE tuple which means $T = e(g,h)^{\alpha^{n+1}}$, the ciphertexts are valid

for $W^*$. In fact, let $h = g^t$. Then

$$C_0^* = h = g^t$$
$$C_1^* = h^a = g^{at}$$
$$= (g^a (\prod_{j \in W^*} g_{n+1-j})^{-1} \prod_{j \in W^*} g_{n+1-j})^t$$
$$= (v \prod_{j \in W^*} g_{n+1-j})^t.$$

$B$ picks randomly $b \in \{0,1\}$ and sets $K_b = K^* = e(g,h)^{\alpha^{n+1}}$. Otherwise, $K_{1-b}$ is a random element. Finally, $B$ sends $(C_0^*, C_1^*, K_1, K_0)$ to $A$.

***Phase* 2**: The adversary continues to issue queries for attribute $\omega'$ with the constraint that $\omega' \neq W^*$.

***Guess*:** Finally, the adversary $A$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b = b'$.

If $b = b'$. then $B$ outputs 1 meaning $T = e(g,h)^{\alpha^{n+1}}$. Otherwise, it outputs 0 meaning $T$ is random in $G_1$.

When the input tuple is sampled from DBDHE tuple (where $T = e(g,h)^{\alpha^{n+1}}$), then $A$'s view is identical to its view in a real attack game and therefore $A$ satisfies

$$|\Pr[b = b'] - \frac{1}{2}| \geq \varepsilon .$$

When the input tuple is sampled from random elements (where $T$ is uniform in $G_1$), then $\Pr[b = b'] = \frac{1}{2}$. Therefore, if an adversary $A$ has advantage $\varepsilon$ in attacking our scheme, then we have

$$|\Pr(B(g,h,\mathbf{Y},e(g,h)^{a^{n+1}}) = 0) - \Pr(B(g,h,\mathbf{Y},T) = 0)|$$
$$\geq |(\frac{1}{2} \pm \varepsilon) - \frac{1}{2}| = \varepsilon ,$$

where $\mathbf{Y} = (y_1, \ldots, y_n, y_{n+2}, \ldots, y_{2n})$.

**Theorem 2** Suppose the *n*-DBDHE assumption holds. Then the second scheme is CCA secure.

The proof is similar with theorem 1. At the Phase 1 and Phase 2, it needs Decryption queries. At the Challenge phase, we can set $s^* = TCR(C_0^*)$ which is used to simulate $C_1^*$.

Note: In our CCA scheme, we used the direct technique in [18]. Hence the full proof can be achieved by combining their proof [18] with our proof in Theorem 1.

## 5. Conclusion

We presented a new HCABE system for fine-grained and flexible access control for cloud computing. The ciphertexts of the new scheme consist of two group elements, which are independent of the hierarchy depth. Encryption algorithm is much more efficient than other HCABE systems. Morever, we show this scheme can be transferred

to a CCA secure scheme.

## References


[1] R. Buyya, C . ShinYeo, J. Broberg and I. Brandic. "Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no 6, pp. 599-616, June, 2009. Article (CrossRef Link)

[2] J.M. Do, Y.J. Song and N. Park. "Attribute Based Proxy Re-encryption for Data Confidentiality in Cloud Computing Environments", in *Proc. of 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering(cnsi)*, Berlin, Springer-Verlag, pp. 248-251, May-23-25, 2011. Articl(CrossRef Link)

[3] A. Sahai, B. Waters. "Fuzzy Identity-Based Encryption," in *Proc. of EUROCRYPT 2005*, vol. 3494, Berlin, Springer-Verlag, pp. 457-473, May 22-26, 2005. Article(CrossRef Link)

[4] V. Goyal, O. Pandey, A. Sahai, etal. "Attribute-based encryption for fine grained access control of encrypted data," in *Proc. of the 13th ACM Conference on Computer and Communications Security*, CCS 2006, ACM Press, pp. 72-81, October 30-November 3, 2006. Article(CrossRef Link)

[5] S. Yu, C. Wang, K. Ren etal. "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. of IEEE INFOCOM*, pp. 534-542, March 14-19, 2010. Article(CrossRef Link)

[6] L. Ibraimi, Q. Tang and P. Hartel etal. "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Proc. of the Information Security Practice and Experience*, Berlin, Springer-Verlag, pp.1−12, April 13-15, 2009. Article(CrossRef Link)

[7] N. Attrapadung, H. Imai. "Dual-Policy attribute based encryption," in *Proc. of the Applied Cryptography and Network Security (ACNS)*. Berlin, Springer-Verlag, pp.168−185, June 2-5, 2009. Article(CrossRef Link)

[8] A. Lewko, T. Okamoto and A Sahai, etal. "Fully secure functional encryption: Attribute-Based encryption and (hierarchical) inner product encryption," in *Proc. of Advances in Cryptology-EUROCRYPT 2010*, LNCS 6110, Berlin, Springer-Verlag, pp. 62−91, May 30- June 3, 2010. Article(CrossRef Link)

[9] D. Boneh, X. Boyen and E. Goh. "Hierarchical identity based encryption with constant size ciphertext," in *Proc. of Advances in EUROCRYPT*, vol. 3494, Berlin, Springer-Verlag, pp. 440-456, May 22-26, 2005. Article(CrossRef Link)

[10] L.Y. Zhang, Y.P. Hu, Q. Wu. "Hierarchical Identity-Based Encryption with Constant size private keys," *ETRI Journal*, 34(1), pp.142-145, February 2012. Article(CrossRef Link)

[11] B. Waters. "Dual key encryption: Realizing fully secure IBE and HIBE under simple assumption," in *Proc of Advances in cryptology-CRYPTO*, vol. 5677, Berlin, Springer-Verlag, pp. 619-636, August 16-20, 2009. Article(CrossRef Link)

[12] D. Cash, D. Hofheinz and E. Kiltz. "How to Delegate a Lattice Basis," *Journal of Cryptology*, vol. 25, no. 4, pp 601-639, October 2012. Article(CrossRef Link)

[13] G. Wang, Q. Liu and J. Wu. "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. of ACM Conference on Computer and Communications Security (CCS)-2010 (Poster)*, ACM Press, pp. 735-737, October 4-8, 2010. Aticle(CrossRef Link)

[14] G. Wang, Q. Liu and J. Wu, et al. "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers." *Computers and Security*, vol. 30, no. 5, pp. 320–331, July 2011. Article(CrossRef Link)

[15] R. Canetti, O. Goldreich and S. Halevi. "The random oracle methodology," *Journal of ACM*, vol. 51, no 4, pp. 557-594, July 2004. Article(CrossRef Link)

[16] J. Liu, Z.G. Wan and M. Gu. "Hierarchical Attribute-Set Based Encryption for Scalable, Flexibleand Fine-Grained Access Control in Cloud Computing," in *Proc. of the Information Security Practice and Experience*.-*ISPEC*, vol. 6672, Berlin, Springer-Verlag, pp. 98-107, May 30- June 1, 2011. Article(CrossRef Link)

[17] Z. B. Zhou, D. J. Huang. "On Efficient Ciphertext-Policy Attribute Based Encryption and Broadcast Encryption," in *Proc. of ACM Conference on Computer and Communications Security*

*(CCS)*, ACM Press, pp. 753-755, October 4-8, 2010. Article(CrossRef Link)

[18] E. Kiltz. "Chosen-ciphertext security from tag-based encryption," in *Proc. of TCC 2006: 3rd Theory of Cryptography Conferenc*e, Berlin, Springer-Verlag, LNCS 3876, pp. 581-600, March 4-7, 2006. Article(CrossRef Link)

**Leyou Zhang** received his Ph.D. from the Xidian University in 2009. Now he is an Associate Professor in the department of Mathematical science of Xidian University. His current research interests include network security, security in Cloud Computing, and public key cryptography. He has published more than 50 papers in international journals and conferences.

**Ying Xia**, received the Ph.D degree in computer science and technology from the Southwest Jiaotong University, China, 2012. Currently, she is a professor at Chongqing University of Posts and Telecommunications, China. Her research interests include spatial database, GIS, cloud computing and cross-media retrieval.

**Yupu Hu** received his Ph.D. from the Xidian University in 1999. Now he is a Professor in the School of Telecommunications Engineering of Xidian University. His current research interests include information security and cryptography. He has published more than a hundred papers in international journals and conferences. He is a Member of China Institute of Communications and a Director of Chinese Association for Cryptologic Research.