A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments

Cheng-Chi Lee[1], Pei-Shan Chung[2], and Min-Shiang Hwang[3]
(Corresponding author: Min-Shiang Hwang)

Department of Library and Information Science, Fu Jen Catholic University[1]
No. 510, Zhongzheng Rd., Xinzhuang Dist., New Taipei City 24205, Taiwan, R.O.C.
Department of Management Information Systems, National Chung Hsing University[2]
250 Kuo Kuang Road, Taichung 402, Taiwan, R.O.C.
Department of Computer Science and Information Engineering, Asia Universtiy[3]
500 Liufeng Road, Wufeng, Taichung 402, Taiwan, R.O.C.
(Email: mshwang@asia.edu.tw)
(Invited Paper)

Abstract

In Attribute-based Encryption (ABE) scheme, attributes play a very important role. Attributes have been exploited to generate a public key for encrypting data and have been used as an access policy to control users' access. The access policy can be categorized as either key-policy or ciphertext-policy. The key-policy is the access structure on the user's private key, and the ciphertext-policy is the access structure on the ciphertext. And the access structure can also be categorized as either monotonic or non-monotonic one. Using ABE schemes can have the advantages: (1) to reduce the communication overhead of the Internet, and (2) to provide a fine-grained access control. In this paper, we survey a basic attribute-based encryption scheme, two various access policy attribute-based encryption schemes, and two various access structures, which are analyzed for cloud environments. Finally, we list the comparisons of these schemes by some criteria for cloud environments.

Keywords: Cloud computing, attribute-based encryption, access control, fine-grained access, revocation.

1 Introduction

Internet technology is growing more and more quickly, and people can process, store, or share with their data by using its ability. Recently, the cloud has emerged to provide various application services to satisfy users' requirement [1]. In the storage service application, the cloud can let the user, data owner, store his data, and share this data with other users via the cloud, because the cloud can provide the pay as you go environment[8] where people just need to pay the money for the storage space they use. It can bring down the cost efficiently for people. But, there is a problem that the data owner has to solve it. The data owner needs to make a flexible and scalable access control policy to command users' access right, so that only the authorized users can access [6, 31].

Besides, for protecting the confidentiality of the stored data, the data must be encrypted before uploading to the cloud [12, 15, 29]. Traditional public key infrastructure can be adopted in the data encryption process, and the data owner uses data users' public key to encrypt this data before uploading to the cloud; if the data user sends through a access request to the cloud, then the cloud would return the corresponding ciphertext to the data user. An user would use his private key to decrypt this data. But this manner would lead to some problems: (1) to be able to encrypt data, the data owner needs to obtain the data user's public key to complete this; (2) a lot of storage overhead would spend because of the same plaintext with different public keys.

For improving these disadvantages, Sahai and Waters proposed an attribute-based encryption (ABE) scheme [27] in 2005, and this paper proposed the first concept of the attribute-based encryption scheme. The ABE scheme used an user's identity as attributes, and a set of attributes were used to encrypt and decrypt data. The ABE scheme can result the problem that data owner needs to use every authorized user's public key to encrypt data. And in the same year, Nail et al. proposed an threshold attribute-based encryption which can prevent the collusion attacks [25].

In 2006, Goyal et al. proposed an key-policy attribute-based encryption (KP-ABE) scheme [11] that built the

access policy into the user's private key and described the encrypted data with user's attributes. The KP-ABE scheme can achieve fine-grained access control and more flexibility to control users than ABE scheme. But the disadvantage of KP-ABE is that the access policy is built into an user's private key, so data owner can't choose who can decrypt the data except choosing a set of attributes which can describe this data. And it is unsuitable in certain application because a data owner has to trust the key issuer. Besides, the access structure in KP-ABE is a monotonic access structure, it can't express the negative attribute to exclude the parties with whom data owner didn't want to share data from memberships.

So Ostrovsky et al. proposed a non-monotonic access structure [26] in 2007, and this scheme can let each attribute attach primed word in front of them. And Bethencourt et al. also proposed an ciphertext-policy attribute based (CP-ABE) scheme [2] in the same year, and the CP-ABE scheme built the access policy into the encrypted data; a set of attributes is in an user's key. The CP-ABE scheme addresses the problem of KP-ABE that data owner only trusts the key issuer. After that, several schemes were proposed based on the CP-ABE scheme [7, 9, 10, 13, 14, 16, 18, 24, 34].

Moreover, Muller et al. proposed an distributed attribute-based encryption scheme [23] in 2008; Yu et al. proposed a fine-grained data access control encryption scheme [35], Tang et al. proposed a Verifiable attribute-based encryption scheme [30], and Wang et al. proposed a hierarchical attribute-based encryption scheme (HABE) [32, 33] in 2010 and 2011. This scheme uses the disjunctive normal form policy and generates the keys hierarchically. And this scheme assumed that all attributes in one conjunctive clause are administered by the same domain authority. In addition to this, there are multi-authorities ABE schemes [3, 4, 5, 19, 20] that use multiple parties to distribute attributes for users.

Based on the type of access structure, attribute-based encryption schemes can be roughly categorized as either monotonic or non-monotonic. And based on the access policy, these schemes also can be roughly categorized as either key policy or ciphertext-policy. In this paper, the survey started from basic attribute-based encryption scheme, followed by monotonic access structure which could be divided into key-policy attribute-based encryption scheme, ciphertext-policy attribute-based encryption scheme. Attribute-based encryption scheme with non-monotonic structure is introduced. Thereafter, and hierarchical attribute-based encryption scheme as the end.

1.1 The Criteria of An Ideal Attribute-based Encryption Scheme

According to these schemes, a summary of the criteria, that ideal attribute-based encryption schemes, are listed as follows.

C1. ***Data confidentiality***
Before uploading data to the cloud, the data was encrypted by the data owner. Therefore, unauthorized parties including the cloud cannot know the information about the encrypted data.

C2. ***Fine-grained access control***
In the same group, the system granted the different access right to individual user. Users are on the same group, but each user can be granted the different access right to access data. Even for users in the same group, their access rights are not the same.

C3. ***Scalability***
When the authorized users increase, the system can work efficiently. So the number of authorized users cannot affect the performance of the system.

C4. ***User accountability*** **[17]**
If the authorized user is dishonest, he would share his attribute private key with the other unauthorized user. It causes the problem that the illegal key would share among unauthorized users.

C5. ***User revocation***
If the user quits the system, the scheme can revoke his access right from the system directly. The revocable user cannot access any stored data, because his access right was revoked.

C6. ***Collusion resistant***
Users cannot combine their attributes to decipher the encrypted data. Since each attribute is related to the polynomial or the random number, different users cannot collude each other.

1.2 Organization

In this paper, we survey several attribute-based encryption schemes including two varied access structures, which are monotonic and non-monotonic. The organization of the paper is organized as follows. In Section 2, we introduce these attribute-based encryption schemes. In Section 3, we compare these schemes by using the criteria illustrated in Section 1, and in Section 4, our conclusions are given.

2 Related Works

In cloud environments, if a data owner wants to share data with users, he will encrypt data and then upload to cloud storage service. Through the encryption step, the cloud cannot know the information of the encrypted data. Besides, to avoid the unauthorized user accessing the encrypted data in the cloud, a data owner uses the encryption scheme for access control of encrypted data. In existing schemes, many encryption schemes can achieve and provide security, assure data confidential, and prevent collusion attack scheme. One of the encryption schemes is attribute-based encryption scheme.

The first concept of attribute-based encryption was proposed in 2005. And then many attribute-based encryption schemes were proposed. According to the access policy, two types of these schemes can be classified, the key-policy and ciphertext-policy attribute-based encryption schemes. The key-policy attribute-based scheme is that the access policy is attached to the user's private key, and a set of descriptive attributes is in the encrypted data. If a set of attributes satisfies the access policy, the user will recover the message. If not, he cannot obtain it. And the ciphertext-policy attribute-based scheme is that the access policy is associated to the encrypted data, and a set of descriptive attributes is in the user's private key. If a set attribute satisfies the access policy, the user can decipher the encrypted data. In this section, we will introduce five attribute-based encryption schemes. And according to the type of access policy, there are monotonic access structure and non-monotonic access structure. Non-monotonic access structure can use the negative word to describe every attribute, but the monotonic access structure cannot.

The notations used in this paper are listed in Table 1.

2.1 Attribute-based Encryption Scheme

Sahai and Waters proposed an attribute based encryption scheme in 2005. There are authority, data owner (also be called sender) and data user (also be called receiver) in this scheme, and authority 's role is to generate keys for data owners and users to encrypt or decrypt data. In this scheme, the authority generates keys according to attributes; and these attributes of public key and master key, which are generated by the authority, should predefine (means that it will list attributes which will be used in the future). If any data user who wants to add to this system, and he owns to attributes don't include predefined attributes. The authority will re-define attributes and generate a public key and master key again. And data owner's role in this scheme is to encrypt data with a public key and a set of descriptive attributes. A data user's role is to decrypt encrypted data with his private key sent from the authority, and then he can obtain the needed data.

For decrypting data, attributes in data user's private key will check by matching with the attributes in encrypted data. If the number of "matching" is at least a threshold value d, the data user's private key will be permitted to decrypt the encrypted data. For example, for a set of descriptive attributes in the encrypted data, $\{MIS, Teacher, Student\}$, the threshold value is 2. If a data user wants to decrypt the encrypted data, his number of attributes in private key will need two or the more than two of attributes in the encrypted data, so that a data user has a private key with attributes, $\{MIS, Student\}$ to decrypt and obtain the data.

In this scheme, there are four algorithms to be executed: Setup, KeyGen, Encrypt, and Decrypt. Let G_1 and G_2 be two bilinear groups of prime order p, and let g be a generator of G_1. In addition , let $e : G_1 \times G_1 \longrightarrow G_2$ denote the bilinear map, and let d be a threshold value.

1) Setup(d): The authority uniformly and randomly chooses t_1, ..., t_n, y from Z_q, and publishes the public key, $PK = (T_1 = g^{t_1}, ..., T_n = g^{t_n}, Y = e(g,g)^y)$. And the master key is $MK = (t_1, ..., t_n, y)$.

2) KeyGen(A_U, PK, MK): The authority executes and generates a private key for the data user U. Choose a $d-1$ degree polynomial q randomly such that $q(0) = y$. The data user's private key D is $\{D_i = g^{\frac{q(i)}{t_i}}\}_{\forall i \in A_U}$.

3) Encrypt(A_{CT}, PK, M): Data owner encrypts message $M \in G_2$ with a set of attributes A_{CT}. Choose a random number $s \in Z_q$, and the encrypted data is published as $CT = (A_{CT}, E = MY^s = e(g,g)^{ys}, \{E_i = g^{t_i s}\}_{\forall i \in A_U})$.

4) Decrypt(CT, PK, D): Data user decrypts the encrypted data CT with the private key D. Choose d attributes from $i \in A_U \bigcap A_{CT}$ to compute $e(E_i, D_i) = e(g,g)^{q(i)s}$ if $|A_U \bigcap A_{CT}| \geq d$. And compute $Y^s = e(g,g)^{q(0)s} = e(g,g)^{ys}$ with the Lagrange coefficient, and the message $M = E/Y^s$ can be obtained.

In KenGen() algorithm, the user's private key is generated with secret sharing [28] in this scheme. The shares of secret y are embedded in the components of the user's private key D_i, and the secret key is associated with the random polynomial $q(i)$. So every user's private key D cannot be combined to a new private key to perform the collusion attack. And in the Encrypt() algorithm, the random number s can avoid user decrypting the data after the first decrypting, when he infers the number. Besides, the component of the encrypted data E_i would be used in Decrypt() algorithm, the needed attributes can be known through this component. The attributes in user's private key and the encrypted data can let this scheme achieve access control. The authorized users can use their private key to decrypt the corresponding data. In addition, application of this scheme would be restricted in the real environment because it use the access of monotonic attributes to control user's access.

2.2 Key-Policy Attribute-based Encryption Scheme

In 2006, Goyal proposed an key-policy attribute-based (KP-ABE) scheme. This scheme uses a set of attributes to describe the encrypted data and builds a access policy in user's private key. If attributes of the encrypted data can satisfy the access structure in user's private key

Table 1: The notations

Notation	Signification
G_x	The bilinear group of prime order p, $x = 1, 2$
g	A generator of G_1
A_U	Attributes of data user U in private key
A_{CT}	Attributes with the encrypted data CT
A_{U-KP}	The access structure in user's private key
A_{CT-CP}	The access structure in the encrypted data
A_{CT-HA}	The DNF access control policy in the encrypted data
$\widetilde{A_U}$	The non-monotonic access structure in user's private key
D	User's private key
M	The message

D, an user can obtain the message through decrypt algorithm. In addition, the KeyGen() algorithm is different from the attribute-based encryption which is introduced at subsection one in this section. The user's private key is according to the access structure to generate. In this algorithm, it adopts secret sharing and chooses a polynomial q_x such that $q_x(0) = q_{parent(x)}(index(x))$, (Note that $parent(x)$ is x's parent node, and $index(x)$ is the number associated with node x that is given by x's parent node.) in a top-down manner which is to start from the root node r for each node x in the access structure. So $q_r(0)$ is equal to the master key y, and the master key y is distributed among the user's private key component D_i which is corresponding to the leaf node (Note that the leaf node represents attribute).

Since the KeyGen() algorithm is different, the Decrypt() algorithm also be different. It use attributes of encrypted data to run decryptnode function in the decryption algorithm. And it can input encrypted data, user's private key, and nodes of the access structure in user's private key; it adopts bottom-up manner in the access structure and recursive manner to decrypt the encrypted data. Beside, this scheme divides nodes of the access structure into the equal the leaf nodes. Finally, it will get a bilinear formula and use polynomial interpolation to get the message. For example, the encrypted data with attributes are $\{MIS \bigwedge Student\}$, and user's private key with access structure is $\{MIS \bigwedge (Teacher \bigvee Student)\}$. The encrypted data with attributes satisfies the access structure of an user's private key, and then user can get the message.

In this scheme, there are four algorithms to be executed: Setup, KeyGen, Encrypt, and Decrypt. And the parameters described in this scheme and parameters of the ABE scheme are the same. It will be depicted as follows.

1) Setup(d): The authority chooses several uniform and random numbers t_1, ..., t_n, y from Z_q, and makes public the public key, $PK = (T_1 = g^{t_1}, ..., T_n = g^{t_n}, Y = e(g,g)^y)$. And keeps the master key, $MK = (t_1, ..., t_n, y)$ be secret.

2) KeyGen(A_{U-KP}, PK, MK): The authority generates private key components for each leaf node x in the access structure. The private key components are $D_x = g^{\frac{q_x(0)}{t_i}}$, where i is equal to a leaf node in the access structure. These components will be merged into the user's private key, and be sent to an user.

3) Encrypt(M, A_{CT}, PK): Data owner chooses a random number s from Z_q and encrypts a message $M \in G_2$ with a set of attributes A_{CT}, and then he generates the encrypted data as $CT = (A_{CT}, E = MY^s = e(g,g)^{ys}, \{E_i = g^{t_i s}\}_{\forall i \in A_{CT}})$.

4) Decrypt (CT, D): This algorithm can be executed by a recursive algorithm, It inputs the encrypted data, user's private key, and nodes of the access structure in user's private key. If i is equal to the leaf node, and i is in the access structure of user's private key, it will call the decryptnode function, $e(D_x, E_i) = e(g,g)^{s \cdot q_x(0)}$. If i is not in the access structure of an user's private key, it will call the decryptnode function; and it outputs invalid. If i is not equal to the leaf node, it will call decryptnode function and input all children nodes of node x, z, and use lagrange coefficient to compute to obtain $e(g,g)^{s \cdot q_x(0)}$. Finally, the decryption algorithm call the decryptnode function on the root of the access structure and compute $e(g,g)^{ys} = Y^s$, if and only if the encrypted data satisfies the access structure of private key. And the message $M = \frac{E}{Y^s}$ can be obtained.

In this scheme, the user's private key is associated with access structure, and the encrypted data with a set of descriptive attributes can be used to be corresponding to the access structure of the user's private key. Since access control is built in user's private key, the attributes of the encrypted data satisfies access structure so to let a data user decrypt the encrypted data. However, the access

control right is owned by user's private key, and some people just obtaining this key can let him decrypt data. It means the user's private key can choose the encrypted data which is satisfied, but the encrypted data cannot choose who can decrypt this data. For a scheme, the ciphertext policy attribute-based encryption is proposed. It builds the access policy in the ciphertext and uses a set of attributes to describe the user's private key.

2.3 Ciphertext-Policy Attribute-based Encryption Scheme

In 2007, Bethencourt et al. proposed a ciphertext policy attribute-based scheme, and the access policy in the encrypted data (ciphertext). The access control method of this scheme is similar to the key policy attribute-based encryption. In key policy attribute-based encryption, the access policy is in user's private key, but the access policy is switched to the encrypted data in ciphertext policy attribute-based encryption. And a set of descriptive attributes are associated with the user's private key, and the access policy is built in the encrypted data. The access structure of the encrypted data is corresponding to the user's private key with a set of descriptive attributes. If a set of attributes in user's private key satisfies the access structure of the encrypted data, the data user can decrypt the encrypted data; if it cannot, the data user cannot obtain the message. For example, the access structure in the encrypted data is $\{MIS \bigwedge (Teacher \bigvee Student)\}$. If a set of attributes in user's private key is $\{MIS \bigwedge Teacher\}$, the user can recover the data.

In the access structure of this scheme, it adopts the same method which was depicted in KP-ABE to build. And the access structure built in the encrypted data can let the encrypted data choose which key can recover the data, it means the user's key with attributes just satisfies the access structure of the encrypted data. And the concept of this scheme is very close to the traditional access control scheme. There are five algorithms in this scheme, Setup(), KeyGen(), Encrypt(), Delegate(), Decrypt(). The Delegate algorithm is in addition more than above schemes, and it can input user's private key and regenerate the new one with another attributes which are in a set of attributes of the original user's private key. And this key is equal to the key generated from the authority.

The parameters in this scheme and in KP-ABE are the same. This scheme will be described as follows.

1) Setup: The authority chooses two random numbers α, β from Z_q as exponents, and generates the public key, $PK = (G_0, g, h = g^{\beta}, f = g^{\frac{1}{\beta}}, e(g,g)^{\alpha})$. The master key is $MK = (\beta, g^{\alpha})$.

2) KeyGen(MK, A_U): The authority chooses a random number s from Z_q, and random s_j for each attribute j in a set of attributes in user's private key. The user's private key, $D = (DK = g^{\frac{(\alpha+s)}{\beta}}, \forall j \in A_U : D_j = g^s \cdot H(j)^{s_j}, D_j^* = g^{s_j})$ is output.

3) Encrypt(PK, M, A_{CT-CP}): Data owner executes this algorithm to encrypt the message M with the access structure A_{CT-CP}. Choose a random number $y \in Z_q$, set $q_r(0) = y$, where r is the root node, and let I be the set of leaf nodes in A_{CT-CP}. The message is encrypted with access structure A_{CT-CP}, and then outputs the encrypted data, $CT = (A_{CT-CP}, \widetilde{C} = Me(g,g)^{\alpha y}), C = h^y, \forall i : C_i = g^{q_i(0)}, C_i^* = H(att(i))^{q_i(0)})$.

4) Delegate$(D, \widetilde{A_U})$: This algorithm takes the user's private key D and a set of attributes whose each attribute is in A_U to create a new user's private key $\widetilde{D}$.

5) Decrypt(CT, D): When data user receives the encrypted data, he can execute this algorithm. The user's private key D and the encrypted data are input in this algorithm and the recursive function, and decryptnode is called. If the node x is a leaf node and let $k = att(x)$, where $k \in A_U$, the decryptnode can be called , then it computes decryptnode$(CT, D, x) = \frac{e(D_k, C_x)}{e(D_k^*, C_x^*)} = e(g,g)^{sq_x(0)}$. If k is not in A_U, decryptnode will output invalid. If x is not the leaf node, the decryptnode function can be called and all children nodes of node x, z can be input to execute. It use Lagrange coefficient to compute and obtain $e(g,g)^{sq_x(0)}$. Hence, if the access structure A_{CT-CP} satisfies A_U, the CT, D, s are input to compute decryptnode$(CT, D, s) = e(g,g)^{sy}$. The algorithm can decrypt by computing $\frac{\widetilde{C}}{e(C, DK)/e(g,g)^{ys}} = M$ to recover the message M.

The CP-ABE builds the access structure in the encrypted data to choose the corresponding user's private key to decipher data. It improves the disadvantage of KP-ABE that the encrypted data cannot choose who can decrypt. It can support the access control in the real environment. In addition, the user's private key is in this scheme, a combination of a set of attributes, so an user only use this set of attributes to satisfy the access structure in the encrypted data. Moreover, the CP-ABE scheme is applied in the proxy re-encryption field to increase security of this field. The CP-ABE scheme can be applied in the scheme which can achieves proxy re-encryption in cloud environments [36].

2.4 Attribute-based Encryption Scheme with Non-Monotonic Access Structures

In 2007, Ostrovsky et al. proposed an attribute-based encryption with non-monotonic access structure. The

access formula of access structure in user's private key can represent any type through attributes such as negative ones. It is different from the previous attribute-based encryption scheme. The previous schemes are like KP-ABE scheme, and the access structure in user's private key has monotonic access formula. No negative attributes exist in it. Apart from this, the access structure of this scheme is the same as the access structure of KP-ABE scheme. There is a Boolean formula such as And, OR, and threshold gates in these access structures, but there is a boolean formula, NOT in access structure of this scheme. However, other schemes do not include it. There is an example for this scheme. If a teacher in department of information management wants to share the data with students, he will set a set of attributes in the encrypted data. And there is an access structure, $\{MIS \bigwedge Student\}$ in students' private key. But the teacher doesn't want graduates to access this data, he adds $NOTgraduate$ to the access structure. So the access structure is $\{MIS \bigwedge Student \bigwedge NOTgraduate\}$. It can let data not be accessed by graduates.

This scheme proposes the first method that can add negative constraints to describe attributes. And it is flexible to use access policy for a data owner. This scheme contains four algorithms: Setup(), KeyGen(), Encrypt(), and Decrypt(), and they will be introduced as follows.

1) Setup(d): A parameter d is decided that how many attributes the encrypted data has. And let G_1 be a bilinear group of prime order p, let g be a generator of G_1, and let $e : G_1 \times G_1 \longrightarrow G_2$ denote the bilinear map. After that, choose two random numbers α, β from Z_q, and denote $g_1 = g^\alpha, g_2 = g^\beta$. Let $h(x), q(x)$ be two polynomials of degree d and constraint that $q(0) = \beta$. Generate the public key, $PK = (g, g_1; g_2 = g^{q(0)}, g^{q(1)}, g^{q(2)}, ..., g^{q(d)}; g^{h(0)}, g^{h(1)}, g^{h(2)}, ..., g^{h(d)})$, and the master key, $MK = \alpha$. In addition, denote and publish two publicly computable function, $T, V : Z_q \longrightarrow G_1$ such as $T(x) \longrightarrow g_2^{x^d} \cdot g^{h(x)}, V(x) \longrightarrow g^{q(x)}$.

2) KeyGen($\widetilde{A_U}, PK, MK$): The authority would execute this algorithm to output a key for users, and let them decrypt the encrypted data only if the attributes of the encrypted data that satisfies the non-monotonic access structure $\widetilde{A_U}$ in user's private key. Let $\widetilde{A_U}$ be a non-monotonic access structure, and choose a random number $s_i \in Z_q$ for each attribute x_i. Moreover, denote a polynomial $p(x)$ randomly such that $p(0) = \alpha$. For x_i is a non-negated attribute, the component of user's private key is $D_i = (D_i^{(1)} = g_2^{p(x_i)} \cdot T(x_i)^{s_i}, D_i^{(2)} = g^{s_i})$. For x_i is a negated attribute, the component of user's private key is $D_i = (D_i^{(3)} = g_2^{p(x_i)+s_i}, D_i^{(4)} = V(x_i)^{s_i}, D_i^{(5)} = g^{s_i})$. And then every user's private key contains each component of private key D_i.

3) Encrypt(M, A_{CT}, PK): when a data owner wants to encrypt a message $M \in G_2$ under a set of attributes $A_{CT} \subset Z_q^*$, he would comply this algorithm. First, the random number s is chosen from Z_q, and is used to compute the encrypted data. And then output the encrypted data as $CT = (A_{CT}, E^{(1)} = M \cdot e(g_1, g_2)^s, E^{(2)} = g^s, \{E_x^{(3)} = T(x)^s\}_{x \in A_{CT}}, \{E_x^{(4)} = V(x)^s\}_{x \in A_{CT}})$.

4) Decrypt(CT, D): Input the encrypted data CT and private key D, and this algorithm is executed. First, a data user checks if $A_{CT} \in \widetilde{A_U}$. If not, its output is invalid. If $A_{CT} \in \widetilde{A_U}$, it will compute $\frac{e(D_i^{(1)}, E^{(2)})}{e(D_i^{(2)}, E_i^{(3)})} = e(g_2, g)^{s \cdot p(x_i)}$ for a non-negated attribute x_i, and compute $\frac{e(D_i^{(3)}, E^{(2)})}{e(D_i^{(5)}, \Pi_{x \in A_{CT}} (E_i^{(4)})^{\sigma_x}) \cdot e(D_i^{(4)}, E^{(2)})^{\sigma x_i}} = e(g_2, g)^{s \cdot p(x_i)}$, where $\{\sigma_x\}_{x \in A_{CT}}$ is a Lagrangian coefficient, for a negated attribute. And use the number of $e(g_2, g)^{s \cdot \alpha}$ is $d+1$ to compute , and obtain $e(g_2, g)^{s \cdot \alpha}$. The message can be recovered by computing $\frac{m \cdot e(g_2, g)^{s \cdot \alpha}}{e(g_2, g)^{s \cdot \alpha}} = M$.

This scheme is undesirable for two reasons. First, there are many negative attributes in the encrypted data, but they don't relate to the encrypted data. It means that each attribute adds a negative word to describe it, but these are useless for decrypting the encrypted data. It can cause the encrypted data overhead becoming huge. In addition, the new attributes may be used after the encrypted data is created, and a data owner can't know all the attributes which may be used to encrypt in the future. In addition, the negative attributes are used to let the setting of access structure be more flexible. Data owner can add a negative word in front of an attribute, and this action can let the person who possesses this attribute be unable to decrypt the data.

2.5 Hierarchical Attribute-based Encryption Scheme

In 2011, Wang et al. proposed a hierarchical attribute-based encryption scheme composed of a hierarchical identity-based encryption scheme (HIBE) and a ciphertext-policy attribute-based encryption scheme. This scheme used the property of hierarchical generation of keys in HIBE scheme to generate keys. Moreover, it used disjunctive normal form (DNF) to express the access control policy, and the same domain authority in this scheme administered all attributes in one conjunctive clause. There are five roles in this scheme: the cloud storage service, data owner, the root authority, the domain authority, and data users. The role of cloud storage service is that let a data owner can store data and share data with users. The role of data owner is encrypting data and sharing data with users. The role of the root authority is generating system parameters and domain

keys, to distribute them. The role of domain authority is managing the domain authority at next level and all users in its domain, to delegate keys for them. Besides, it can distribute secret keys for users. And users can use their secret keys to decrypt the encrypted data and obtain the message.

The key generation in this scheme adopts a hierarchical method. The root authority generates a root master key for domain authority at the first level. The system public key and the master key of the domain authority at first level are used to create the master keys for the domain authorities at the next level by the root authority or the domain authority at the first level. In addition, the domain authority generates the user identity secret key and the user attribute secret key for the authorized user. The processes of this scheme will be introduced as follows.

1) Setup(K): The security parameter K is input, mk_0 is chosen from Z_q, two bilinear groups G_1, G_2 of order p, and a bilinear map $e : G_1 \times G_1 \longrightarrow G_2$ are chosen by the root authority. And then three random oracle H_1, H_2, H_A, and a generator $P_0 \in G_1$ are picked. The system public key is $PK = (p, G_1, G_2, e, P_0, Q_0, H_1, H_2, H_A)$, and the system master key is $MK_0 = mk_0$ which will be kept secret.

2) CreateDM(PK, MK_i, PK_{i+1}): The root authority or the domain authority generates mater keys $MK_{i+1} = (mk_{i+1}, Hmk_{i+1}, D_{i+1}, Q - tuple_{i+1})$ for domain authorities DM_{i+1} by using system public key PK, the public key of domain authorities DM_{i+1}, PK_{i+1}, and its master key MK_i. Where mk_{i+1} is the index of the random oracle $H_{mk_{i+1}}$, $D_{i+1} = D_i + mk_i P_{i+1}$, $P_{i+1} = H_1(PK_{i+1}) \in G_1$, $Q - tuple_{i+1} = (Q - tuple_i, Q_{i+1})$, and $Q_{i+1} = mk_{i+1}P_0 \in G_1$. Assume that D_0 is an identity element of G_1, and $Q - tuple_0 = (Q_0)$.

3) CreateUser(PK, MK_i, PK_u, PK_a): The domain authority first checks whether the user u is authorized for attribute a which is monitored by itself, when a user sends a request to domain authority for the user identity secret key $D_{i,u}$ and the user attribute secret key on a, $D_{i,u,a}$. If so, it creates the user identity secret key $D_{i,u} = (Q - tuple_{i-1}, mk_i \cdot mk_u P_0)$, and the user attribute secret key $D_{i,u,a} = D_i + mk_i \cdot mk_u P_a \in G_1$ by computing $mk_u = H_A(PK_u) \in Z_q$ and $P_a = H_{mk_i}(PK_a)P_0 \in G_1$. If not, it outputs "NULL".

4) Encrypt($PK, M, A_{CT-HA}, \{PK_a \mid a \in A_{CT-HA}\}$): The data owner encrypts data M with a DNF access control policy $A_{CT-HA} = \bigvee_{i=1}^{N}(CC_i) = \bigvee_{i=1}^{N}(\bigwedge_{j=1}^{n_i} a_{ij})$ and public keys of all attributes in access control policy $\{PK_a \mid a \in A_{CT-HA}\}$. The encrypted data is $CT = [A_{CT-HA}, C_f = (U_0, U_{12}, ..., U_{1t_1}, U_1, ..., U_{N2}, ..., U_{Nt_N}, U_N, V)]$.

   Where $U_0 = rP_0$, r is a random number which is selected from Z_q, $U_0 = rP_0$, $U_{12} = rP_{12}$, $U_{1t_1} = rP_{1t_1}$, $U_1 = r\sum_{j=1}^{n_1} P_{a_{1j}}$, $U_{N2} = rP_{N2}$, $U_{Nt_N} = rP_{Nt_N}$, $U_N = r\sum_{j=1}^{n_N}$, $P_{ij} = H_1(PK_{ij}) \in G_1$ for $1 \le i \le N$ and $1 \le j \le t_i$, $P_{a_{ij}} = H_{mk_{it_i}}(PK_{a_{ij}})P_0 \in G_1$ for $1 \le i \le N$ and $1 \le j \le n_i$, and $V = M \oplus H_2(e(Q_0, rn_A P_1))$. Note that n_A is the lowest common multiple of $n_1, ..., n_N$.

5) Decrypt($PK, CT, D_{i,u}, \{D_{i,u,a} \mid a \in CC_j\}$): The user want to obtain the message M , and his attributes satisfies the access policy in the encrypted data A_{CT-HA}. He recovers the message M by computing
$V \oplus H_2(\frac{e(U_0, \frac{n_A}{n_i}\sum_{j=1}^{n_i} D_{it_i,u,a_{ij}})}{e(mk_u mk_{it_i} P_0, \frac{n_A}{n_i} U_i)\prod_{j=2}^{t_i} e(U_{ij}, n_A Q_{i(j-1)})}) = M.$

This scheme can satisfy the property of fine-grained access control on the cloud by combining HIBE scheme and CP-ABE scheme , and full delegation to cloud computation. It can share data for users in the cloud in an enterprise environment. Furthermore, it can apply to achieve proxy re-encryption [21, 22]. But in practice, it is unsuitable to implement. Since all attributes in one conjunctive clause in this scheme may be administered by the same domain authority, the same attribute may be administered by multiple domain authorities.

3 Comparisons

In this section, we compare these schemes which we survey. First, we compare these schemes by the criteria that we listed in Section 1. And the second, we compare these schemes by the length of their user's private key and ciphertext, and by the operation of the encrypt and decrypt algorithm.

3.1 Security Analysis

These schemes which we survey were compared by the criteria listed in Section 1. The criteria contain C1- fine-grained access control, C2- data confidentiality, C3- scalability, C4- user accountability, C5- user revocation, and C6- collusion resistant. This comparison table is listed in Table 2.

We can know that these schemes almost cannot satisfy the criteria of scalability and user accountability, and they all can achieve the data confidentiality. The ABE scheme only satisfies one criteria. Because it uses the attributes in the user's private key to match the attributes in the encrypted data, it only achieve the basic security requirement. But it provides the first concept to develop the attribute-based encryption scheme. After that, these schemes cannot satisfy all the criteria except HABE. Besides, the criteria of user accountability is hard to achieve.

Table 2: The criteria of an ideal attribute-based encryption scheme

Item	ABE	KP-ABE	CP-ABE	ABE with non-monotonic	HABE
C1	N	Y	Y	Y	Y
C2	Y	Y	Y	Y	Y
C3	N	N	N	N	Y
C4	N	N	Y	N	Y
C5	N	Y	Y	Y	Y
C6	Y	Y	Y	Y	Y

Table 3: The comparison in the length of user's key and ciphertext

Item	ABE	KP-ABE	CP-ABE	ABE with non-monotonic	HABE
User's Private Key	$\|A_U\| + L_{G_1}$	$\|A_U\|L_{G_1}$	$(2\|A_U\| + 1)L_{G_1}$	$6\|A_U\|L_{G_1}$	$\|A_U\|L_{G_1}$
Ciphertext	$\|A_{CT}\|L_{G_1} + L_{G_2}$	$\|A_{CT}\|L_{G_1} + L_{G_2}$	$(2\|A_{CT}\| + 1)L_{G_1} + L_{G_2}$	$(2\|A_{CT}\| + 1)L_{G_1} + L_{G_2}$	$\|A_{CT}\|L_{G_1} + L_{G_2}$

Table 4: Performance comparison

Item	ABE	KP-ABE	CP-ABE	ABE with non-monotonic	HABE
Encryption	$\|A_{CT}\|G_1 + 2G_2$	$\|A_{CT}\|G_1 + 2G_2$	$(2\|A_{CT}\| + 1)G_1 + 2G_2$	$(2\|A_{CT}\| + 1)G_1 + 2G_2$	$\|A_{CT}\|G_1 + G_2$
Decryption	$dC_e + 2dG_2$	$\|A_{CT}\|C_e + 2\|m\|G_2$	$2\|A_U\|C_e + (2\|m\| + 2)G_2$	$(3\|A_{CT}\| + 2)C_e + G_2$	$3\|A_U\|C_e + (\|A_U\| - 1)G_2$
Policy	Threshold	AND, OR, threshold	AND, OR, threshold	AND, OR, NOT, threshold	AND, OR, threshold
Based on	ABE	KP-ABE	CP-ABE	KP-ABE	CP-ABE

Preventing the problem of illegal key sharing among users is difficult to solve, because it is hard to trace who shares the key. So almost all ABE schemes that we introduce cannot achieve two criteria.

3.2 Performance Analysis

L_{G_1} denotes the bit-length of element in G_1, L_{G_2} denotes the bit-length of element in G_2, C_e denotes a pairing operation, G_1, G_2 denotes two bilinear group operation, m denotes least node of the tree which can satisfy the access structure, and $|*|$ denotes the number of the element $*$. Table 3 and Table 4 list the comparison result [29]. We can found out, the length of user's private key and the ciphertext are corresponding to the number of attributes; if the number of attributes is too many. the length would increase. Moreover, the length of user's private key in ABE with a non-monotonic access structure scheme is more than other schemes, because the component of the user's private key in this scheme is including non-negated and negated attributes. Besides, we can find out in Table 4; if the scheme is based on the CP-ABE scheme, the decryption computation time is more than basic ABE scheme and KP-ABE scheme. In addition, the policy in ABE with a non-monotonic access structure is different, because it can use the negated word to describe attributes. But it causes a problem that the length of user's private key is longer than other schemes.

4 Conclusions

In this paper, we survey five different attribute-based encryption schemes: ABE, KP-ABE, CP-ABE, ABE with non-monotonic access structure, and HABE, and illustrate their schemes and compare them. These schemes can be classified according to their access policy. The access policy in the user's private key is KP-ABE, and the access policy in the encrypted data is CP-ABE. Besides, we can find these schemes that are hard to satisfy user accountability. Moreover, the access structure is pre-defined in these schemes; if a new user wants to access data and his attributes are not in the access structure, these encrypted data will be re-generated.

Thus, based on the discussion above, these existing attribute-based encryption schemes have properties: (1) These schemes are encrypted with attributes, so a data owner just needs to predefine these attributes that he would use, he doesn't need to care about the number of users in the system; (2) Each attribute has public key, secret key, and a random polynomial, so different users cannot combine their attributes to recover the data, and different users cannot carry out collusion attacks; (3) Only the user who possesses the authorized attributes can satisfy the access policy to decrypt data; (4) The access policy contains a boolean formula such as AND, OR et al. which can let the access structure be flexible to control users' access. However, almost all schemes exist that the authority is used to generate keys. Since these schemes contain the authority that just suits the private cloud environments, the authority should be removed in the fu-

ture. Furthermore, ABE schemes (like KP-ABE or CP-ABE scheme) are generally applied in the field of proxy re-encryption.

Acknowledgments

This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC101-2221-E-030-018.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50–58, 2010.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 321V334, 2007.

[3] V Bozovic, D Socek, R Steinwandt, and V. I. Villanyi, "Multi-authority attribute-based encryption with honest-but-curious central authority," *International Journal of Computer Mathematics*, vol. 89, pp. 3, 2012.

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

[5] M. Chase and S. 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*, pp. 121–130, 2009.

[6] C. C. Chang, I. C. Lin, and C. T. Liao, "An access control system with time-constraint using support vector machines," *International Journal of Network Security*, vol. 2, no. 2, pp. 150–159, 2006.

[7] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the ACM conference on Computer and communications security*, pp. 456–465, 2007.

[8] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and Athena Vakali, "Cloud computing: Distributed internet computing for it and scientific research," *IEEE Internet Computing*, vol. 13, pp. 10 –13, 2009.

[9] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proceedings of the Information Security Practice and Experience*, pp. 13–23, 2009.

[10] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proceedings of the ICALP*, pp. 579–591, 2008.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, 2006.

[12] M. S. Hwang and I. C Lin, "Introduction to Information and Network Security (4ed, in Chinese)," in *Mc Graw Hill. In Taiwan*, 2011.

[13] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," *Information Security Applications*, vol. 5932 of LNCS, pp. 309–323, 2009.

[14] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Proceedings of the Information Security Practice and Experience*, pp. 1–12, 2009.

[15] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proceedings of the 14th international conference on Financial cryptograpy and data security*, pp. 136–149, 2010.

[16] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," *Advances in Cryptology V EUROCRYPT*, vol. 6110 of LNCS, pp. 62–91, 2010.

[17] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," *Information on Security*, vol. 5735 of LNCS, pp. 347–362, 2009.

[18] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 343–352, 2009.

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

[20] Q Li, H. Xiong, F. Zhang, and S. Zeng, "An expressive decentralizing KP-ABE scheme with constant-size ciphertext," *International Journal of Network Security*, vol. 15, no. 3, pp. 161–170, 2013.

[21] Q. Liu, C. C. Tan, J. Wu, and GuojunWang, "Reliable re-encryption in unreliable clouds," in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–5, 2011.

[22] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Information Sciences. In Press*, 2012.

[23] S. Muller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Proceedings of ICISC*, pp. 20–36, 2008.

[24] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proceedings of the Applied Cryptography and Network Security*, pp. 111–129, 2008.

[25] 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.
[26] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 195–203, 2007.
[27] A. Sahai and B. Waters, "Fuzzy identity based encryption," *Advances in Cryptology V EUROCRYPT*, vol. 3494 of LNCS, pp. 457–473, 2005.
[28] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612–613, 1979.
[29] J. S. Su, D. Cao, X. F. Wang, Y. P. Su, and Q. L. Hu, "Attribute-based encryption schemes," *Journal of Software*, vol. 6, pp. 1299–1315, 2012.
[30] Q. Tang and D. Ji, "Verifiable attribute-based encryption," *International Journal of Network Security*, vol. 10, no. 2, pp. 114–120, 2010.
[31] S. F. Tzeng, C. C. Lee, and T. C. Lin, "A novel key management scheme for dynamic access control in a hierarchy," *International Journal of Network Security*, vol. 12, no. 3, pp. 178–180, 2011.
[32] G. Wang, Q. Liu, and J.Wu, "Hierachical attibute-based encryption for fine-grained access control in cloud storage services," in *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 735–737, 2010.
[33] G.Wang, Q. Liu, J.Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," *Computer & Security*, vol. 30, pp. 320–331, 2011.
[34] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," *Public Key Cryptography V PKC*, vol. 6571 of LNCS, pp. 53–70, 2011.
[35] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of IEEE INFOCOM*, pp. 534–542, 2010.
[36] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261–270, 2010.

Cheng-Chi Lee received the Ph.D. degree in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently an Associate Professor with the Department of Library and Information Science at Fu Jen Catholic University. Dr. Lee is currently as an editorial board member of International Journal of Network Security and Journal of Computer Science. He also served as a reviewer in many SCI-index journals, other journals, other conferences. His current research interests include data security, cryptography, network security, mobile communications and computing, wireless communications. Dr. Lee had published over 100+ articles on the above research fields in international journals.

Pei-Shan Chung received her B. M. in information Management from Chung Yuan Christian University, Jungli, Taiwan, ROC, in 2010. She is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. Her research interests include information security, cloud computing, and cryptography.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China (ROC), in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He is currently a professor of the department of Management Information System, National Chung Hsing University, Taiwan, ROC. He obtained 1997, 1998, 1999, 2000, and 2001 Outstanding Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published 170+ articles on the above research fields in international journals.