Verifiable Attribute Based Encryption

Qiang Tang and Dongyao Ji
(Corresponding author: Qiang Tang)
The State Key Lab of Information Securrity, Graduate University of Chinese Academy of Sciences
No. 19A, Yuquan Road, Beijing, 100049, China (Email: qtang84@gmail.com)
(Received Dec. 10, 2008; revised and accepted Mar. 13, 2009)

Abstract

In this paper, we propose the notion of Verifiable Attribute-Based Encryption (VABE) and give two constructs of key-policy VABE. 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. This could not be trivially solved, such as trying random decryption. Adding the verification property has a few advantages: first, it allows the user to immediately check the correctness of the keys, if not, he only needs 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; 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; third, the trick used in this paper could also be used in the ciphertext-policy scenario. We formalize the notion of VABE and prove our schemes in our model.

Keywords: Attribute-based encryption, verifiable, provable security, multi-authority

1 Introduction

Identity Based Encryption (IBE), introduced by Shamir [14], 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 [1], after that, IBE has received a lot of attention [4, 6, 8, 15].

To better express identity and allow for a certain amount of error-tolerance, Sahai and Waters proposed fuzzy IBE [12], in their scheme, identity is viewed as a set of descriptive attributes, and a user with the secret key for the identity is able to decrypt a ciphertext encrypted with the public key if and only if and are with a certain distance of each other as judged by some metric.

In the paper [7], 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 [2, 5], 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 [3] to answer an open question in [12], 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. The load of the single authority is enormously reduced.

In ABE [2, 3, 5, 7, 9], user goes to the central authority for the key according to his access structure, and stores it for future use. So, the correctness of the key should be verified the first time the user gets the key.

The trivial way of randomly choosing a message then encrypting it, and try decrypting, could not solve the problem in the multi-authority case. As the key of each authority is only known to the central authority, using the trivial method, the user could not know which authority is responsible for the mistake or cheating even if he finds that there is a flaw in the key. Further, in the future development of ABE schemes, when the attributes or access structure is hidden, then, the trivial way of decrypting randomly chosen encrypted message even could not be implemented, because only encrypting the random message with the corresponding attributes or policy would make sense.

1.1 Our Contribution

We add a verifiable property to the single authority key-policy ABE schemes, the core idea of the method is using some trick to change secret sharing [13] in the ABE schemes with verifiable secret sharing [10]. This trick can also be used in constructing ciphertext-policy VABE. Not only as a building block for multi-authority VABE, but also the single authority VABE has its own merits:

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.

4) It would be useful in the scenario when the attributes set or policy is hidden.

We give a formalized definition of VABE, and its security model, and prove the security under the new model.

1.2 Structure of This Paper

The paper is organized as follows: we give out the preliminaries in Section 2, and definitions of VABE and its security in Section 3, then we give a concrete construction with single authority and we prove its security in Sections 4 and 5, we also give out a construction with multi-authorities in Section 6, and at last, we draw a conclusion in Section 7.

2 Preliminaries

2.1 Bilinear maps

Let G_1 and G_2 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_1 and let e be a bilinear map, $e : G_1 \times G_1 \rightarrow G_2$. 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 A 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 A is $|Pr[A(A, B, C, e(g, g)^{abc} = 0] - Pr[A(A, B, C, e(g, g)^z) = 0]|$ 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 A.

3 VABE and Its Security Model

3.1 Definitions

Definition 1. *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, the decryption key D for Γ and the public parameter PK. It outputs the message M if $\gamma \in \Gamma$ (or $\Gamma(\gamma) = 1$).*

Definition 2. *If both of the conditions below are satisfied, then we say the ABE scheme is a VABE scheme.*

- *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 $\Gamma(\gamma) = 1$, B is the PPT algorithm to reconstruct the secret message from a given access structure, we define the games as follows.*

Setup. *The challenger runs the setup algorithm of VABE, gives the adversary all public information and parameter k_0;*

Challenge. *The adversary computes k_1, k_2 from $K(k_0)$, and $\Gamma_1 \leftarrow_R A(k_1, r, \gamma)$, $\Gamma_2 \leftarrow_R A(k_2, r, \gamma)$, gives Γ_1, Γ_2 to the challenger, the challenger flips a coin β, then runs the reconstruct algorithm B, gives the adversary $B(\Gamma_\beta)$;*

Guess. *The adversary returns a guess β' to the challenger.*

For any k_0, the quantity $Adv = |Pr(\beta' = \beta) - 1/2|$ is a negligible function of k_0.

- *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).*

3.2 Security Model for VABE

Our security model adds a verification query to the selective-set model in [7].

Initial. 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 [7] and gives the public parameters to the adversary.

Phase 1. 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_0 and M_1. The challenger flips a random coin b, and encrypts M_b with γ. The ciphertext is passed to the adversary.

Phase 2. Phase 1 is repeated.

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

The advantage of an adversary 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 Phase 1 and Phase 2, 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 [11]. A VABE 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 constructed by Goyal et al. [7]. 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 $\mathrm{att}(x)$.

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 $e : G_1 \times G_1 \rightarrow G_2$ denotes 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 $Z_p : \Delta_{i,S}(x) = \prod_{j \in S, j \neq i}(x - j)/(i - j)$. We will associate each attribute with a unique element in $Z_p{}^*$.

Setup. Define the universe of attributes $U = 1, 2, \cdots, n$. Randomly choose $t_1, \cdots, t_n$, y from Z_p. The published public keys PK are $T_1 = g^{t_1}, \cdots, T_{|U|} = g^{t_{|U|}}$, $Y = e(g, g)^y$. The master key MK is: $t_1, \cdots, t_n$, 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: $E = (\gamma, E' = MY^s, \{E_i = T_i^s\}_{i \in \gamma})$.

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 $q_x(\cdot)$ 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 $q_x(\cdot)$, then compute $h_x = e(g, g)^{q_x(0)}$ and $C_x : \{e(g, g)^{a_i}\}_{i=1,\cdots,k_x-1}$, $\{a_i\}$ are the non constant coefficients of the polynomial $q_x(\cdot)$ 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 give 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$(\Gamma, PK, \{h_x\}, \{C_x\}, D)$**.** For leaf node x, after getting $\{D_x\}$, the user firstly checks whether $e(D_x, T_i) = h_x$, and then, verifies:

$$\begin{aligned} h_x &= e(g,g)^{q_{parent(x)}(index(x))} \\ &= e(g,g)^{q_{parent(0)}+a_1(index(x))+..+a_{i-1}(index(x)^{i-1})} \\ &= h_{parent(x)} \times \prod_{i=1}^{k-1}(e(g,g)^{a_i})^{index(x)^i} \end{aligned} \quad (1)$$

Where, k is the degree of the polynomial $q_{parent(x)}(\cdot)$, 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 $x_1, \cdots, x_l$, if any node x_i's ancestor node is in this nodes set, we remove x_i from this set, at last, we get $z_1, \cdots, z_t$, we ask the authority to resend the shares of the subtree $\Gamma_{z_1}, \cdots, \Gamma_{z_t}$.

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:

$$\begin{aligned} e(D_x, E_i) &= e(g^{q_x(0)/t_i}, g^{st_i}) \\ &= e(g,g)^{sq_x(0)}, \quad \text{if } i \in \gamma; \end{aligned}$$

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 $\perp$, otherwise, we compute:

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x(0)}}, \text{where } i = index(z), \\ S'_x &= \{index(z) : z\}) \\ &= e(g,g)^{sq_x(0)}, \end{aligned}$$

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

Theorem 1. *The concrete construction of verifiable ABE scheme is verified, which means it satisfies the two conditions in Definition 2 of Section 3.1, and the scheme is also secure in the selective-set model defined in Section 3.2 under the decisional BDH assumption.*

Proof. First, we observe the Conditions 1 and 2 in Definition 1 of Section 3.1. 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 $h_x = h_{parent(x)} \times \prod_{i=1}^{k-1}(e(g,g)^{a_i index(x)^i})$ 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 $q_{parent(x)}(\cdot)$, 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 in Condition 1 in Definition 1, if any of this quantity generated 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

$$\begin{aligned} Q &= e(g,g)^a \\ &= e(g,g)^{q_{x(0)}} \\ &= h_{parent(x)} \times \prod_{i=1}^{k-1}(e(g,g)^{a_i index(x)^i}, \end{aligned}$$

then we will get $\log_{e(g,g)} 1 = a - q_{x(0)} < q$ (all the computation is under the modular arithmetics)with a non-negligible quantity, however, $e(g,g)$ is in a prime order q group,its order can only be q, 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 Condition 2 is satisfied.

Next, we show the security of our scheme. As in [5], 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 $\epsilon/2$. The main difference between our proof and the one in [5] is the check query.

The challenger flips a fair binary coin, if it equals to 0, sets the tuple $(A, B, C, Z) = (g^a, g^b, g^c, e(g,g)^{abc})$, else 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:

Initial. B runs A, A chooses 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 $i \in \gamma$, set $T_i = g^{r_i}$, r_i is

randomly chosen from Z_p, otherwise set $T_i = B^{k_i}$, k_i is randomly chosen from Z_p, then B gives the public parameters to A.

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

- $Satpoly(\Gamma_x, \gamma, \lambda_x)$ constructs the polynomials for the sub tree Γ_x and $\Gamma_x(\gamma) = 1$. It first sets up a polynomial q_x of degree d_x for the root node x and satisfying $q_x(0) = \lambda_x$. For each child node x' of x, we defines its polynomial by calling $Satpoly(\Gamma'_x, \gamma, q_x(index(x')))$.
- $Unsatpoly(\Gamma_x, \gamma, g^{\lambda_x})$ sets up polynomials for the sub tree Γ_x and $\Gamma_x(\gamma) = 0$. For h_x satisfied children nodes, B randomly choose λ_y and ensures $q_y(0) = q_x(index(y)) = \gamma_y$, for another $d_x - h_x$ child nodes, B randomly chooses a value λ_z for each, which satisfying $g^{q_x(index(z))} = g^{\lambda_z}$, then the polynomial $q_x(\cdot)$ is decided in this way:

$$g^{q_x(0)} = g^{a + a_1 x + .. + a_k x^k},$$

B knows $k = d_x$ different value $g^{q_x(i)}$, so could compute all $g^{a_i}, i = 1, \cdots, d_x$. For the rest unsatisfied child nodes, B fixes the value using $g^{\lambda_z} = g^{q_x(index(z))}$ or direct interpolation. Next, 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 $Q_x(\cdot) = bq_x(\cdot)$, the simulator B then computes all the values needed to send to A: for leaf node $x, i = att(x)$.

If $i \in \gamma$, B computes: $D_x = B^{q_x(0)/r_i} = g^{bq_x(0)/r_i} = g^{Q_x(0)/t_i}$;

If $i \notin \gamma$, 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:

$$h = e(B, g^{q_x(0)}) = e(g, g)^{Q_x(0)}$$

and

$$C_x : \{e(B, g^{a_i})\}_{i=1,\cdots,k_{x-1}} = \{e(g, g)^{ba_i}\}_{i=1,..k_{x-1}}$$

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

Challenge. A sends B two messages M_0, M_1. The simulator B flips a coin ν, returns the encryption of M_ν, the ciphertext is as: $E = (\gamma, E' = M_Z, \{E_i = C^{r_i}\}_{i\in\gamma})$, 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 $\mu = 1$, $Z = e(g,g)^z$, E' will be a random number.

Phase 2. The simulator repeats Phase 1.

Guess. A submits his guess $\nu'$for ν, if $\nu' = \nu$, the simulator B outputs $\mu' = 0$, otherwise, it outputs $\mu' = 1$.

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

$$\begin{aligned}
& |Pr[\mu' = \mu] - 1/2| \\
= \; & |Pr[\mu' = \mu | \mu = 0].Pr[\mu = 0] \\
& \quad + Pr[\mu' = \mu | \mu = 1] Pr[\mu = 1] - 1/2| \\
= \; & |1/2(Pr[\mu' = \mu | \mu = 0] + Pr[\mu' = \mu | \mu = 1]) - 1/2| \\
= \; & |1/2(1/2 + \epsilon) + 1/21/2 - 1/2| \\
= \; & \epsilon/2.
\end{aligned}$$

□

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 [3].

6.1 The Algorithms of Multi-Authority VABE and Its 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 Γ_c^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 $\Gamma_c^k(A_c) = 1$ for all authorities k.

The security model only adds a verification query in each single authority and the central authority to the model in [3].

6.2 Concrete Construction

Setup. Fix prime order group G, G_1, bilinear map $e : G \times G \to G_1$ and generator $g \in G$. Choose seeds $s_1, \cdots, s_k$ for all authorities, also randomly choose y_0, $\{t_{k,i}\}_{k=1,\cdots,K,i=1,\cdots,n} \in Z_q$. System public key $Y_0 = e(g,g)^{y_0}$.

Attribute Authority k. Authority Secret Key $t_{k,1}, \cdots, t_{k,n}, s_i$; Authority Public Key $T_{k,1}, \cdots, T_{k,n}$ 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 $y_{k,u}$ as its secret input to provide user with $\{D_x\}$.

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 Section 3.1, 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 $Y_0 = Y_{CA} \times \prod_{k-1}^{K} Y_k$, 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 $Y_0^s = Y_{CA}^s \times \prod_{k-1}^{K} Y_k^s$. Then decrypt to get m.

6.3 The Security Proof for VABE with Multi-Authority

Theorem 2. *If all the checks pass, the Multi Authority Verifiable ABE scheme satisfies the two conditions in Section 3.1, and based on the DBDH assumption, the scheme is secure in the selective-set model defined in Section 4.1.*

Proof. The proof of this theorem is very similar to the proof of Theorem 1, First, checks the two conditions in Section 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 [3], 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 concrete constructions of VABE schemes with a single authority and multi authorities respectively, and prove their security under the model.

8 Acknowledgements

This work has been supported in part by National Natural Science Foundation No.90604010, National Hi-Tech Research and Development Program (also 863 Program) of China No.2006AA01Z427, and National Basic Research Program (also 973 Program) of China No.2007CB311202.

References

[1] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Proceedings of Crypto'01*, LNCS 2139, pp. 213-229, Springer, 2001.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption,"

IEEE Symposium on Security and Privacy, pp. 321-334, 2007.

[3] M. Chase. "Multi-authority attribute-based encryption," *Proceedings of TCC'07*, LNCS 4392, pp. 515-534, Springer, 2007.

[4] X. Cheng, L. Guo, and X. Wang, "An identity-based mediated signature scheme from bilinear pairing," *International Journal of Network Security*, vol. 2, no. 1, pp. 29-33, 2006.

[5] L. Cheung and C. Newport. "Provably secure ciphertext policy ABE," *Proceedings of CCS'07*, pp. 456-465, New York, ACM Press, 2007.

[6] S. Cui, P. Duan, C. W. Chan, and X. Cheng, "An efficient identity-based signature scheme and its applications," *International Journal of Network Security*, vol. 5, no. 1, pp. 89-98, 2007.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proceedings of CCS'06*, pp. 89-98, New York, ACM Press, 2006.

[8] Z. Li, C. F. Chong, L. C. K. Hui, S. M. Yiu, K. P. Chow, W. W. Tsang, H. W. Chan, and K. K. H. Pun, "An attack on Libert et al.' s iD-based indeniable signature scheme," *International Journal of Network Security*, pp. 220-223, vol. 5, no. 2, 2007.

[9] D. Nali, C. Adams, and A. Miri, "Using threshold attribute-based encryption for practical biometric-based access control," *International Journal of Network Security*, vol. 1, no. 3, pp. 173-182, 2005.

[10] T. P. Pederson. "Non-interactive and information-theoretic secure verifiable secret sharing," *Proceedings of Crypto'91*, LNCS 576, pp. 129-140, Springer, 1991.

[11] A. Sahai, "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security," *Proceedings of Eurocrypt'99*, LNCS 3494, pp. 457-473, Springer, 1999.

[12] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Proceedings of Eurocrypt'05*, LNCS 3494, pp. 457-473, Springer, 2005.

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

[14] A. Shamir, "Identity-based cryptosystems and signature schemes," *Proceedings of Crypto'84*, LNCS 196, pp. 47-53, Springer, 1984.

[15] H. Xiong, Z. Qin, and F. Li, "Identity-based threshold signature secure in the standard model," *International Journal of Network Security*, vol. 10, no. 1, pp. 75-80, 2010.

Qiang Tang is a Master student in Graduate University of Chinese Academy of Sciences. His research interests focus on applied cryptography, including: attribute based encryption, privacy preserving authentication, cryptographic protocol, and digital signatures. He has published 3 scientific papers.

Dongyao Ji is an associate professor in The State Key Lab of Information Security, Graduate University of Chinese Academy of Sciences. He received his Ph.D. degree from Xidian Univeristy, China, 2001. His research interests include: Design and analysis of cryptographic schemes, and formal methods. He has published more than 20 scientific papers.