Mitigating Key Escrow in Attribute-based Encryption

Yongtao Wang, Xiaonan Liu, Lulu Liang, Weiduan Feng, and Guang Yang
(Corresponding author: Yongtao Wang)

China Information Technology Security Evaluation Center, Beijing, P. R. China
(E-mail: wyt.itsec@gmail.com)

(Received Mar. 22, 2013; revised and accepted July 16, 2013)

Abstract

The notion of accountable authority introduced by Goyal (Crypto 2007) in identity-based encryption (IBE) setting is a novel approach to mitigate the (inherent) key escrow problem in identity based cryptosystems. As far as we know, the (inherent) key escrow problem also exists in attribute based encryption (ABE), for example ciphertext policy ABE (CP-ABE). In this paper, the concept of accountable authority is generalized to ABE setting. We first formalize the definitions and security models for accountable authority attribute-based encryption (A-ABE), and then present two concrete constructions. One is designed for the threshold ABE with large universe attributes, and the other is built for ciphertext policy ABE. In our scheme, a user will be identified by a pair (id, ω), where id denotes the identity of a user and ω denotes a attribute set associated to the user. In addition, both constructions are shown to be secure in the standard model under some reasonable assumptions.

Keywords: Accountable authority, attribute-based encryption, key escrow, standard model, traceability

1 Introduction

The concept of attribute-based encryption first introduced by Sahai and Waters [20] in 2005 provides one-to-many communication and is a new means for encrypted access control. In an ABE scheme, an access structure is integrated with the identity-based encryption. The original system of Sahai and Waters is a Threshold ABE system, in which both user's private key and ciphertext are associated with a set of attributes, respectively. A user can decrypt a ciphertext when at least d (threshold parameter) attributes overlap between the attribute set associated to the ciphertext and the attribute set associated to the user's private key. At present, there are many extensions for supporting more complex access structure, for example, the key policy ABE (KP-ABE) in [8, 17] and the ciphertext policy ABE (CP-ABE) in [2, 14]. We refer the reader to [4, 10, 15, 22] for some related results about ABE.

The motivation to consider accountable authority attribute-based encryption system is as follows. ABE can be viewed as a generalization of identity-based encryption, and it inherits the key escrow problem from IBE [3, 5, 21, 25]. In an ABE system, all users' private keys are issued by an unconditionally trusted authority. Such an authority possesses the master secret key of the system, and can decrypt all ciphertexts encrypted to any user. Moreover, the authority can redistribute users' private keys for malicious use. Thus, it has great significance that reducing the trust in the authority in an ABE system.

Goyal [6] first introduced the notion of accountable authority identity based encryption (A-IBE) for mitigating the key escrow problem in IBE. In an A-IBE scheme, if the authority redistributes some user's private key for malicious use, it will run the risk of being caught and sued by the user. The above goal can be achieved by the following approach: (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, that the generated key belongs to, 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, two keys from distinct families for an identity give evidence of the authority's misbehavior. At present, there are several constructions for A-IBE [1, 6, 7, 12, 13, 19]. In ABE setting, we notice that Li et al. [11] introduced a new type of ABE, which achieves the accountability of the authority. However, their scheme assumes that each user has a higher level secret before requesting an attribute private key. This work is orthogonal to ours.

In this paper, we formalize the definitions and security notions for accountable authority ABE and present two constructions. One is based on the large universe construction for ABE in [20], and the other is built on the ciphertext policy ABE scheme in [14]. We refer the reader to [23, 24] for some other constructions. Our approach to achieve the notion of accountable authority ABE is inspired by the A-IBE scheme proposed by Libert et al. [12, 13]. In our constructions, we introduce identity id to label a user's private key, and a user will be identified by a pair (id, ω), where ω is a set of attributes. Furthermore, we modify the scheme in [12], and achieve that the ability of decryption of a user is independent of his identity. After that, we non-trivially integrate the modify scheme with an ABE scheme. Thus, the notion of A-ABE is achieved. We describe our two constructions in the white-box model [6]. However, notice that the weak black-box tracing algorithm described in [12] can be trivially extended to our schemes. In addition, our constructions are shown to be secure in the standard model under some reasonable assumptions.

The rest of the paper is organized as follows. In Section 2 we recall some preliminaries. In Section 3, we give the construction for accountable authority threshold ABE with large universe attributes and its security proofs. We describe the construction for accountable authority ciphertext policy ABE and give its security proofs in Section 4. Finally, we conclude in Section 5.

2 Preliminaries

2.1 Bilinear Maps and Complexity Assumptions

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, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which has the following properties [3]:

- 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 $u, v \in \mathbb{G}_1$.

We require the following two assumptions holds which have been used before in [9, 12].

Definition 1. (Modified Diffie-Hellman Assumption, MDH). *Given a tuple (g, h, g^x), where g is a generator of $\mathbb{G}_1$, and $x \in \mathbb{Z}_p$, $h \in \mathbb{G}_1$ are random, there is no polynomial-time algorithm to output $h^{1/x}$.*

Definition 2. (Modified Bilinear Decision Diffie-Hellman Assumption, MBDDH). *This assumption states that the two distributions $(A = g^a, B = g^b, C = g^c, C' = g^{c^2}, D = e(g, g)^{abc})$ and $(A = g^a, B = g^b, C = g^c, C' = g^{c^2}, D = e(g, g)^z)$ 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 $\mathrm{Adv}_{\mathcal{B}}^{mbddh}(\kappa)$ of $\mathcal{B}$ is defined as*

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

We say that the MBDDH assumption holds if $\mathrm{Adv}_{\mathcal{B}}^{mbddh}(\kappa)$ is negligible for all polynomial-time adversaries.

Note that the MDH assumption is equivalent to the Diffie-Hellman assumption [12], and the MBDDH assumption is equivalent to the Bilinear Decision Diffie-Hellman assumption (BDDH) [9].

2.2 Syntax

We now describe the syntax of an A-ABE scheme. In our setting, a user will be identified by a pair (id, ω). It is allowable for different users with the same set of attributes. However, we require that a user should be assigned only one set of attributes. Formally, an accountable authority attribute-based encryption scheme consists of five polynomial-time algorithms described as follows:

- **Setup:** This algorithm takes as input a security parameter κ, and outputs a master public key mpk and a master secret key msk.
- **KeyGen:** This is an interactive protocol implemented between a user U and the authority. The public input to the authority and U consists of the mpk and (id, ω) (of U). The private input to the authority is the msk. In addition, a sequence of random coin tosses may be used by the authority and U as private inputs. At the end of the protocol, U can extract a private key $d_{id,\omega}$.
- **Encryption:** Takes as input the mpk, a set of attributes ω' (in CP-ABE, ω' is replaced by an access structure W) and a message M, this algorithm outputs a ciphertext E.
- **Decryption:** Takes as input a user secret key $d_{id,\omega}$ and a ciphertext E encrypted under ω' (or W), this algorithm outputs a plaintext message M if $|\omega \cap \omega'| \geq d$ (in CP-ABE, ω should satisfy the access structure W), where d is a threshold parameter.
- **Trace:** This algorithm takes a well-formed decryption key $d_{id,\omega}$ as input, and outputs the decryption key family number n_F.

Note that the above algorithms are described in the white-box traceability model. In this model, the Trace algorithm can only deal with a well-formed key. However, the Trace algorithm needs to trace a decryption box in the black-box traceability model [7].

2.3 Security Models

In this section, we describe the security models of our A-ABE scheme in the white-box setting. The reader is referred to [6, 7, 12] for further extensions in the black-box model.

The IND-SS-CPA game. We simply extend the selective model of [20] to our setting. Let $\mathcal{A}$ be an adversary.

- **Init.** $\mathcal{A}$ declares a set of attributes ω^* (in CP-ABE, $\mathcal{A}$ declares an access structure W^*).
- **Setup.** The challenger runs the Setup algorithm of A-ABE and gives the public parameters to $\mathcal{A}$.
- **Phase 1.** $\mathcal{A}$ runs the key generation protocol with the challenger for many pairs (id_j, ω_j), where $|\omega_j \cap \omega^*| < d$ (in CP-ABE, this condition should state that ω_j does not satisfy W^*) for all j. There is no limitation on id_j.
- **Challenge.** $\mathcal{A}$ submits two equal length messages M_0, M_1. The challenger flips a random coin, ν, and encrypts M_ν with ω^* (replaced by the access structure W^* in CP-ABE). The ciphertext is passed to $\mathcal{A}$.
- **Phase 2.** Phase 1 is repeated.
- **Guess.** $\mathcal{A}$ outputs a guess ν' of ν.

In the above game, the advantage of the adversary $\mathcal{A}$ is defined as $|\Pr[\nu' = \nu] - \frac{1}{2}|$.

The FindKey game. This game follows from [6] except slight modification for our setting. Let $\mathcal{A}$ be a malicious authority.

- **Setup.** $\mathcal{A}$ generates and gives the mpk with a pair (id, ω) 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 (id, ω). The challenger gets the key $d_{id,\omega}$ 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 $d'_{id,\omega}$. The challenger runs a key sanity check on it. It aborts if the check fails.

Let K_1 denote the event that $\mathrm{Trace}(d'_{id,\omega}) = \mathrm{Trace}(d_{id,\omega})$. Define $\mathcal{A}$' advantage as $\Pr[K_1]$. The above game emulates the attack that a malicious authority try to produce a private key belonging to the same family as the user's key generated in the key generation protocol.

The ComputeNewKey game. This game is defined along the line of [6]. Here, we describe it in the selective model. Let $\mathcal{A}$ be an adversary.

- **Init.** $\mathcal{A}$ declares an identity id^*.
- **Setup.** The challenger runs the Setup algorithm of A-ABE and gives the public parameters to $\mathcal{A}$.
- **Key Generation.** $\mathcal{A}$ runs the KeyGen with the challenger for many pairs (id_j, ω_j), where id_j must be distinct. There is no limitation on ω_j.
- **New Key Computation.** $\mathcal{A}$ outputs two decryption keys $d_{id^*,\omega}$ and $d'_{id^*,\omega}$. The challenger runs a key sanity check on them. It aborts if any check fails.

Let K_2 denote the event that $\mathrm{Trace}(d'_{id^*,\omega}) \neq \mathrm{Trace}(d_{id^*,\omega})$. In this game, define $\mathcal{A}$' advantage as $\Pr[K_2]$. Intuitively, this game emulates the attack that a user try to compute a new key belonging to a different family from his key generated in the key generation protocol.

Definition 3. *An A-ABE scheme is IND-SS-CPA secure if all polynomial time adversaries have at most a negligible advantage in the above three games.*

3 Accountable Authority Threshold ABE with Large Universe Attributes

We now describe our first construction, which mainly borrows ideas from the A-IBE scheme [12] and is based on the large universe construction of the attribute-based encryption scheme proposed by Sahai and Waters [20].

3.1 Description

In this construction, we assume n be the maximum size attribute set for a user, d be the threshold value and N be the set of $\{1, 2, \ldots, n+1\}$. For simplicity, let ω be a set of n elements of $\mathbb{Z}_p^*$. In addition, we can apply a collision-resistant hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, which allows arbitrary strings as attributes. 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 follows:

- **Setup:** First, generate a set of pairing groups at the security level κ. Next, choose randomly $x, y, y_1 \in \mathbb{Z}_p^*$, $h, Z, g_2, t_1, \ldots, t_{n+1} \in \mathbb{G}_1$, and set $X = g^x, Y_1 = g_2^{y_1}, g_1 = g^y$. Now, define a function T as in [20]: $T(x) = g_2^{x^n} \prod_{i=1}^{n+1} t_i^{\Delta_{i,N}(x)}$. Finally, set the master public key mpk as

  $$\{X, h, Z, t_1, \ldots, t_{n+1}, K = e(g_1, g_2), K_1 = e(g, Y_1)\},$$

  and the master private key msk as $\{x, y, y_1\}$.
- **KeyGen:** To generate a private key for a user U with (id, ω). The following protocol will be executed between U and the authority.
  - U chooses $s_0, \theta \in \mathbb{Z}_p^*$ at random, and provides a commitment $R = h^{s_0} \cdot X^\theta$ with an interactive witness indistinguishable proof of knowledge of the (s_0, θ) to the authority. In addition, U retains (s_0, θ).
  - The authority verifies the proof of knowledge, outputs $\perp$ if fails. Otherwise, it chooses randomly $s_1, r' \in \mathbb{Z}_p^*$ and a $d-1$ degree polynomial $q(x)$ with $q(0) = y - y_1$. The authority returns $d'_{id,\omega} = (d'_1, d'_2, d'_3, d'_4)$ as

    $$((g_2^{y_1} R h^{s_1})^{\frac{1}{x}} \cdot (g^{id} Z)^{r'}, X^{r'}, s_1,$$
    $$\{D_i = g_2^{q(i)} T(i)^{r_i}, \quad F_i = g^{r_i}\}_{i \in \omega}),$$

where r_i are chosen randomly from $\mathbb{Z}_p^*$.

- U chooses $r'' \in \mathbb{Z}_p^*$ at random and computes $d_{id,\omega} = (d_1, d_2, d_3, d_4)$ as $(d'_1/g^{\theta} \cdot (g^{id}Z)^{r''}, d'_2 \cdot X^{r''}, d'_3 + s_0, d'_4)$, which should equal

$$((g_2^{y_1}h^s)^{\frac{1}{x}} \cdot (g^{id}Z)^r, X^r, s, \\ \{D_i = g_2^{q(i)}T(i)^{r_i}, \quad F_i = g^{r_i}\}_{i\in\omega}),$$

where $r = r' + r''$ and $s = s_0 + s_1$. Now U checks $d_{id,\omega}$ as follows. First, the consistence of every element of d_4 should be checked. Choose an arbitrary d-element subset, S, of ω and compute $R_i = e(D_i, g)/e(F_i, T(i))$ for $i \in S$. Then compute $R_0 = \prod_{i\in S}(R_i^{\Delta_{i,S}(0)})$. Let Γ be an arbitrary subset of S such that $|\Gamma| = d-1$. Define $S' = \Gamma \cup \{0\}$. U checks the following relation for each $j \in \omega - S$:

$$\prod_{i\in S'}(R_i^{\Delta_{i,S'}(j)}) = \frac{e(D_j, g)}{e(F_j, T(j))}.$$

Second, U checks

$$\frac{e(d_1, X)}{e(g, h)^{d_3} \cdot e(g^{id}Z, d_2)} = K_1,$$

and $R_0 \cdot K_1 = K$. U outputs $\perp$ if any check fails. Otherwise, U sets his private key as $d_{id,\omega}$ and the key family number as $n_F = d_3 = s$.

- **Encryption:** 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'. Compute $E = (\omega', E_1, E_2, E_3, E_4, E_5)$ as

$$(\omega', g^t, Z^t, X^t, M \cdot e(g_1, g_2)^t, \{T(i)^t\}_{i\in\omega'}).$$

- **Decryption:** Let E be a valid encryption of M under ω'. E can be decrypted by a user with the private key $d_{id,\omega}$, where $|\omega \cap \omega'| \geq d$. First compute $E'_1 = E_1^{id} \cdot E_2 = (g^{id} \cdot Z)^t$ and $E'_5 = \prod_{i\in S}(\frac{e(D_i,E_1)}{e(F_i,T(i)^t)})^{\Delta_{i,S}(0)}$, where S be an arbitrary d-element subset of $\omega \cap \omega'$. Next decrypt E as:

$$M = E_4 \cdot \frac{e(E_1, h)^{d_3} \cdot e(E'_1, d_2)}{e(d_1, E_3) \cdot E'_5},$$

- **Trace:** Takes as input a well-formed decryption key $d_{id,\omega} = (d_1, d_2, d_3, d_4)$, this algorithm outputs the decryption key family number $n_F = d_3$.

3.2 Analysis of the Construction

If the cryptosystem is operated as specified, we have $E'_5 = e(g^t, g_2)^{y-y_1}$ and

$$\begin{aligned} & E_4 \cdot \frac{e(E_1, h)^{d_3} \cdot e(E'_1, d_2)}{e(d_1, E_3) \cdot E'_5} \\ &= E_4 \cdot \frac{e(g^t, h)^s \cdot e((g^{id}Z)^t, X^r)}{e((g_2^{y_1}h^s)^{\frac{1}{x}} \cdot (g^{id}Z)^r, X^t) \cdot e(g^t, g_2)^{y-y_1}} \\ &= E_4 \cdot \frac{1}{e(g^{y_1}, g_2^t) \cdot e(g^{y-y_1}, g_2^t)} \\ &= M. \end{aligned}$$

The above construction is based on the large universe construction in [20]. However, we notice that one can use our approach to construct an A-ABE based on the large universe construction of KP-ABE [8]. Now we present the security proofs for our construction.

Theorem 1. *The above construction is IND-SS-CPA secure 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 MBDDH instance with advantage $\frac{\epsilon}{2}$. First, let the challenger set the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with an efficient bilinear map e. Second, the challenger flips a fair binary coin μ outside of $\mathcal{B}$'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 $\alpha, \beta, \gamma, y_1 \in \mathbb{Z}_p^*$ and sets $h = A^{\alpha}, X = C^{\beta}, Z = X^{\gamma}, K = e(A, B)$ and $K_1 = e(g, B^{y_1})$ (we have implicitly set $g_1 = A, g_2 = B$). $\mathcal{B}$ chooses a random n degree polynomial $f(x)$ and computes an n degree polynomial $u(x)$ such that $u(x) = -x^n$ for all $x \in \omega^*$ and $u(x) \neq -x^n$ for $x \notin \omega^*$. For $i \in \{1, \ldots, n+1\}$, $\mathcal{B}$ sets $t_i = g_2^{u(i)}g^{f(i)}$. Now $\mathcal{B}$ gives the public parameters to $\mathcal{A}$.

- **Phase 1.** Suppose $\mathcal{A}$ requests a private key (id, ω). We assume that id is non zero. 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 succeeds to be verified, $\mathcal{B}$ prepares the private key as follows. Firstly, it chooses randomly $s_1 \in \mathbb{Z}_p^*$ and defines $W = Y_1 \cdot R \cdot h^{s_1}$, $d'_3 = s_1$. Then, d'_1 and d'_2 are generated as:

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

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

$$\begin{aligned} d'_1 &= W^{1/(c\beta)} \cdot (g^{id} \cdot Z)^{\tilde{r}'} \\ &= W^{1/(c\beta)} \cdot (g^{id} \cdot X^{\gamma})^{r'} \cdot (g^{id} \cdot X^{\gamma})^{-\frac{w}{c\beta\cdot id}} \\ &= (g^{id} \cdot Z)^{r'} \cdot W^{-\frac{\gamma}{id}}, \end{aligned}$$

and $d'_2 = X^{\tilde{r}'} = X^{r'} \cdot (g^{c\beta})^{-\frac{w}{c\beta\cdot id}} = X^{r'} \cdot W^{-\frac{1}{id}}$, where $w = \log_g(W)$. To generate d'_4, $\mathcal{B}$ proceeds as follows. First, define three sets $\Gamma = \omega^* \cap \omega$, Γ' such that $\Gamma \subseteq \Gamma' \subseteq \omega$ and $|\Gamma'| = d-1$, and $S = \Gamma' \cup \{0\}$. For $i \in \Gamma'$, Define $D_i = g_2^{\lambda_i}T(i)^{r_i}, F_i = g^{r_i}$, where r_i, λ_i are chosen randomly in $\mathbb{Z}_p^*$. For $i \in \omega - \Gamma'$, $\mathcal{B}$ chooses $r'_i \in \mathbb{Z}_p^*$ and defines D_i as

$$(\prod_{j\in\Gamma'} g_2^{\lambda_j\Delta_{j,S}(i)})((g_1 g^{-y_1})^{\frac{-f(i)}{i^n+u(i)}}(g_2^{i^n+u(i)}g^{f(i)})^{r'_i})^{\Delta_{0,S}(i)},$$

and $F_i = ((g_1 g^{-y_1})^{\frac{-1}{i^n+u(i)}} g^{r'_i})^{\Delta_{0,S}(i)}$. Let $r_i = (r'_i - \frac{a-y_1}{i^n+u(i)})\Delta_{0,S}(i)$, the above values have the correct distributions. D_i is equal to

$$\begin{aligned}
&\big(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\big)\big((g_1 g^{-y_1})^{\frac{-f(i)}{i^n+u(i)}}(g_2^{i^n+u(i)} g^{f(i)})^{r'_i}\big)^{\Delta_{0,S}(i)}\\
&= \big(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\big)\big(g^{\frac{-(a-y_1)f(i)}{i^n+u(i)}}(g_2^{i^n+u(i)} g^{f(i)})^{r'_i}\big)^{\Delta_{0,S}(i)}\\
&= \big(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\big) g_2^{(a-y_1)\Delta_{0,S}(i)}\big((g_2^{i^n+u(i)} g^{f(i)})^{r_i}\big)^{\Delta_{0,S}(i)}\\
&= g_2^{q(i)} T(i)^{r_i}
\end{aligned}$$

and

$$\begin{aligned}
F_i &= ((g_1 g^{-y_1})^{\frac{-1}{i^n+u(i)}} g^{r'_i})^{\Delta_{0,S}(i)}\\
&= (g^{\frac{-(a-y_1)}{i^n+u(i)}} g^{r'_i})^{\Delta_{0,S}(i)} = g^{r_i}.
\end{aligned}$$

Finally, $d'_{id,\omega}$ is returned to $\mathcal{A}$.

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

  $$E^* = (\omega^*, C^\rho, C'^{\beta\gamma\rho}, C'^{\beta\rho}, M_\nu D^\rho, \{C^{\rho f(i)}\}_{i\in\omega^*}).$$

  Let $t = c\rho$. If $D = e(g,g)^{abc}$, the above ciphertext is a valid encryption of M_ν. Otherwise, if D is random, E_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 the simulator in solving MBDDH instance is $\frac{\epsilon}{2}$. □

Theorem 2. *The construction is secure in the FindKey game in the information theoretic sense.*

The above theorem directly follows from [12], due to the uses of the perfect hiding property of Pedersen's commitment [18] and the perfect witness indistinguishability of the protocol [16].

Theorem 3. *The construction is secure in the Compute-NewKey game under the MDH assumption.*

Proof. Let $\mathcal{A}$ be an adversary against the Compute-NewKey game. We build a simulator $\mathcal{B}$ which can find $h^{1/x}$ given $(g, h, X = g^x)$. $\mathcal{B}$ proceeds as follows:

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

- **Setup.** To prepare the public parameters. First, $\mathcal{B}$ takes X, h from the MDH instance. Second, it chooses $\alpha, \beta, \gamma \in \mathbb{Z}_p^*$, $t_1, \ldots, t_{n+1} \in \mathbb{G}_1$ and sets $K = e(g^\alpha, g^\beta), Z = g^{-id^*} X^\gamma$. $\mathcal{B}$ chooses ρ, s'_1 and defines $Y_1 = X^\rho h^{-s'_1}$ and computes K_1. Finally, $\mathcal{B}$ gives the public parameters to $\mathcal{A}$.

- **Key Generation.** Suppose $\mathcal{A}$ requests a private key (id, ω). 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 succeeds to be verified, $\mathcal{B}$ prepares the private key as follows:

  - For $id \neq id^*$, $\mathcal{B}$ 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

    $$(d'_1, d'_2) = ((g^{id} Z)^{r'} W^{-\frac{\gamma}{id-id^*}}, X^{r'} W^{-\frac{1}{id-id^*}}).$$

    If we let $\tilde{r}' = r' - \frac{\log_g(W)}{x\cdot(id-id^*)}$, the above components have the correct distribution. Additionally, d'_4 can be generated as follows. By our setting of Y_1, we have $y_1 = \frac{x\rho - s'_1 \log_g h}{\beta}$. It first chooses an arbitrary subset Γ of ω such that $|\Gamma| = d-1$. For $i \in \Gamma$, it chooses randomly $\lambda_i, r_i \in \mathbb{Z}_p^*$ and sets $D_i = g_2^{\lambda_i} T(i)^{r_i}$, $F_i = g^{r_i}$. We implicitly select a $d-1$ degree polynomial $q(x)$ with $q(i) = \lambda_i$ and $q(0) = \alpha - \frac{x\rho - s'_1 \log_g h}{\beta}$. For $i \in \omega - \Gamma$, it chooses $r_i \in \mathbb{Z}_p^*$ and sets

    $$D_i = \big(\prod_{j\in\Gamma} g_2^{\lambda_j \Delta_{j,S}(i)}\big)\big(g_2^\alpha X^{-\rho} h^{s'_1}\big)^{\Delta_{0,S}(i)} T(i)^{r_i},$$

    and $F_i = g^{r_i}$. Notice that the above values have correct distributions.

  - For $id = id^*$, $\mathcal{B}$ uses the knowledge extractor to find (s_0, θ) of R by rewinding $\mathcal{A}$. It sets $s_1 = s'_1 - s_0$. Now it chooses $r \in \mathbb{Z}_p^*$ and computes

    $$(d'_1, d'_2, d'_3) = (g^{\rho+\theta} \cdot (g^{id^*} \cdot Z)^r, X^r, s_1).$$

    For d'_1, we have $(Y_1 R h^{s_1})^{1/x} \cdot (g^{id^*} \cdot Z)^r = g^{\rho+\theta} \cdot (g^{id^*} \cdot Z)^r$. The above values have correct distributions. To prepare d'_4, $\mathcal{B}$ does as the above description for the case of $id \neq id^*$.

  Finally, $d'_{id,\omega}$ is returned to $\mathcal{A}$.

- **New Key Computation.** At this point, $\mathcal{A}$ outputs two well-formed private keys for id^*, i.e., $d^{(1)}_{id^*,\omega} = (d_1^{(1)}, d_2^{(1)}, d_3^{(1)}, d_4^{(1)})$ and $d^{(2)}_{id^*,\omega} = (d_1^{(2)}, d_2^{(2)}, d_3^{(2)}, d_4^{(2)})$, 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 $h^{1/x} = \big(\frac{d_1^{(1)}/(d_2^{(1)})^\alpha}{d_1^{(2)}/(d_2^{(2)})^\alpha}\big)^{\frac{1}{s-s'}}$.

This completes the proof. □

4 Accountable Authority Ciphertext Policy ABE

This construction is built on the ciphertext policy ABE scheme in [14]. We assume n be the maximum size attribute set for a user and N be the set of $\{1, 2, \ldots, n\}$. We refer to attributes i and their negations $\neg i$ as literals. The ciphertext policy achieved is AND gate access structure $W = \bigwedge_{i \in I} \underline{i}$, where $I \in N$ and every i is a literal, i.e., i or $\neg i$.

4.1 Description

- **Setup:** First, generate a set of pairing groups at the security level κ. Next, choose randomly $x, y, y_1, t_1, \ldots, t_{3n} \in \mathbb{Z}_p^*$, a generator g of $\mathbb{G}_1$ and $h, Z \in \mathbb{G}_1$. Set $X = g^x, Y_1 = g^{y_1}, K = e(g, g)^y, K_1 = e(g, Y_1)$ and $T_i = g^{t_i}$ for $i \in \{1, \ldots, 3N\}$. Finally, set the master private key msk and the master public key mpk as:
$$msk = \{x, y, y_1, t_1, \ldots, t_{3n}\},$$
$$mpk = \{X, h, Z, K, K_1, T_1, \ldots, T_{3n}\}.$$
- **KeyGen:** To generate a private key for a user U with (id, ω). The following protocol will be executed between U and the authority.
  - U chooses $s_0, \theta \in \mathbb{Z}_p^*$ at random, and provides a commitment $R = h^{s_0} \cdot X^\theta$ with an interactive witness indistinguishable proof of knowledge of the (s_0, θ) to the authority. In addition, U retains (s_0, θ).
  - The authority verifies the proof of knowledge, outputs $\perp$ if fails. Otherwise, it selects $r_1, \ldots, r_n \in \mathbb{Z}_p^*$ such that $\sum_{i=1}^n r_i = y - y_1$. The authority first generates d'_4. For each $i \in N$, set $D_i = g^{\frac{r_i}{t_i}}$ if $i \in \omega$; otherwise, let it be $g^{\frac{r_i}{t_{n+i}}}$. In addition, set $F_i = g^{\frac{r_i}{t_{2n+i}}}$ for all $i \in N$. Now it chooses randomly $s_1, r' \in \mathbb{Z}_p^*$ and returns $d'_{id,\omega} = (d'_1, d'_2, d'_3, d'_4)$ as
$$((g^{y_1} R h^{s_1})^{\frac{1}{x}} \cdot (g^{id} Z)^{r'}, X^{r'}, s_1, \{D_i, F_i\}_{i \in N}).$$
  - U chooses $r'' \in \mathbb{Z}_p^*$ at random and computes $d_{id,\omega} = (d_1, d_2, d_3, d_4)$ as $(d'_1/g^\theta \cdot (g^{id} Z)^{r''}, d'_2 \cdot X^{r''}, d'_3 + s_0, d'_4)$, which should equal
$$((g^{y_1} h^s)^{\frac{1}{x}} \cdot (g^{id} Z)^r, X^r, s, \{D_i, F_i\}_{i \in N}),$$
where $r = r' + r''$ and $s = s_0 + s_1$. Now U checks $d_{id,\omega}$ as follows. First, the consistence of every element of d_4 should be checked. For each $i \in N$, compute $R_i = e(D_i, T_i)$ if $i \in \omega$, otherwise, compute $R_i = e(D_i, T_{n+i})$. Set $R_0 = \prod_{i \in N} R_i$. In addition, compute $R'_i = e(F_i, T_{2n+i})$ for $i \in N$ and check $R_i = R'_i$ for $i \in N$. Next, check
$$\frac{e(d_1, X)}{e(g, h)^{d_3} \cdot e(g^{id} Z, d_2)} = K_1,$$
and $R_0 \cdot K_1 = K$. U outputs $\perp$ if any check fails. Otherwise, U sets his private key as $d_{id,\omega}$ and the key family number as $n_F = d_3 = s$.
- **Encryption:** The sender chooses $t \in \mathbb{Z}_p^*$ at random to encrypt a message $M \in \mathbb{G}_2$ under $W = \bigwedge_{i \in I} \underline{i}$. For each $i \in I$, let E_i be T_i^t if $\underline{i} = i$, or be T_{n+i}^t if $\underline{i} = \neg i$. For each $i \in N \setminus I$, let E_i be T_{2n+i}^t. Compute $C = (W, C_1, C_2, C_3, C_4, C_5)$ as
$$(W, g^t, Z^t, X^t, M \cdot K^t, \{E_i\}_{i \in N}).$$
- **Decryption:** Let C be a valid encryption of M under $W = \bigwedge_{i \in I} \underline{i}$. C can be decrypted by a user with the private key $d_{id,\omega}$, where ω satisfies W. First compute $C'_1 = C_1^{id} \cdot C_2 = (g^{id} \cdot Z)^t$. Then compute
$$C'_5 = \prod_{\underline{i} = i \wedge i \in \omega} e(D_i, E_i) \prod_{\underline{i} = \neg i \wedge i \notin \omega} e(D_i, E_i) \prod_{i \notin I} e(F_i, E_i).$$
Next decrypt C as:
$$M = C_4 \cdot \frac{e(C_1, h)^{d_3} \cdot e(C'_1, d_2)}{e(d_1, C_3) \cdot C'_5},$$
- **Trace:** Takes as input a well-formed decryption key $d_{id,\omega} = (d_1, d_2, d_3, d_4)$, this algorithm outputs the decryption key family number $n_F = d_3$.

4.2 Analysis of the Construction

If the cryptosystem is operated as specified, we have $C'_5 = e(g^t, g)^{y - y_1}$ and

$$\begin{aligned} & C_4 \cdot \frac{e(C_1, h)^{d_3} \cdot e(C'_1, d_2)}{e(d_1, C_3) \cdot C'_5} \\ &= C_4 \cdot \frac{e(g^t, h)^s \cdot e((g^{id} Z)^t, X^r)}{e((g^{y_1} h^s)^{\frac{1}{x}} \cdot (g^{id} Z)^r, X^t) \cdot e(g^t, g)^{y - y_1}} \\ &= C_4 \cdot \frac{1}{e(g^{y_1}, g^t) \cdot e(g^{y - y_1}, g^t)} \\ &= M. \end{aligned}$$

Now we present the security proofs for our second construction.

Theorem 4. *The above construction is IND-SS-CPA secure 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 MBDDH instance with advantage $\frac{\epsilon}{2}$. First, let the challenger set the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with an efficient bilinear map e. Second, the challenger flips a fair binary coin μ outside of $\mathcal{B}$'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 structure $W^* = \bigwedge_{i \in I} \underline{i}$ from $\mathcal{A}$.

- **Setup.** $\mathcal{B}$ chooses $\alpha, \beta, \gamma, \delta \in \mathbb{Z}_p^*$ and sets $K = e(A, B), K_1 = e(g, B^{\delta}), h = A^{\alpha}, X = C^{\beta}, Z = X^{\gamma}$. For each $i \in N$, $\mathcal{B}$ selects $\alpha_i, \beta_i, \gamma_i \in \mathbb{Z}_p^*$. For $i \in I$, set $T_i = g^{\alpha_i}, T_{n+i} = B^{\beta_i}$ and $T_{2n+i} = B^{\gamma_i}$ if $\underline{i} = i$; otherwise, set $T_i = B^{\alpha_i}, T_{n+i} = g^{\beta_i}$ and $T_{2n+i} = B^{\gamma_i}$. For $i \notin I$, set $T_i = B^{\alpha_i}, T_{n+i} = B^{\beta_i}$ and $T_{2n+i} = g^{\gamma_i}$. Now $\mathcal{B}$ gives the public parameters to $\mathcal{A}$.

- **Phase 1.** Suppose $\mathcal{A}$ requests a private key (id, ω), where id is non zero and ω does not satisfy W^*. 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 succeeds to be verified, $\mathcal{B}$ prepares the private key as follows. Firstly, it chooses randomly $s_1 \in \mathbb{Z}_p^*$, and defines $W = B^{\delta} \cdot R \cdot h^{s_1}$ and $d'_3 = s_1$. Then, d'_1 and d'_2 are generated as:

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

  where r' is chosen randomly from $\mathbb{Z}_p^*$. If we let $\tilde{r}' = r' - \frac{\log_g(W)}{(c\beta)\cdot id}$, d'_1 and d'_2 have the correct distributions. To generate d'_4, $\mathcal{B}$ proceeds as follows. There exist $j \in I$ such that: either $j \in \omega$ and $\underline{j} = \neg j$, or $j \notin S$ and $\underline{j} = j$, due to ω does not satisfy W^*. For simplicity, assume $\mathcal{B}$ chooses such j: $j \notin S$ and $\underline{j} = j$. For each $i \in N, i \neq j$, $\mathcal{B}$ selects randomly $r'_i \in \mathbb{Z}_p^*$ and sets $r_i = r'_i b$. It sets $r_j = ab - b\delta - \sum_{i \in N \wedge i \neq j} r_i$. Then D_j and F_j can be computed as:

  $$D_j = g^{\frac{r_j}{b\beta_j}} = A^{\frac{1}{\beta_j}} g^{\frac{-\delta - \sum_{i \in N \wedge i \neq j} r'_i}{\beta_j}},$$
  $$F_j = g^{\frac{r_j}{b\gamma_j}} = A^{\frac{1}{\gamma_j}} g^{\frac{-\delta - \sum_{i \in N \wedge i \neq j} r'_i}{\gamma_j}}.$$

  For $i \neq j$, we have follows. When $i \in \omega$, set $D_i = B^{\frac{r'_i}{\alpha_i}}$ if $i \in I \wedge \underline{i} = i$ or set $D_i = g^{\frac{r'_i}{\alpha_i}}$ if $(i \in I \wedge \underline{i} = \neg i) \vee i \notin I$. When $i \notin \omega$, set $D_i = g^{\frac{r'_i}{\beta_i}}$ if $(i \in I \wedge \underline{i} = i) \vee i \notin I$ or set $D_i = B^{\frac{r'_i}{\beta_i}}$ if $i \in I \wedge \underline{i} = \neg i$. For F_i, we set $F_i = g^{\frac{r'_i}{\gamma_i}}$ if $i \in I$; otherwise, set $F_i = B^{\frac{r'_i}{\gamma_i}}$. It is easy to see the above values have correct distributions. Finally, $d'_{id,\omega}$ is returned to $\mathcal{A}$.

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

  $$C^* = (W^*, C^{\rho}, C'^{\beta\gamma\rho}, C'^{\beta\rho}, M_\nu D^{\rho},$$
  $$\{C^{\alpha_i} | i \in I \wedge \underline{i} = i\}, \{C^{\beta_i} | i \in I \wedge \underline{i} = \neg i\}, \{C^{\gamma_i} | i \notin I\}).$$

  Let $t = c\rho$. If $D = e(g, g)^{abc}$, the above ciphertext is a valid encryption of M_ν. Otherwise, if D is random, the challenge ciphertext 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 the simulator in solving MBDDH instance is $\frac{\epsilon}{2}$. □

Theorem 5. *The construction is secure in the FindKey game in the information theoretic sense.*

This theorem directly follows from [12], due to the uses of the perfect hiding property of Pedersen's commitment [18] and the perfect witness indistinguishability of the protocol [16].

Theorem 6. *The construction is secure in the Compute-NewKey game under the MDH assumption.*

Proof. Let $\mathcal{A}$ be an adversary against the Compute-NewKey game. We build a simulator $\mathcal{B}$ which can find $h^{1/x}$ given $(g, h, X = g^x)$. $\mathcal{B}$ proceeds as follows:

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

- **Setup.** To prepare the public parameters. First, $\mathcal{B}$ takes X, h from the MDH instance. Second, it chooses $\alpha, \gamma, t_1, \ldots, t_{3n} \in \mathbb{Z}_p^*$, and sets $K = e(g, g)^{\alpha}, Z = g^{-id^*} X^{\gamma}$ and $T(i) = g^{t_i}$ for $i \in \{1, \ldots, 3n\}$. $\mathcal{B}$ chooses ρ, s'_1 and defines $Y_1 = X^{\rho} h^{-s'_1}$ and computes K_1. Finally, $\mathcal{B}$ gives the public parameters to $\mathcal{A}$.

- **Key Generation.** Suppose $\mathcal{A}$ requests a private key (id, ω). 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 succeeds to be verified, $\mathcal{B}$ prepares the private key as follows:

  – For $id \neq id^*$, $\mathcal{B}$ 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 $(d'_1, d'_2) = ((g^{id} Z)^{r'} W^{-\frac{\gamma}{id - id^*}}, X^{r'} W^{-\frac{1}{id - id^*}})$. If we let $\tilde{r}' = r' - \frac{\log_g(W)}{x \cdot (id - id^*)}$, the above components have the correct distribution. Additionally, d'_4 can be generated as follows. It chooses randomly $j \in \omega$, $r_i \in \mathbb{Z}_p^*$ for $i \in N \wedge i \neq j$. Set $r_j = \alpha - x\rho + s'_1 \log_g h - \sum_{i \in N \wedge i \neq j} r_i$. Then D_j and F_j can be computed as:

  $$D_j = g^{\frac{\alpha - \sum_{i \in N \wedge i \neq j} r_i}{t_j}} X^{\frac{-\rho}{t_j}} h^{\frac{s'_1}{t_j}},$$
  $$F_j = g^{\frac{\alpha - \sum_{i \in N \wedge i \neq j} r_i}{t_{2n+j}}} X^{\frac{-\rho}{t_{2n+j}}} h^{\frac{s'_1}{t_{2n+j}}}.$$

  For $i \neq j \wedge i \in N$, D_i and F_i can be computed as the description of the scheme. Notice that the above values have correct distributions.

  – For $id = id^*$, $\mathcal{B}$ uses the knowledge extractor to find (s_0, θ) of R by rewinding $\mathcal{A}$. It sets $s_1 = s'_1 - s_0$. Now it chooses $r \in \mathbb{Z}_p^*$ and computes

  $$(d'_1, d'_2, d'_3) = (g^{\rho + \theta} \cdot (g^{id^*} \cdot Z)^r, X^r, s_1).$$

For d'_1, we have $(Y_1 R h^{s_1})^{1/x} \cdot (g^{id^*} Z)^r = g^{\rho+\theta} \cdot (g^{id^*} Z)^r$. Thus, the above values have correct distributions. To prepare d'_4, $\mathcal{B}$ does as the above description for the case of $id \neq id^*$.

Finally, $d'_{id,\omega}$ is returned to $\mathcal{A}$.

- **New Key Computation.** At this point, $\mathcal{A}$ outputs two well-formed private keys for id^*, i.e., $d^{(1)}_{id^*,\omega} = (d^{(1)}_1, d^{(1)}_2, d^{(1)}_3, d^{(1)}_4)$ and $d^{(2)}_{id^*,\omega} = (d^{(2)}_1, d^{(2)}_2, d^{(2)}_3, d^{(2)}_4)$, such that $s = d^{(1)}_3 \neq d^{(2)}_3 = s'$. Then, we have $d^{(1)}_1 = (Y_1 h^s)^{1/x} \cdot X^{\alpha r}, d^{(1)}_2 = X^r$ and $d^{(2)}_1 = (Y_1 h^{s'})^{1/x} \cdot X^{\alpha r'}, d^{(2)}_2 = X^{r'}$, where $r, r' \in \mathbb{Z}^*_p$ are unknown to $\mathcal{B}$. Now, the simulator can compute $h^{1/x} = (\frac{d^{(1)}_1/(d^{(1)}_2)^\alpha}{d^{(2)}_1/(d^{(2)}_2)^\alpha})^{\frac{1}{s-s'}}$.

This completes the proof. □

5 Conclusions

To mitigate the key escrow problem existed in threshold attribute based encryption and ciphertext policy attribute based encryption, we introduce the notion of accountable authority attribute based encryption and present related constructions. In addition, we proof the security of our constructions in the standard model under some reasonable assumptions.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (No. 61202493). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. H. Au, Q. Huang, J. K. Liu, W. Susilo, D. S. Wong, and G. Yang, "Traceable and retrievable identity-based encryption," in *Applied Cryptography and Network Security*, pp. 94–110, 2008.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology - Crypto '01*, pp. 213–229, 2001.

[4] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[5] C. Gentry, "Practical identity-based encryption without random oracles," in *Advances in Cryptology - Eurocrypt '06*, pp. 445–464, 2006.

[6] V. Goyal, "Reducing trust in the pkg in identity based cryptosystems," in *Advances in Cryptology - Crypto '07*, pp. 430–447, 2007.

[7] V. Goyal, S. Lu, A. Sahai, and B. Waters, "Black-box accountable authority identity-based encryption," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 427–436, 2008.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[9] E. Kiltz and Y. Vahlis, "Cca2 secure ibe: standard model efficiency through authenticated symmetric encryption," in *Topics in Cryptology - CT-RSA '08*, pp. 221–238, 2008.

[10] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[11] J. Li, K. Ren, and K. Kim. "A2be: Accountable attribute-based encryption for abuse free access control,". Cryptology ePrint Archive, http://eprint.iacr.org/2009/118, 2009.

[12] B. Libert and D. Vergnaud, "Towards black-box accountable authority ibe with short ciphertexts and private keys," in *Public Key Cryptography - PKC '09*, pp. 235–255, 2009.

[13] B. Libert and D. Vergnaud, "Towards practical black-box accountable authority ibe: Weak black-box traceability with short ciphertexts and private keys," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 7189–7204, 2011.

[14] C. Ling and N. Calvin, "Provably secure ciphertext policy abe," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 456–465, 2007.

[15] D. Nali, C. M. 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.

[16] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes," in *Advances in Cryptology - Crypto '92*, pp. 31–53, 1993.

[17] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 195–203, 2007.

[18] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology - Crypto '91*, pp. 129–140, 1992.

[19] A. Sahai and H. Seyalioglu, "Fully secure accountable-authority identity-based encryption," in *Public Key Cryptography - PKC '11*, pp. 296–316, 2011.

[20] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - Eurocrypt '05*, pp. 457–473, 2005.

[21] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology - Crypto '84*, pp. 47–53, 1985.

[22] Q. Tang and D. Ji, "Verifiable attribute based encryption.," *International Journal of Network Security*, vol. 10, no. 2, pp. 114–120, 2010.

[23] Y. Wang, K. Chen, and Y. Long, "Towards accountable authority attribute-based encryption," *High Technology Letters*, vol. 19, no. 1, pp. 82–87, 2013.

[24] Y. Wang, K. Chen, Y. Long, and Z. Liu, "Accountable authority key policy attribute-based encryption," *Science China Information Sciences*, vol. 55, no. 7, pp. 1631–1638, 2012.

[25] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology - Eurocrypt '05*, pp. 114–127, 2005.

Yongtao Wang was born in 1980. He received his Ph.D. degree in Computer Science and Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently a Research Assistant at China Information Technology Security Evaluation Center, Beijing, China. His research interests include information security and modern cryptography, etc.

Xiaonan Liu was born in Changchun city of Jilin province in June 1982.Graduated from College of Computer Science and Technology of Jilin University in June 2010 and received a doctors degree in the same year. The main research field is Computer Network, Wireless Communication Network and Information Security. She works in China Information Technology Security Evaluation Center, her position is Assistant Researcher. Dr. Liu published academic papers more than 10 articles, is the member of China Computer Federation and ACM. Articles have been published in publications such as Lecture Notes in Computer Science and Journal of Computational Information Systems.

Lulu Liang reveived the B.S and Ph.D. degree from Beijing Jiaotong University, China in 2007, and 2012 respectively. Since 2012, he has been with China Information Technology Security Evaluation Center as a researcher. His current research interests include wireless sensor networks, risk assessment, web security.

Weiduan Feng was born in 1982. He received his Ph.D. degree in Department of Mathematics from Shanghai Jiao Tong University, Shanghai, China, in 2010. He is currently an associate researcher at China Information Technology Security Evaluation Center, Beijing, China. His research interests include information security and modern cryptography, etc.

Guang Yang was born in 1980. She received her Ph.D. degree in Computer Application from Harbin Engineering University, Harbin, China, in 2009. She is currently an associate professor at China Information Technology Security Evaluation Center, Beijing, China. Her research interests include information security and wireless network security, etc.