Verifiable Attribute-Based Encryption

Qiang Tang and Dongyao Ji
State Key Lab of Information Security ,
Graduaction University of Chinese Academy of Sciences
Beijing 100049
qtang84@gmail.com dyji@gucas.ac.cn

Abstract. In this paper, we construct two verifiable attribute-based encryption (VABE) schemes. One is with a single authority, and the other is with multi authorities. Not only our schemes are proved secure as the previous ABE schemes, they also provide a verification property. Adding the verification property has a few advantages: first, it allows the user to immediately check the correctness of the keys, if not, we only need the authority to resend the corresponding shares, especially, in multi-authority case, if the key does not pass the check, the user only needs to ask the particular authority to resend its own part, without need to go to all the authorities, this saves a lot of time when error appears, second, if the keys pass the verification but the user still does not rightly decrypt out the message, something might be wrong with the attributes or ciphertexts, then, the user has to contact with the encryptor. We formalize the notion of VABE and prove our schemes in our model.

Keywords: ABE, verifiable secret sharing, multi-authority, provable security

1 Introduction

Identity Based Encryption (IBE), introduced by Shamir [1], is a novel encryption which allows users to use any string as their public key (for example, an ID card number or an email address). Encrypting messages without access to a public key certificate reduces the load of creating and storing certificates. The first provably secure and elegantly designed IBE scheme was given by Boneh and Franklin [2], after that, IBE has received a lot of attention.

To better express identity and allow for a certain amount of error-tolerance, Sahai and Waters proposed fuzzy IBE [3], in their scheme, identity is viewed as a set of descriptive attributes, and a user with the secret key for the identity ω is able to decrypt a ciphertext encrypted with the public key ω' if and only if ω and ω' are with a certain distance of each other as judged by some metric.

In the paper [4], Goyal et al. developed a much richer type of ABE cryptosystem and demonstrated its applications. In their system each ciphertext is labeled by the encryptor with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of ciphertexts the key can decrypt. The access policy in their work is described by an access tree, which is more general than simple t-out-of-n threshold, and thus well suits for fine-grained access control of encrypted data and some other kind of applications.

In the paper [5] and [6], Cheung et al. and Bethencourt et al. respectively constructed a ciphertext policy ABE scheme, in which attributes are used to describe a user’s credentials, and a party encrypting data determines a policy for who can decrypt. This conception is closer to traditional access control methods.

All ABE schemes mentioned are with single authority, so Chase presented multi-authority ABE in [7] to answer an open question in [3], in multi-authority scenario, more than one authority are responsible for maintaining one kind of attributes, they operate simultaneously, and handle out secret keys for different set of attributes, and the load of the single authority is enormously reduced.

ABE schemes are mostly used in the scenario below: the encryptor creates the ciphertext and put it to the public environment, the user who wants to decrypt goes to the authority (below, we assume the authority and the key generation center are the same) to ask for his decryption key satisfying the policy. Thus, a key of a user would be stored and used many times later.

Our Contributions: we add a verifiable property to the ABE schemes, the core idea of the method is using some trick to change secret sharing [8] in the ABE schemes with verifiable secret sharing [9]. Compared to decrypting out the wrong message, this method has at least three advantages, especially in the multi-authority case:

1. If the key does not pass the verification, there must be something wrong with the process of generating the key, and we can just ask the authority to resend the corresponding shares, without need to repeat the whole process of generating the key.

2. If the keys pass the verification but the user still does not rightly decrypt out the message, there is a explicit notification that something might be wrong with the attributes or ciphertexts, in this situation, the user goes to the creator of the ciphertext.

3. In the case of multi-authority, each authority uses the VABE scheme, if the check does not passes in some single authorities, the user does not need to ask all authorities to resend the shares, but only needs to ask for the particular authorities to resend the corresponding shares computed by those authorities.

VABE may also potentially be used to construct other schemes as a building block. The security of these schemes have not been influenced, we only need to make some modifications that computing the values for verification to answer the new queries in the proof in previous ABE schemes to finish our security proof.

The paper is organized as follows: we give out the preliminaries in part 2, and definitions of VABE and its security in part 3, then we give a concrete construction with single authority and we prove ite security in part 5, we also give out a construction with multi-authorities in part 6, and at last, we draw a conclusion in part 7.

2 Preliminaries

2.1 Bilinear maps

Let G_1andG_2be two multiplicative cyclic groups of prime order p. Let g be a generator of G_1 and e be a bilinear map, $e: G_1 \times G_2 \to G_2$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in G_1$ and$a, b \in Z_p$, we have $e(u^a, v^b) = e(u^b, v^a) = e(u, v)^{ab}$
2. Non-degeneracy: $e(g, g) \neq 1$.
3. Efficiently computable.

2.2 The decisional bilinear Diffie-Hellman (BDH) assumption

Let $a, b, c, z \in Z_p$be chosen at random and g be a generator of G_1. The decisional BDH assumption is that no probabilistic polynomial-time algorithm $\mathfrak{I}$ can distinguish the tuple:

(A= g^a *, B=* g^b *, C=* g^c *,* $e(g,g)^{abc}$) from the tuple *(A=* g^a *, B=* g^b *, C=* g^c *,* $e(g,g)^z$ *)* with more than a negligible advantage. The advantage of $\Im$ is:

| *Pr[* $\Im$ *(A,B,C,* $e(g,g)^{abc}$ *)=0]-Pr[* $\Im$ *(A,B,C,* $e(g,g)^z$ *)=0]* |, where the probability is taken over the random choice of the generator g, the random choice of a,b,c,z in Z_p , and the random bits consumed by $\Im$.

3 VABE and its security model

3.1 Definition1: A VABE scheme is a special kind of key policy ABE scheme in which the correctness of key could be verified when the user gets the key from the key generation center, it includes following basic algorithms:

SETUP: This is a randomized algorithm that takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK

ENCRYPTION: This is a randomized algorithm that takes as input a message m, the public parameters PK, and attributes set γ . It outputs the ciphertext E.

KEY GENERATION: This is a randomized algorithm that takes as input the master key MK, the public parameter PK and policy (or access structure) Γ. It outputs a decryption key D, and verification information V. It is executed by the authority.

VERIFICATION: This is a deterministic algorithm that takes V and all public information as input, outputs 1 when V passes the check, else outputs 0.

DECRYPTION: If VERIFICATION outputs 1, then this algorithm is executed. It takes as input- the ciphertext E that was encrypted under the descriptive information D1, the decryption key D for Γ and the public parameter PK. It outputs the message M if $\gamma \in \Gamma$ *(or* $\Gamma(\gamma)$*=1).*

3.2 Definition2: If both of the conditions below are satisfied, then we say the ABE scheme is verified.

1. Assume that K is a PPT algorithm to generate two independent random numbers, K outputs different results even with the same input at different times. A is a PPT algorithm taking a node r, a parameter k, and attributes set γ as input, to create an authorized sub access structure Γ rooted at node r, and Γ satisfies that Γ (γ)=1, B is a PPT algorithm to reconstruct the secret message from a given access structure, for any k_0 and at anytime, the quantity $Adv(A, B, \gamma, r, k_0)$:

$$1 - Pr\ [k_{1,}k_2 \leftarrow K(k_0), \Gamma_1 \leftarrow_R A(k_1, r, \gamma), \Gamma_2 \leftarrow_R A(k_2, r, \gamma), B(\Gamma_1) = B(\Gamma_2)]$$

is a negligible function of k_0 .

2. If all the shares are right, the user could reconstruct the secret with probability 1, which means the user could decrypt out the right message.(we assume there is nothing wrong with the ciphertext and attributes)

3.3 security model for VABE

Our security model adds a verification query to the selective-set model in [4],

Init: The adversary declare the set of attributes, γ , that he wishes to be challenged upon.

Setup: The challenger runs the SETUP algorithm of GPSW ABE in [4] and gives the public parameters to the adversary.

Phase1: The adversary is allowed to issue queries for verification information and private keys for many access structure Γ_j, where $\Gamma_j(\gamma) \neq 1$for all j, and the adversary checks the correctness of the keys.

Challenge: The adversary submits two equal length messages M_0andM_1. The challenger flips a random coin b, and encrypts $M_b$with γ. The ciphertext is passed to the adversary.

Phase2: Phase1 is repeated.

Guess: The adversary outputs a guess b' of b.

The advantage of an adversary $\Im$ in this game is defined as *Pr[b'=b]- 1/2*

This model can be easily extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase1 and Phase2, and a scheme secure in this model is also easily be extended to be secure in chosen-ciphertext model using simulation sound NIZK proofs which presented in [10]. A GPSW ABE scheme is secure in the selective-set model of security if all polynomial time adversaries have at most a negligible advantage in the selective-set game.

4 A Concrete construction of VABE with single authority

The ABE scheme we use as a building block is the construction of Goyal et al in [4].

We first describe the tree structure used in this scheme, the access tree Γ, each non-leaf node represents a threshold gate, described by its children and a threshold value. A node x, has num_x children and a threshold value k_x. We also define the function parent(x) to return the parent node of x, and index(x) to return the index of x as a child of its parent. a leaf node x is defined by an attribute att(x).

Let Γ_x be the sub tree rooted at the node x, we compute $\Gamma(\gamma)$ in a recursive manner:

If x is a leaf node, $\Gamma_x(\gamma)$ returns 1 if and only if att(x)$\in \gamma$

If x is a non-leaf node, evaluate $\Gamma_{x'}(\gamma)$for all children x' of node x, $\Gamma_x(\gamma)$ returns 1 if and only if at least k_x children returns 1.

Now we demonstrate the construction as follows:

Let G_1 be a bilinear group of prime order p, and let g be a generator of G_1. In addition, let $e: G_1 \times G_2 \rightarrow G_2$ denote the bilinear map. A security parameter, k, will determine the size of the groups. We also define the Lagarange coefficient $\Delta_{i,S}$ for $i \in Z_p$ and a set S, of elements in Z_p: $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. We will associate each attribute with a unique element in Z_p^*.

Setup Define the universe of attributes U= {1, 2, … , n}. Randomly choose $t_1, .. t_n, y$ from Z_p. The published public parameters PK are ：

$T_1 = g^{t_1}, \ldots, T_{|U|} = g^{t_{|U|}}, Y = e(g,g)^y$. The master key MK is: $t_1, .. t_n, y$.

Encryption (M, γ, PK) To encrypt a message $M \in G_2$ under a set of attributes γ, choose a random number $s \in Z_p$ and publish the ciphertext as : $E = \left(\gamma, E' = MY^s, \{E_i = T_i^s\}_{i \in \gamma}\right)$.

Key Generation (Γ, MK) this process shares the secret y in a top-down manner with Shamir's threshold secret sharing scheme , for each non leaf node x, we choose a polynomial q_x with degree $d_x = k_x - 1$, make the polynomial satisfy $q_x(0) = q_{parent(x)}(index(x))$,and randomly fix other $d_x$points to completely define q_x, then compute $h_x = e(g,g)^{q_x(0)}$ and $C_x: \{e(g,g)^{a_i}\}_{i=1,..,k_x-1}$, $\{a_i\}$ are the non constant coefficients of the polynomial $q_x(\quad)$ used to share the secret of the node x. After all the polynomials are decided, for each leaf node x, we give the following set of secret values D to the user: $\Gamma_x, D_x = g^{q_x(0)/t_i}$, where i= att(x) and the additional values $h_x = e(g,g)^{q_x(0)}$, and for every other node x, we h_x and C_x. This process enables the user to decrypt a message encrypted under a set of attributes γ if and only if $\Gamma(\gamma)$=1.

Verification（Γ，**PK, {** h_x**}, {** C_x**}, D**） for leaf node x, after getting { D_x}, the user firstly checks whether $e(D_x, T_i) = h_x$, and then, verifies ：

$$h_x = e(g,g)^{q_{parent(x)}(index(x))},$$

$$= e(g,g)^{q_{parent(0)}+a_1(index(x))+..+a_{i-1}(index(x)^{i-1})}$$
$$= h_{parent(x)} \times \prod_{i=1}^{k-1} (e(g,g)^{a_i})^{index(x)^i}$$
k is the degree of the polynomial $q_{parent(x)}(\,)$, (1) if all the leaf nodes pass the verification, using h_x and equation (1) to verify the correctness of all other nodes level by level, until to the root node and at last checks whether $h_r = Y$. Assume the check fails at nodes $x_1, .., x_l$, if any node x_i's ancestor node is in this nodes set, we remove x_i from this set, at last, we get $z_1, .. z_t$, we ask the authority to resend the shares of the subtree $\Gamma_{z_1}, .., \Gamma_{z_t}$.

Decryption (E, D) we specify the decryption procedure in a bottom-up manner: Let i= att (x)

If x is a leaf node, then:

DecryptNode (E, D, x) returns $e(D_x, E_i) = e\left(g^{\frac{q_x(0)}{t_i}}, g^{st_i}\right) = e(g,g)^{sq_x(0)}$ if $i \in \gamma$, otherwise, returns $\perp$;

If x is a non-leaf node, we recursively compute the DecryptNode (E, D, x), the output of it is denoted as F_x, for all nodes z that are children of x, 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, the function returns _|_, otherwise, we compute:

$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)}$, where i=index(z), S_x'={index(z):z}

$= e(g,g)^{sq_x(0)}$, using Lagrange polynomial interpolation.

We can know that, at last when reaching the root node r, we get:

DecryptNode *(E, D, r)*= $e(g,g)^{sy} = Y^s$, so if the condition $\Gamma(\gamma)$=1 is satisfied, the user can decrypt.

5 Security proof for the VABE scheme

Theorem1. The concrete construction of verifiable ABE scheme is verified, which means it satisfies the two conditions in 3.2 Definition2, and the scheme is also secure in the selective-set model defined in 3.3 under the decisional BDH assumption.

Proof sketch: First, we observe the condition1 and condition2 in 3.2 definition1.

The additional information for verification of each leaf are commitments for the coefficients of the polynomials used in key generation phase. The real secret is y, but the user can not directly get the shares of y, so in the first step of check whether *e* (D_x, T_i) equals h_x to ensure that the very $q_x(0)$ in $D_x$is the same as that in h_x. The sharing process is to share y, so the polynomial in each step to finally get $D_x$and $h_x$is the same, if:

$$h_x = h_{parent(x)} * \prod_{i=1}^{k-1} (e(g,g)^{a_i})^{index(x)^i} \qquad (*)$$

passes, we can be sure that $h_x$is rightly computed from the polynomial of parent(x), namely, $q_x(0) = q_{parent(x)}(index(x))$ is rightly computed from that polynomial, thus, $q_{parent(x)}(0)$is shared without mistakes, while for degree d polynomial $q_{parent(x)}(\,)$, every qualified structure in this level at least contains $k_{parent(x)}$ values, while $k_{parent(x)} = d + 1$, so any qualified set of values uniquely decide the polynomial. Now, let's check the $Adv(A, B, \gamma, r, k_0)$in condition1 in definition1, if any of this quantity generated by some k_0 at some time is non-negligible, it means that from two qualified sub structure with the same root node, we get different secrets with a non-negligible quantity, and the difference must be from some level of sharing, then, in this level of sharing, at least two qualified sets of values reconstruct different secrets, which means at least one of them passes the test in (*) but is wrong, assume the node is x, that is, the key generation center intentionally or not computes a value a, and satisfying:

$$Q = e(g,g)^a = e(g,g)^{q_x(0)} = h_{parent(x)} * \prod_{i=1}^{k-1} (e(g,g)^{a_i})^{index(x)^i}$$

then the KGC could solve $log_{e(g,g)}Q = a - q_x(0)$ with a non-negligible quantity, however, it is negligible, that is a contradiction, so condition1 is satisfied.

And at last, we check $h_r = Y$ to ensure the initial secret is the same as the one in the public key. If all these checks are valid, the initial secret y is rightly shared in each step to the final pieces of sharing, the user could decrypt, then condition2 is satisfied.

Next, we show the security of our scheme. As in [4], we use A the adversary of attacking the ABE scheme with advantage ε to build a simulator B for solving the DBDH problem with advantage ε /2. The main difference between our proof and the one in [4] is the check query.

The challenger flips a fair binary coin μ , if μ =0, it sets the tuple (A,B,C,Z)= $(g^a, g^b, g^c, e(g,g)^{abc})$, else, it sets the tuple (A,B,C,Z)= $(g^a, g^b, g^c, e(g,g)^z)$,, for random a,b,c,z. The simulation proceeds as follows:

Init B runs A, Achooses the attributes set γ he wishes to attack.

Setup B sets the parameter *Y=e(A,B)=* $e(g,g)^{ab}$, for all i in the universe, if i∈ γ, set $T_i = g^{r_i}$, $\mathrm{r_i}$is randomly chosen from $\mathrm{Z_p}$, otherwise, set $T_i = B^{k_i}$, $\mathrm{k_i}$ is randomly chosen from $\mathrm{Z_p}$, then B gives the public parameters to A.

Phase1 A makes requests for keys corresponding to access structure Γ' that$\Gamma'(\gamma) \neq 1$. We define two functions Satpoly and Unsatpoly.

Satpoly$(\Gamma_\mathrm{x}, \gamma, \lambda_\mathrm{x})$ constructs the polynomials for the sub tree Γ_x and $\Gamma_\mathrm{x}(\gamma) = 1$. It first sets up a polynomial $\mathrm{q_x}$ of degree $\mathrm{d_x}$for the root node x and satisfying $\mathrm{q_x}(0) = \lambda_\mathrm{x}$. For each child node x' of x, we defines its polynomial by calling Satpoly($\Gamma_x, \gamma, q_x(index(x'))$).

Unsatpoly$(\Gamma_\mathrm{x}, \gamma, \mathrm{g}^{\lambda_\mathrm{x}})$ sets up polynomials for the sub tree$\Gamma_\mathrm{x}$and $\Gamma_\mathrm{x}(\gamma)$=0.For $\mathrm{h_x}$satisfied children nodes, B randomly choose λ_y and ensures $q_y(0) = q_x\big(index(y)\big) = \lambda_y$, for another $d_x - h_x$ child nodes, B randomly chooses a value$\lambda_\mathrm{z}$for each, which satisfying $g^{q_x(index(z))} = g^{\lambda_z}$, then the polynomial $\mathrm{q_x}(\,)$is decided in this way: $g^{q_x(0)} = g^{a+a_1x+..+a_kx^k}$, B knows *k=*$d_x$ different value $g^{q_x(i)}$, so B could compute all g^{a_i}, *i=1,…*. d_x. For the rest unsatisfied child nodes, B fixes the value using $g^{\lambda_z} = g^{q_x(index(z))}$or direct interpolation. Next, B defines the polynomials for the child nodes recursively as follows, if child node x' is satisfied, B calls Satpoly($\Gamma_x, \gamma, q_x(index(x'))$), If child node x' is unsatisfied, B calls Unsatpoly(($\Gamma_x, \gamma, g^{q_x(index(x'))}$)).

The final polynomial $\mathrm{Q_x}(\,) = \mathrm{bq_x}(\,)$, the simulator B then computes all the values needed to send to A : for leaf node x, i = att(x).

if i ∈ γ, B computes: $D_x = B^{q_x(0)/r_i} = g^{bq_x(0)/r_i} = g^{Q_x(0)/t_i}$

if i ∉ γ, B computes: $D_x = g^{q_x(0)/k_i} = g^{bq_x(0)/bk_i} = g^{Q_x(0)/t_i}$

Therefore, the simulator is able to construct the private key for Γ', and the distribution is identical to that in the original scheme. Further, for all every node x, B can compute :

$h = e\big(B, g^{q_x(0)}\big) = e(g,g)^{Q_x(0)}$ *and* $C_x\colon \{e(B, g^{a_i})\}_{i=1,..k_x-1} = \{e(g,g)^{ba_i}\}_{i=1,..k_x-1}$

so all values for verification are ready. A then checks the correctness.

Challenge A sends B two messages$\mathrm{M_0}, \mathrm{M_1}$. The simulator B flips a coin ν ,returns the encryption of $\mathrm{M_\nu}$, the ciphertext is as: *E= (*γ *, E'=*M_ν*Z,* $\{E_i = C^{r_i}\}_{i\in\gamma}$*)*, Z is from the DBDH challenger, if $e(g,g)^{abc}$*=0, Z=*$e(g,g)^{abc}$. Then, here, *Y=*$e(g,g)^{ab}$, *s=c,* $E_i = C^{r_i} = (g^{r_i})^c = T_i^s$, therefore, E is a valid encryption. If μ=1, Z=e(g, g)$^{\mathrm{z}}$, E' will be a random number.

Phase2 the simulator repeats phase1

Guess A submits his guess ν′for ν, if ν′=ν, the simulator B outputs μ′=0, otherwise, it outputs μ′=1.

The overall advantage of the simulator in the DBDH game is:

$Pr[\mu'=\mu] - 1/2 = Pr[\mu'=\mu \mid \mu=0].Pr[\mu=0]+Pr[\mu'=\mu \mid \mu=1]Pr[\mu=1]-1/2$

$=1/2(Pr[\mu'=\mu \mid \mu=0]+Pr[\mu'=\mu \mid \mu=1])-1/2$

$=1/2(1/2+\varepsilon)+1/2.1/2-1/2=\varepsilon/2.$

6 VABE with Multi-Authorities

VABE is much more useful in multi-authority environment, if any mistake is detected; user only needs to communicate with the particular authority, if there is no verification algorithm, he has to contact with all authorities. We give a concrete construction, taking the trick in [7].

6.1 The algorithms of Multi Authority VABE and security model

A Multi Authority ABE scheme is composed of K attribute authorities and one central authority, the scheme uses the following algorithms:

SETUP: A randomized algorithm which must be run by some trusted part (e.g CA). Takes as input the security parameter. Outputs a public key, secret key pair for each of the attribute authorities, and also outputs a system public key and master secret key which will be used by the central authority.

ATTRIBUTE KEY GENERATION: A randomized algorithm run by an attribute authority. Takes as input the authority secret key, the authority's value d_{k}, a user's ID, and an access structure $\Gamma_{\mathrm{c}}^{\mathrm{k}}$. Output secret key for the user.

CENTRAL KEY GENERATION: A randomized algorithm run by the central authority. Take as input the master secret key and a user's ID and outputs secret for the user.

ENCRYPTION: A randomized algorithm run by a sender. Takes as input a set of attributes for each authority, a message, and the system public key. Outputs the ciphertext.

VERIFICATION: A deterministic algorithm run by a user. Takes all public information and verification information as input. Outputs 1 if all checks pass, or else outputs 0.

DECRYPTION: If verification outputs 1, this deterministic algorithm is run by a user. Takes as input a ciphertext, which was encrypted under attribute set A_{c} . Output a message M if $\Gamma_{\mathrm{c}}^{\mathrm{k}}$ $(\mathrm{A}_{\mathrm{c}})$=1 for all authorities k.

The security model only adds a verification query in each single authority and the central authority to the model in [7].

6.2 concrete construction

Setup Fix prime order group G, G_1, bilinear map $e: G_1 \times G_2 \rightarrow G_2$, and generator $\mathrm{g} \in \mathrm{G}$. Choose seeds $\mathrm{s}_1, .. \mathrm{s}_{\mathrm{k}}$ for all authorities, also randomly choose y_0 ,$\{t_{k,i}\}_{k=1,..K,i=1,..n} \in Z_q$. System public key $Y_0 = e(g,g)^{y_0}$.

Attribute Authority k

Authority Secret Key $t_{k,1}, .. t_{k,n}, s_i$

Authority Public Key $T_{k,1}, .. T_{k,n}$where $T_{k,i} = g^{t_{k,i}}$

Secret Key for User u: Let $y_{k,u} = F_{S_k}(u)$.

To use a single authority verifiable ABE scheme as sub function with $\mathrm{y}_{\mathrm{k,u}}$ as its secret input to provide user with $\{\mathrm{D}_{\mathrm{x}}\}$.

Central Authority

Central Authority Secret Key s_{k} for all authorities k, y_0

Secret Key for User u: Let $y_{k,u} = F_{S_k}(u)$.for all k, Secret Key: $D_{CA} = g^{(y_0 - \sum_{k=0}^{K} y_{k,u})}$, at the same time, CA constructs a table, storing information related to the secret of each authority, and publish the table, the table has K+1 columns and the row is labeled by user identification u, in each row, the CA put $Y_{k,u} = e(g,g)^{y_{k,u}}$, k is from 1 to K, the last one in a row is $Y_{CA} = e(g,g)^{(y_0 - \sum_{k=0}^{K} y_{k,u})}$, once a new user makes a query for decryption key, the CA adds a new row to the table.

Encryption for Attribute set A_c Choose random s from Z_q . $E = Y_0^s m$, $E_{CA} = g^s$, and $\{E_{k,i} = T_{k,i}^s\}_{i \in A_C^k, \forall k}$

Verification: After getting the $\{D_x\}$from each authority, the user verifies as in 3.2, if any mistake is detected within a authority's shares, the user asks the authority to resend the corresponding shares without need to contact with other authorities, and in the last step of verification, take the value in the table to compare, if passes for all authorities, check an equation in the row labeled by his user identification $Y_0 = Y_{CA} * \prod_{k-1}^{K} Y_k$, if this also passes, then the key the user got is a right one which could be used to decrypt.

Decryption: For each authority k, the authorized user could interpolate to reconstruct $Y_{k,u} = e(g,g)^{y_{k,u}}$, compute $Y_{CA}^s = e(E_{CA}, D_{CA})$. Combine all these values to obtain $Y_0^s = Y_{CA}^s * \prod_{k-1}^{K} Y_k^s$. Then decrypt to get m.

6.3 The security proof for VABE with multi-authority

Theorem2. If all the checks pass, the Multi Authority Verifiable ABE scheme satisfies the two conditions in 3.2, and based on the DBDH assumption, the scheme is secure in the selective-set model defined in 4.1.

Proof sketch: the proof of this theorem is very similar to the proof of theorem1,

First, checks the two conditions in 2.3, at each authority, the verification makes sure that the sharing process is correct, this part is the same as in the proof in Theorem1, and then, checks in the table ensures all the shares of the authorities are rightly shared by the CA from the top secret y_0.

Next, the security proof only needs a few modifications of the original proof in [7], and the modification method is like that in the proof in theorem1, computing the verification information when emulating the oracle, to answer the new queries.

7 Conclusions

We introduce the definition of VABE, which allows the user checks the correctness of the key, using to decrypt all qualified ciphertext, doing this kind of verification reduces the trust of the authority, it is helpful when some error happens in creating or sending the secret, especially in multi-authority scenario, and it may potentially be useful as a building block to construct other kinds of cryptographic application. We make a proper security model for it, and give two concept constructions of VABE schemes with a single authority and multi authorities respectively, and prove their security under the model.

References

[1] Adi Shamir. Identity-based cryptosystems and signature schemes. In Proc. of CRYPTO84, volume 196, LNCS, 47-53. Springer, 1984

[2] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. In Proc. of CRYPTO01, volume 2139, LNCS, 213-229. Springer, 2001

[3] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Proc. of EUROCRYPT 05, volume 3494, LNCS, 457-473. Springer, 2005

[4] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for

fine-grained access control of encrypted data. In Proc. of CCS06, 89-98, New York, ACM Press, 2006

[5]. L. Cheung, and C. Newport. Provably secure ciphertext policy ABE, In Proc. of CCS07, 456 - 465, New York, ACM Press, 2007

[6]. J. Bethencourt, A .Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In IEEE Symposium on Security and Privacy, 321-334, 2007

[7] Melissa Chase. Multi-Authority Attribute-Based Encryption In TCC07, volume 4392, LNCS, 515-534. Springer, 2007

[8] Adi Shamir. How to share a secret. Communications of the ACM, volume22, 612-613, 1979

[9] Torben Pryds Pederson. Non-interactive and Information-theoretic Secure Verifiable Secret Sharing. In Proc. of CRYPTO 1991, volume 576, LNCS, 129-140, Springer, 1991

[10] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In Proc. of EUROCRYPT99, volume 3494, LNCS, 457-473 Springer, 1999

创新性说明：

本文提出了可验证的基于属性的加密的概念，即对于近期出现的研究热点基于属性的加密增加了密钥可验证的性质，增加该性质在密钥分发时有几个好处：

1、 检验出错，可以知道具体出错的地方，密钥分发中心便不需要重新计算整个所有的秘密信息，

2、 特别是在多密钥分发中心的情况下，检查出错，只需要找相应部分的密钥分发中心，而不用去找所有的分发中心都重发一次

3、 如果密钥通过检验，解密仍然出错，那么便可确定是密文或者属性出错，避免了解密出错不知道该找密钥分发方还是找信息发送方的尴尬

并且，我们给出了可验证属性加密的形式化定义，可验证的形式化定义，修正的选择属性模型，给出了两个具体构造方案，并在模型下证明了其安全性。