Research on Attribute Based Encryption Schemes and Its Advanced Functions

Kunlun Gao[1], Xingkun Xu[1], Chi Chen[2], Wei Yuan[2]*

[1] State Grid Information System Security Lab., Information & Communication Dept., Information Space Security Technology Section, Beijing, China.
[2] SKLOIS, IIE, CAS, Beijing, China.

* Corresponding author. Tel.: 008618910673762; email: yuanwei@iie.ac.cn

Manuscript submitted September 10, 2014; accepted February 3, 2015.
doi: 10.17706/ijcce.2015.4.3.145-155

Abstract: In this paper, we summarize the basic schemes of attribute based encryption and analyze the differences and relationships of their structures. Further, we point how to add new functions to these basic ABE schemes, such that new functions can be combined into basic algorithms. Then we introduce and analyze some existing ABE schemes with functions of hidden policy, multi-authority, and traitor tracing. We analyze the characteristic of these schemes and show their embedded methods in detail.

Key words: Attribute based encryption, hidden policy, multi-authority, traitor tracing.

1. Introduction

With the development of computer networks and modern communication technology, dissemination of information becomes faster and more convenient. People are easy to communicate with others in transnational manner. Accordingly, a new challenge on how to protect the security of information communication and storage should be overcome.

The core technology to protect information security is encryption. An encryption scheme usually consists of two algorithms: encrypt and decrypt. The encrypt algorithm transform a plaintext into a ciphertext with a key, and the decrypt retransform the ciphertext back to the plaintext. The condition to run the decrypt algorithm owns a league decrypt key. In traditional encryption schemes, the key used in encrypt algorithm is the same as in decrypt algorithm. When we use these algorithms, the encryptor and decryptor should share the same key, and this key cannot be gained by others. Thus, the encryptor needs to send the key to the decryptor. To protect the security of the key, a secure channel between them is necessary. The costs to achieve this condition are very large in reality and sometime nearly hard to reach.

To overcome the shortages in this model, Diffie and Hellman [1] proposed the public key cryptography (PKC). In PKC, encrypt algorithm uses different key with decrypt algorithm. The key used in encrypt algorithm can be open, and a one-way trapdoor function associates the key used in decrypt algorithm to it, such that anyone cannot compute the decrypt key according to the encrypt key. Only the owner of the decrypt key can decrypt the ciphertext encrypted by the encrypt key. As a result, the encryptor does not need to send encrypt key in secure manner. Later, the first practical public key scheme, RSA [2], was proposed by Rivest, Shamir and Adleman, and many public key schemes based on different computational problem were proposed, subject to Rabin scheme [3], ElGamal scheme [4], and Elliptic curve (ECC) scheme [5].

Although PKC eliminates secure channel between encryptor and decryptor, it brings a new problem: how to confirm the owner of a encrypt key. If an attacker pretends him to be a normal decryptor and give his encrypt key to others, no method can differ him. Thus, in practical PKC systems, we need a trusted method to bind a user identifier with his encrypt key. To achieve this goal, Kohnfelder [6] proposes certificate based scheme. A certificate authority (CA) is introduced as a third party trusted by all users in the system. CA opens its encrypt key and signs a user's identifier and associated encrypt key as his certificate. Anyone wants to communicate with that user needs to get corresponding certificate to confirm his real identity. However, the storage, distribution, query, validation to certificates need a lot of resources in certificate base systems. To simplify this system, Shamir [7] proposed identity based cryptography (IBE). IBE also assumes that a trusted third party (called PKG) exists in the system. The user identifier or the hash value of that identifier is his encrypt key. The decrypt key is computed based on the user identifier by PKG. Therefore, we do not need to generate certificates to bind identifier and encrypt key. The overheads on managing certificates can be eliminated.

Although IBE was proposed in 1984, Shamir did not give useable scheme but pointed out that RSA cannot construct IBE. The first practical IBE scheme was proposed in 2001 by Boneh and Franklin [8]. This scheme first introduces bilinear pairing into public key system, and can be proved CPA secure under RO model [9]. Then, bilinear pairing is widely used to construct IBE schemes with different additional functions. IBE greatly reduces the costs on certificates and has been used in many applications. In modern distributed networks, users usually dynamically join or leave the network. Encryptor cannot get all users' identifiers in advance.

To meet new application requirements, Sahai and Waters [10] proposed attribute based encryption (ABE). In ABE, a user is identified by a set of attributes but not a single identifier. If a user satisfies parts of required attributes, he can decrypt the ciphertext. We can image that many users may meet same attribute set. So the encryptor is able to send secret information to many decryptors one time. For example, considering following scenario. A company includes for departments: human resource department, research and development department, production department, and sales department. Each department has a manager and some staffs. If the general manager of that company needs to send a sensitive e-mail to each one of the human resource department, he needs to generate different ciphertext for each of them under IBE scheme. In addition, he may not know some of staffs' identity. If he encrypts the sensitive e-mail under ABE, only one ciphertext encrypted by the attribute 'human resource department' is needed. Thus, ABE is more flexible in practical applications, subject to remote file management, broadcast encryption, pay-tv.

Starting from Sahai and Waters' research, the concept of attribute is introduced into cryptography. User can be described with a series of attributes, such as sex, age, height, weight, position. Meanwhile, ciphertext is also described by some attributes. The match of user and ciphertext can be defined in more flexible manner. As a result, ABE becomes a hot research area in cryptography rapidly.

Goyal *et al*. [11] first classified ABE into two kinds: key-policy (KP) ABE and ciphertext-policy (CP) ABE. Policy can be regarded as a threshold to restrain attributes. For example, male, 30 years old, and manager are three attributes. Policy = {"male" and "manager"} represents that the manager of male can decrypt ciphertext. If the policy is combined with the ciphertext and the user decrypt key is described by a set of attributes, that scheme is CP-ABE. Otherwise, if the policy is used to generate decrypt key and the attributes set is used to generate the ciphertext, that scheme is KP-ABE. Reference [11] supports more flexible policy. That is, the policy can be described by multiple attributes connected with "and" and "or". Ostrovsky *et al.* [12] further introduced "not" into KP-ABE. Bethencourt *et al*. [13] implement CP-ABE, which supports "and" and "or". Cheung *et al*. [14] constructed an efficient CP-ABE that only support "and" operation in policy. Later, many kinds of ABEs with different functions are proposed.

In this paper, we summarize the basic schemes of attribute based encryption and analyze the differences and relationships of their structures. Further, we point how to add new functions to these basic ABE schemes, such that new functions can be combined into basic algorithms. Then we introduce and analyze some existing ABE schemes with functions of hidden policy, multi-authority, and traitor tracing. We analyze the characteristic of these schemes and show their embedded methods in detail.

2. Analysis on Basic ABEs

In this section, we list the model of initial ABE, KP-ABE, and CP-ABE. Then we analyze their relationships and differences. Finally, we summarize the matters to notice when we design similar schemes with other function.

2.1. Model of ABE

ABE has two main forms: CP-ABE and KP-ABE. Their differences are the positions of the policy and attributes set.

1) **KP-ABE**. A key-policy attribute based encryption scheme includes following four algorithms:
   - **Setup** (λ)→(PK, MK). The setup algorithm takes as input a security parameter λ. It outputs the public parameters PK and the master secret key MK.
   - **Encrypt** (PK, M, S)→CT. The encryption algorithm takes as input the public parameters PK, a message M and a set of attributes S. It outputs a ciphertext CT associated with the attributes set.
   - **KeyGen** (MK, $\mathbb{A}$)→SK. The key generation algorithm takes as input the master secret key MK and a policy $\mathbb{A}$. It outputs a private key SK associated with the policy.
   - **Decrypt** (SK, CT)→M. The decryption algorithm takes as input a private key SK associated with policy $\mathbb{A}$ and a ciphertext CT associated with attribute set S. It outputs a message M if S satisfies $\mathbb{A}$ or an error message $\perp$ otherwise.

The correctness property of KP-ABE requires that for all sufficiently large λ, all (PK, MK) $\in$ Setup (λ), all SK $\in$ KeyGen (MK, $\mathbb{A}$), and all CT $\in$ Encrypt (PK, M, S), if S satisfies $\mathbb{A}$, then Decrypt (SK, CT) outputs M.

2) **CP-ABE**. A ciphertext-policy attribute based encryption scheme also includes following four algorithms:
   - **Setup** (λ)→(PK, MK). The setup algorithm takes as input a security parameter λ. It outputs the public parameters PK and the master secret key MK.
   - **Encrypt** (PK, M, $\mathbb{A}$)→CT. The encryption algorithm takes as input the public parameters PK, a message M and a policy $\mathbb{A}$ over the universe of attributes. It outputs a ciphertext CT associated with the policy.
   - **KeyGen** (MK, S)→SK. The key generation algorithm takes as input the master secret key MK and a set of attributes S. It outputs a private key SK associated the attributes set.
   - **Decrypt** (SK, CT)→M. The decryption algorithm takes as input a private key SK associated with attributes set S and a ciphertext CT associated with policy $\mathbb{A}$. It outputs a message M if S satisfies $\mathbb{A}$ or an error message $\perp$ otherwise.

The correctness property of CP-ABE requires that for all sufficiently large λ, all (PK,MK) $\in$ Setup (λ), all SK $\in$ KeyGen (MK, S), and all CT $\in$ Encrypt (PK, M, $\mathbb{A}$), if S satisfies $\mathbb{A}$, then Decrypt (SK, CT) outputs M.

2.2. Basic ABE Schemes

1) Basic ABE. Suppose G_1 is a bilinear group with prime order p, g is a generator of G_1. $e: G_1 \times G_1 \to G_2$ is the bilinear pairing on G_1.

$$\Delta_{i,S}(X) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$$

Which is Lagrange parameter, S represent a set, whose elements belong to Z_p. U represents the attributes universe. The basic ABE scheme is as follows:

- **Setup**. Select random numbers $t_i \in Z_p$, and computes public parameters PK $= \{T_1 = g^{t_1}, \cdots, T_{|U|} = g^{t_{|U|}}\}, Y = e(g,g)^y$, and master secret key MK $= \{t_1, \cdots, t_{|U|}\},\ y$.
- **KeyGen.** For an attributes set $\omega \subseteq U$, randomly select a (d-1)-dimension polynomial $q(x)$, such that $y = q(0)$. Then compute decrypt key $D = \{D_i\}_{i\in\omega}$, $D_i = g^{q(i)/t_i}$
- **Encrypt**. For an attributes set $\omega' \subseteq U$, and a plaintext $M \in G_2$, select random number $s \in Z_p$, compute the ciphertext $E = (\omega', E' = MY^s, \{E_i = T_i^s\}_{i\in\omega'})$.
- **Decrypt**. If the set $S = |\omega \cap \omega'| \geq d$, then select d elements from S, based on Lagrange' interpolation formula, decryptor computes $M = E'/\sum_{i\in S}(e(D_i, E_i))^{\Delta_{i,S}(0)}$
- KP-ABE. The concepts and notations are the same as the basic ABE. KP-ABE is described as follows:
- **Setup**. Select random numbers $t_i \in Z_p$, and compute public parameters PK $= \{T_1 = g^{t_1}, \cdots, T_{|U|} = g^{t_{|U|}}\}, Y = e(g,g)^y$, and master secret key MK $= \{t_1, \cdots, t_{|U|}\}, y$.
- **KeyGen**. Organize the policy as a tree. For each non-leaf node x in the policy tree, select a d_x dimension polynomial q_x, the value of d_x is the threshold value of that node minus one. Suppose the threshold value of a node x is k_x, we know that $d_x = k_x - 1$. For the root node r, we have $y = q_r(0)$. For other non-leaf nodes, we have $q_x(0) = q_{\text{parent}}(\text{index}(x))$. Parent represents the parent node of node x, index associates the order of the children node. For the leaf nodes, compute the associated decrypt key $K_x = g^{q_x(0)/t_i}$, $i = att(x)$.
- **Encrypt**. For an attributes set $\omega' \subseteq U$, and a plaintext $M \in G_2$, select a random number $s \in Z_p$, compute the ciphertext $E = (\omega', E' = MY^s, \{E_i = T_i^s\}_{i\in\omega'})$.
- **Decrypt**. For all leaf nodes, decryptor computes $D_x = e(K_x, E_i) = e(g,g)^{sq_x(0)}$, For the non-leaf nodes, based on the return value of its children, decryptor computes the values from down to top:

$$F_x = \sum\nolimits_{z\in S'_x} (F_z)^{\Delta_{i,S'_x}(0)} = e(g,g)^{sq_x(0)}$$

$i = \text{index}(z)$, $S'_x = \{\text{index}(z): z \in S_x\}$. Finally, compute $e(g,g)^{sy}$ with Lagrange interpolation formula, and get the plaintext $M = E'/\,e(g,g)^{sy}$

2) CP-ABE. The concepts and notations are the same as the basic ABE. CP-ABE is described as follows:

- **Setup**. Select random numbers $\alpha, \beta \in Z_p$, and compute public parameters PK $= (g, h = g^\beta,\ h = g^{1/\beta}, e(g,g)^\alpha)$, and master secret key MK $= (\beta, g^\alpha)$.
- **Encrypt**. Organize the policy as a tree. To encrypt a plaintext $M \in G_2$. For each non-leaf node x in the policy tree, select a d_x dimension polynomial q_x, the value of d_x is the threshold value of that node minus one. Suppose the threshold value of a node x is k_x, we know that $d_x = k_x - 1$. For the root node r, select a random number $s \in Z_p$, such that $s = q_r(0)$. For other non-leaf nodes, compute $q_x(0) = q_{parent}(\text{index}(x))$. parent represents the parent node of node x, index associates the order of the children node. Finally, compute the ciphertext

$$CT = (\omega, C = Me(g,g)^{\alpha s}, C_0 = h^s,$$
$$\forall\ i \in \omega: C_i = g^{q_i(0)}, C'_i = H(att(y))^{q_i(0)})$$

- **KeyGen**. For an attributes set S, and, select a random number $r \in Z_p$, for each attribute $j \in S$, select a random number $r_j \in Z_p$, and compute the decrypt key as:

$$SK = (S, D_0 = g^{(\alpha+r)/\beta}, \forall\, j \in S{:}\, D_j = g^r H(j)^{r_j}, D_j' = g^{r_j})$$

- **Decrypt**. For all leaf nodes, decryptor computes

$$D_x = \frac{e(\,D_i, C_x)}{e(D_i', C_x')} = \ e(g,g)^{rq_x(0)}$$

For the non-leaf nodes, based on the return value of its children, decryptor computes the values from down to top:

$$F_x = \sum\nolimits_{z \in S_x'} (F_z)^{\Delta_{i,S_x'}(0)} = \ e(g,g)^{rq_x(0)}$$

$i = \mathrm{index}(z)$, $S_x' = \{\mathrm{index}(z){:}\, z \in S_x\}$. Finally, compute $e(g,g)^{rs}$ with Lagrange' interpolation formula, and get the plaintext $M = C/(e(\,C_0,\ D_0)/\ e(g,g)^{rs})$.

2.3. Analysis of the Basic ABEs

We can see that the setup algorithm and encryption algorithm of the initial ABE and KP-ABE are nearly the same. Their differences are mainly in the key generation algorithm and decryption algorithm. KP-ABE introduces a new tree structure. The leaf nodes are given same decrypt key as in initial ABE, and then the non-leaf nodes further abstract the policy associated key components from the key components of these leaf nodes in hierarchical manner. The decryption algorithm can be regarded as the reverse process of the key generation. Thus, KP-ABE actually recursive generates decrypt associated to a user-drawn policy.

CP-ABE follow the tree based structure of KP-ABE. The difference is the recursive tree is embedded into the encryption algorithm. Viewing from the decryption algorithms, CP-ABE also has different points with KP-ABE. It needs two components in both decrypt key and ciphertext. KP-ABE only needs one component. The reason is that the secret value $y = q_r(0)$ is selected by authority as part of master secret key in KP-ABE, and y is constant for different users. However, $s = q_r(0)$ is selected by the encryptor in CP-ABE, and it changes each time the encryption is executed. Thus, to design CP-ABE, author introduce two constants C_0 and D_0. The other parts are the same as in KP-ABE.

When we add new functions of ABE, we have two notices to consider:

1) Whether the tree based structure is the best way to achieve KP-ABE or CP-ABE, can we further develop new functions based on this structure.
2) We may use more than two components to achieve new functions.

3. ABE with Advanced Functions

ABE is a powerful cryptographic tool and suitable for network access control in cloud environment. Thus, it has become the current research focus in cryptography. Many scholars add advance functions on ABE to fit a variety of practical applications.

We summarize these advanced functions on ABE and analyze how these functions are integrated with the basic ABEs. In this paper, we mainly focus on adding functions of hiding encryption policy, multi-authority, and traitor tracing to ABE.

3.1. ABE with Hidden Encryption Policy

Encryption policy is to describe which recipients can decrypt received ciphertext with own decrypt key, and which recipients cannot decrypt. Hiding this policy means that each recipient does not know whether other recipients can decrypt the ciphertext he received. It can be considered as a recipient-anonymous targeted broadcast encryption. Many research papers [15], [16] has contributed to this area. We introduce reference [17] and analyze how this scheme hides the encryption policy from its basic scheme [14].

Suppose $N = \{1, \cdots, n\}$ represents the attribute universe. G_1 is a bilinear group with prime order p, g is a generator of G_1. $e: G_1 \times G_1 \to G_2$ is the bilinear pairing on G_1. Following algorithms describe a CP-ABE scheme that hides the encryption policy.

- **Setup**. Select random numbers $\omega \in Z_p$, for each attribute i in N, select random numbers $\left\{a_{i,t}, b_{i,t} \in Z_p\right\}_{1 \le t \le n_i}$ and random points $\left\{A_{i,t} \in G_1\right\}_{1 \le t \le n_i}$. The public parameters PK includes $G_1, G_2, p, Y, g, e, \{\{A_{i,t}^{a_{i,t}}, A_{i,t}^{b_{i,t}}\}_{1 \le t \le n_i}\}_{1 \le t \le n}$, and the master secret key MK is $\omega, \{\{a_{i,t}, b_{i,t}\}_{1 \le t \le n_i}\}_{1 \le t \le n}$.
- **KeyGen.** Suppose $L = [L_1, \cdots, L_n] = [v_{1,t_1}, \cdots, v_{1,t_n}]$ represents the attributes set of a user. For each attribute i in L, select a random number $s_i, \lambda_i \in Z_p$. Then compute $s = \Sigma_{i=1}^{n} s_i$, $D_0 = g^{\omega - s}$. When $L_i = v_{i,t_i}$, compute $[D_{i,0}, D_{i,1}, D_{i,2}] = [g^{s_i}(A_{i,t_i})^{a_{i,t_i} b_{i,t_i} \lambda_i}, g^{a_{i,t_i} \lambda_i}, g^{b_{i,t_i} \lambda_i}]$. Finally, the decrypt key of that user is generated as follows $\mathrm{SK}_L = D_0, \{D_{i,0}, D_{i,1}, D_{i,2}\}_{1 \le i \le n}$.
- **Encrypt.** For a encryption policy represented by $W = [W_1, \cdots, W_n]$, and a plaintext $M \in G_2$, select a random number $r \in Z_p$, and compute $C = MY^r$ and $C_0 = g^r$. Later, for each attribute i, select a random number $\{r_{i,t} \in Z_p\}_{1 \le t \le n_i}$, and compute $\{C_{i,t,1}, C_{i,t,2}\}_{1 \le t \le n_i}$ as follows: If $v_{i,t} \in W_i$, we have $[C_{i,t,1}, C_{i,t,2}] = [(A_{i,t_i}^{b_{i,t_i}})^{r_{i,t_i}}, (A_{i,t_i}^{a_{i,t_i}})^{r - r_{i,t_i}}]$. Otherwise, if $v_{i,t} \notin W_i$, pad $[C_{i,t,1}, C_{i,t,2}]$ with random numbers. Finally, ciphertext is generated as follows. $\mathrm{CT} = C, C_0, \{\{C_{i,t,1}, C_{i,t,2}\}_{1 \le t \le n_i}\}_{1 \le t \le n}$.
- **Decrypt**. After receiving the ciphertext that does not include the policy W, decryptor tries to decrypt the ciphertext as follows: For each attribute i, when $L_i = v_{i,t_i}$, compute $[C_{i,1}^{'}, C_{i,2}^{'}] = [C_{i,t_i,1}, C_{i,t_i,2}]$. If the attribute set L satisfied the policy W, decryptor is able to get the plaintext

$$M = \frac{C \prod_{i=1}^{n} e(C_{i,1}^{'}, D_{i,1}) e(C_{i,2}^{'}, D_{i,2})}{e(C_0, D_0) \prod_{i=1}^{n} e(C_0, D_{i,0})}$$

Comparing with its basic scheme, this scheme extends a vector in the key generation algorithm and encryption algorithm of the basic scheme to two matrices. The sum of the values in some line of one matrix equals to an element of the vector in basic scheme. The elements in another matrix are random. The encryption policy is the standard to choose element in which matrix. Thus, if the attribute list satisfies the hidden policy, the recipient can decrypt the ciphertext.

3.2. Multi-authority ABE

In commercial applications, many organizations may set up their own cloud system, and uses ABE to control users' access privileges. To connect these isolated systems, how to communicate between users of different authorities becomes a problem must be solved. We introduce and analyze a multi-authority scheme [18] to show how this function achieved on basic ABE.

Suppose G_1 is a bilinear group with prime order p, g is a generator of G_1. $e: G_1 \times G_1 \to G_2$ is the bilinear pairing on G_1. Each user owns a globe unique identifier GID besides their attributes. A multi-authority ABE scheme is described as follows:

- **Setup**. Suppose k authorities in the system. Select a random numbers $y_i \in Z_p$ for each user, and then compute $y_0 = \sum_{i=1}^{k} y_i$. Finally, open $Y = e(g,g)^{y_0}$, and keep $y_1, \cdots, y_k \in Z_p$ as the secret key of each authority.

- **Authority-KeyGen**. An authority k generates the master secret key set, $\{t_{k,i}\}$, associated to the attributes that can be granted to its users. Then it computes and publishes associated public key $\{T_{k,i} = g^{t_{k,i}}\}$.
- **User-KeyGen**. An authority k selects a random number $y_k \in Z_p$ and a d_k-1 dimension polynomial q, such that $q(0) = y_k$. For an attributes set A_k, authority generates $\{D_{k,i} = g^{q(i)/t_{k,i}}\}_{i \in A_k}$ for that user.
- **Encrypt**. For an attributes set $\omega' \subseteq U$ and plaintext $M \in G_2$, decryptor selects $s \in Z_p$, and computes the ciphertext $E = (\omega', E' = MY_0^s, \{E_{k,i} = T_{k,i}^s\}_{i\in\omega'})$.
- **Decrypt**. Decryptor selects d_k attributes of authority k, which are used in encrypt algorithm, and then computes $Y_k^s = e(g,g)^{q(0)s} = e(g,g)^{y_k^s}$ by Lagrange interpolation formula. Later, decryptor combines his decrypt key gained from each authority to compute $\Pi_{i=1}^{k} Y_i^s = Y_0^s$. Finally, decryptor gets plaintext $M = E / Y_0^s$.

In multi-authority scheme, the secret y in setup is separated into k shares, and each share is allocated to one authority. Then each authority generate its public parameters, distributes decrypt key for its users with its secret share. The encryption algorithm and decryption algorithm are similar with basic ABE, but following two points should be noticed:

In practical applications, since each y_i has been selected, all authorities, which want to connect with others, need to share its secret with others.

An obvious difference between multi-authority scheme and basic scheme is the global unique identifier GID. It means that two user must have different identifier although they may have same name or other similar features. This requirement is easy to achieve. For example, two users are both called Tom. One is belongs to PKU, and another is come from THU. We can name the first one PKU|Tom, and identifies the second one THU|Tom. Then their identifier is different.

3.3. ABE with Traitor Tracing

In some special application scenarios, a decrypt key is only granted to one user, e.g. pay-tv. However, basic ABE does not consider the situation that user give his key to others. The traitor tracing is to find these users of giving own decrypt key to others.

Hinek *et al.* [19] introduce this concept and analyze how to prevent illegal key clone. Wang *et al.* [20] further research this problem based on common security code. We introduce [20] and analyze how it works on basic ABE.

Suppose G_1 and G_2 are bilinear groups with prime order p. $e: G_1 \times G_2 \to G_T$ is the bilinear pairing. All the elements in Z_p consist the attributes universe. A user at most owns n_1 attributes and N users in the system. $h: \{0,1\}^* \to Z_p^*$ is a hash function to map any string to attributes in the system. The system threshold value is d. Suppose the size of the input symbol is s, and each input string contains l characters. Thus, the length of an attribute is $n_2 = \lceil \log_2 s \rceil \cdot l$. π is a one-way permutation on $\{1, \cdots, N\}$, $\omega_r^{\pi(i)}$ can be computed and given to user with identifier id_i, r is a random string selected by trusted authority.

Suppose $cw \in \{0,1\}^{n_2}$ represents an attribute, cw_j is the j bit of *cw*. Define a vector $v = (v_i), i \in \{0, \cdots, n_2\}$, whose length is $(n_2 + 1)$. All the elements in this vector are selected from G_2. Suppose $u_i = \phi(v_i)$, we have Waters hash as follows

$$H(cw) = v_0 \prod_{j \in B} v_j$$

B represents the set, in which $cw_j = 1$. The above hash function can be simplified to

$$G(cw) = u_0 \prod_{j \in B} u_j = \varphi(H(cw))$$

The ABE with traitor tracing can be described as follows:

- **Setup**. Select a random element h from G_2, and compute its isomorphic element $g = \phi(h)$ in G_1. Select a random number $\alpha \in Z_p$, and then compute $g_1 = g^{\alpha}, h_1 = h^{\alpha}$. Next, select random element $h_2 \in G_2$, and compute $g_2 = \phi(h_2)$. Let N represent set $\{1, \cdots, n_1 + 1\}$, select $t_1, \cdots, t_{n_1+1}$ from G_2, for an attribute $i \in N$, compute $c_i = \phi(t_i)$. Define a function

$$T(x) = h_2^{x^{n_1}} \prod_{i=1}^{n_1+1} t_i^{\Delta_{i,N}(x)}$$

Suppose $K(x) = \phi(T(x))$. It is equivalent to define a function

$$K(x) = g_2^{x^{n_1}} \prod_{i=1}^{n_1+1} c_i^{\Delta_{i,N}(x)} = \varphi(T(x))$$

At last, public parameters PK and master secret key are generated as follows: $\mathrm{PK} = \{g, g_1, h_2, u = (u_0, \cdots, u_{n_2}), c_1, \cdots, c_{n_1+1}\}$, $\mathrm{MK} = \{\alpha, h, v, t_1, \cdots, t_{n_1+1}\}$.

- **KeyGen**. For an attributes set ω, randomly select a $(d$-1)-dimension polynomial $q(x)$, such that $\alpha = q(0)$. For each $i \in \omega$, select $r_i, r_i' \in Z_p$. Then compute the decrypt key $\{D_i\}_{i\in\omega}$, $\{d_i\}_{i\in\omega}$, and $\{d_i'\}_{i\in\omega}$ as follows:

$$D_i = h_2^{q(i)} T(i)^{r_i} H(cw)^{r_i'}$$

$$d_i = h^{r_i}$$

$$d_i' = h^{r_i'}$$

- **Encrypt**. For an attributes set ω', and a plaintext $M \in G_T$, select random number $t \in Z_p$, compute the ciphertext $C = (\omega', C_1 = g^t, C_2 = Me(g_1, h_2)^t, C_3 = \{B_i = K(i)^t\}_{i\in\omega'}, C_4 = \{u_j^t\}_{j=0,\cdots,n_2})$.

- **Decrypt**. If the set $S = |\omega \cap \omega'| \geq d$, decryptor computes

$$C_4' = C_4^{(0)} \prod_{j \in B} C_4^{(j)} = G(cw)^t$$

B represents the set of elements, which $cw_j = 1$ for each element *j*. Then select d elements from S, and compute

$$M = C_2 \prod_{i \in S} (\frac{e(B_i, d_i)e(G(cw)^t, d_i')}{e(C_1, D_i)})^{\Delta_{i,S}(0)}$$

- **Tracing**. This algorithm can only be executed by authority. Suppose this algorithm has a decoder $\mathbb{D}$ for attributes set ω. $\delta(\kappa)$ represents a non-negligible function of κ. $\mathbb{D}$ can decrypt ciphertext that encrypted under ω with probability of $\delta(\kappa)$. Let $C_4^{(i,j)}$ represent the $(\lceil log_2 s \rceil (i-1) + j)$ element of C_4. For each $1 \leq i \leq l$ and $1 \leq j \leq \lceil \log_2 s \rceil$, authority initials a counter $ctr_{i,j} = 0$, and runs following test repeatedly:

Select a plaintext *m*, and encrypt *m* with ω. Then it replaces $C_4^{(i,j)}$ with a random element in G_1. Next, it query $\mathbb{D}$ with the new ciphertext. If $\mathbb{D}$ answers m, $ctr_{i,j} = ctr_{i,j} + 1$. Finally, it reconstructs a n_2 bits string cw', and trace the identifier of the traitor with common security code.

This scheme first introduces a serial symbols relative to common security code and use asymmetric group to replace metric group. The other parts in setup algorithm are similar to the basic ABE. The key generation algorithm is a bit different. Common security code is embedded into user decrypt key, and a paired random number r_i' is selected and given to user in exponential form. Function $K()$ is used in C_3 and does not participate in decryption algorithm. The other parts are similar to the basic ABE. In tracing algorithm, authority can verify whether the decrypt key comes from its owner using C_3. We can see from the traitor tracing ABE that new function of tracing can be combined into ABE in a natural manner.

4. Conclusion and Future Works

In this paper, we summarize the basic schemes of attribute based encryption and analyze the differences and relationships of their structures. Further, we point how to add new functions to these basic ABE schemes, such that new functions can be combined into basic algorithms. Then we introduce and analyze some existing ABE schemes with functions of hidden policy, multi-authority, and traitor tracing. We analyze the characteristic of these schemes and show their embedded methods in detail. In future works, we will add new functions in our applications demand to basic ABE.

Acknowledgment

This paper is supported by natural science foundation of China (Grant No.61402468).

References

[1] Diffie, D., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22(6)*, 644-654.

[2] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystem. *Communications of the ACM*, *21(2)*, 120-126.

[3] Rabin, M. O. (1979). Digital signatures and public-key functions as intractable as factorization.

Proceedings of MIT Library for Computer Science.

[4] ElGamal, T. (1985). A Public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, *31(4)*, 469-472.

[5] Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, *48*, 203-209.

[6] Kohnfelder, L. M. (1978). Towards a practical public-key cryptosystem. B.S. thesis, MIT, Cambridge, MA.

[7] Shamir, A. (1984). Identity-based cryptosystems and signature schemes. *Advances in Cryptology CRYPTO* (pp. 47-53). Berlin: Springer.

[8] Boneh, D., & Franklin, M. (2001). Identity-based encryption from the Weil pairing. *Advances in Cryptology CRYPTO* (pp. 213-229). Berlin: Springer-Verlag.

[9] Bellare, M., & Rogaway, P. (1993) Random oracles are practical: A paradigm for designing efficient protocols. *Proceedings of the ACM CCS* (pp. 62-73).

[10] Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. *Advances in Cryptology EUROCRYPT* (pp. 457-473). Berlin: Springer, Aarhus, Denmark.

[11] Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of the ACM CCS* (pp. 89-98).

[12] Ostrovsky, R., Sahai, A., & Waters, B. (2007). Attribute-based encryption with non-monotonic access structures. *Proceedings of the ACM CCS* (pp. 195-203). Alexandria, Virginia, USA.

[13] Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-policy attribute-based encryption. *Proceedings of IEEE Symposium on Security and Privacy* (pp. 321–334).

[14] Cheung, L., & Newport, C. (2007). Provably secure ciphertext policy ABE. *Proceedings of the ACM CCS* (pp. 456-465).

[15] Boneh, D., & Waters, B. (2007). Conjunctive, subset, and range queries on encrypted data. *Proceedings of the TCC*: *LNCS 4392* (pp. 535-554). Springer.

[16] Katz, J., Sahai, A., & Waters, B. (2008). Predicate encryption supporting disjunctions, polynomial equations, and inner products. *Advances in Cryptology EUROCRYPT* (pp. 146-162). Springer.

[17] Nishide, T., Yoneyama, K., & Ohta, K. (2008). Attribute-based encryption with partially hidden encryptor-specified access structures. *Proceedings of the ACNS*: *LNCS 5037* (pp. 111-129). Berlin: Springer.

[18] Chase, M. (2007). Multi-authority attribute-based encryption. *Proceedings of the Fourth Theory of Cryptography Conference*.

[19] Hinek, M. J., Jiang, S., Safavi-Naini, R., & Shahandashti, S. F. (2008). Attribute based encryption with key cloning protection. Report 2008/478. from http://eprint.iacr.org/2008/478.

[20] Wang, Y. T., Chen, K. F., & Chen, J. H. (2001). Attribute based traitor tracing. *Journal of Information Science and Engineering*, *27(1)*, 181-195.

Kunlun Gao received the B.S degree from Jilin University, Changchun, China, in 1993. He received the M.S. and Ph.D. degrees from China Electric Science Research Institute, Beijing, China, in 1997 and 2012, respectively. He is the vice-director at the Department of Information and Communication, CEPRI and the head of Information Security Lab of State Grid Corporation of China (SGCC). His research interests are electric power system and automation, smart grid information security and computer applications.

He has presided and undertook five national and more than ten provincial technology projects in the field of information security, and he has won National Scientific and Technological Progress Award once, China Electric Power Science and Technology Award four times, Scientific and Technological Progress Award of

SGCC eight times, National Energy Science and Technology Progress Award once and the twelfth Information Industry Important Technological Invention Award once.

Xingkun Xu received the B.S and M.S. degrees from Beijing Institute of Technology, Beijing, China, in 2005 and 2007, respectively. In 2010, he received his Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China. He is now a senior engineer of State Grid Information System Security Lab., Beijing, China. His research interest includes the cloud security and database security.