SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2012 Vol. 55 No. 7: 1631–1638
doi: 10.1007/s11432-012-4594-7

Accountable authority key policy attribute-based encryption

WANG YongTao[1,2*], CHEN KeFei[1,3], LONG Yu[1,4] & LIU ZhaoHui[2]

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[2]*China Information Technology Security Evaluation Center, Beijing* 100085*, China;*
[3]*Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, Shanghai,* 200240*, China;*
[4]*State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences,* 100039*, China*

Received September 13, 2010; accepted October 25, 2011

Abstract We present an accountable authority key policy attribute-based encryption (A-KPABE) scheme. In this paper, we extend Goyal's work to key policy attribute-based encryption setting. We first generalize the notion of accountable authority in key policy attribute-based encryption scenario, and then give a construction. In addition, our scheme is shown to be secure in the standard model under the modified Bilinear Decisional Diffie-Hellman (mBDDH) assumption.

Keywords accountable authority, key policy, attribute-based encryption, standard model, access structure

Citation Wang Y T, Chen K F, Long Y, et al. Accountable authority key policy attribute-based encryption. Sci China Inf Sci, 2012, 55: 1631–1638, doi: 10.1007/s11432-012-4594-7

1 Introduction

In 2005, Sahai and Waters [1] introduced the notion of attribute-based encryption, which integrates an access structure with the public key encryption scheme. In their system, a user is described by a set of attributes and the private key for a user is generated by the authority according to his attribute set. In the same way, a ciphertext is also associated with a set of attributes, i.e., the sender can generate a ciphertext according to an attribute set. A user can decrypt a ciphertext when at least d (threshold parameter) attributes overlap between the attribute set of his private key and the attribute set of the ciphertext.

The above scheme has achieved threshold access structure and is a Threshold ABE system. At present, more complex access structures deployed in users' private keys or ciphertexts are achieved. Those schemes extend the original system mainly along two lines. One is the key policy attribute-based encryption (KP-ABE) first proposed by Goyal et al. [2], and the other is the ciphertext policy attribute-based encryption (CP-ABE) [3–5]. In KP-ABE, each ciphertext is labeled with a set of attributes and each private key is identified by an access structure. On the contrary, each private key is identified by a set of attributes and each ciphertext is labeled with an access structure in CP-ABE. Both types of ABE systems have found many uses, such as the sharing of audit-log and broadcast encryption. In addition, we notice that

*Corresponding author (email: wyt_sjtu@yahoo.com.cn, wangyt@itsec.gov.cn)

the Dual policy ABE [6] is composed of the above two approaches. However, all the above schemes are shown to be secure in the Selective model. More recently, Lewko et al. [7] present a fully secure ABE scheme.

Attribute based encryption scheme has many appealing properties and can be viewed as a generalization of identity based encryption (IBE) [8–10]. However, it inherits the key escrow problem from IBE. That is, all users' private keys are issued by an unconditionally trusted authority. In such a system, the authority can easily distribute users' private keys for malicious use. Additionally, it can decrypt any ciphertext encrypted to a user and can forge signature on any user's behalf. In some situations, this problem may stunt applications. In IBE setting, there are several approaches to mitigate the key escrow problem. For example, the master key is shared among multi authorities by using threshold [9]. But this approach is not practical. Here, we consider the approach of accountable authority. Goyal [11] first introduced the notion of accountable authority identity-based encryption (A-IBE) to mitigate the key escrow problem in IBE. In A-IBE, if the authority distributes some user's key for malicious use, it runs the risk of being caught and sued by the user. This can happen if the following conditions satisfied: (1) For every identity id, there will be an exponential number of possible keys. (2) To generate a user's private key, an interactive key generation protocol will be implemented between the authority and the user. This protocol will ensure that the family to which this key belongs is concealed to the authority. (3) With this single key, it is intractable for the user to find any other keys from a different family. Thus, the evidence of the authority's misbehavior will be given by two keys from distinct families for an identity.

In IBE setting, there are several related researches. In [11], Goyal gave two constructions. The first one is based on the IBE scheme of [12], but this construction only allows tracing well-formed decryption keys. The second one is built on the fuzzy IBE scheme [1] and provides weak black-box traceability. However, this construction is somewhat inefficient. Moreover, Goyal [13] proposed a new scheme with the black-box traceability. In addition, Au et al. [14] proposed a scheme which reveals the PKG's master secret key if more than one key is released for per identity. At PKC'09, Libert et al. [15] proposed an efficient construction for A-IBE, which has short ciphertexts and provides a very simple weak black-box tracing mechanism. In ABE setting, Li et al. [16] introduced an ABE scheme which achieves the accountability of the authority. However, this scheme assumes that each user has a higher level secret before requesting an attribute private key. In addition, Yu et al. [17] proposed a scheme to deal with the key abuse problem.

In this work, we first generalize the concept of accountable authority to key policy ABE setting, and then give a construction. The construction is shown to be secure in the standard model under the modified bilinear decisional Diffie-Hellman (mBDDH) assumption. Comparing to Li et al.'s work [16], our construction does not need such a high level secret and is very efficient. Our scheme non-trivially integrates an A-IBE scheme proposed by Libert et al. [15] with the first key policy ABE scheme of [2]. Firstly, we identify a user in the system by a pair $(\mathrm{id}, \mathcal{T})$, where $\mathcal{T}$ is an access tree and id is the user's identity. Secondly, some secret value Y will be split into two parts. One will be used for generating partial key related to id, and the other will be used for generating partial key related to access tree. Thirdly, we require that the ability to decrypt be independent of the id. To solve this, an additional element is introduced in the ciphertexts. Furthermore, the weak black-box tracing mechanism, which is described in [15], can be extended to our scheme.

The rest of the paper is organized as follows. In Section 2, we recall some preliminaries. In Section 3, we formalize the definition and security notions for accountable authority key policy attribute-based encryption scheme. We describe our construction and prove its security in Section 4. Finally, we draw some conclusions in Section 5.

2 Definitions

2.1 Bilinear maps

We now review the notion of bilinear maps. Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}_1$. Let e denote a bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, which has the following properties [9]:

- Bilinearity: For all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, it satisfies $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: $e(g, g) \neq 1_{\mathbb{G}_2}$.
- Computability: There exists an efficient algorithm to compute $e(u, v)$, for all $u, v \in \mathbb{G}_1$.

2.2 Complexity assumption

We require the modified bilinear decisional Diffie-Hellman (mBDDH) assumption should hold. This assumption has been used before in [18]. It states that the two distributions $(g^a, g^b, g^c, g^{c^2}, e(g, g)^{abc})$ and $(g^a, g^b, g^c, g^{c^2}, e(g, g)^z)$ are indistinguishable for any polynomial-time adversary $\mathcal{B}$, where $a, b, c, z \in \mathbb{Z}_p$ are random. Let κ be the security parameter. The advantage function $\mathrm{Adv}(\kappa)$ of $\mathcal{B}$ is defined as

$$\left|\Pr\left[\mathcal{B}(g^a, g^b, g^c, g^{c^2}, e(g, g)^{abc}) = 1\right] - \Pr\left[\mathcal{B}(g^a, g^b, g^c, g^{c^2}, e(g, g)^z) = 1\right]\right|.$$

We say that the modified BDDH assumption holds if $\mathrm{Adv}(\kappa)$ is negligible for all polynomial-time adversaries.

2.3 Syntax and security models

An accountable authority key policy attribute-based encryption scheme consists of five polynomial-time algorithms described as follows:

- **Setup:** Taking a security parameter κ, this algorithm outputs a master public key mpk and a master secret key msk.
- **KeyGen:** An interactive protocol is implemented between the authority and a user P. The public input to the authority and P consists of the mpk and $(\mathrm{id}, \mathbb{A})$, where $\mathbb{A}$ is an access structure associated to P's identity id. The private input to the authority is the msk. Additionally, either the authority or P may use a sequence of random coin tosses as his private inputs. At the end of the protocol, P can extract a private key $d_{\mathrm{id},\mathbb{A}}$ as his private output.
- **Encryption:** Taking the mpk, a set of attributes ω and a message M, this algorithm outputs a ciphertext C.
- **Decryption:** Taking a user secret key $d_{\mathrm{id},\mathbb{A}}$ and a ciphertext C encrypted under ω, this algorithm outputs a plaintext message M if $\omega \in \mathbb{A}$.
- **Trace:** Taking a well-formed decryption key $d_{\mathrm{id},\mathbb{A}}$, this algorithm outputs the decryption key family number n_F.

We describe the security models of our A-KPABE scheme in the white-box setting. Please refer to [13,15] for further extensions in the black-box setting. The models are as follows.

The IND-SS-CPA game. The proposed accountable authority key policy attribute-based encryption scheme is indistinguishable under chosen plaintext attack (IND-CPA). Here, we simply extend the Selective-set (SS) model described in the KP-ABE [2] to our setting.

- Init. The adversary declares a set of attributes, ω^*.
- Setup. The challenger runs the setup algorithm of A-KPABE and gives the public parameters to the adversary.
- Phase 1. The adversary runs the key generation protocol with the challenger for many pairs $(\mathrm{id}_j, \mathbb{A}_j)$, where $\omega^* \notin \mathbb{A}_j$ for all j. There is no limitation on id_j.
- Challenge. The adversary submits two equal length messages M_0, M_1. The challenger flips a random coin, ν, and encrypts M_ν with ω^*. The ciphertext is passed to $\mathcal{A}$.
- Phase 2. Phase 1 is repeated.
- Guess. The adversary outputs a guess ν' of ν.

The advantage of the adversary in this game is defined as $\Pr[\nu' = \nu] - \frac{1}{2}$.

The FindKey game. This game follows from [11] except for slight modification in our setting. Let $\mathcal{A}$ be the malicious authority.

- Setup. $\mathcal{A}$ generates mpk and gives it with a pair $(\mathrm{id}, \mathbb{A})$ to the challenger. The challenger runs a sanity check on mpk and aborts if the check fails.

• Key generation. The challenger and $\mathcal{A}$ run the key generation protocol to generate a decryption key for $(\mathrm{id}, \mathbb{A})$. The challenger gets the key $d_{\mathrm{id},\mathbb{A}}$ as private output and runs a key sanity check on it. It aborts if the check fails.

• Find key. $\mathcal{A}$ outputs a decryption key $d'_{\mathrm{id},\mathbb{A}}$. The challenger runs a key sanity check on it and aborts if the check fails.

Let K_1 denote the event that $\mathrm{Trace}(d'_{\mathrm{id},\mathbb{A}}) = \mathrm{Trace}(d_{\mathrm{id},\mathbb{A}})$. The advantage of $\mathcal{A}$ in this game is defined as $\Pr[K_1]$. If the advantage is negligible, the above game will imply that it is intractable for the malicious authority to find user's key family.

The ComputeNewKey game. The game is defined along the line of [11]. Here, we describe it in Selective-ID model.

• Init. The adversary declares a challenge of identity id^*.

• Setup. The challenger runs the Setup algorithm of A-KPABE and gives the public parameters to the adversary.

• Key generation. The adversary runs the KeyGen protocol with the challenger for many pairs $(\mathrm{id}_j, \mathbb{A}_j)$, where id_j must be distinct. There is no limitation on $\mathbb{A}_j$.

• New key computation. The adversary outputs two decryption keys $d_{\mathrm{id}^*,\mathbb{A}}$ and $d'_{\mathrm{id}^*,\mathbb{A}}$. The challenger runs a key sanity check on them. It aborts if the check fails.

Let K_2 denote the event that $\mathrm{Trace}(d'_{\mathrm{id}^*,\mathbb{A}}) \neq \mathrm{Trace}(d_{\mathrm{id}^*,\mathbb{A}})$. The advantage of the adversary in this game is defined as $\Pr[K_2]$. If the advantage is negligible, this game will imply that it is intractable for the user to find any other keys from a different family with his legal key.

Definition 1. An A-KPABE scheme is IND-SS-CPA secure if all polynomial time adversaries have at most a negligible advantage in the above three games.

3 Our scheme

In the construction, we instantiate the access structure $\mathbb{A}$ by a access tree structure described in [2]. We identify a user in the system by a pair $(\mathrm{id}, \mathcal{T})$, where id denotes the user's identity and $\mathcal{T}$ is the access tree associated to id. The following condition must be satisfy, i.e., a user should be assigned only one access tree. Notice that it is allowable for different users with the same access tree $\mathcal{T}$. The construction borrows idea from the accountable authority identity-based encryption scheme [15] and is based on the first construction of key policy attribute-based encryption scheme [2].

Let $\mathcal{U}$ denote the universe attributes. Each element of $\mathcal{U}$ will be mapped to a unique integer in $\mathbb{Z}_p^*$. We also define the Lagrange coefficient $\Delta_{k,S}$ for $k \in \mathbb{Z}_p$ and a set, S, of elements in $\mathbb{Z}_p$: $\Delta_{k,S}(x) = \prod_{j \in S, j \neq k} \frac{x-j}{k-j}$. Our construction is as follows.

• **Setup** (1^κ)**:** Input the security parameter κ and generate a set of pairing groups. Choose randomly $x, y, y_1 \in \mathbb{Z}_p^*$ and h, $Z \in \mathbb{G}_1$, and set $X = g^x$, $Y = g^y$, $Y_1 = g^{y_1}$. Take the first $|\mathcal{U}|$ elements of $\mathbb{Z}_p^*$ as the universe, and then choose randomly $t_1, \ldots, t_{|\mathcal{U}|}$ from $\mathbb{Z}_p^*$. Set $T_i = g^{t_i}$ for $i = 1, \ldots, |\mathcal{U}|$. Now set the master secret key as msk$= (x, y, y_1, t_1, \ldots, t_{|\mathcal{U}|})$, and the master public key as mpk$= (g, X, h, Z, T_1, \ldots, T_{|\mathcal{U}|}, E = e(g, Y), E_1 = e(g, Y_1))$.

• **KeyGen:** To generate a private key for a user P with $(\mathrm{id}, \mathcal{T})$, the following protocol will be executed between P and the authority.

$-P$ provides a commitment $R = h^{s_0} \cdot X^\theta$, where $s_0, \theta \in \mathbb{Z}_p^*$ are chosen randomly, with an interactive witness indistinguishable (WI) proof of knowledge of the (s_0, θ) to the authority. In addition, P retains (s_0, θ).

$-$ Having received R, the authority verifies the proof of knowledge, and outputs $\perp$ if it fails to verify. Otherwise, it generates private key component for $\mathcal{T}$ by using the algorithm **Key Generation** described in [2]. The algorithm proceeds as follows.

$*$ Define a polynomial q_{x} for each node x in $\mathcal{T}$. Set $b_{\mathrm{x}} = k_{\mathrm{x}} - 1$, where b_{x} is the degree of q_{x}, and k_{x} is the threshold value.

∗ For the root node, define q_{root} by setting $q_{\text{root}}(0) = y - y_1$ and b_{root} other points randomly. For any other node x, define q_{x} by setting $q_{\text{x}}(0) = q_{\text{parent(x)}}(\text{index(x)})$ and b_{x} other points randomly.

∗ For each leaf node x, set $D_{\text{x}} = g^{\frac{q_{\text{x}}(0)}{t_i}}$, where i is the attribute associated to leaf node x. Let D be the set of the above D_{x}.

The authority chooses randomly $s_1, r' \in \mathbb{Z}_p^*$ and returns

$$d'_{\text{id},\mathcal{T}} = (d'_1, d'_2, d'_3, d'_4) = ((Y_1 R h^{s_1})^{\frac{1}{x}} \cdot (g^{\text{id}} Z)^{r'}, X^{r'}, s_1, D). \tag{1}$$

– Having received $d'_{\text{id},\mathcal{T}}$, P chooses $r'' \in \mathbb{Z}_p^*$ at random and computes $d_{\text{id},\mathcal{T}} = (d_1, d_2, d_3, d_4)$ as $((d'_1/g^{\theta}) \cdot (g^{\text{id}} Z)^{r''}, d'_2 \cdot X^{r''}, d'_3 + s_0, d'_4)$, which should be equivalent to

$$((Y_1 h^{s_0+s_1})^{\frac{1}{x}} \cdot (g^{\text{id}} Z)^{r}, X^{r}, s_0 + s_1, D), \tag{2}$$

where $r = r' + r''$. P checks $d_{\text{id},\mathcal{T}}$ as follows. First, for simplicity, choose arbitrary subset S from the set of the leaf node in $\mathcal{T}$ such that $\mathcal{T}(S) = 1$, and then compute $R_i = e(T_i, g^{\frac{q_{\text{x}}(0)}{t_i}})$ for x $\in S$, where i is the attribute associated to the leaf node x. Now P can compute $R_0 = e(g, g)^{q_{\text{root}}(0)}$ for the root node by using polynomial interpolation in exponential. Second, P checks

$$\frac{e(d_1, X)}{e(g, h)^{d_3} \cdot e(g^{\text{id}} Z, d_2)} = E_1, \tag{3}$$

and $R_0 \cdot E_1 = E$. P outputs $\perp$ if checking fails. Otherwise, P sets his private key as $d_{\text{id},\mathcal{T}}$ and the key family number as $n_F = d_3 = s_0 + s_1$.

• **Encryption**(M, ω)**:** The sender chooses $t \in \mathbb{Z}_p^*$ at random to encrypt a message $M \in \mathbb{G}_2$ under a set of attributes ω. It computes $C = (\omega, C_1, C_2, C_3, C_4, C_5)$ as

$$(\omega, X^t, g^t, Z^t, M \cdot e(g, Y)^t, \{T_i^t\}_{i \in \omega}). \tag{4}$$

• **Decryption**$(C, d_{\text{id},\mathcal{T}})$**:** Let C be a valid encryption of M under ω. C can be decrypted by a user with the private key $d_{\text{id},\mathcal{T}}$, where $\mathcal{T}(\omega) = 1$. The user first computes $e(g, g)^{q_{\text{root}}(0)t}$ by using the algorithm Decryption described in [2] and $C'_2 = C_2^{\text{id}} \cdot C_3 = (g^{\text{id}} \cdot Z)^t$. Then the user can decrypt C as

$$M = C_4 \cdot \frac{e(C_2, h)^{d_3} \cdot e(C'_2, d_2)}{e(d_1, C_1) \cdot e(g, g)^{q_{\text{root}}(0)t}}. \tag{5}$$

• **Trace:** Taking a well-formed decryption key $d_{\text{id},\mathcal{T}} = (d_1, d_2, d_3, d_4)$, this algorithm outputs the decryption key family number $n_F = d_3$.

4 Analysis of the scheme

4.1 Correctness

Notice that if the cryptosystem is operated as specified, we have $e(g, g)^{q_{\text{root}}(0)t} = e(C_2, g^{y-y_1})$ and

$$C_4 \cdot \frac{e(C_2, h)^{d_3} \cdot e(C'_2, d_2)}{e(d_1, C_1) \cdot e(g, g)^{q_{\text{root}}(0)t}} = C_4 \cdot \frac{e(g^t, h)^{s_0+s_1} \cdot e((g^{\text{id}} \cdot Z)^t, X^r)}{e((Y_1 h^{s_0+s_1})^{\frac{1}{x}} \cdot (g^{\text{id}} Z)^r, X^t) \cdot e(g^{q_{\text{root}}(0)}, g^t)}$$
$$= C_4 \cdot \frac{1}{e(Y_1, g^t) \cdot e(g^{q_{\text{root}}(0)}, g^t)} = M. \tag{6}$$

4.2 Security proofs

In this section, we give the security proofs for the above A-KPABE scheme.

Theorem 4.1. The scheme is IND-SS-CPA under the mBDDH assumption.

Proof. Let $\mathcal{A}$ be an adversary against our scheme with advantage ϵ. We build a simulator $\mathcal{B}$ that can solve a modified DBDH instance with advantage $\epsilon/2$.

The challenger sets the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with an efficient bilinear map e and flips a fair binary coin μ outside $\mathcal{B}$'s view. If $\mu = 0$, the challenger sets (A, B, C, C′, D) $= (g^a, g^b, g^c, g^{c^2}, e(g,g)^{abc})$; otherwise it sets (A, B, C, C', D)$= (g^a, g^b, g^c, g^{c^2}, e(g,g)^z)$ for random a, b, c, $z \in \mathbb{Z}_p^*$. The simulator proceeds as follows:

• Init. $\mathcal{B}$ receives the target set of attributes ω^* from $\mathcal{A}$.

• Setup. $\mathcal{B}$ chooses $\alpha, \beta \in \mathbb{Z}_p^*$ and sets $X = \mathrm{C}, h = \mathrm{B}, Z = X^\alpha = g^{c\alpha}, E = e(\mathrm{A}, \mathrm{B}), E_1 = e(g, \mathrm{B}^\beta)$ (this implicitly sets $y = ab$). For $i \in \omega^*$, $\mathcal{B}$ chooses $w_i \in \mathbb{Z}_p^*$ and sets $T_i = g^{w_i}$. For $i \notin \omega^*$, $\mathcal{B}$ chooses $\beta_i \in \mathbb{Z}_p^*$ and sets $T_i = \mathrm{B}^{\beta_i}$. Now $\mathcal{B}$ gives the public parameters to $\mathcal{A}$.

• Phase 1. Suppose $\mathcal{A}$ requests a private key $(\mathrm{id}, \mathcal{T})$, where $\mathcal{T}(\omega^*) = 0$. The simulator will receive an element $R = h^{s_0} \cdot X^\theta$ with a WI proof of knowledge of (s_0, θ) from $\mathcal{A}$. If the proof proves valid, $\mathcal{B}$ prepares the private key as follows ($\mathcal{B}$ does not need to rewind $\mathcal{A}$). Firstly, it picks $s_1 \in \mathbb{Z}_p^*$ at random and defines $W = Y_1 \cdot R \cdot h^{s_1}$, $d'_3 = s_1$. Then, d'_1 and d'_2 are generated as (we assume that id is non-zero)

$$(d'_1, d'_2) = ((g^{\mathrm{id}} \cdot Z)^{r'} \cdot W^{-\frac{\alpha}{\mathrm{id}}}, X^{r'} \cdot W^{-\frac{1}{\mathrm{id}}}), \tag{7}$$

where r' is chosen randomly from $\mathbb{Z}_p^*$. Letting $\tilde{r}' = r' - \frac{\log_g(W)}{c \cdot \mathrm{id}}$, d'_1 and d'_2 have the correct distribution. We have

$$d'_1 = W^{1/c} \cdot (g^{\mathrm{id}} \cdot Z)^{\tilde{r}'} = W^{1/c} \cdot (g^{\mathrm{id}} \cdot X^\alpha)^{r'} \cdot (g^{\mathrm{id}} \cdot X^\alpha)^{-\frac{w}{c \cdot \mathrm{id}}} = (g^{\mathrm{id}} \cdot Z)^{r'} \cdot W^{-\frac{\alpha}{\mathrm{id}}}, \tag{8}$$

and $d'_2 = X^{\tilde{r}'} = X^{r'} \cdot (g^c)^{-\frac{w}{c \cdot \mathrm{id}}} = X^{r'} \cdot W^{-\frac{1}{\mathrm{id}}}$, where $w = \log_g(W)$. To generate d'_4, $\mathcal{B}$ proceeds as follows. It assigns a polynomial q_{x} of degree b_x for every node in the access tree $\mathcal{T}$. Recall that we have implicitly set $g^{q_{\mathrm{root}}(0)} = Y/Y_1 = g^{ab-b\beta}$. The simulator first defines two algorithms $\mathbf{PolySat}(\mathcal{T}_{\mathrm{x}}, \omega^*, \lambda_{\mathrm{x}})$ and $\mathbf{PolyUnsat}(\mathcal{T}_{\mathrm{x}}, \omega^*, g^{\lambda_{\mathrm{x}}})$ as described in Theorem 1 [2], where $\mathcal{T}_{\mathrm{x}}$ denotes the sub-tree with root node x and λ_{x} denotes the secret value for node x. Then, the simulator invokes $\mathbf{PolyUnsat}(\mathcal{T}, \omega^*, A \cdot g^{-\beta})$ to define a polynomial Q_{x} for each node x of $\mathcal{T}$. Notice that this implicitly sets $Q_{\mathrm{root}}(0) = a - \beta$. Now, the simulator sets $q_{\mathrm{x}} = b \cdot Q_{\mathrm{x}}$ for each node of $\mathcal{T}$. Thus, we have $q_{\mathrm{root}}(0) = ab - b\beta$, which has the correct value. $\mathcal{B}$ can generate private key for $\mathcal{T}$ as follows. Let i be the attribute associated to leaf node x. If $i \in \omega^*$, set

$$D_{\mathrm{x}} = g^{\frac{q_{\mathrm{x}}(0)}{w_i}} = g^{\frac{bQ_{\mathrm{x}}(0)}{w_i}} = \mathrm{B}^{\frac{Q_{\mathrm{x}}(0)}{w_i}}. \tag{9}$$

Otherwise, set

$$D_{\mathrm{x}} = g^{\frac{q_{\mathrm{x}}(0)}{b\beta_i}} = g^{\frac{bQ_{\mathrm{x}}(0)}{b\beta_i}} = g^{\frac{Q_{\mathrm{x}}(0)}{\beta_i}} = (g^{Q_{\mathrm{x}}(0)})^{\frac{1}{\beta_i}}. \tag{10}$$

At last, $d'_{\mathrm{id},\mathcal{T}}$ is returned to $\mathcal{A}$.

• Challenge. In this stage, $\mathcal{A}$ will submit two challenge messages M_1 and M_0 to the simulator. The simulator chooses randomly $\gamma \in \mathbb{Z}_p^*$ and flips a fair binary coin, ν. It returns the challenge ciphertext of an encryption of M_ν as

$$C^* = (\omega^*, \mathrm{C}'^{\gamma}, \mathrm{C}^{\gamma}, \mathrm{C}'^{\alpha\gamma}, M_\nu \mathrm{D}^{\gamma}, \{\mathrm{C}^{\gamma w_i}\}_{i \in \omega^*}). \tag{11}$$

Let $s = c\gamma$. If $\mathrm{D} = e(g,g)^{abc}$, the above ciphertext is a valid encryption of M_ν. Otherwise, if D is random, C_4^* gives no information about M_ν.

• Phase 2. The simulator proceeds as it did in Phase 1.

• Guess. $\mathcal{A}$ will submit a guess ν' of ν. If $\nu = \nu'$ the simulator will output $\mu' = 0$. Otherwise it will output $\mu' = 1$.

The advantage of $\mathcal{B}$ in solving the modified DBDH instance is $\epsilon/2$. This completes the proof.

Theorem 4.2. In the information theoretic sense, the scheme is secure in the FindKey-CPA game.

This theorem directly follows from [15], owing to the use of the perfect witness indistinguishability of the protocol [19] and the perfect hiding property of Pedersen's commitment [20].

Theorem 4.3. The scheme is secure in the Selective-ID ComputeNewKey game under the Diffie-Hellman assumption.

Proof. Let $\mathcal{A}$ be an adversary against the Selective-ID ComputeNewKey game. We build an simulator $\mathcal{B}$ which can find $h^{1/x}$ given $(g, h, X = g^x)$ (notice that it is equivalent to the Diffie-Hellman assumption). $\mathcal{B}$ proceeds as follows:

• Init. $\mathcal{B}$ receives the target identity id^* from $\mathcal{A}$.

• Setup. $\mathcal{B}$ prepares the public parameters as follows. Firstly, it takes X and h from the given instance. Secondly, it chooses $\alpha, \beta \in \mathbb{Z}_p^*$ and sets $Z = g^{-\mathrm{id}^*} X^\alpha$, $E = e(g,g)^\beta$. Thirdly, it chooses $\gamma, s_1' \in \mathbb{Z}_p^*$ and sets $Y_1^* = X^\gamma h^{-s_1'}, E_1 = e(g, Y_1)$. For all $i \in \mathcal{U}$, $\mathcal{B}$ chooses $w_i \in \mathbb{Z}_p^*$ and sets $T_i = g^{w_i}$. Finally, $\mathcal{B}$ gives the public parameters $(g, X, h, Z, T_1, \ldots, T_{|\mathcal{U}|}, E, E_1)$ to $\mathcal{A}$.

• Key generation. Suppose $\mathcal{A}$ requests a private key $(\mathrm{id}, \mathcal{T})$. The simulator will receive an element $R = h^{s_0} \cdot X^\theta$ with a WI proof of knowledge of (s_0, θ) from $\mathcal{A}$. If the proof is verified, $\mathcal{B}$ prepares the private key as follows:

– For $\mathrm{id} \neq \mathrm{id}^*$, $\mathcal{B}$ picks $s_1, r' \in \mathbb{Z}_p^*$ at random and defines $W = Y_1 R h^{s_1}$, $d_3' = s_1$. Then, d_1', d_2' can be generated as

$$(d_1', d_2') = ((g^{\mathrm{id}} Z)^{r'} W^{-\frac{\alpha}{\mathrm{id}-\mathrm{id}^*}}, X^{r'} W^{-\frac{1}{\mathrm{id}-\mathrm{id}^*}}). \tag{12}$$

Letting $\tilde{r}' = r' - \frac{\log_g(W)}{x \cdot (\mathrm{id}-\mathrm{id}^*)}$, the above components have the correct distribution. Recall that we have implicitly set

$$g^{q_{\mathrm{root}}(0)} = g^{\beta - x\gamma + s_1' \log_g(h)}. \tag{13}$$

To generate d_4', $\mathcal{B}$ first defines a procedure $\mathbf{Poly}(\mathcal{T}_\mathrm{x}, \lambda_\mathrm{x})$ as follows.

∗ This procedure takes an access tree $\mathcal{T}_\mathrm{x}$ and an integer $\lambda_\mathrm{x} \in \mathbb{Z}_p^*$ as inputs. Define a $b_{\mathrm{x}'}$ degree polynomial $Q_{\mathrm{x}'}$ for each node x' in $\mathcal{T}_\mathrm{x}$, where $b_{\mathrm{x}'} = k_{\mathrm{x}'} - 1$ and $k_{\mathrm{x}'}$ is the threshold value.

∗ For the root node, define Q_x by setting $Q_\mathrm{x}(0) = \lambda_\mathrm{x}$ and other b_x random points. For any other node x' in $\mathcal{T}_\mathrm{x}$, define $Q_{\mathrm{x}'}$ by setting $Q_{\mathrm{x}'}(0) = Q_{\mathrm{parent}(\mathrm{x}')}(\mathrm{index}(\mathrm{x}'))$ and other $b_{\mathrm{x}'}$ random points.

∗ Return those $Q_{\mathrm{x}'}$.

Next, $\mathcal{B}$ invokes $\mathbf{Poly}(\mathcal{T}, \beta)$, $\mathbf{Poly}(\mathcal{T}, \gamma)$, and $\mathbf{Poly}(\mathcal{T}, s_1')$ to get three polynomials $Q_\mathrm{x}^{(1)}, Q_\mathrm{x}^{(2)}, Q_\mathrm{x}^{(3)}$ with the same degree b_x for each node x in $\mathcal{T}$. Define the final polynomial $q_\mathrm{x} = Q_\mathrm{x}^{(1)} - x Q_\mathrm{x}^{(2)} + Q_\mathrm{x}^{(3)} \log_g h$. Notice that q_x have the correct shape due to $q_{\mathrm{root}}(0) = \beta - x\gamma + s_1' \log_g h$. Now $\mathcal{B}$ generates private key for each leaf node x in $\mathcal{T}$ as follows:

$$D_\mathrm{x} = g^{\frac{q_\mathrm{x}(0)}{w_i}} = g^{\frac{Q_\mathrm{x}^{(1)}(0) - x Q_\mathrm{x}^{(2)}(0) + Q_\mathrm{x}^{(3)}(0) \log_g h}{w_i}} = g^{\frac{Q_\mathrm{x}^{(1)}(0)}{w_i}} X^{-\frac{Q_\mathrm{x}^{(2)}(0)}{w_i}} h^{\frac{Q_\mathrm{x}^{(3)}(0)}{w_i}}, \tag{14}$$

where i is the attribute associated to leaf node x. Finally, $\mathcal{B}$ returns the private key to $\mathcal{A}$.

∗ For $\mathrm{id} = \mathrm{id}^*$, $\mathcal{B}$ uses the knowledge extractor to find (s_0, θ) of R by rewinding $\mathcal{A}$. Now it sets $s_1 = s_1' - s_0$ and returns as

$$(d_1', d_2', d_3') = (g^{\gamma+\theta} \cdot (g^{\mathrm{id}^*} \cdot Z)^r, X^r, s_1), \tag{15}$$

where $r \in \mathbb{Z}_p^*$ is chosen randomly. Additionally, d_4' can be generated as in the case of $\mathrm{id} \neq \mathrm{id}^*$.

• New key computation. At this point, $\mathcal{A}$ outputs two well-formed private keys for id^*, i.e., $d^{(1)}_{\mathrm{id}^*,\mathcal{T}} = (d_1^{(1)}, d_2^{(1)}, d_3^{(1)}, d_4^{(1)})$ and $d^{(2)}_{\mathrm{id}^*,\mathcal{T}} = (d_1^{(2)}, d_2^{(2)}, d_3^{(2)}, d_4^{(2)})$, such that $s = d_3^{(1)} \neq d_3^{(2)} = s'$. Then, we have $d_1^{(1)} = (Y_1 h^s)^{1/x} \cdot X^{\alpha r}, d_2^{(1)} = X^r$ and $d_1^{(2)} = (Y_1 h^{s'})^{1/x} \cdot X^{\alpha r'}, d_2^{(2)} = X^{r'}$, where $r, r' \in \mathbb{Z}_p^*$ are unknown to $\mathcal{B}$. Now, the simulator can compute $h^{1/x} = \left(\frac{d_1^{(1)}/(d_2^{(1)})^\alpha}{d_1^{(2)}/(d_2^{(2)})^\alpha}\right)^{\frac{1}{s-s'}}$.

5 Conclusions

We introduce the notion of accountable authority key policy ABE with a construction, which mitigates the key escrow problem in KP-ABE. The scheme is described in white-box setting and is very efficient. In addition, the construction is proved to be secure under mBDDH assumption in the standard model.

In the future work, we intend to construct an accountable authority key policy ABE scheme which enjoys the black-box traceability and security in the full model.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 61133014, 60970111, 60903189, 60903020), and National Basic Research Program of China (Grant No. 2007CB311201).

References

1 Sahai A, Waters B. Fuzzy identity based encryption. In: Proceedings of EUROCRYPT'05. LNCS, 3494. Berlin: Springer, 2005. 457–473
2 Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security. New York: ACM Press, 2006. 89-98
3 Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proceedings of 2007 IEEE Symposium on Security and Privacy. Washington: IEEE Computer Society, 2007. 321–334
4 Cheung L, Newport C. Porvably secure ciphertext policy ABE. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007. 456–465
5 Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290
6 Attrapadung N, Imai H. Dual-policy attribute based encryption. In: Applied Cryptography and Network Security. LNCS, 5536. Berlin: Springer, 2009. 168–185
7 Lewko A, Okamoto T, Sahai A, et al. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Proceedings of EUROCRYPT 2010. LNCS, 6110. Berlin: Springer, 2010. 62–91
8 Shamir A. Identity-based cryptosystems and signature schemes. In: Proceedings of CRYPTO 1984. LNCS, 196. Berlin: Springer, 1984. 47–53
9 Boneh D, Franklin M. Identity based encryption from the Weil pairing. In: Proceedings of CRYPTO'01. LNCS, 2139. Berlin: Springer, 2001. 213–229
10 Waters B. Efficient identity-based encryption without random oracles. In: Proceedings of EUROCRYPT'05. LNCS, 3494. Berlin: Springer, 2005. 114–127
11 Goyal V. Reducing trust in the PKG in identity based cryptosystems. In: Proceedings of CRYPTO'07. LNCS, 4622. Berlin: Springer, 2007. 430–447
12 Gentry C. Practical identity-based encryption without random oracles. In: Proceedings of EUROCRYPT'06. LNCS, 4004. Berlin: Springer, 2006. 445–464
13 Goyal V, Lu S, Sahai A, et al. Black-box accountable authority identity-based encryption. In: Proceedings of the 15th ACM Conference on Computer and Communications Security. New York: ACM Press, 2008. 427–436
14 Au M H, Huang Q, Liu J K, et al. Traceable and retrievable identity-based encryption. In: Applied Cryptography and Network Security–ACNS'08. LNCS, 5037. Berlin: Springer, 2008. 94–110
15 Libert B, Vergnaud D. Towards black-box accountable authority IBE with short ciphertexts and private keys. In: Proceedings of PKC'09. LNCS, 5443. Berlin: Springer, 2009. 235–255
16 Li J, Ren K, Kim K. A2BE: accountable attribute-based encryption for abuse free access control. Cryptology ePrint Archive, Report 2009/118
17 Yu S, Ren K, Lou W, et al. Defending against key abuse attacks in KP-ABE enabled broadcast systems. In: Proceedings of SecureComm 2009. Berlin: Springer, 2009. 311–329
18 Kiltz E, Vahlis Y. CCA2 Secure IBE: standard model efficiency through authenticated symmetric encryption. In: Proceedings of CT-RSA'08. LNCS, 4964. Berlin: Springer, 2008. 221–238
19 Okamoto T. Provably secure and practical identification schemes and corresponding signature schemes. In: Proceedings of CRYPTO 1992. LNCS, 740. Berlin: Springer, 1993. 31–53
20 Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing. In: Proceedings of CRYPTO 1991. LNCS, 576. Berlin: Springer, 1992. 129–140