A Survey Of Attribute-based Encryption Schemes

Qxx

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

***Index Terms*—The keywords goes here[?].**

I. INTRODUCTION

introduction goes here.

II. DEFINITION

introduction of framework and security model.

A. Framework Of ABE

Framework Of ABE goes here.

B. Security Model

introduction of security model.

1) Random Oracle And standard model: Random Oracle And standard model goes here.

2) Select Security And Full Security: Select Security And Full Security goes here.

3) CPA and CCA: CPA and CCA goes here.

III. DESIGN PHILOSOPHY OF CLASSICAL ABE SCHEME

some introduction goes here.

A. Access Control

As mentioned in introduction, access control and attribute as object of authorization are most essential feature of ABE.

Firstly, the least fineness which still has the function of object of authorization in ABE turn into "attribute", instead of "identity" in IBE. each attribute only have two state that users have or not have one, in the beginning of ABE schemes, which implies more states can be concluded in rear schemes. and users are authorized, capacity to decrypt legitimately, in the way that proper access policy are designed by data owners, according to the attributes users have.

Then, the "access control" is just that access policy mentioned above. Attributes of an user often are concluded in a set due to the convenience for the access control, And in order to express what attribute sets of users can be used to decrypt, intuitionally, we can enumerate all situation, and the access structure $\mathbb{A}$ is the collection of those sets, the attribute set can be used to decrypt, if and only if it *belongs to* the set $\mathbb{A}$, which trends to be a specifical interpretation of the statement, "attributes *satisfy* the policy", from most present ABE schemes. we call expression above as *enumeration*. However, this expression on access control is not brief at all. In consideration of the case above, Boolean Formula is thought of. In Boolean Formula, the various relations, which are designed by data owners, among attributes are pointed out directly. Those relations contain threshold, *AND*, *OR* and *NOT*. As far as comprehension on relevance among attributes, Boolean Formula perform most directly. Therefore, many explorations on how to realize in specific schemes have been done since the time of birth of ABE. So far, The *access tree* and *circuits* have been accepted as the solution. More consideration that whether there is a black-box way to achieve access control is made. In this way, inputs are attributes from each user, while output is the result that policy is satisfied or not, i.e. it is possible that we can pay little attention to the concrete relationship among attributes. The Boolean Function, f: $\{1,0\}^n \longrightarrow (0,1)$, is recalled as theory model of this method. The input of the Boolean Function is a n-dimension vector, and every dimension is a Boolean variable with the value of "1" or "0". In ABE context, from the perspective of the enumeration, each value of Boolean variable from an user depends on the relation between each attribute and authorized set. Then the output is a boolean variable with the values of "1" or "0". Analogously, the value depends on whether a user's attributes set belongs to the access structure. As for how to realize Boolean Function in specific ABE with monotone access structure, [**?**] utilized the conclusion by [**?**], [**?**], [**?**], [**?**], [**?**], [**?**] that for every linear secret-sharing scheme(LSSS) realizable access structure, there exists a monotone span program (MSP) that computes the corresponding Boolean function and vice versa. Monotonicity of access structure and LSSS are introduced in our below. The *access tree* and *circuits* are also surveyed in this subsection.

1) Monotonicity: Monotonicity is defined as follows:

Definition 1 (Monotonicity[?], [?], [?], [?], [?], [?]) *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*, therefore the non-monotone access structure should support *NOT*.

In the access control of present ABE schemes, most access structure is monotone, due to that normal methods to express attribute in access structure, such as $\rho(i)$ in LSSS and leaf node in access tree, only support the situation that each attribute is in a certain authorized set or not in, and don't support the one that an attribute is not in any authorized sets, which however is necessary to non-monotone access structure. So, in order to achieve non-monotone access structure, "negative attribute" is supposed ([**?**]). Main idea of the supposition is that attributes are divided into two values, positive and negative, and preinstall two kinds of SK and ciphertext so that all attributes(with positive and negative values) of users authorized

this is thanks

can be expressed in a certain authorized set. Nevertheless, the main problem of this method inclines to inefficient. Two values of each attribute imply the number of attributes in the system will be doubled too ([**?**]). As for this problem,[**?**] has found a solution that all attributes have positive values, and negative values of some attributes are added according to specific systems. Additionally, [**?**] has implicitly proposed a transformation technology to meet the monotone required by LSSS. i.e.this technology can transform non-monotone access structure into monotone one(not real), so that 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 from most of present ABE schemes, *access tree*, *circuits* and *LSSS*, and unless stated otherwise, by an access structure we mean a monotone access structure. Moreover, the comparison among the three methods has been made.

Before our survey, the *secret sharing* in access control should be firstly introduced. To put it simply, secret sharing in ABE schemes is the operation that information distributed amongst an authorized attribute set achieve reconstructing, while each or many but not all attributes in an authorized set(suppose these attributes can not be a collection of another authorized set) are of no use on their own. We can say achieving access control is just for the purpose realizing secret share in all concrete ABE schemes, and detailedly obtaining the secret value, usually s. After acquire s in certain form, user can decrypt successful. For the reason that the ABE syntax has been provided, now the discusses here is just up to gain of secret value s. Additionally, we denote several common notions in our three expressions as shown on Table I.

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 I 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 authorized 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 detailedly 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 authorized set. In fact, process above is the reverse of secret sharing, the distribution for secret value s.

The set part holds the authorized 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 in Table I. For each non-leaf node x, evaluate $mathcalT_{x\prime}(\gamma)$ 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 authorized 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

Firstly, "attributes *satisfy* the policy" can be interpret on the expression of access tree as that attributes from the authorized set 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, the satisfaction companies with the achieving secret sharing as our discuss in the beginning of this subsection. Thirdly, general access tree narrated as above is monotone for the reason that if a attribute set γ can achieve secret share, another set, which contains all elements of γ plus an arbitrary attribute having been defined by system, can also achieve that simply by not using the additional attribute. Fourthly, each node of a layer of access tree is just the structure of threshold secret sharing[**?**] 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

TABLE I
SEVERAL COMMON NOTIONS

Notions	Descriptions
p	the order of prime group
$P = \{P_1, P_2 \cdots P_u\}$	$P_i, i \in 1, 2 \cdots u$, denotes $i-th$ attribute as a unique number defined by system as a selected randomly number P denotes the set of u numbers corresponding all u attributes
γ	authorized set that can be used to decrypt successfully
$\mathbb{A}$	access structure in *enumeration*
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 correspondence between $\mathbb{A}$ and other forms blow.

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 [**?**][**?**], [**?**][**?**], [**?**], 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 is a 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 I is the number of inputs corresponding the set of subscripts of P_i shown on Table I, 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. $A : GatesWires$ is a function to identify each $gate's$ first incoming wire, and $B : GatesWires$ is a function to identify each $gate's$ second incoming wire. Finally, GT: $GatesAND, OR$ 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 ω is the same as the label of the outgoing wire from ω. For convenience of description, an extra function are defined here. First is layer of gate, $layer(\omega)$: the shortest path from gate ω 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 I.

we assume that the set part have known $e_{i,j}$ defined above, g_1, and for each attribute in the authorized 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 authorized 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 case one and case two, and the calculation is same as either case one or case two of one. The calculating do not stop until gets the $g_{l+1}^{r_{u+q}}$.

B. Design philosophy of classical ABE scheme

some introduction goes here.

1) KP-ABE: analyze article: *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data* Vipul Goyal Omkant Pandeyy Amit Sahaiz Brent Waters in Proc of Acmccs- 2006

2) CP-ABE: analyze article: *Ciphertext-Policy Attribute-Based Encryption* John Bethencourt Amit Sahai Brent Waters in 2007 IEEE Symposium on Security and Privacy 2007

IV. State-of-the-Art Of the ABE

A. the birth of ABE

Attribute-based encryption is firstly mentioned in[**?**],this idea originates from Hierarchical identity-based encryption schemes[**?**]and the schemes of [**?**] can be achieved due to the inspiration from threshold secret share technology by[**?**].After that,researchers pay much attention on general policy,which ties data owner with data user as a series of formalized constraint,compared with traditional point-to-point constraint(corresponding the privilege management infrastructure technology).So the access control technology is recalled,and ABE has perfectly formed just when this technology and the attribute as object of authorization are used in public-key scheme.When access control technology occurs to ABE,there are two types of models realized,i.e KP-ABE and CP-ABE.[**?**] is the first KP-ABE scheme and [**?**] is the first CP-ABE,so these two schemes proposed sign the perfect formation of ABE.After then,apart from research on more practical access policies,which are surveyed in section 3,the directions of development of ABE can be summarized those:fuction,efficiency and security.

B. efficiency

1:Survey on original ABE,and some efficient technology such us constant ciphertext and constant cost in decryption.

2:You describe these articles in time sequence.

3:And also need a table to compare efficiency among articles mentioned by you.

C. security

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

1:Survey on original ABE,and some articles,which have promoted in function,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 if possible.

V. Extension

some introduction goes here.

A. Key-abuse Problem

Key-abuse Problem goes here.

B. Outsource

Key-abuse Problem goes here.

C. Proxy Re-encrypton

Proxy Re-encrypton goes here.

D. Multi-authority And distributed ABE

Multi-authority And distributed ABE goes here.

E. Revocation

Revocation goes here.

F. hide-policy ABE

hide-policy ABE goes here.

VI. Related work

With the exploration on ABE from aspects of efficiency, security and function, more and more ABE variants has been growing. However, there has been no integrated overview published that contains not only various ABE schemes, but also analysis of design philosophy. [**?**] does the survey on ABE of access control, but just enumerate 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, and also Boolean formula of LSSS and circuit do not be mentioned. From the standpoint of difficulties in ABE, [**?**] describes the access structure, attribute revocation, key-abuse problem and multi-authority scheme, but we cannot find the Non-Monotonic Access Structures and Boolean formula of circuit in the part of access structure from this article. Besides, the ABE variants such as proxy re-encryption, outsource ABE are not concluded. [**?**] points 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.

PLACE PHOTO HERE

aaa Biography text here.

bbb Biography text here.

ccc Biography text here.