# A Survey Of Attribute-based Encryption Schemes

Qxx

TABLE I
NOTIONS FOR COMPARISON

| Notions | Descriptions |
|---|---|
| $n$ | the number of the attributes in ciphertext |
| $\omega$ | the number of the attributes in secret key |
| $U$ | the number of the attributes in the whole system |
| $C$ | the number of nodes ( in access tree) or gates ( in circuits) |
| $k$ | the number of the authorities |
| $\|G_i\|$ | the size of the element in group $G_i$ |
| $t_{e_i}$ | the time cost of every exponential operation in $G_i$ |
| $t_p$ | the time cost of every pairing operation |

***Abstract*—The abstract goes here.**

***Index Terms*—The keywords goes here.**

## I. Introduction

introduction goes here.

## II. Definition

We first show the notions in our paper. Then formal framework of attribute based encryption (ABE) are given. Finally, based on different attack models, we describe the security definition for ABE.

### *A. Notations (by Lei Xu)*

Let $\mathbb{N}$ denote the set of natural numbers. if $\lambda \in \mathbb{N}$ then $1^\lambda$ denotes the string of $\lambda$ ones. Let $(y, z, \ldots) \leftarrow A(w, x, \ldots)$ or $A(w, x, \ldots) \rightarrow (y, z, \ldots)$ denotes the operation of running an algorithm $A$ with inputs $(w, x, \ldots)$ and output $(y, z, \ldots)$. We also list the notations appearing in our comparison on Table I.

### *B. Framework (by Xiping Zhang)*

Attribute Based Encryption can be divided into two categories: Key-Policy Attribute-Based Encryption(KP-ABE) and Cipertext-Policy Attribute-Based Encryption(CP-ABE). In a CP-ABE scheme,an access structure (i.e. policy) would be associated to each ciphertext, while a users private key would be associated with a set of attributes. In KP-ABE scheme,on the contrary,the access structure is specified in the private key, while the ciphertexts are simply labeled with a set of descriptive attributes.

**Definition 1** *An attribute-based encryption is defined by the following algorithms*

- **Setup**$(\lambda, U)$ This is a randomized algorithm that takes the inputs, security parameter $\lambda$ and system attribute set $U$. It outputs the public parameters $PK$ and a master key $MK$.
- **KeyGen**$(MK, PK, X)$ The key generation algorithm takes as input the master key $MK$, public parameters $PK$ and a permission $X$, It outputs a private key $SK_X$.
- **Encrypt**$(\mathcal{M}, PK, Y)$ This is a randomized algorithm that takes as input a message $\mathcal{M}$, the public parameters $PK$ and a ciphertext index $Y$. It outputs the ciphertext $CT_Y$.
- **Decrypt**$(PK, CT_Y, SK_X)$ The decryption algorithm takes as input the public parameters $PK$, a ciphertext $CT_Y$, the decryption key $SK_X$. It outputs the result of decrypt $\mathcal{M}$.

**correctness**. The correctness of a attribute based encryption scheme requires that the following equations hold with probability one:
**Decrypt**$(PK$,**KeyGen**$(MK, PK, X)$,**Encrypt**$(\mathcal{M}, PK, Y))$ $= \mathcal{M}$

It should be noted that the permission $X$ of CP-ABE is a set of attributes that describe the key while in KP-ABE is an access structure. The ciphertext index $Y$ is an access structure in CP-ABE and is a set of attribute in KP-ABE. In addition, for the sake of avoiding distinguishing KP-ABE with CP-ABE, denote $f_{as}(X, Y) = 1$ as that attributes satisfy access policy.

### *C. Security Model (by Lei Xu)*

According to the theory of provable security in public key encryption [1], security definitions are generally developed from the antagonistic relationship between the security goal and the adversary model possess the specific attack capacities. With semantic security ( i.e, the indistinguishability of encryptions, IND) goal and three attacks (chosen-plaintext attack (CPA), non-adaptive chosen-ciphertext attack (CCA1) and adaptive chosen-ciphertext attack (CCA2)). There are three security models usually taken into consideration: **IND-CPA** security, **IND-CCA1** security and **IND-CCA2** security in order of increasing strength [2]. Nevertheless, as for ABE, the present schemes offer only two security models, **IND-ABE-CPA** security [3] and **IND-ABE-CCA2** security [4], due to the fact that the direct construct of ABE schemes can usually be achieved the weakest **IND-ABE-CPA** security of above three securities, and that compound construct with a certain assistant technology, such as Canetti-Halevi-Katz approach [5] for stronger security, has already been competent for **IND-ABE-CCA2**.

Compared with normal public key encryption, ABE has some subtle variations for the reason of the fine-grained property mentioned in our introduction. Fine-grained property couludes two sides: granting differential access rights to a set of users and allowing flexibility in specifying the access rights

this is thanks

of individual users. Then both of the two sides naturally result in that it is most probable that the adversary can obtain the private keys for the many permission $X$( i.e, attribute sets for CP-ABE or access structure for KP-ABE) from other users or from the adversary himself. For example, there are two data encrypted to $S1$ and $S2$ respectively in a KP-ABE system. the adversary, who is qualified to data1 and not to data2, obtains the the private key for his access structure from himself, in order to break the data1 . Therefore, when the weakest **IND-CPA** of three security models is straightforward defined on ABE, the ABE scheme under normal **IND-CPA** is completely insecure. Implementing in ABE, other security models have the same problem in the same way. This issue bears some resemblance to the one of identity-based encryption [6]. So, for the most basic purpose that the security model can evaluate the security, the capacity to query private keys for the many permission $X$ must be given to the adversary in any attack model of ABE, and the private key generation oracle $\mathcal{O}_{keygen}$ is just used to simulate such query. Thus, different attack models depend on the different levels to access the decryption oracle $\mathcal{O}_{dec}$.

The most difference between ABE and IBE arises in the inputs of the private key generation oracle $\mathcal{O}_{keygen}$ and the decryption oracle $\mathcal{O}_{dec}$ in security model. In ABE $\mathcal{O}_{keygen}$ put the permission $X$ as input, but the ciphertext as the input of $\mathcal{O}_{dec}$ is associated with ciphertext index $Y$, instead of the same thing, identity strings ID in IBE [6], appearing the positions of $X$ and $Y$. The reason is that in IBE the encryption and key generation both operate on the identity strings ID, while ciphertext is generated from $Y$ and SK is the product of $X$ for ABE. Based on the discussion above, in following parts, we first describe the details of key generation oracle $\mathcal{O}_{keygen}$ and decryption oracle $\mathcal{O}_{dec}$ for ABE. Then, we will discuss different attack models in ABE on the basis of the availability of above two oracles in each phase of the security game. Finally, formal security definitions will be given.

*1) Oracles:* There are two oracles involved in the security games for ABE scheme, which can be queried by the adversary in an adaptive manner in the security game. The aim of these oracles defined here is to provide the adversary with the attack capacity in attack model.

Despite the capability to query private keys can be simulated by the private key generation oracle $\mathcal{O}_{keygen}$, accessing to this oracle without any restriction is not necessary for the adversary to obtain such capability. In an effort to avoid the trivial success for the adversary, which makes no sense for demonstrating the security, the definition of the private key generation oracle must contain the non-trivial restriction. Particularly, on condition that the challenge ciphertext index $Y^*$ associated with the challenge ciphertext has been submitted, private key generation oracle $\mathcal{O}_{keygen}$ is not allowed to issue queries for private keys for the permission $X_i$, where $f_{as}(X_i, Y^*) = 1$ for all $j$. Similarly, the decryption oracle $\mathcal{O}_{dec}$ for ABE have the analogous restriction. formal definition on $\mathcal{O}_{keygen}$ and $\mathcal{O}_{dec}$ are shown as follows.

- $\mathcal{O}_{keygen}$: On input a permission $X$, outputs a private key $SK_X \leftarrow$ **KeyGen**$(MK, PK, X)$. If the challenge ciphertext index $Y^*$ has not been submitted yet, the inputs $X$ do not have any constraint. Otherwise, those inputs $X_i$ satisfying $f_{as}(X_i, Y) = 1$ are invalid for $\mathcal{O}_{keygen}$.
- $\mathcal{O}_{dec}$: On input a ciphertext $CT_Y$ encrypted to $Y$, $\mathcal{O}_{dec}$ create a $X$, which make $f_{as}(X, Y) = 1$ hold. Then runs $SK_X \leftarrow$ **KeyGen**$(MK, PK, X)$ and $\mathcal{M} \leftarrow$ **Decrypt**$(PK, CT_Y, SK_X)$. Finally, outputs the $\mathcal{M}$. Note that the input can be the ciphertext encrypted to $Y^*$, while the challenge ciphertext $CT^*$ submitted is invalid input for $\mathcal{O}_{keygen}$.

*2) Attack Model:* For the briefness, We do not adopt the mnemonic way in [2] to describe the attack model for ABE. Additionally, due to the fact that the practical combinations of oracles in ABE is not so many, instead of the way by [7] to , We simply enumerate these reasonable combinations. We first assume $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively to be the sets of the $\mathcal{O}_{keygen}$ oracle and $\mathcal{O}_{dec}$ oracle that are available in **Phase** 1 and **Phase** 2. As the discussion at the beginning of this section, any security model should conclude the private key generation oracle $\mathcal{O}_{keygen}$, so $\mathcal{O}_1$ and $\mathcal{O}_2$ may be one of the two values, $\{\mathcal{O}_{keygen}\}$ and $\{\mathcal{O}_{keygen}, \mathcal{O}_{dec}\}$. The attack models have been enumerated on Table II.

TABLE II
DIFFERENT ATTACK MODELS FOR ABE

| $\mathcal{O}_1$ | $\mathcal{O}_2$ | Attack Models |
|---|---|---|
| $\{\mathcal{O}_{keygen}\}$ | $\{\mathcal{O}_{keygen}\}$ | CPA |
| $\{\mathcal{O}_{keygen}, \mathcal{O}_{dec}\}$ | $\{\mathcal{O}_{keygen}\}$ | CCA1 |
| $\{\mathcal{O}_{keygen}, \mathcal{O}_{dec}\}$ | $\{\mathcal{O}_{keygen}, \mathcal{O}_{dec}\}$ | CCA2 |

Although there has not been a present ABE scheme with CCA1 security, we also describe CCA1 attack model in order to deal with the situation that the ABE scheme under CCA1 attack without any assistant technologies appears.

*3) Security Definition:* The definition of security in ABE processes by incorporating the indistinguishability game and the family of the above-mentioned attack models. For indistinguishability game with respect to three kinds attack models, the only difference lies in whether or not $\mathcal{A}_1$ and $\mathcal{A}_2$ are given decryption oracles.

**Definition 2** *Let $\Sigma$ = (**Setup**, **KeyGen**, **Encrypt**, **Decrypt**) be a attribute-based encryption scheme, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a probabilistic polynomial-time adversary and $\mathcal{O}_1$ and $\mathcal{O}_2$ be the set of available oracles for $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. For the security parameter $\lambda \in \mathbb{N}$ and $atk \in$ {CPA, CCA1, CCA2}, the indistinguishability game is defined by the following experiment.*

$\mathbf{Exp}_{\Sigma,\mathcal{A},b}^{IND-atk}(\lambda)$:

$$\begin{aligned}
(MK, PK) &\leftarrow \mathbf{Setup}(\lambda, U);\\
(\mathcal{M}_0, \mathcal{M}_1, Y^*) &\leftarrow \mathcal{A}_1^{\mathcal{O}_1}(PK);\\
b &\leftarrow \{0, 1\};\\
CT^* &\leftarrow \mathbf{Encrypt}(PK, Y^*, \mathcal{M}_b);\\
b' &\leftarrow \mathcal{A}_2^{\mathcal{O}_2}(\mathrm{PK}, CT^*, \mathcal{M}_0, \mathcal{M}_1, Y^*).
\end{aligned}$$

where $\mathcal{O}_1$ and $\mathcal{O}_2$ are defined according to the attack model atk, as described on Table II.

**Definition 3** *For $\lambda \in \mathbb{N}$, the ABE scheme $\Sigma$ is said to be **IND-ABE-atk** secure if all probabilistic polynomial-time adversaries $\mathcal{A}$ have at most negligible advantage to win the indistinguishability experiment $\textbf{Exp}_{\Sigma,\mathcal{A},b}^{IND-atk}(k)$, where advantage of $\mathcal{A}$ is given by*

$$\begin{aligned}&\textbf{Adv}_{\Sigma,\mathcal{A},b}^{IND-atk}(\lambda)\\&=|Pr[\textbf{Exp}_{\Sigma,\mathcal{A},1}^{IND-atk}(t)=1]-Pr[\textbf{Exp}_{\Sigma,\mathcal{A},0}^{IND-atk}(t)=1]|.\end{aligned}$$

Different from the security models under three attack models, *selective security* or *full security* depends on whether or not the challenge ciphertext index $Y^*$ is committed by adversary before the stage **Setup** in $\textbf{Exp}_{\Sigma,\mathcal{A},b}^{IND-atk}(k)$.

Now we illustrate the defined security models with examples of representative ABE schemes. Although Goyal $et\ al.$ and Bethencourt $et\ al.$ content that their KP-ABE scheme [3] and CP-ABE scheme [8] respectively can be converted to **IND-ABE-CCA2** security by revising the Canetti-Halevi-Katz (CHK) approach [5], indeed, the two schemes themselves are **IND-ABE-CPA** security. Cheung and Newport [4] apply the CHK technique to construct a IND-ABE-CCA2-secure CP-ABE scheme. Given above IND-ABE-CCA2-secure ABE scheme involve one-time signatures due to the CHK technique, Liu $et\ al.$ [9] exploit Chameleon hash to achieve IND-ABE-CCA2-secure KP-ABE scheme without one-time signature. As mentioned above, as far as we know there has been not a ABE scheme just satisfying the **IND-ABE-CCA1** security. The concrete comparison of the representative ABE schemes involved in security notions can be found in the Table **??**.

## III. State-of-the-Art

### A. *the birth of ABE (by Lei Xu)*

Attribute-based encryption originates from fuzz identity based encryption [10] by Sahai and Waters. Fuzz identity based encryption supports decrypting by the attributes with selectable lower-limit number, due to use of the threshold secret sharing technology [11]. After that, beyond the threshold policy, much attention on general policy has been paid. The general policy can tie data owner with data user by a series of formalized constraint, which supports general relationship among attributes. Access control technology just performs well, and attribute-based encryption has perfectly formed only when this technology are put into use. More specifically, by using access control technology, ABE inherently includes two types of manners, KP-ABE and CP-ABE. Goyal *et al.* [3] propose the first KP-ABE scheme, while Bethencourt *et al.* [8] constructs the first CP-ABE scheme. These two schemes concretely achieve the access control, so we contend that they sign the birth of ABE. Since then, the researches on ABE tend to be booming, and we summarize the trunk directions of development on ABE, i.e, function, efficiency, security and expressions on access control.

### B. *efficiency (by Hao Zhang)*

ABE brings a new idea of encryption. Whereas in today's life it is not widely used, the traditional ABE encryption and decryption speed can not make people satisfied in nowadays big data situation on the Internet. Based on this dilemma, many new ABE schemes are proposed to improve the efficiency of ABE. The first is ABE with constant ciphertext length. The length of the ciphertext depends on the number of attributes in previous ABE schemes. Emura [12] first put forward the ABE with constant ciphertext length, unlike the previous ABE schemes they put the user attributes in one set, the whole set is used as the generation of ciphertext, so that fixed the ciphertext length. Then Herranz [13] propose the collusion-resistant ABE scheme with constant ciphertext length. Even if the ABE with constant ciphertext length is not secure enough, thus in 2013 Doshi [14] proposed a fully secure ABE with constant ciphertext length, that makes the ABE with constant ciphertext length can be in the security with a higher efficiency. However, in order to improve the efficiency of the ABE scheme, not only the constant ciphertext length of this scheme. Guo [15] in 2014 proposed a constant keys length rather than the ciphertext length of the ABE scheme makes ABE scheme can be applied to lightweight devices. In addition to constant ciphertext and constant keys way, there is another way to improve efficiency. Ibraimi [16] proposed a highly efficient CP-ABE scheme with "?" and "?" access structure. By operating on the access structure, decryption is used only once to improve efficiency. Attrapadung [**?**] proposed the size of the ciphertext and the size of the key tradeoff scheme can be carried out directly to improve the efficiency of ABE scheme. In the ABE scheme there is a common method to enhance efficiency, that is outsource. The ABE program outsourcing the complex encryption process and decryption process outsourcing to the cloud server.Let high-performance server to help us to compute the most complicated part of the scheme, and we only accept the results of the final operation out.

### C. *security (by Yi Wang)*

1:Survey on original ABE,and some articles,which have promoted in security,based on original ABE.

2:You describe these articles in time sequence.

3:Also need a table to compare security among articles mentioned by you.

### D. *function (by Lei Xu)*

Although original ABE schemes are of great significance as mentioned in our introduction, there are still some limitation for some specific scenario. For example, the principal of the university wants to send a encrypted file to those who satisfy following constraint: ((“school: EECS” and “career: teacher”) or (“career: administrator”)) and “age: 35-45”. Obviously, original ABE schemes are not able or inefficient to deal with above situation. So, some schemes with additional functions are proposed in order to support the various applications.

In this subsection, we survey the development on additional function compared with original ABE schemes. First constraint in original ABE schemes is that each attribute only has two state: a attribute is held by somebody or not. Fan *et al.* [18] straightforward propose the Arbitrary-State Attribute-Based

Encryption in 2014. In [18], “teacher” and “administrator” are two values of the only one attribute “career” in our example, i.e, the scheme has the function supporting attribute with many values. This kind function leads to the fact that the number of designed attributes somebody receives is comparable to the number of natural attributes it has. In fact, in 2009, Bobba *et al.* [19] have achieved multiple value assignment on an attribute, which arbitrary-state attribute is somewhat similar to.

Additionally, “numerical attribute”, “age” in our instance, means that the value of an attribute is numerical, i.e, we can compare those values using operations such us $\leq, >, \neq$ and so on. In the original CP-ABE scheme[8], “numerical attribute” and the operation on values has been posed, but not applied in the concrete scheme. Lang *et al.* [20] in 2013 achieves the construct of an ABE scheme supporting “numerical attribute” and the operation on its values. Having the function supporting “numerical attribute” and the operation on values, ABE schemes can achieve more effective data self-protection mechanisms in open environments such as Cloud computing [20].

Moreover, Weighted Attribute Based Encryption is proposed by Liu *et al.* [21] to deal with the situation that in some circumstances the attributes are not always in the same position, i.e, different attributes have different importance.

## IV. DESIGN PHILOSOPHY

some introduction goes here.

### A. Access Control (by Lei Xu)

As mentioned in introduction, “attribute” and “access control” are most essential difference of ABE scheme from other schemes. Before our formal discuss, we firstly introduce a new notion, qualification. Informally, the qualification signifies the process of obtaining the permission to decrypt. Then, we claim that “access control” is that if and only if some objects from user satisfy a certain rule from data owner, the user can be qualified. On the one hand, contrasted with “identity” in IBE, “attribute” in ABE is the least fineness which still has the function of objects on qualification. On the other hand, the rule mentioned above means access policy or access structure, which will be surveyed below in detail. Based above, the prerequisite of qualification in ABE can be explained as “attributes satisfy access structure ( or access policy)”, as many ABE schemes state [22][23][24][25][26].

For convenience, attributes the participant owns are usually collected in a set called attribute set. Moreover, the attribute set which satisfies specific access policy is called qualified attribute set. As for access policy, however, there are various expressions. We will survey on three main expressions in most present ABE schemes.

Before our survey, the conception, secret sharing, should be firstly introduced. To put it simply, secret sharing in ABE schemes is the operation that information distributed amongst attributes in each qualified attribute set achieve being reconstructed, while each or many but not all attributes in a qualified set (suppose these attributes can not be a collection of another qualified set) are of no use on their own. It is easy to find that achieving access control is just for the purpose realizing secret share in all ABE schemes, and more concretely obtaining the secret value, usually $s$, or an implicit expression on secret value. After acquiring $s$ in certain form, user can decrypt successful. For example, sometimes $A^{B \cdot s}$ is enough for decryption. Due to the fact that the ABE syntax has been provided, the discusses here is just up to gain of secret value $s$. For a clear statement, we also list several common notions as shown on Table III.

To put the statement “attributes satisfy access structure (or access policy)” into practice, Firstly, we can intuitionally enumerate qualified attribute sets, then the access structure $\mathbb{A}$ can be denoted as the collection of these sets. Therefore, the statement above can be interpreted as that a qualified attribute set belongs to the set $\mathbb{A}$. We call this expression as *enumeration*. As far as the easier comprehension of access structure as well as access control, *enumeration* performs well, but obviously this expression is not brief at all.

Then, new methods to express access structure are proposed by considering boolean formula, because boolean formula can directly point out the relations among attributes in access structure. These relations contain threshold, *AND*, *OR* and *NOT*. In fact, explorations on how to realize Boolean Formula in ABE schemes have been made since the time of birth of ABE [3][27][28], and so far the *access tree* and *circuits* have been accepted as the solution [3] [28].

More consideration that whether there is a black-box way to achieve access control is made by researches. In this way, inputs are associated with attributes, while output is the result that the policy is satisfied or not, which means that the way pays little attention to the concrete relationship among attributes. The boolean function, $f$: $\{1, 0\}^n \rightarrow (0, 1)$, is as theory model of this method. The input of the boolean function is a n-dimension vector, and each dimension is a boolean variable with the value of “1” or “0”. In ABE context, each value of boolean variable from input depends on whether user possess corresponding attribute. Then the output is also a boolean variable with the values of “1” or “0”. Analogously, the value of output depends on whether the attributes of user set satisfy the access structure. Linear secret-sharing scheme(LSSS) is used to realize Boolean Function in practical ABE [29].

In addition, monotonicity of access structure should be taken into account, and it will be discuss in our next. *access tree*, *circuits* and LSSS, three main expressions mentioned above are also surveyed in this subsection.

*1) Monotonicity:* Monotonicity is defined as follows:

**Definition 4 (Monotonicity[30])** *access structure $\mathbb{A}$ is a collection of sets, and $\mathbb{A}$ is monotone if $\forall$A,B, such that A is in $\mathbb{A}$ and A$\subseteq$ B ,then B is also in $\mathbb{A}$. Otherwise $\mathbb{A}$ is non-monotone.*

In Boolean Formula, monotone access structure don’t involve *NOT*, while the non-monotone access structure should support *NOT*.

In the access control of present ABE schemes, the majority of access structures are monotone, due to the fact that normal methods to express attribute in access structure, such as $\rho(i)$

TABLE III
SEVERAL COMMON NOTIONS

| Notions | Descriptions |
|---|---|
| $p$ | the order of prime group |
| $P = \{P_1, P_2 \cdots P_u\}$ | $P_j, j \in 1, 2 \cdots u$, denotes a selected random and unique number associated with the $j-th$ attribute<br>P denotes the set of $u$ numbers corresponding u attributes appearing access structure |
| $\gamma$ | qualified attribute set |
| $\mathbb{A}$ | access structure |
| $s$ | the secret to share |
| $X$ | the independent variable of polynomials |

As our discuss above, access control has various forms in expression, so does access structure. we let $\mathbb{A}$ be the substance of access structure for all expressions because of briefer formulation and easier comprehension of it. Also we will point out the consistency between $\mathbb{A}$ and other format of access structure

in LSSS and leaf node in access tree, only support the situation that each attribute is in a certain qualified set or not in, and don't support the one that an attribute is not in any qualified sets. However, it is necessary to non-monotone access structure that supporting each attribute being not in any qualified attribute sets. In order to address non-monotone access structure with mature technology to achieve monotone access structure, "negative attribute" is supposed ([27]). Based the comprehension of boolean formula "$B\ and\ \overline{A}$", main idea of this supposition is shown as follows. Firstly, attributes are divided into two state, positive and negative. If a user have a attribute $A$, he holds the positive state $A$, otherwise holds the negative state $-A$. Then, the formula of both privacy key and ciphertext has two options for every attribute, so that the boolean formula containing $\overline{A}$ can be expressed. Nevertheless, the main problem of this method inclines to inefficient. Two states of each attribute imply the number of attributes in the system will be doubled such us [31]. As for this problem, Ostrovsky *et al.* [27] propose a solution. In [27], each attribute has positive state, and negative states of some attributes are added according to specific systems. Additionally, the method in [27] can be also regard as a general transformation technology, and this technology can make non-monotone structure into the monotone format to satisfy the requirement by LSSS. Therefore, more efficient expressions only supporting monotonic access structure have possibility to address non-monotone access structure using this technology.

*2) expressions on access control:* In this part, we survey three main expressions on access control from most of present ABE schemes, *access tree*, *circuits* and *linear secret sharing scheme(LSSS)*. Unless stated otherwise, by an access structure we mean a monotone access structure. Moreover, the comparison among the three methods to achieve access control has been by us.

**Access Tree**

Let $\mathcal{T}$ be a tree as an access structure. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. The $num_x$ labels the number of children of a node $x$ and $k_x$ labels its threshold value. Define that $k_x = 1$ if the threshold gate is an OR gate and that $k_x = num_x$ if it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$.

For convenience of description on secret sharing, some extra function are defined here. Define function parent($x$) as the parent of the node $x$ in the tree, and the function $att(x)$ the attribute same as $P_i$ in Table III associated with the leaf node $x$ only if $x$ is a leaf node. The access tree $\mathcal{T}$ also defines an ordering between the children of every node, i.e, the children of a node are numbered from 1 to $num$, and the function $index(x)$, which returns a value associated with the node $x$, and the index values are uniquely assigned to nodes in the access tree for inputs of secret sharing in an arbitrary manner.

**Achieving Secret Sharing In Access Tree**

As far as satisfaction between attributes and policy, there are two parts. One part called set part holds the qualified attribute set, and another called structure part holds the access structure.

In access tree, structure part chooses a polynomial $q_x$ for each node $x$ by the degree of the polynomial and points with the number of one more than degree. These polynomials are chosen concretely in a top-down manner as follows. Firstly, beginning at the root node, for each node x, set the degree $d_x$ of the polynomial $q_x$ to be one less than the threshold value $k_x$ of that node, i.e, $d_x = k_x - 1$. Then, for the root node r, set $q_r(0) = s$ and $d_r$ other points of the polynomial $q_r$ randomly. After that, for other node x, set $q_x(0) = q_{parent(x)}(index(x))$ and choose $d_x$ other points randomly, too. Up to every leaf node in this way, we are finally aware of that the degree of each leaf node is 1, and it only has one point, i.e, each $q_x$ of leaf node has been determined as a constant by the polynomial of its parent.

We can see the secret value $s$ finally transmits to the constants in leaf nodes associating attributes in qualified set. In fact, process above is the reverse of secret sharing, the distribution for secret value $s$.

The set part holds the qualified attribute set as mentioned above, and we assume that the part is provided the constants corresponding attributes in the set of it own after the distribution for secret value, and that set part have known the structure of access tree. Label the access tree with root $r$ as $\mathcal{T}$, and the subtree of $\mathcal{T}$ rooted at the node x as $\mathcal{T}_x$. In this way, $\mathcal{T}$ is the same as $\mathcal{T}_r$. Every node have two case, being leaf node and non-leaf one. For each leaf node $x$, specific constant is given if and only if $att(x) \in \gamma$, and denote $\mathcal{T}_x(\gamma) = 1$, so the $q_x(X)$ can be calculated by the degree $d_x = 0$ and the point containing an arbitrary number coupled with constant. After that, set part can find a point in $q_{parent(x)}(X)$ containing $index(x)$ coupled with $q_x(0)$. where $X$ is shown on Table III. For each non-leaf node $x$, evaluate $\mathcal{T}_{x\prime}(\gamma)$ for all children $x\prime$ of node $x$, and if and only if at least $k_x$ children equal to 1, i.e. at least $k_x$ points of $q_x(X)$ can be obtained, $\mathcal{T}_x(\gamma) = 1$, so the $q_x(X)$ can be calculated by the degree $d_x = k_x - 1$ and optional $k_x$ points gotten from children. After that, set part can again find a point in $q_{parent(x)}(X)$ containing $index(x)$ coupled with $q_x(0)$. Process above continues until acquires the value of $q_r(0)$, the secret value $s$. Additionally, if an extra part without qualified set attempts to pursue the secret value, it can't go on in certain step of obtaining $q_x(X)$ due to the lack of some points.

**Some Analyses And Remarks On Access Tree**

Firstly, "attributes *satisfy* the policy" can be interpret on the expression of access tree as that leaf nodes have the consistency in logic with gates of tree layer by layer, such that finally $\mathcal{T}_r(\gamma) = 1$. So, $\mathbb{A}$ can be interpret here as the leaf node with the constraints from the structure of the tree. Secondly, general access tree narrated as above is monotone for the reason that if a attribute set $\gamma$ can achieve secret share, another set, which contains all elements of $\gamma$ plus an arbitrary attribute having been defined by system, can also achieve that simply by not using the additional attribute. Thirdly, each node of a layer of access tree is just the structure of threshold secret sharing[11] for the reason that $\mathcal{T}_{x\prime}(\gamma) = 1$ with the number of at least $k_x$ children. More specifically, "or" gate implies the threshold value 1, and "and" gate implies the threshold value number of children of the node, i.e, the threshold secret sharing is done by obtaining points with the number of one more that degree of target polynomial. Between two layers, massage is delivered by the point $(0, q_x(0))$ from nodes of low layer to a node of up layer, and essentially achieving the point $(0, q_x(0))$ is just the specific form in access tree of outcome that the Boolean Formula of this node $x$ are satisfied by the nodes contacted with $x$ from closest lower layer.

Finally, In many specific schemes [3],[8],[32], the method to build polynomials by Lagrange's interpolation. So whether there are other methods such us Newton interpolation to build polynomials in ABE schemes and whether those interpolation method can achieve schemes more efficiently are possible direction of future work. Moreover, every layer achieves threshold secret sharing by building polynomial, so we can consider the possibility that achieve sharing through other mathematical technologies.

**Circuits**

For concision in exposition, we restrict that gates of *circuits* are either *AND* or *OR* two inputs. Define the circuit structure as a 5-tuple $f = (u, q, A, B, GT)$. where $u$ shown on Table III is the number of inputs corresponding the set of subscripts of $P_i$ shown on Table III, and $q$ is the number of gates. Label $Inputs = \{1, ..., u\}$, $Wires = \{1, \ldots, u + q\}$, and $Gates = \{u+1, \ldots, u+q\}$. The wire $n+q$ is output wire of the whole circuit. $A : Gates \rightarrow Wires$ is a function to identify each $gate's$ first incoming wire, and $B : Gates \rightarrow Wires$ is a function to identify each $gate's$ second incoming wire. Finally, $GT$: $Gates \rightarrow AND, OR$ is a function to identifies a gate as either an *AND* or *OR* gate.

Specify that $\omega > B(\omega) > A(\omega)$, where $\omega \in Gates$, so that the label of a gate $\omega$ is the same as the label of the outgoing wire from $\omega$. For convenience of description, an extra function are defined here. First is layer of gate, $layer(\omega)$: the shortest path from gate $\omega$ to an input belonging to the set $Input$ plus 1, and naturally if $\omega \in Inputs$, $layer(\omega) = 1$. we also define the layer of wires,$layer\prime$ ,as follows. If $layer(\omega) = m$ then $layer\prime(A(\omega)) = layer\prime(B(\omega)) = m - 1$ and specially define the layer of output wire as $layer\prime(u + q)$,

**Achieving Secret Sharing In Circuit**

In circuit bounded with $layer\prime(u+q) = l$, the structure part firstly produces groups $G = (G_1, \ldots, G_{l+1})$ of prime order $p$, with canonical generators $g_1, \ldots, g_{l+1}$, and find out a set of bilinear maps $\{e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \longrightarrow \mathbb{G}_{i+j} | i, j \geq 1, i + j \leq l + 1\}$, so that the map $e_{i,j}$ satisfies the following relation: $e_{i,j}({g_i}^a, {g_j}^b) = {g_{ab}}^{i+j} \forall a, b \in Z_p$.

Then, the structure part chooses randomly $r_1, \ldots, r_{u+q-1}, r_{u+q} \in Z_p$, where the $u + q$ values have one-to-one correspondence with $u + q$ wires. Then, when those the numbers of subscript $k$ of $r$ are greater than $u$, where $k$ denote a gate or their output wire, structure part does the calculation as followings.

When $k$ is a *AND* gate, calculates $choose1(k) : r_k - r_{A(k)} - r_{B(k)}$, then calculate $g_m^c hoose1$; when $k$ is a *OR* gate, calculates $choose2(k) : r_k - r_{A(k)}$ and $choose3(k) : r_k - r_{B(k)}$, then calculate $g_m^c hoose2$ and $g_m^c hoose3$, where $m = layer(k)$. And the secret value is $g_{l+1}^{r_{u+q}}$ same as $s$ on Table III.

we assume that the set part have known $e_{i,j}$ defined above, $g_1$, and for each attribute in the qualified set of its own, the part is also provided the corresponding $r_k$, where $k \geq u$ is a $input$. Additionally, each *AND* gate $k$ have been attached $choose1(k)$; For each *OR* gate $k$, attach $choose2(k)$ to $k\prime$ first incoming wire and $choose3(k)$ to $k\prime$ second incoming wire.

The set part achieve secret sharing from the bottom up as follows.

For simple narration, denote that function $C_k(x) = 1$, if and only if the $input$ $k$ is in the qualified set or logic of gate $k$ is true. there are three case for $k$: $input$, gate *AND* and *OR*. Only when each $k$ let $C_k(x) = 1$ holds, following calculation will be do. When $k$ is a $input$, calculates $e_{1,1}(g_1^{r_k}, g_1) = g_2^{r_k}$; When $k$ is a gate *AND*, calculates $e_{m,1}(g_m^{r_{A(k)}}, g_1) \cdot e_{m,1}(g_m^{r_{B(k)}}, g_1) \cdot e_{m,1}(g_m^{choose1(k)}, g_1) = g_{m+1}^{r_k}$; When $k$ is a gate *OR*, there are three cases. Case one is that $C_t(x) = 1$ where $A(k)$ is output wire of gate $t$ and that $C_y(x) \neq 1$ where $B(k)$ is output wire of gate $y$, and calculates $e_{m,1}(g_m^{r_{A(k)}}, g_1) \cdot e_{m,1}(g_m^{choose2(k)}, g_1) = g_{m+1}^{r_k}$. Case two is revise on case one, i.e, $C_t(x) \neq 1$ where $A(k)$ is output wire of gate $t$ and that $C_y(x) = 1$ where $B(k)$ is output wire of gate $y$, and calculates $e_{m,1}(g_m^{r_B(k)}, g_1) \cdot e_{m,1}(g_m^{choose3(k)}, g_1) = g_{m+1}^{r_k}$. Case three is both of conditions of cases above, and the calculation is chosen as either case one or case two of one. The calculating do not stop until obtains the $g_{l+1}^{r_{u+q}}$.

**Some Analyses And Remarks On Circuit**

Firstly, "attributes *satisfy* policy" can be interpreted on the expression of circuits similar with that of access tree, i.e, $inputs$ have the consistency in logic with gates of circuit layer by layer, such that finally $C_{u+q}(x) = 1$. So, $\mathbb{A}$ can be interpret here as the inputs with the constraints from the structure of the circuit. Secondly, the circuit narrated as above is monotone. As for non-monotone circuit, there is a simple transformation that uses De Morgan's rule to transform any general Boolean circuit into an equivalent monotone Boolean

TABLE IV
CHOOSING DEPENDS ON DIFFERENT SITUATION IN INCOMING WIRES

| Incoming Wire | | gate type | output | choosings |
|---|---|---|---|---|
| The First | The Second | | | |
| 1 | 1 | AND | 1 | choose1 |
| 1 | 0 | OR | 1 | choose2 |
| 0 | 1 | OR | 1 | choose3 |
| 1 | 1 | OR | 1 | choose2 or choose3 |
| * | * | * | 0 | $\diagdown$ |

circuit with negation gates only allowed at the inputs, and just ignore the layer of the negation gates then non-monotone circuit become our familiar form of the monotone circuit. Thirdly, each gate of a layer in circuit is straightforward Boolean formula that emulate all situations on the incoming wires for that output of the gate's Boolean formula equal to 1, and that do nothing for that output of the gate's Boolean formula equal to 0. More specifically, for the gate in circuit above, according to different situations in incoming wires make the corresponding choices, which is shown on Table IV. Between two layers, massage is delivered in the way that logical value as output of lower layer performs meanwhile as input of the gate of higher layer, and accordingly $g_k$ by pairing operation with a jump element $g_1$ becomes $g_{k+1}$.

Finally, we can see the circuit as well as access tree is designed by fully considering the Boolean Formula, i.e, finding resolutions of that how to achieve the building of specific relationship among attributes. But for circuit, there is apparent drawback that if each gate has more input, then the situations mentioned above becomes more and more. For example, each gate has 3 inputs, so that there are 8 the situations that output of the gate's Boolean formula equals to 1, and 8 choices for all situations despite of 4 distinct choices. In addition, circuit used in ABE schemes bring more cost of communication between two parts due to more prepositive parameter such us those groups and pairing operation.

**Linear Secret Sharing Scheme**

Access structure in the LSSS is described as follows. Firstly, define a surjection, $\rho : \{1, \ldots, \ell\} \rightarrow \{P_1, \ldots, P_u\}$, where $P_j$ is shown on Table III. There is a matrix $M$ with $\ell$ rows and $n$ columns, where $\ell \geq n$. Row index $i \in \{1, \ldots, \ell\}$ represents the label of $ith$ row of $M$, and $M_i$ represents the $ith$ row of $M$. Let $\rho(i) = P_j$ so that the row index of $M$ map to attributes. Then define a set $I = \{i|\rho(i) \in \gamma\}$, where $\gamma$ denotes a qualified set shown on Table III, and the $M_\gamma$. $M_\gamma$ is constituted by combining $M_i$ for all $i \in I$.

Now, we reveal how to achieve secret sharing. Structure part generate a column vector $\boldsymbol{v}= (s, r_2, \ldots, r_n)^T$, where $s$ is the secret to be shared, and $r_2, \ldots, r_n \in Z_p$ are randomly chosen. Then computes $M_i{\cdot}\boldsymbol{v}$ for all $i \in \{1, \ldots, \ell\}$.

$$\sum_{i \in I}(\omega_i \cdot M_i) = \boldsymbol{e} \tag{1}$$

For set part, $M_i{\cdot}\boldsymbol{v}$ for $i \in I$ have been known. This part can find out constant $\omega_i$, and make equation (1) holds, where $\boldsymbol{e}$ is a n-dimension row vector and $\boldsymbol{e}= (1, 0, \ldots, 0)$, so that $\sum_{i \in I}(\omega_i \cdot (M_i{\cdot}\boldsymbol{v})) = (\sum_{i \in I}(\omega_i \cdot M_i)){\cdot}\boldsymbol{v}{=}\boldsymbol{e}{\cdot}\boldsymbol{v}{=}\ s$

Amos Beimel [30] points that each $\gamma$ can reconstruct the secret in the method by using a linear function of its pieces, attributes in $\gamma$. We informally call the secret sharing scheme using linear function is Linear Secret Sharing Scheme. the span program [33] is the known method. Monotone span program (MSP) is associated monotone Boolean Function, and M. Karchmer and A. Wigderson [33] proved that if there is a monotone span program for some Boolean Function then there exists a linear secret sharing scheme for the corresponding access structure, and vice versa [29], so matrix $M$, and those constants $\omega$ can be found for certain.

**Some Analyses And Remarks On LSSS**

Firstly, "attributes *satisfy* policy" can be interpreted in LSSS as that the row indexes achieve finding out those $\omega$ such that equation (1) holds. So, $\mathbb{A}$ is interpreted here as row indexes with the constraints from matrix $M$ and mapping $\rho$, and actually between these two objects there is a intermediate, the collection of input boolean vectors of Boolean Function.

As above mentioned, general LSSS can only address monotone access structure. When it occur to non-monotone case, use technology of [27].

LSSS achieve access control though the Boolean Function, so that there have no logical judgments or other consideration on relation among attributes. Therefore, LSSS is somewhat more expressive than above two expressions as far as automaticity and mathematics. which implies that not only can general monotone access structure be expressed by Boolean Function but also LSSS can achieve access control based that.

**Comparison**

We give a comparison among three different expressions on a monotone access structure. For clear, we list the items we concern.

- **Boolean Formula**. We consider how to embody the Boolean Formula in the three expressions.
- **Fan-in**. In this item, we survey whether each node in access tree or each gate in circuits supports multiple( more than two) inputs at a negligible price.
- **Extension Cost**. For view on scalability, We find out the most costs of adding an attribute on access structure in the three expressions.
- **Key Size**. For view on efficiency, we pick out three schemes based on different expressions, and point out these sizes of secret key of schemes (KP-ABE). For the sake of comparing as fair as possible , these three schemes have same type of attacks, chosen-plaintext attack, same model used to prove security, standard model, same security, selective security, and similar assumptions that are BDDH assumption and natural multilinear generalization of the BDDH assumption. Access tree and LSSS are both in [3], while circuit is in [28].
- **Backtracking Resistant** We call an expression resisting backtracking attack backtracking resistant. Backtracking attack firstly posed by [28] is that decryption algorithm learns some format of the value $q_B(0)$ for gate B even though the decryption algorithm do not know $q_B(X)$ on input $att(x)$ on the condition that $parent(B)$ is a OR gate and the decryption algorithm have been aware of $q_A(0)$, the value of one of other children of $parent(B)$.

TABLE V
A COMPARISON AMONG THREE EXPRESSIONS

| Expression | Boolean Formula | Fan-in | Extension Cost | Key Size | Backtracking Resistant |
|---|---|---|---|---|---|
| Access Tree [3] | direct | unrestricted | change a polynomial | $\omega\|G_1\|$ | NO |
| Circuit [28] | direct | restricted | add $(C+1)$ gates | at least $\|G_C\|$ | YES |
| LSSS [3] | black-box | $\diagdown$ | row size of $M$ add 1 | $\omega\|G_1\|$ | YES |

where $|G_C| = 3\frac{\omega}{2}|G_2| + \ldots + 3\frac{\omega}{2^{i-1}}|G_i| + \ldots + 3 \cdot 2|G_{log_2\omega}| + 2\omega|G_1| + |G_{log_2\omega+1}|$, $(2 \le i \le log_2\omega)$

We specify that those gates in access tree and in circuit only have two inputs for simpleness to compute the size of secret key. Comparison is shown on Table V, where $C$ denotes the number of gates in circuit and $\omega$ denotes the number of the $inputs$ of circuit or the number of leaf nodes in access tree or the one of rows of matrix $M$ in LSSS.[1] All notions used in Table V can be found in Table I.

### B. Design philosophy of classical ABE scheme

some introduction goes here.

*1) KP-ABE by (Li Peng):* As defined in [3], a KP-ABE scheme consists of four algorithms: **Setup**, **Encrypt**, **KeyGen** and **Decrypt**. Let a tuple$(p, g, \mathbb{G}_1, \mathbb{G}_2, e)$ denote a bilinear map, where $\mathbb{G}_1, \mathbb{G}_2$ are multiplicative cyclic groups of prim order $p$ and $e$ denote $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. A security parameter, $\lambda$, denote the size of the groups. The Lagrange coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set, $S$, of elements is defined in $\mathbb{Z}_p : \Delta_{i,S} = \Pi_{j\in S, j\neq i}\frac{x-j}{i-j}$. Each attribute is associated with a unique element in $\mathbb{Z}_p^*$. The construction follows.

**Setup**$(\lambda, U)$ The setup algorithm takes $\lambda, U$ as input parameters. Firstly, let $U = \{1, 2...n\}$ denote the universe of attributes, and define a tuple $(p, g, \mathbb{G}_1, \mathbb{G}_2, e)$ as above mentioned. Then choose a number $t_i$ uniformly at random for each attribute $i \in U$ and generate a random?value $y \in \mathbb{Z}_p$. Finally, the public parameters are published as $PK = (T_1, T_2, ..., T_{|U|}, Y)$, where $T_i = g^{t_i}, i \in \{1, 2...|U|\}, Y = e(g, g)^y$. The master key is $MK = (t_1, ..., t_{|U|}, y)$.
**Encrypt**$(\mathcal{M}, S, PK)$ The encryption algorithm takes as input a message $\mathcal{M} \in \mathbb{G}_2$, a set of attributes $S$ and the public parameters $PK$. To encrypt the message $\mathcal{M}$ under a set of attributes $S$, it chooses a a random value $\upsilon \in \mathbb{Z}_p$. Then the ciphertext is published as $CT = (S, C' = MY^{\upsilon}\{C_i = T_i^{\upsilon}\}_{i\in S})$;
**KeyGen**$(\mathcal{T}, MK)$ The Key Generation algorithm takes an access structure $\mathcal{T}$ and the master key $MK$, then outputs a secret key $SK$. The $SK$ enables the user to decrypt a message encrypted under a set of attributes $S$ if and only if $\mathcal{T}(S) = 1$. First let the access structure $\mathcal{T}$ be a tree, then choose a polynomial $q_x$ for each node $x$ (including the leaves) in the tree $\mathcal{T}$.

For each node $x$ in the tree, choose a polynomial $q_x$ whose degree $d_x$ is one less than the threshold value $k_x$ of that node, that is, $d_x = k_x - 1$. Firstly, for the root node $r$, set $q_r(0) = y$ and choose $d_r$ other points of the polynomial $q_r$ randomly to get the complete polynomial $q_x$.Then, for any other node $x$, set $q_x(0) = q_{parent(x)}(index(x))$ , where $parent(x)$ denote the parent node of $x$ and $index(x)$ denote the number of the node, and choose $d_x$ other points randomly to completely define $q_x$. As a result, for each leaf node $x$, $q_x(0)$ is defined.Finally, the secret key is published as $SK = (D_x)$, where $D_x = g^{\frac{q_x(0)}{t_i}}, i = att(x)$.
**Decrypt**$(CT, SK)$ Firstly, define a recursive algorithm DecryptNode $(CT, SK, x)$ that takes ciphertext $CT$, the private key $SK$ and a node $x$ in the tree as input. Then it outputs a group element of $\mathbb{G}_2$ or $\perp$. Let $i = att(x)$. If the node $x$ is a leaf node then:

$$\text{DecryptNode}(CT, SK, x) = \begin{cases} e(D_x, C_i) = & e(g^{\frac{q_x(0)}{t_i}}, g^{\upsilon \cdot t_i}) \\ \qquad\quad = & e(g, g)^{s \cdot q_x(0)} \quad \text{if} i \in S \\ \qquad \perp & \text{otherwise} \end{cases}$$

If $x$ is a non-leaf node, it calls DecryptNode$(CT, SK, z)$ for each child node $z$. Then it stores the output as $F_z$. Let $S_x$ be an arbitrary $k_x$-sized set of child nodes $z$ such that $F_z \neq \perp$. If no such set exists then the node was not satisfied and the function returns $\perp$.
Otherwise, we compute:

$$\begin{aligned} F_x &= \prod_{z\in S_x} F_z^{\Delta_{t,s'_x}(0)} \\ &= \prod_{z\in S_x} \left(e(g,g)^{\upsilon \cdot q_z(0)}\right)^{\Delta_{t,s'_x}(0)} \\ &= \prod_{z\in S_x} \left(e(g,g)^{\upsilon \cdot q_{parent(z)}(index(z))}\right)^{\Delta_{t,s'_x}(0)} \\ &= \prod_{z\in S_x} e(g,g)^{\upsilon \cdot q_x(i) \cdot \Delta_{t,s'_x}(0)} \\ &= e(g,g)^{\upsilon \cdot q_x(0)} \text{ (using polynomial interpolation)} \end{aligned}$$

and return the result.
After computing, the equation DecryptNode$(CT, SK, r) = e(g, g)^{y\upsilon} = Y^{\upsilon}$ is true if and only if the ciphertext satisfies the tree. Since, $C' = \mathcal{M}Y^{\upsilon}$s the decryption algorithm simply divides out $Y^{\upsilon}$ and recovers the message $\mathcal{M}$.

*2) CP-ABE (by Hao Zhang):* In Attribute-Based Encryption scheme is generally divided into two categories. One is key-policy attribute based encryption, in key-policy attribute based en- cryption, ciphertexts are associated with sets of descriptive attributes, and users? keys are associated with policies. Other is ciphertext-policy attribute-based encryption, in ciphertext-policy attribute-based encryption user?s private

[1]Access tree can be convert to LSSS for the reason that LSSS can express general access structure. For two-input gate in access tree, we can construct a matrix $M$, let the number of rows be same with the number of the leaf nodes [34]

key will be associated with an arbitrary number of attributes expressed as strings. The most important to CP-ABE is access structure, CP-ABE has a variety of access structures. For example, the tree in mathematics is one of its access structure, where nodes of the access structure are composed of threshold gates and the leaves de- scribe attributes. However, for CP-ABE scheme there is a very important property that is the collusion resistance. There are a number of ways to implement collusion resistance, John Bethencourt[] in his article gives one of the solutions of specific methods. An ciphertext-policy attribute based encryption scheme consists of four fundamental algorithms: **Setup**, **Encrypt**, **KeyGen**, and **Decrypt**.

**Setup**. The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters $PK$ and a master key $MK$.

**Encrypt**$(PK, \mathcal{M}, \mathbb{A})$.The encryption algorithm takes as input the public parameters $PK$, a message $\mathcal{M}$, and an access structure $\mathbb{A}$ over the universe of attributes. The algorithm will encrypt $\mathcal{M}$ and produce a ciphertext $CT$ such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains $\mathbb{A}$.

**Key Generation**$(MK, S)$. The key generation algorithm takes as input the master key $MK$ and a set of attributes S that describe the key. It outputs a private key $SK$.

**Decrypt**$(PK, CT, SK)$. takes as input the public parameters $PK$, a ciphertext $CT$, which contains an access policy $\mathbb{A}$, and a private key $SK$, which is a private key for a set S of attributes. If the set $S$ of attributes satisfies the access structure $\mathbb{A}$ then the algorithm will decrypt the ciphertext and return a message $\mathcal{M}$.

## V. Extension

some introduction goes here.

### A. Outsource (By Li Peng)

With the rapid development of cloud computing, a growing number of users choose to share their sensitive data on the clouds. To keep the data security and privacy, encrypting this data is a valid approach before uploading. In addition, this data also should be only accessed to the authenticated users. But it is difficult for traditional public key encryption to realize this demand because it can only realize one-to-one encryption. To handle this problem, ABE is proposed, which provides one-to-many encryption and fine-grained access control of encrypted data.

However, almost all proposed ABE schemes have a main efficiency drawback, that is, computational cost for encryption and decryption is very high. More seriously, the computational cost grows with the complexity of access formula. In a resource-limited device, such as a smart phone, decrypting a ABE ciphertext is very difficult. Suppose that, you are traveling and only taking a smart phone, but your leader suddenly need you to do something immediately and send you a message which is encrypted by ABE. Its a terrible experience because you cant decrypt this message using a smart phone.

To leverage computation burden on the user side, Green *et al.* [35] suggested to outsource decryption in attribute-based encryption. The schemes in [35] allow a cloud to translate an ABE ciphertext satisfied by that users attributes into a (constant-size) ElGamal-style ciphertext. Of course, the coulds who perform the translation operation cannot read any part of messages and the users private keys. To outsource the decryption to a could, a user need to split his private key into a transformation key (denoted by TK), and an El Gamal-type secret key (denoted by DK). The user give the TK to the cloud, then the could translate the ABE ciphertext into an ElGamal-style ciphertex and send it to the user. Finally, the user utilize the DK to recover the encrypted message. In [35], Green *et al.* also proposed replayable CCA(RCCA) security schemes in which the user can check the correctness of recovered message in random oracles(RO). But in their RCCA schemes,a malicious cloud could replace the original ciphertext and its tag which is used to check the correctness. So this method can not strictly guarantee the correctness of transformed ciphertext, to solve this problem, Attribute-Based Encryption with verifiable outsourced decryption was proposed by Lai *et al.* [36]. In their schemes, a tag computed by a real message and a random message is used to verify the correctness of the real message. In addition, the original untransformed ciphertext also need to be inputted in the final decryption stage. As a result, this method causes approximately double overhead in both ciphertext size and decryption operation compared with [35]. To increase the efficiency of ABE with verifiable outsourced decryption in [36], Qin *et al.* [37] and Lin *et al.* [38] introduced a key encapsulated mechanism (KEM). Their methods reduce the ciphertext and the computation costs almost by half compared with Lai *et al.*s scheme [36]. On the other hand, in view of the huge overhead computation in the attribute authority, Li *et al.* [39] proposed a new Secure Outsourced ABE system, which outsource the partial key-issuing and decryption. In their construction, the users private keys are associated with users attributes and a default attribute. The computation for users attributes is outsourced to the clouds and the computation for the default attribute is performed by the attribute authority. The final users private key can be gained by combining these two parts. And the method for outsourced decryption is the same as the method in [35]. Furthermore, this construction also provide the checkability, but it also suffers from the attack existed in Green *et al.*s schemes [35]. In [25], Ma *et al.* proposed two ciphertext-policy attribute-based key encapsulation mechanism (CP-AB-KEM) schemes. Their schemes for the first time achieve both outsourced encryption and outsourced decryption. Moreover, their schemes not only offer the method to efficiently check the correctness of the outsourced encryption and decryption, but also exculpability, which means a user cannot accuse a Decryption Service Providers(DSP) of returning incorrect results when it return right results. The efficiency and communication cost comparison is presented in Table VI and

TABLE VI
EFFICIENCY COMPARISON FOR OUTSOURCE ABE

| Schemes | Enc(Server) | Enc(USER) | Dec(Server) | Dec(USER) | Enc.Verify | Dec.Verify |
|---|---|---|---|---|---|---|
| [35] | × | $(3\ell+1)E_{G_1}+E_{G_2}$ | $(I+2)P+2\|I\|E_{G_1}$ | $E_{G_2}$ | × | × |
| [36] | × | $(6\ell+4)E_{G_1}+2E_{G_2}$ | $(4\|I\|+2)P+2\|I\|E_{G_2}$ | $2E_{G_1}+2E_{G_2}$ | × | ✓ |
| [37] | × | $(3\ell+1)E_{G_1}+E_{G_2}$ | $(\|I\|+2)P+2\|I\|E_{G_1}$ | $E_{G_2}$ | × | ✓ |
| [38] | × | $(3\ell+3)E_{G_1}+E_{G_2}$ | $(2\|I\|+1)P+\|I\|E_{G_2}$ | $2E_{G_1}+E_{G_2}$ | × | ✓ |
| scheme 1 in [25] | $(5\ell+1)E_{G_1}$ | $2E_{G_1}+E_{G_2}$ | $(3\|I\|+2)P+(4\|I\|+2)E_{G_1}+\|I\|E_{G_2}$ | $2E_{G_1}+E_{G_2}$ | ✓ | ✓ |
| scheme 2 in [25] | $(10\ell+2)E_{G_1}$ | $2E_{G_1}+E_{G_2}$ | $(3\|I\|+2)P+(\|I\|+1)E_{G_1}+\|I\|E_{G_2}$ | $2E_{G_1}+E_{G_2}$ | ✓ | ✓ |

TABLE VII
COMMUNICATION COST COMPARISON

| Schemes | Transfer Size(Enc) | Full Size | Transfer Size(Enc) |
|---|---|---|---|
| [35] | × | $(2\ell+1)\|G_1\|+\|G_2\|$ | $2\|G_2\|$ |
| [36] | × | $(4\ell+3)\|G_1\|+2\|G_2\|$ | $\|G_1\|+2\|G_2\|$ |
| [37] | × | $(2\ell+1)\|G_1\|+\|G_2\|$ | $\|G_2\|$ |
| [38] | × | $(2\ell+2)\|G_1\|+\|G_2\|$ | $\|G_2$ |
| scheme 1 in [25] | $(3\ell+3)\|G_1\|+3\ell\|Z_p\|$ | $(3\ell+1)+\|G_1\|+(3\ell+1)\|Z_p\|$ | $\|G_1\|+\|G_2\|$ |
| scheme 2 in [25] | $(6\ell+4)\|G_1\|+4\ell\|Z_p\|$ | $(3\ell+1)\|G_1\|+2\ell\|Z_p\|$ | $\|G_1\|+\|G_2\|$ |

Table VII respectively.

## B. Proxy Re-Encryption (by Yi Wang)

Proxy Re-Encryption (PRE) is a cryptographic primitive first introduced by Blaze et.al in 1998. In a PRE scheme, a proxy is allowed to transform a ciphertext encrypted by Alice's public key into another one that can be decrypted by Bob's secret key, this process is called re-encryption. Considering this case, an encrypted email is send to Alice, but Alice is off-line. Alice wishes Bob could still read the message in her encrypted email. Therefore, Alice delegates a re-encryption key to the proxy. With the key, the proxy could transform the email to one that could be decrypted by Bob's secret key. With the PRE system, Bob could decrypt a re-encrypted email of the same message with his own secret key.

In PRE system, the proxy is semi-trusted, but the user(Alice) doesn't need to delegate part of her decryption capability to others including the proxy. There is also an advantage that the user could finish any decryption only with his own secret key. So storing any additional decryption key is unnecessary. Otherwise, the proxy authority can only translate one ciphertext into another one under another public key. It is impossible for the delegation to obtain the corresponding plaintext and the original secret keys. With these advantages, PRE can be widely used into many public key cryptosystem. It also have many practical applications. For example, encrypted email forwarding, distribute file system, and the DRM(Digital Rights Management) of Apple's iTunes.

Since PRE have been introduced in 1998, there are many related works on it. In the paper [40], it proposed the first concrete bidirectional PRE scheme. This scheme allows the key holder to publish the proxy function. It applied by untrusted parties without further involvement by the original key holder. Their scheme also has multi-use property. Ateniese *et al.* [41] presented the first unidirectional. It is a single-use proxy re-encryption scheme. In 2007, Green and Ateniese[42] provided identity-based PRE. The security of it is in the random oracle model. Chu *et al.* [43] proposed new identity-based proxy re-encryption schemes in the standard model. Matsuo [44] also proposed new identity-based proxy re-encryption system, the solution needs a re-encryption key generator (RKG) to generate re-encryption keys. Guo *et al.* [45] proposed the first attribute-based proxy re-encryption(ABPRE) scheme, but their scheme is based on key policy and bidirectional. Liang *et al.* [31] proposed the first ciphertext policy attribute-based proxy re-encryption(CP-ABPRE) scheme which has the above properties except re-encryption control. In TCC 2012, Chandran *et al.* [46] proposed an obfuscation for functional re-encryption with collusion resistant property. Recently, a CCA secure CP-ABPRE was proposed in [47], in this paper the scheme is proven in the random oracle model. In[48], a CP-ABPRE system was built and proven secure against CCA in the standard model.

Next, we will give an ABPRE model in[31].

Definition: *An* ABPRE *scheme is a tuple of probabilistic polynomial time algorithms* (Setup, KeyGen, RKGen, Enc, ReEnc, Dec).

- **Setup**($1^k$)→(*pp*,*mk*): On input a security parameter $1^k$, the setup algorithm SETUP outputs a system public parameter *pp* and a master key *mk*.
- **KeyGen**(*S;mk*)→(*usk*): On input an index set $S^1$ and a master key *mk*, the key generation algorithm KEYGEN outputs a secret key *usk*.
- **RKGen**(*usk*;*AS*)→(*rk*): On input a secret key *usk* and an access structure *AS*, the re-key generation algorithm RKGEN outputs a re-key *rk*.
- **Enc**(*AS*;*m*)→(*C*): On input an access structure *AS* and a message *m*, the encryption algorithm ENC outputs a ciphertext *C*.
- **ReEnc**(*rk*;*C*)→($C'$): On input a re-key *rk* and a ciphertext *C*, the re-encryption algorithm REENC first checks if the

index set in *rk* satisfies the access structure of *C*. Then, if check passes, it outputs a re-encrypted ciphertext $C'$; otherwise, it outputs "reject".

- **Dec**(*usk*;*C*)→(*m*): On input a secret key *usk* and a ciphertext *C*, the decryption algorithm DEC first checks if the index set in *usk* satisfies the access structure of *C*. Then, if check passes, it outputs a message *m* in the message space; otherwise, it outputs "reject".

Then, we present the construction with six algorithms Setup, KeyGen, ENC, RKGen, ReEnc, Dec.

**Setup**$(1^k)$ Generate a bilinear group $G$ of prime order $p$, with bilinear map $e$:$G\times G \to G_T$. Next, it selects elements $y$, $t_i(1\leq i\leq 3n)$ in $Z_p$ and two generators $g$, $h$ of $G$ at random. Let $Y$:=$e(g,h)^y$ and $T_i$:=$g^{t_i}$, $T_i'$:=$h^{\frac{1}{t_i}}$, for each $1\leq i\leq 3n$. The public parameter *pp* includes $\langle$ *e, g, h, Y,* $\{T_i, T_i'\}_{1\leq i\leq 3n}$ $\rangle$. The master key *mk* is $\langle y, \{t_i\}_{1\leq i\leq 3n}\rangle$.

**KeyGen**(*S*, *mk*) Let *S* denote an index set of attributes. It chooses random $r_1 \cdots r_n$ from $Z_p$ and sets $r = r_1 + r_2 + \cdots + r_n$. Compute $\hat{D}$=$h^{y-r}$, and for each $i \in \mathcal{N}$($\mathcal{N}$={1, 2, $\cdots$, $n$}): if $i \in S$, $D_{i,1}$=$h^{\frac{r_i}{t_i}}$, $D_{i,2}$=$h^{\frac{r_i}{t_{2n+i}}}$; otherwise, $D_{i,1}$=$h^{\frac{r_i}{t_{n+i}}}$, $D_{i,2}$=$h^{\frac{r_i}{t_{2n+i}}}$. It outputs a user's secret key *usk*=$\langle S, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}}, \hat{D}\rangle$.

**Enc**(*m*, *AS*) Let *AS* denote an access structure. To encrypt a message $m \in G_T$, it selects random $s \in Z_p$ and computes $\tilde{C}$=$m \cdot Y^s$, $\hat{C}$=$g^s$, $\breve{C}$=$h^s$. For $i \in \mathcal{N}$: if $+d_i$ appears *AS*, $C_i = T_i^s$; if $-d_i$ appears *AS*, $C_i = T_{n+i}^s$; otherwise $C_i = T_{2n+i}^s$. It outputs *C*=$\langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{(i\in\mathcal{N})}\rangle$.

**RKGen**(*usk*, *AS*) Let *usk* denote a valid secret key consisting of $\langle S, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}}, \hat{D}\rangle$. and *AS* denote an access structure.It selects random $d \in Z_p$ and set $\mathfrak{D} = g^d, \hat{D}' = \hat{D}$. For $i \in \mathcal{N}$: if $i \in S$, $D_{i,1}'$=$D_{i,1} \cdot (T_i')^d$, $D_{i,2}'$=$D_{i,2} \cdot (T_{2n+i}')^d$; otherwise, $D_{i,1}'$=$D_{i,1} \cdot (T_{n+i}')^d$, $D_{i,2}'$=$D_{i,2} \cdot (T_{2n+i}')^d$; $\mathfrak{C}$ is the ciphertext of $\mathfrak{D}$ under the access structure *AS*.

It outputs *rk*=$\langle S, AS, (D_{i,1}', D_{i,2}')_{i\in\mathcal{N}}, \hat{D}', \mathfrak{C}\rangle$.

**ReEnc**(*rk*, *C*) Let *rk* denote a valid re-key consisting of $\langle S, AS', (D_{i,1}', D_{i,2}')_{i\in\mathcal{N}}, \hat{D}', \mathfrak{C}\rangle$ and *C* denote a well-formed ciphertext $\langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{(i\in\mathcal{N})}\rangle$, It checks if *S* satisfies *AS*, if not, output $\perp$; otherwise, for $i \in \mathcal{N}$:

- $+d_i$ appears in *AS*, $E_i = e(C_i, D_{i,1}') = e(g^{t_i s}, h^{\frac{r_i+d}{t_i}}) = e(g,h)^{s(r_i+d)}$;
- $-d_i$ appears in *AS*, $E_i = e(C_i, D_{i,1}') = e(g^{t_{n+i}s}, h^{\frac{r_i+d}{t_{n+i}}}) = e(g,h)^{s(r_i+d)}$;
- otherwise, $E_i = e(C_i, D_{i,1}') = e(g^{t_{2n+i}s}, h^{\frac{r_i+d}{t_{2n+i}}}) = e(g,h)^{s(r_i+d)}$;

It then computes $\bar{C}$=$e(\hat{C}, \hat{D}')\prod_{i\in\mathcal{N}} E_i = e(g^s, h^{y-\sum_{i=1}^n r_i}) \cdot e(g,h)^{nds+s\sum_{i=1}^n r_i} = e(g,h)^{nds+ys}$; output a re-encrypted ciphertext $C'$=$\langle AS', \tilde{C}, \bar{C}, \breve{C}, \mathfrak{C}\rangle$.

**Dec**(*C*, *usk*) Let *usk* denote a valid secret key $\langle S, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}}, \hat{D}\rangle$. It checks if *S* satisfies *AS*, if not, output $\perp$; otherwise, do

1) If C is an original well-formed ciphertext consisting of $\langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{(i\in\mathcal{N})}\rangle$, for $i \in \mathcal{N}$:
   - $+d_i$ appears in *AS*, $E_i = e(C_i, D_{i,1}) = e(T_i^s, h^{\frac{r_i}{t_i}}) = e(g,h)^{sr_i}$;
   - $-d_i$ appears in *AS*, $E_i = e(C_i, D_{i,1}) = e(T_{n+i}^s, h^{\frac{r_i}{t_{n+i}}}) = e(g,h)^{sr_i}$;
   - otherwise, $E_i = e(C_i, D_{2n+i,1}) = e(T_i^s, h^{\frac{r_i}{t_{2n+i}}}) = e(g,h)^{sr_i}$;

   It outputs $\frac{\tilde{C}}{e(\hat{C},\hat{D})\cdot\prod_{i\in\mathcal{N}} E_i} = \frac{m\cdot e(g,h)^{ys}}{e(g,h^{y-r})\cdot e(g,h)^{sr}} = m$.
2) Else if *C* is a re-encrypted well-formed ciphertext consisting of $\langle AS', \tilde{C}, \bar{C}, \breve{C}, \mathfrak{C}\rangle$, it decrypts $\mathfrak{C}$ using *usk* and obtains $\mathfrak{D} = g^d$. Then, it outputs $\frac{\tilde{C}e(\mathfrak{D},\breve{C})^n}{\bar{C}} = \frac{m\cdot e(g,h)^{ys}\cdot e(g^d,h^s)^n}{e(g,h)^{ys+nds}} = m$.
3) Else if *C* is a multi-time re-encrypted well-formed ciphertext, decryption is similar with the above phases.

### C. Multi-authority And distributed ABE

*1) Multi-authority And distributed ABE( by Prince):* In Distributed systems, it will not be user-friendly for access rules for objects to be based on identities considering the various dynamic sets of users in todays computing environment. In light of the above, the incipient proposal of Attribute Based Encryption in a landmark work by A. Sahai and B. Waters in 2005[10] and later by Goyal *et al.* [3] opened up various interests to the research community.

In the traditional ABE scheme, there exist a central authority (CA) in charge of all attributes and responsible for the issuance of secret keys to users for decryption. Consequently, the CA can decrypt every ciphertext in the system by calculating the required secret keys at any time, this is the *key escrow problem* of ABE[49]. This problem triggered the conceptualization of multi-authority and distributed ABE schemes. Imperatively, there are two major concepts under the ABE scheme, namely:

Key-Policy ABE (KP-ABE). In these schemes, the secret keys are associated with an access structure, while the ciphertext is labeled with a set of attributes[10], [3], [50], [27].

Ciphertext-Policy ABE (CP-ABE). In these schemes, the ciphertext is associated with an access structure, while the secret keys are labeled with a set of attributescite[8],[22], [51].

Even though some prior researchers[52][53][54][55] proposed some form of multi-authority, it was noted by Mller *et al.* [56] that the techniques used in these applications are not collusion-resistant, so they can not be classified as ABE.

**The first Multi-Authority**

In 2007 Chase[50] introduced what could be considered as the first Key-Policy ABE with multi-authority since she introduced various authorities with global identifiers to keep a users' keys in sync in her scheme. However, these authorities were fixed during initialization and threshold gates were used as access policy. One dominant downside of his scheme was that, it depended on a *Central Authority* (CA).

**Distributed Attribute-Based Encryption**

One year later (2008), Mller *et al.* [56], introduced the concept of Distributed Attribute-Based Encryption (DABE) using Cipertext-Policy, that supports an arbitrary number of attribute authorities where both the authorities and users could join the system any time. Nonetheless, one central authority*(Master)* is dedicated to the distribution of user secret keys and an arbitrary number of Attribute authorities responsible for the verification of user eligibility and the distribution of secret attribute keys to the users.

Their scheme constructs access policies using the Disjointed Normal Form (DNF). Compared to earlier ABE schemes, the DABE is fairly efficient as most of its computation uses group operations in bilinear group except during the decryption stage where it uses two time of the bilinear pairing for the computation. Mller *et al.* secured their scheme in the random oracle model and provided a security proof using the adaptive model.

*2) Decentralized Attribute Based Encryption( by Prince):* One natural feature of decentralized ABE systems is the absence of a central authority (CA). Lin *et al.* [57] proposed the first non-CA multi-authority ABE taking advantage of Distributed Key Generation (DKG) protocol and Joint Zero Secret Sharing (JZSS) protocol to replace the CA. However, this scheme can only resist collusion up to collusion of $m$ users. Where $m$ was a fixed system parameter chosen at setup. In 2009, Chase *et al.* [58] constructed a CA-free multi-authority ABE which solved the key escrow problem in[50] using distributed Pseudo Random Functions (PRF) and key-policy with AND-gate access structure. The scheme is limited to handling only a set of fixed number of authorities at system initialization.

Lewko *et al.* [34] introduced a novel decentralized multi-authority scheme which is CA-free. The scheme which is a ciphertext-policy ABE, functions entirely independently where any party can become an authority and authorities need not even be aware of each other. He based his scheme on the concept of global identifier introduced by Chase[50] to bind attribute-related secret keys of a user from different authorities together, thereby preventing collusion attack. The scheme is secured in the random oracle model and proven secured using the adaptive model. Liu *et al.* [59] came up with a LSSS multi-authority CP-ABE system which has multiple CAs and authorities. The scheme is adaptively secure without random oracles unlike Lewko and Waters.

**Decentralized Ciphertext-Policy Attribute-Based Encryption Scheme with Fast Decryption**

In most of the multi-athorities mentioned above, the ciphertext size are huge likewise the linear-size of the bilinear pairing. To help solve this problem,[56] propose an efficient multi-authority decentralized Ciphertext-Policy Attribute-Based Encryption scheme for Monotone Access Structures (MAS) with fast decryption, where no Central Authority exists and all authorities function independently without being aware of each other.

The security of this scheme is built on the generic bilinear group model[60] [61] and proven secured adaptively. It is collusion resistance and also secured against the escrow problem. In comparison with[34], this scheme's decryption time is constant for general MAS giving it fast performance. Furthermore, it provides a mechanism for packing multiple messages in a single ciphertext which makes it more efficient to use.

**Large universe decentralized key-policy attribute-based encryption**

Li *et al.* [62] present a large universe decentralized key-policy ABE scheme on prime order groups without any central authority where attribute authority executes independently from the others and can join or depart the system anytime.

This system supports a large universe of attributes and does not impose any bounds on the set of attributes, which will be used in encryption. It is constructed in the standard security model and proven secured with the selective model.

*3) Privacy-Preserving in Attribute-Based Encryption( by Prince):* Over the past years Multi-Authority ABE has been expanded and improved upon by various researchers in this field. Attribute-Based Encryption, where a user in the system can share his encrypted data based on access policy defined by him and only users whose attributes satisfy the policy can subsequently decrypt and have access to the shared data. This can be considered as an efficient and convenient primitive to use in pervasive computing environments.

Even though this primitive brings flexibility, it also materialized security concerns. Notably among them is the problem where authorities can collect the attributes of users and consequently decrypt messages or in worse case, impersonate the user.

Considering the security of attributes of users, Han *et al.* [63] introduced the first privacy-preserving decentralized key-policy ABE scheme where each authority can issue secret keys to a user independently without knowing anything about his GID. Thereby protecting the user's privacy. Their scheme is designed in the standard security model and proven secure in the selective model. Using privacy-preserving key extraction protocol method and a global identifier (GID), they tied the user's access to all authorities to prevent collusion attacks.

Unfortunately, by breaking the weak ties between authorities, (to remove such a connection by changing the identifier associated with particular secret keys) Ge *et al.* [64], proved that the scheme "*Privacy-Preserving Decentralized Key-Policy Attribute-Based Encryption*" proposed by Han *et al.* in 2012 is open collusion attacks in the standard model.

In 2014, Han *et al.* [64] proposed a privacy-preserving decentralized CP-ABE (PPDCP-ABE) scheme where the central authority is not required and each authority can work independently without the cooperation to initialize the system and the user can convince the authorities that the attributes for which he is obtaining secret keys are monitored by them without compromising the GID and the attributes of the user. This scheme is built on standard model and proven secured in the selective model. Furthermore the scheme is based on the Privacy-Preserving Key Extract Protocol and Linear Secret Sharing Schemes.

On the other hand, Wang *et al.* [65] gave a security analysis of PPDCP-ABE scheme of Han *et al.* [64] and point out the security weakness of their scheme. It came out that their basic decentralized ciphertext-policy ABE scheme cannot resist collusion attacks. Also, the privacy-preserving key extract protocol proposed by[64], allows the authority to reveal users' credentials, hence, the privacy protection of attributes cannot be provided.

**User Collusion Avoidance Scheme for Privacy-Preserving Decentralized Key-Policy Attribute-Based Encryption**

More recently in september 2016, Rahulamathavan *et al.* [66] have proposed a user collusion avoidance scheme which preserves the user's privacy when they interact with multiple authorities to obtain decryption credentials. This they achieved by tying secret known for Attribute Authority and secret known for the user in a non-linear fashion. Further, by modify the scheme in [64] using the anonymous key issuing protocol in [58] to secure the bind between decryption keys and GID as well as to preserve the user's privacy while the Attribute Authority is guaranteed that the decryption-keys are the only information that the user learns from the transaction using blind IBE schemes [67].

This scheme has been designed on the standard security model (decisional bilienar Deffie-Hellman assumption. It is proven to be secured in the selective ID model agains collusion attacks and chosen plain attacks. However, in terms of performance, the proposed scheme works a bit slower compared to other schemes like Han *et al.*

*4) Revocable Decentralized Attribute-Based Encryption( by Prince):* The introduction of decentralized multi-authority thwarted most drawbacks of the single authority and brought about flexibility where different users with arbitrary attributes are given various forms of access to different types of encrypted data. In the real world, attributes of users could change periodically. This prompted researchers to investigate revocation in ABE where occasional key updates would allow only the eligible non-revoked users to be able to decrypt recently encrypted data.

Revocation in ABE can be described in two major ways:

1. *indirect revocation* [68]. This form of revocation invokes key updates from the authority periodically, such that only non-revoked users' keys can receive the available updates, thereby rendering revoked users' keys useless.

2. *direct revocation* [68]. This form of revocation invokes key updates from the sender directly. This he does by stipulating the revocation list when encrypting the ciphertext.

Attrapadung *et al.* [68] proposed a user-revocable ABE scheme by combining broadcast encryption schemes with ABE schemes where the data owner must take complete responsibility of maintaining the membership lists for each attribute group to ensure the direct user revocation. This scheme will not work on data outsourcing platform, since the data owner will no longer have direct control of data distribution after outsourcing the data to the external service providers.

Liang *et al.* [69] offered a CP-ABE scheme with efficient revocation. Their construction is built on the linear secret sharing and binary tree techniques, and proven secure in the standard model. Besides the attribute set, users are also tagged with a unique identifier which can be used to revoke easily.

All the above schemes[68][69] support user revocation, but they have no effect on attribute revocation.

**Revocable and Decentralized Attribute-Based Encryption**

Recently Cui *et al.* [70]presented a decentralized ciphertext-policy ABE (CP-ABE) system supporting indirect revocation which splits the exclusive AA's role across multiple AAs such that the AAs can indirectly accomplish revocation by stopping updating the keys for the revoked users. The splitting of roles across multiple AAs, reduces the computational overhead at the same time keeping the system decentralized where by any party can become an authority by creating a public and private key pair.

To prevent revoked users from combining their keys with non-revoked users (collusion), Cui *et al.* bonded the time period and the global identifier during the key generation process at the same time keeping the global identifier away during the encryption stage.

Comparatively, Cui *et al.*'s scheme could be considered scalable as against [68] and [71]. this is because an attribute can freely be added or revoked at any time without modifying the operation of the system.

This scheme is securely constructed in the standard model. The dual encryption security technique is applied during the security proof. The keys and the ciphertexts are divided as normal and semi-functional: the normal keys can decrypt the semi-functional ciphertexts, the semi-functional keys can decrypt the normal ciphertexts, but the semi-functional keys cannot decrypt the semi-functional ciphertexts [72].

*5) Multi-authority And distributed ABE (by Ziheng Ding):* In traditional Attribute Based Encryption(ABE) scheme, it cannot solve the problem when attributes belong to diverse servers, since these servers cannot be totally trusted by others. For instance, if attributes which are ID number and major of one user belong to governmental authority and college respectively, information is unable to be shared because government and college do not believe each other. To solve this problem, the first solution titled Multi-Authority Attribute Based Encryption[50] was proposed. Two methods was introduced in this scheme. The first one is Global Identifier(GID), which means every receiver is given a unique number illustrating their identity. GID are able to be verified by all authorities, while no one would have access to it expect the user himself. Another tool is Central Authority which is totally trusted by all users and other authority. The specific scheme was divided into 5 steps:

**System** It output the system public key and security parameter and system public key $\mathrm{Y}_0$.

**Attribute Authority k** For authority k, authority secret key ($\mathrm{s}_k$,$\mathrm{t}_{k,1}$...$\mathrm{t}_{k,n}$,) and authority public key ($\mathrm{T}_{k,1}$...$\mathrm{T}_{k,n}$) is established in this step.Besides, user u would have a unique secret key $\mathrm{D}_{k,i}$ here.

**Central Authority** Every authority gets a central authority secret key $\mathrm{s}_k$ here. Like other authority, a secret key $\mathrm{D}_{CA}$ would be established. Noteworthy is that includes the secret key of all authorities.

**Encrypt** this algorithm that takes a message m as input and the cipher text as output (E,$\mathrm{E}_(CA)$).

**Decrypt** Receiver has to recover the message according to its attributes.

The advantages are obvious. Central authority ensure that the scheme is safe even if some normal authorities are corrupted. To be more specific,when some are dishonest,the secret key of Central authority could prevent illegal decryption

TABLE VIII
THE COMPARISON IN THE LENGTH OF USER'S KEY AND CIPHERTEXT

| scheme | private key | ciphertext | encryption | decryption | security |
|---|---|---|---|---|---|
| [50] | $nG_1$ | $nG_1+G_2$ | $t_p+nt_{e1}+t_{e2}$ | $t_p+kt_{e2}$ | Bilinear BDH assumption |
| [57] | $nG_1$ | $nG_1+G_2$ | $t_p+nt_{e1}+t_{e2}$ | $2t_p+(2k+1)t_{2e}$ | Decisional BDH ssumption |
| [59] | $nG_1$ | $(2n+1)G_1+G_2$ | $t_p+3nt_{e1}+t_{e2}$ | $(n+1)t_{e2}+(2n+1)t_p$ | Number-Theoretic assumptions |
| [76] | $4G_1$ | $(2n+1)G_2+4nG_1$ | $t_p+4kt_{e1}+(2k+1)t_{e2}$ | $2t_p+3t_{e2}$ | Complexity assumption |
| [74] | $(3n+k)G_1$ | $(2n+1)G_1+G_2$ | $t_p+(2n+1)t_{e1}+t_{e2}$ | $(2k+5)t_p+(2k+5)t_{e2}$ | Decisional BDH assumption |
| [77] | $2G_1$ | $(5n+1)G_1+G_2$ | $t_p+(3w+1)t_{e1}+t_{e2}$ | $(3n+1)t_{e2}+(3n+1)t_p$ | z-type assumption |

by other attributes which are still honest. It is also arguable that authorities are able to decrease their burden distributing secret keys as well as monitoring attributes. Since each attribute is monitored by a diverse authority and central authority monitor none. In addition,an increasing number of schemes was proposed to perfect Multi-Authority ABE with more functions. For example,in MA Verifiable ABE[73], user has ability to check which authority gives wrong secret key and when key of authority fails in verification, he only need to send corresponding part again. In MA-ABE Scheme with Revocation [74], revocation could be achieved by using attributes classification management.Cloud computing is also utilized in MA-ABE which is introduced in Online/offline unbounded MA-ABE for data sharing in mobile cloud computing [75]. On the other hand, central authority are prone to bring certain risks that if it was corrupted, the whole scheme would be broken. Due to reducing the severe reliance on central authority, the Secure threshold multi authority attribute based encryption without a central authority was introduced[57]. Two protocols, distributed key generation protocol (DKG) and joint zero secret sharing protocol (JZSS), make the major contribution.Another approach is that several authorities could be in charge of the qualified specific attributes' key distribution in Efficient Statically-Secure Large-Universe Multi-Authority Attribute-Based Encryption[76].

In terms of security,a wide range of security models is used in these proposals,which lies in the table1.

table2 and table 3 contains the comparison in the length of user's key and ciphertext and performance comparison. $t_{e1}$,$t_{e2}$ denotes the time spent on two different bilinear group operations, $t_p$ denotes the time spent on pairing operation, k means the number of authority. The table compares the efficiency of diverse scheme.

## D. Revocation (by Xiping Zhang)

Revocation mechanism is necessary for any encryption schemes that involve many users, since a user′s permissions change and key leakage may happen with time. Attrapadung, Imai *et al.* [68] first divided revocation mechanism into two types: direct revocation and indirect revocation according to the different executor. In an indirect revocation scheme, the key authority, who possesses the current revocation list, periodically announces a key update material at each time slot so that only non-revoked users can update their key and use it to decrypt ciphertexts encrypted at the present time. In a direct revocation scheme, senders are able to specify the revocation list directly when encrypting. But both schemes have some problem: The indirect revocation scheme the key update phase can be a bottleneck since it requires communication from the key authority to all non-revoked users at all time slots.The direct revocation scheme requires senders to possesses the current revocation list. It could be a troublesome task to manage the revocation list. Attrapadung,Imai *et al.* [68] designed a hybrid revocable ABE scheme that allows the senders to select when encrypting whether to use either direct or indirect revocation mode, and the reciever possesses only one key but will be able to decrypt ciphertexts that were constructed in either modes.

The directly revocation CP-ABE scheme is as follows:

**Setup** The algorithm first picks a random generator g, v, $h_0, \ldots, h_{m'} \in G$ and random $\alpha$, a, $b \in Z_p$. The public key is pk=(g, $g^a$, $g^{b^2}$, v, $v^b$, $g^a$, $h_0$,..., $h'_m$,e(g,g)). The master key is msk=($\alpha$,b). It outputs(pk,msk). Define a function F:Zp$\to$G by $F(x)=\prod_{j=0}^{m'} h_j^{(x^j)}$

**Encrypt**(S,(M,),M,pk) Inputs to the encryption algorithm are a user index set S$\supset$U and a LSSS access structure (M, ) for subjective policy. Let M be $l_s \times k_s$ matrix. Let R=U$\backslash$S. Denote R= /$ID_1$,...$ID_r$/. The algorithm first randomly chooses s, $y_2$,...,$y_{k_s} \in Z_p$ and lets u=(s,$y_2$,...,$y_{k_s}$). For i= 1 to $l_s$, it calculates $_i$=$M_i \cdot$u, where $M_i$ is the vector corresponding to ith row of M. It also chooses random $s_1$,...$s_r \in Z_p$, such that s= $s_1$+ ...+ $s_r$. The ciphertext ct is set to ct=(C,$C_1$,$\{C_i^{(2)}\}_{i\in[1,l_s]}$, $\{C_i^{(3)}\}_{j\in[1,r]}$,$\{C_i^{(4)}\}_{j\in[1,r]}$), where

$$C = \mathcal{M} \cdot (e(g,g)^{\alpha})^s, \quad C^{(1)} = g^s, \quad C_i^{(2)} = g^{a\lambda_i} F(\rho(i))^{-s}, \quad C_j^{(3)} = g^{b \cdot s_j}, \quad C_j^{(4)} = (g^{b^2 \cdot ID_j} v^b)^{s_j}.$$

**KeyGen**(ID, $\rho$, msk, pk) Inputs to the encryption algorithm are a user index ID$\in$U and an attribute set$\psi$ $\supset$N. The algorithm randomly chooses t,r$\in$ $Z_p$. It outputs the private key as sk=($D^{(1)}$,$D^{(2)}$,$\{D_x^{(3)}\}_{x\in\psi}$,$D^{(4)}$,$D^{(5)}$) where

$D^{(1)}=g^{\alpha+b^2 t} \cdot g^{ar}$, $D^{(2)}=g^r$, $D_x^{(3)}=F(x)^r$, $D^{(4)}=(g^{b \cdot ID} v)^t$, $D^{(5)}=g^t$.

**Decrypt**(ct,(S,(M,$\rho$)),sk,(ID,$\psi$),pk) Suppose that the attribute set $\psi$ satisfies the access structure (M, $\rho$)) and the user index ID$\in$ S (so that the decryption is possible). Let $I_s=\{i\rho(i) \in \psi\}$.It then calculates corresponding sets of reconstruction constants$\{(i,\mu_i)\}_{i\in I_s}$=$Recon_{(M,\rho)}(\psi)$.Then it computes

TABLE IX
COMPARISON AMONG REVOCATION

| Scheme | Types | Directly | Indirectly | Ciphertext Size | SK Size | Decryption cost |
|---|---|---|---|---|---|---|
| [82] | CP | $\surd$ | $\times$ | (2r+1)$\lvert G_1\rvert$ | $3\lvert G_1\rvert+\lvert G_2\rvert$ | 3P |
| [83](scheme1) | KP | $\surd$ | $\times$ | $(\omega+2)\lvert G_1\rvert+\lvert G_2\rvert$ | $2l_0\lvert G_1\rvert$ | $(2l_0+2)P+l_0E_{G2}$ |
| [83](scheme2) | KP | $\surd$ | $\times$ | $(2r+\omega+1)\lvert G1\rvert+\lvert G2\rvert$ | $(2l_0+2)\lvert G1\rvert$ | $(2l_0+2r)P+(l_0+r)E_{G2}$ |
| [83](scheme1) | CP | $\surd$ | $\times$ | $(l_s+2)\lvert G1\rvert+\lvert G2\rvert$ | $(2l_s+3)P+l_sE_{G2}$ | $3\lvert G1\rvert$ |
| [83](scheme2) | CP | $\surd$ | $\times$ | $4\lvert G1\rvert+\lvert G2\rvert$ | $5\lvert G1\rvert$ | $(2l_s+2r+1)P+(l_s+r)E_{G2}$ |
| [69] | KP | $\surd$ | $\times$ | $(3+l_{n_{max}})\lvert G1\rvert+\lvert G2\rvert$ | $(n_{max}+\lvert S\rvert+3)\lvert G1\rvert$ | $(n_{max}+6)P+(n_{max}\lvert I\rvert+\lvert I\rvert+2)E_{G2}$ |
| [79] | CP | $\surd$ | $\times$ | $(2\omega-j+2)\lvert G1\rvert+\lvert G2\rvert$ | $(4l+n)\lvert G1\rvert$ | $2(\omega-j)P+(\omega-j)E_{G2}$ |

$$K=\frac{e(C^{(1)},D^{(1)})}{\prod_{i=1}^{l_s}\left(e(C_i^{(2)},D^{(2)})\cdot e(C^{(1)},D_{\rho(i)}^{(3)})\right)^{\mu_i}}\cdot$$
$$\prod_{j=1}^{r}\left(\frac{e(D^{(5)},C_j^{(4)})}{e(D^{(4)},C_j^{(3)})}\right)^{1/(ID-ID_j)}$$
where it can compute since ID=$\mathrm{ID}_j$ for j=1,...,r. It then obtains M=C/K.

In order to reduce the costs during the update phase,Liang, Xiaohui, Lu, Rongxing [69] designed a efficient scheme which used the binary tree technique to build a revocation tree. The revocation tree corresponds to time t and the identifier of revoked user is uid which is associated with one leaf node. In comparison with the traditional ciphertext policy attribute based encryption,the size of user secret key is increased by multiplying logn. In this scheme,the system manager should only publishes the revocation information according to a time stamp.The primary trigger control of the users access ability is the update information.In direct revocation, Pratish Datta, Ratna Dutta [78] combined some existing encrypt technology and revocation technology, achieve very short ciphertext size without imposing any extra overhead on the decryption key for the added revocation functionality and reduce the number of group elements in the public parameters to log$N_{max}$

The revocation can also execute on uesrs$'$ one attribute. Li, Qiang, Feng, Dengguo [79] implant attributes into usersprivate key.It can revoke a user$'$s one attribute without influencing his private key, if the revoked users other attributes also satisfy the access structure, he can decrpyt the cipertext successfully.This realize fine-gained attribute revocation under direct revocation model.

There are also a direct revocation scheme called third-part revocation. The bring in of third-part to execute revocation can reduce the authoritys work. Shi, Yanfeng,Zheng, Qingji *et al.* researched this scheme and get ahead. In [80], the ciphertext is divided into two part: the data and the authorize(identity)part,the trusted authority is allowed to revoke users by updating the revocation list and the third part is allowed to update ciphertexts with public information.

In addition,its important to avoid the abuse of key, that is to trace the malicious user, Liu, Zhen,Wong, DuncanS [81] make great progress on blackbox trace, which is highly expressive and achieves the most efficient level to date.
Some comparison of the efficiency of the schemes are shown on Table IX.

## VI. Related work(by Lei xu)

With the exploration on ABE from various aspects concluding efficiency, security and function, the variety of ABE schemes have been growing. However, there has been no integrated overview that contains not only various ABE schemes, but also analysis of design philosophy. Cheng-Chi Lee *et al.* [84] do the survey on ABE of access control, but just enumerates the CP-ABE, KP-ABE, ABE scheme with Non-Monotonic Access Structures and so on, so that we cannot obtain the clear relationship among those schemes from the paper. Moreover, the significant LSSS and circuits do not be mentioned. From the standpoint of difficulties in ABE, Jin-Shu Su *et al.* [85] describe the access structure, attribute revocation, multi-authority scheme and so on. Nevertheless, we cannot find the Non-Monotonic access Structure and circuits in the part of access structure from this article. Besides, the ABE variants such as proxy re-encryption, outsource ABE are not concluded. Deng-Guo Feng and Cheng Chen [1] point out the direction of development on hot topics of ABE, However, the survey on the most essential access policy is not embodied.

## VII. Future work

Future work goes here.

## VIII. Conclusion

The conclusion goes here.

## References


[1] C. C. Deng-Guo Feng, “Research on attribute-based cryptography,” *Journal of Cryptologic Research*, vol. 1, no. 1, pp. 1–12, 2014.

[2] D. P. P. R. Mihir Bellare, Anand Desai, “Relations among notions of security for public-key encryption schemes,” *Advances in Cryptology - CRYPTO: Annual International Cryptology Conference*, pp. 26–45, 1998.

[3] A. S. B. W. Vipul Goyal, Omkant Pandey, “Attribute-based encryption for fine-grained access control of encrypted data,” *ACM conference on Computer and communications security*, pp. 89–98, 2006.

[4] L. Cheung and C. Newport, “Provably secure ciphertext policy abe,” *ACM conference on Computer and communications security*, pp. 456–465, 2007.

[5] J. K. Ran Canetti, Shai Halevi, “Chosen-ciphertext security from identity-based encryption,” *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 207–222, 2004.

[6] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” *Advances in Cryptology - CRYPTO: Annual International Cryptology Conference*, pp. 213–229, 2001.

[7] S. W. J. B. Zhiguang Qin, Hu Xiong, “A survey of proxy re-encryption for secure data sharing in cloud computing,” *IEEE Transactions on Services Computing*, 2016.

[8] B. W. John Bethencourt, Amit Sahai, "Ciphertext-policy attribute-based encryption," *IEEE symposium on security and privacy*, pp. 321–334, 2007.

[9] Q. W. B. Q. Y. Z. Weiran Liu, Jianwei Liu, "Practical direct chosen ciphertext secure key-policy attribute-based encryption with public ciphertext test," *Computer Security - ESORICS: European Symposium on Research in Computer Security*, pp. 91–108, 2014.

[10] B. W. Amit Sahai, "Fuzzy identity based encryption," *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, 2005.

[11] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612–613, Nov. 1979.

[12] A. N. K. O. M. S. Keita Emura, Atsuko Miyaji, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," *International Conference on Information Security Practice and Experience*, pp. 13–23, 2009.

[13] C. R. Javier Herranz, Fabien Laguillaumie, "Constant size ciphertexts in threshold attribute-based encryption," *Public Key Cryptography-PKC: International Workshop on Public Key Cryptography*, pp. 19–34, 2010.

[14] N. Doshi and D. C. Jinwala, "Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption," *Security and Communication Networks*, vol. 7, pp. 1988–2002, 2014.

[15] W. S. D. S. W. V. V. Fuchun Guo, Yi Mu, "Cp-abe with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, 2014.

[16] P. H. W. J. Luan Ibraimi, Qiang Tang, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," *International Conference on Information Security Practice and Experience*, pp. 1–12, 2009.

[17] T. Teruya and S. Yamada, "Attribute based encryption with direct efficiency tradeoff," *Applied Cryptography and Network Security*, p. 249, 2016.

[18] H.-M. R. Chun-I Fan, Vincent Shi-Ming Huang, "Arbitrary-state attribute-based encryption with dynamic membership," *IEEE Transactions on Computers*, vol. 63, pp. 1951–1961, 2014.

[19] M. P. Rakesh Bobba, Himanshu Khurana, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," *Computer Security - ESORICS: European Symposium on Research in Computer Security*, pp. 587–604, 2009.

[20] Y. D. Bo Lang, Xu Runhua, "Extending the ciphertext-policy attribute based encryption scheme for supporting flexible access control," *Security and Cryptography (SECRYPT), 2013 International Conference on*, pp. 1–11, 2013.

[21] M. J. M. J.-M. S. Liu Ximeng, Zhu Hui, "Ieee international conference on communications workshops," *IEEE International Conference on Communications Workshops*, pp. 694–699, 2014.

[22] C. N. Ling Cheung, "Provably secure ciphertext policy abe," *ACM conference on Computer and communications security*, pp. 456–465, 2007.

[23] J. W. Guojun Wang, Qin Liu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," *ACM conference on Computer and communications security*, pp. 735–737, 2010.

[24] D. H. Zhibin Zhou, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," *ACM conference on Computer and communications security*, pp. 753–755, 2010.

[25] Z. W. Y. L.-S. L. Hui Ma, Rui Zhang, "Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, 2015.

[26] W. S. Tran Viet Xuan Phuong, Guomin Yang, "Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 35–45, 2016.

[27] B. W. Rafail Ostrovsky, Amit Sahai, "Attribute-based encryption with non-monotonic access structure," *ACM conference on Computer and communications security*, pp. 195–203, 2007.

[28] S. H. A. S. Sanjam Garg, Craig Gentry, "Attribute-based encryption for circuits from multilinear maps," *Advances in Cryptology - CRYPTO: Annual International Cryptology Conference*, pp. 479–499, 2013.

[29] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," *Public Key Cryptography-PKC: International Workshop on Public Key Cryptography*, pp. 53–70, 2011.

[30] A. Beimel, "Secure schemes for secret sharing and key distribution," *PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel*, 1996.

[31] H. L. Xiaohui Liang, Zhenfu Cao, "Attribute based proxy re-encryption with delegating capabilities," *International Symposium on Information, Computer, and Communications Security*, pp. 276–286, 2009.

[32] W. Z. W. M. Jung Taeho, Li Xiang-Yang, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 190–199, 2015.

[33] M. Karchmer and A. Wigderson, "On span programs," *The Eighth Annual Structure in Complexity Theory*, pp. 102–111, 1993.

[34] B. W. Allison Lewko, "Decentralizing attribute-based encryption," *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588, 2011.

[35] B. W. Matthew Green, Susan Hohenberger, "Outsourcing the decryption of abe ciphertexts," *Usenix Security Symposium*, 2011.

[36] C. G. J. W. Junzuo Lai, Robert H. Deng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, 2013.

[37] S. L. S. M. Baodong Qin, Robert H Deng, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, 2015.

[38] H. M. M. W. Suqing Lin, Rui Zhang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, 2015.

[39] J. L. X. C. Y. X. Jin Li, Xinyi Huang, "Securely outsourcing attribute-based encryption with checkability," *IEEE transactions on parallel and distributed systems*, 2014.

[40] M. B. B. Strauss, "Divertible protocols and atomic proxy cryptography," *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 127–144, 1998.

[41] G. A. K. F. G. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, pp. 1–30, 2006.

[42] M. Green and G. Ateniese, "Identity-based proxy re-encryption," *Applied Cryptography and Network Security*, pp. 288–306, 2007.

[43] C.-K. Chu and W.-G. Tzeng, "Relaxing chosen-ciphertext security," *Information Security: International Information Security Conference*, pp. 189–202, 2007.

[44] T. Matsuo, "Proxy re-encryption systems for identity-based encryption," *International Conference on Pairing-Based Cryptography*, pp. 247–267, 2007.

[45] J. W. Q. X. Shanqing Guo, Yingpei Zeng, "Attribute-based re-encryption scheme in the standard model," *Wuhan University Journal of Natural Sciences*, vol. 13, pp. 621–625, 2008.

[46] V. V. Nishanth Chandran, Melissa Chase, "Functional re-encryption and collusion-resistant obfuscation," pp. 404–421, 2012.

[47] W. S. D. S. W. Kaitai Liang, Liming Fang, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," *IEEE International Conference on Intelligent Networking and Collaborative Systems*, pp. 552–559, Sep. 2013.

[48] W. S. D. S. W. G. Y. Y. Y. Kaitai Liang, Man Ho Au, "An adaptively cca-secure ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *International Conference on Information Security Practice and Experience*, pp. 448–461, 2014.

[49] R. D. Y. Sreenivasa Rao, "Decentralized ciphertext-policy attribute-based encryption scheme with fast decryption," *IFIP International Conference on Communications and Multimedia Security*, vol. 8099, pp. 66–81, 2013.

[50] M. Chase, "Multi-authority attribute based encryption," *Theory of Cryptography Conference*, pp. 515–534, Feb. 2007.

[51] A. S. K. T. B. W. Allison Lewko, Tatsuaki Okamoto, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 62–91, 2010.

[52] E. Yuan and J. Tong, "Attributed based access control (abac) for web services," *International Conference on Web Services*, p. 569, 2005.

[53] A.-R. S. Andr Adelsbach, Ulrich Huber, "Property-based broadcast encryption for multi-level security policies," *International Conference on Information Security and Cryptology*, pp. 15–31, 2006.

[54] S. W. S. Apu Kapadia, Patrick P. Tsang, "Attribute-based publishing with hidden credentials and hidden policies," *ISOC Network and Distributed System Security Symposium*, pp. 179–192, March 2007.

[55] K. E. S. Robert W. Bradshaw, Jason E. Holt, "Concealing complex policies with hidden credentials," *ACM conference on Computer and communications security*, pp. 146–157, 2004.

[56] C. E. Sascha Mller, Stefan Katzenbeisser, "Distributed attribute-based encryption," *International Conference on Information Security and Cryptology*, pp. 20–36, 2008.

[57] X. L. J. S. Huang Lin, Zhenfu Cao, "Secure threshold multi authority attribute based encryption without a central authority," *Progress in Cryptology - INDOCRYPT: International Conference on Cryptology in India*, pp. 426–436, 2008.

[58] S. S. Melissa Chase, "Improving privacy and security in multi-authority attribute-based encryption," *ACM conference on Computer and communications security*, pp. 121–130, Nov. 2009.

[59] Q. H. D. S. W. T. H. Y. Zhen Liu, Zhenfu Cao, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," *Computer Security - ESORICS: European Symposium on Research in Computer Security*, pp. 278–297, 2011.

[60] V. Shoup, "Lower bounds for discrete logarithms and related problems," *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 256–266, 1997.

[61] E.-J. G. Dan Boneh, Xavier Boyen, "Hierarchical identity based encryption with constant size ciphertext," *Advances in Cryptology - EUROCRYPT: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 440–456, 2005.

[62] R. L. J. X. X. L. Qi Li, Jianfeng Ma, "Large universe decentralized key-policy attribute-based encryption," *Security Comm. Networks*, vol. 8, p. 501509, 2014.

[63] Y. M. Jinguang Han, Willy Susilo, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE transactions on parallel and distributed systems*, vol. 23, pp. 2150–2162, 2012.

[64] R. Z. C. M. Z. Z. Aijun Ge, Jiang Zhang, "Security analysis of a privacy-preserving decentralized key-policy attribute-based encryption scheme," *IEEE transactions on parallel and distributed systems*, vol. 24, pp. 2319–2321, 2013.

[65] Y. M. J. Z. M. H. A. Jinguang Han, Willy Susilo, "Ppdcp-abe: privacy-preserving decentralized ciphertext-policy attribute-based encryption," *Computer Security - ESORICS: European Symposium on Research in Computer Security*, pp. 73–90, 2014.

[66] C. C. Minqian Wang, Zhenfeng Zhang, "Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme," *Concurrency And Computation: Practice And Experience*, vol. 28, pp. 1237–1245, 2015.

[67] A. R. C. S. Jan Camenisch, Markulf Kohlweiss, "Blind and anonymous identity-based encryption and authorized private searches on public key encrypted data," *Public Key Cryptography-PKC: International Workshop on Public Key Cryptography*, pp. 196–214, 2009.

[68] H. I. Nuttapong Attrapadung, "Attribute-based encryption supporting direct/indirect revocation modes," *IMA International Conference on Cryptography and Coding*, pp. 278–300, 2009.

[69] X. L. X. S. S. Xiaohui Liang, Rongxing Lu, "Ciphertext policy attribute based encryption with efficient revocation," 2010.

[70] R. H. D. Hui Cui, "Revocable and decentralized attribute-based encryption," *The Computer Journal*, 2016.

[71] V. K. Alexandra Boldyreva, Vipul Goyal, "Revocable and decentralized attribute-based encryption," *ACM conference on Computer and communications security*, pp. 417–426, 2008.

[72] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," *Advances in Cryptology - CRYPTO: Annual International Cryptology Conference*, pp. 619–636, 2009.

[73] Q. TANG and D. JI, "Multi-authority verifiable attribute-based encryption," *Journal of Wuhan University (Natural Science Edition)*, vol. 5, p. 024, 2008.

[74] B. Q. Z. L. XiaoFang Huang, Qi Tao, "Multi-authority attribute based encryption scheme with revocation," *IEEE International Conference on Computer Communications and Networksks*, pp. 1–5, 2015.

[75] Q. L. J. L. H. L. Yinghui Zhang, Dong Zheng, "Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing," *Security and Communication Networks*, vol. 9, p. 36883702, Nov. 2016.

[76] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," *International Conference on Financial Cryptography and Data Security*, pp. 315–332, July 2015.

[77] R. L. J. X. X. L. Qi Li, Jianfeng Ma, "Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption," *Security and Communication Networks*, vol. 8, p. 40984109, Aug. 2015.

[78] S. M. Pratish Datta, Ratna Dutta, "General circuit realizing compact revocable attribute-based encryption from multilinear maps," *Information Security: International Information Security Conference*, pp. 336–354, Aug. 2015.

[79] L. Z. Qiang Li, Dengguo Feng, "An attribute based encryption scheme with fine-grained attribute revocation," *IEEE Global Communications Conference*, pp. 885–890, 2012.

[80] J. L. Z. H. Yanfeng Shi, Qingji Zheng, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Information Sciences*, vol. 295, pp. 221–231, 2015.

[81] Z. Liu and D. S. Wong, "Practical attribute-based encryption: Traitor tracing, revocation and large universe," *The Computer Journal*, p. bxv101, Nov. 2015.

[82] S. D. G. Kopeetsky, "Permanent revocation in attribute based broadcast encryption," *Cyber Security, 2012 International Conference on*, pp. 203–208, Dec. 2012.

[83] N. A. Imai, "Conjunctive broadcast and attribute-based encryption," *International Conference on Pairing-Based Cryptography*, pp. 248–265, 2009.

[84] M.-S. H. Cheng-Chi Lee, Pei-Shan Chung, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 14, pp. 231–240, 2013.

[85] X.-F. W. Y.-P. S. Q.-L. H. Jin-Shu Su, Dan Cao, "Attribute based encryption schemes," *Journal of Software*, vol. 22, pp. 1299–1315, June 2011.


PLACE PHOTO HERE

**aaa** Biography text here.

PLACE PHOTO HERE

**bbb** Biography text here.

PLACE PHOTO HERE

**ccc** Biography text here.