Traceable Ciphertet-Policy Attribute-Based Encryption with Constant Decryption

Guangbo Wang[1*], Feng Li[1], Pengcheng Wang[1], Yixiao Hu[1]
[1] Beijing Science and Technology Information Research Center
Beijing 100036, Beijing – P. R. China
[e-mail: 691759571@qq.com]
*Corresponding author: Guangbo Wang

Received December 4, 2016; revised September 13, 2020; accepted July 9, 2021; published September 30, 2021

Abstract

We provide a traceable ciphertext-policy attribute based encryption (CP-ABE) construction for monotone access structures (MAS) based on composite order bilinear groups, which is secure adaptively under the standard model. We construct this scheme by making use of an "encoding technique" which represents the MAS by their minimal sets to encrypt the messages. To date, for all traceable CP-ABE schemes, their encryption costs grow linearly with the MAS size, the decryption costs grow linearly with the qualified rows in the span programs. However, in our traceable CP-ABE, the ciphertext is linear with the minimal sets, and decryption needs merely three bilinear pairing computations and two exponent computations, which improves the efficiency extremely and has constant decryption. At last, the detailed security and traceability proof is given.

Keywords: ciphertext-policy, traceability, constant decryption

http://doi.org/10.3837/tiis.2021.09.017 **ISSN : 1976-7277**

1. Introduction

The notation of ABE, firstly put forward by Sahai et al. [1], is a summarization of fuzzy IBE[2]. Then Goyal et al. put foward a Key-Policy ABE, and further defined a CP-ABE [3]. In the CP-ABE, the private key is constructed over the attributes, and the user designs the access control policy over some attributes when encrypting data. If the attributes associating with private key achieve the access control policy, then the decryption is successful. CP-ABE provides a new method to carry out the fine-grained access policy to encrypted data so that a lot of researches have been carried out about it including [3-8], which focused on different properties including expressiveness, performance or security. Especially, Lewko et al. provided a CP-ABE scheme [8] by using the monotone span program (MSP) respectively, which were both proved adaptively secure.

As is described above, the key is related to attributes owned by multiple users so that distinct users may have the same attribute sets, namely the same decryption keys. If a key is leaked, it is hard to identify who exactly leaks it. Liu et al. [9] provided a traceable CP-ABE scheme which can identify the malicious users leaking their decryption keys. This scheme was constructed based on Lewko et al's CP-ABE [8] construction over composite order bilinear groups by drawing on Boneh and Boyen's signature technique [10]. Later, Ning et al. provided a traceable CP-ABE construction[11] in large universe by using the similar method based on Rouselakis et al's construction [12] that was proved selectively secure over prime order bilinear groups. Zhang et al. [13] took it a step further by adding authority accountability. In addition, some relevant papers are proposed also[14-16]. However, in all these traceable CP-ABE constructions, the ciphertext grows linearly with the MSP, similarly, the number of pairing computations in decryption algorithm also grows linearly with that of qualified rows in the MSP.

Note that, constant computation and low communication generally has more practical significance for some applications with limited computing resources and bandwidth. Therefore, we put forward a traceable CP-ABE based on [9] for MAS. In this scheme, we make use of an "encoding technique" which represents the MAS by their minimal sets to encrypt the messages, so the ciphertext size is polynomial with the number of minimal sets. For some access policies, this scheme may have shorter ciphertext and lower encryption cost (see Section 3.5). Additionally, the most important thing is that our construction has only three bilinear pairing operations and two exponent operations in the decryption process, which improves the efficiency extremely.

2. Preliminaries

In this part, we will take view of several facts that will be applied in our scheme including access structures, minimal set, CP-ABE, composite order bilinear maps, $l-SDH$ assumption and traceability game.

2.1 Access Structures

Definition 1.(Access Structures). *Suppose* $\mathcal{M}=\{m_1,m_2,...,m_n\}$ *is a set of members and* $\mathbb{S}\subset 2^{\mathcal{M}}$ *is a collection. If for each set* A *and* B, *it satisfies if* $A\in\mathbb{S}$ *and* $A\subseteq B$, *then* $B\in\mathbb{S}$ *holds, then we can say* $\mathbb{S}$ *is monotone. An access structure is defined as collection* $\mathbb{S}\subseteq 2^{\mathcal{M}}\setminus\{0\}$.

2.2 Minimal Set

Definition 2.(Minimal Set of a MAS). *Suppose* $\mathbb{S}$ *is a MAS with the attribute set* $U=\{u_1,u_2,...,u_n\}$. *For all sets* A *and* B *in* $\mathbb{S}$, *if* $\forall B\in\mathbb{S}\setminus\{A\}$, *we have* $B\not\subset A$, *then* A *is a minimal authorized set. The collection of minimal set in* $\mathbb{S}$ *composes the base of* $\mathbb{S}$.

Theorem 1. [17] *If there exists a linear secret sharing scheme (LSSS) for a concrete MAS, then there must be a smallest MSP for the same MAS.*

2.3 CP-ABE

Assuming the set of attributes is $U=\{u_1,u_2,...,u_n\}$, then the tracable CP-ABE is defined as follows:

Setup $(1^{\lambda},U)$.The input is security parameter 1^{λ} and attribute universe U, then it will output the public parameters PP and the master key MK. In addition, it sets the initial tracing table as $T=\phi$, where ϕ denotes empty.

KeyGen(MK,id,S). The input is the MK, identity id and attributes set S, then it will output the secret key $SK_{id,S}$. Finally, it will add id to the tracing table T.

Encrypt$(PP,\mathcal{D},M)$. The input is the PP, access structure $\mathcal{D}$ for attributes universe U and plaintext M needing to be encrypted, then it will output the ciphertext $CT_{\mathcal{D}}$ with $\mathcal{D}$ implicitly contained.

Decrypt$(CT_{\mathcal{D}},SK_{id,S})$. The input is the $CT_{\mathcal{D}}$ and $SK_{id,S}$. If S satisfies $\mathcal{D}$, then it will output the valid message M.

Trace$(T,SK_{id,S})$. The input is the T and $SK_{id,S}$. Then it will output an identity id or a special symbol ϕ.

The above algorithms can be generalized as **Fig. 1**.

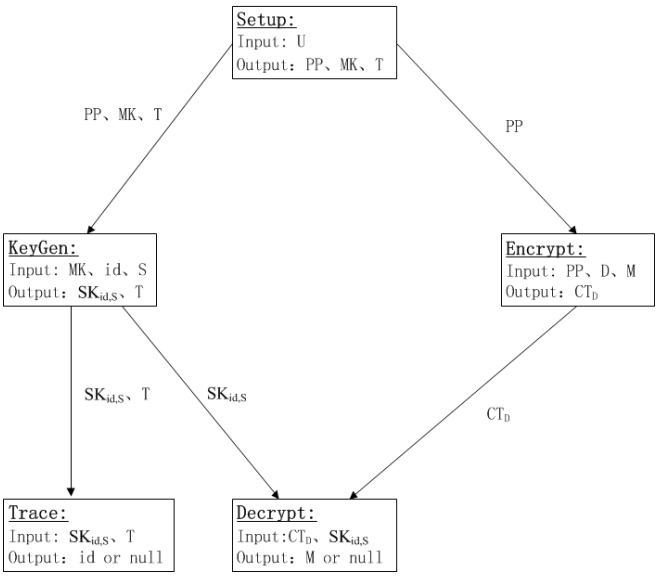

Fig. 1. Steps involved in various phases of our proposed scheme

Next, we will provide concrete definition of adaptively secure against chosen plaintext attacks for the CP-ABE construction, which is represented as a security game $Game_{Real}$ between challenger $\mathcal{C}$ and attacker $\mathcal{A}$.

Setup: $\mathcal{C}$ carries out **Setup** algorithm to generate the public parameters PP and master key MK. $\mathcal{C}$ begins to interact with $\mathcal{A}$ by giving PP.

Phase 1: $\mathcal{A}$ makes a number of key queries for the identity-attribute tuples $\{(id_1, S_1), (id_2, S_2), ..., (id_{q_1}, S_{q_1})\}$.

Challenge: $\mathcal{A}$ submits two plaintexts M_0, M_1 with the equal length and access policy $\mathcal{D}^*$ that cannot be satisfied by $\{S_1, S_2, ..., S_{q_1}\}$. Then $\mathcal{C}$ chooses a random $\beta \in \{0,1\}$ and encrypts the plaintext M_β using $\mathcal{D}^*$. At last, it outputs generated ciphertext $CT_{\mathcal{D}^*}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ proceeds to make a number of key queries for the identity-attribute tuples $\{(id_{q_1+1}, S_{q_1+1}), (id_{q_1+2}, S_{q_1+2}), ..., (id_q, S_q)\}$ requiring that no S_i satisfies $\mathcal{D}^*$.

Guess: $\mathcal{A}$ outputs a guess β' for β.

$\mathcal{A}$' advantage is described to be $Adv^{\mathcal{A}}_{Game_{Real}}(\lambda) = |\Pr[\beta = \beta'] - 1/2|$.

Definition 3. *We can know a CP-ABE is adaptively secure assuming for polynomial time attackers $\mathcal{A}$, the advantage $Adv^{\mathcal{A}}_{Game_{Real}}(\lambda)$ is negligible.*

2.4 Composite Order Bilinear Maps

Next, we will describe the composite order bilinear group firstly proposed in [18]. Suppose $\mathcal{G}$ is a group generator, and it outputs the parameters $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e)$, in which p_1, p_2

and p_3 denote three different big primes, $\mathbb{G}$ and $\mathbb{G}_T$ denote the groups with order $N = p_1 p_2 p_3$. Additionally, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ represents a bilinear map which satisfies that

1.For any $x, y \in \mathbb{Z}_N^*$ and $u, v \in \mathbb{G}$, we have $e(u^x, v^y) = e(u, v)^{xy}$.

2.There exists $e(g, h)$ having order N in $\mathbb{G}_T$, where $g, h \in \mathbb{G}$.

Suppose $\mathbb{G}_{p_1}$, $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ are the subgroups of $\mathbb{G}$ with order p_1, p_2 and p_3 respectively. Let $h_i \in \mathbb{G}_{p_i}$ and $h_j \in \mathbb{G}_{p_j}$ be random parameters with $i \neq j$, then we have $e(h_i, h_j) = 1$ according to orthogonal property [19].

Now, we will state three complexity assumptions proposed by Lewko et al. [19], on which the security proof of our scheme is based, as follows:

Assumption 1. Provided the group generator $\mathcal{G}$, the problem of Assumption 1 is defined to be:

$$(N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda),$$
$$g \xleftarrow{R} \mathbb{G}_{p_1}, X_3 \xleftarrow{R} \mathbb{G}_{p_3},$$
$$L = ((N, \mathbb{G}, \mathbb{G}_T, e), g, X_3),$$
$$T \xleftarrow{R} \mathbb{G}_{p_1 p_2}, T' \xleftarrow{R} \mathbb{G}_{p_1}$$

$\mathcal{A}$'s advantage to break the above assumption is:

$$Adv1_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(L, T) = 1] - \Pr[\mathcal{A}(L, T') = 1]|$$

Assumption 2. Provided the group generator $\mathcal{G}$, the problem of Assumption 2 is defined to be:

$$(N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda),$$
$$g, X_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3, Y_3 \xleftarrow{R} \mathbb{G}_{p_3},$$
$$L = ((N, \mathbb{G}, \mathbb{G}_T, e), g, X_1 X_2, X_3, Y_2 Y_3),$$
$$T \xleftarrow{R} \mathbb{G}, T' \xleftarrow{R} \mathbb{G}_{p_1 p_3}$$

$\mathcal{A}$'s advantage to break the above assumption is:

$$Adv2_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(L, T) = 1] - \Pr[\mathcal{A}(L, T') = 1]|$$

Assumption 3. Provided the group generator $\mathcal{G}$, the problem of Assumption 3 is defined to be:

$$(N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \alpha, s \xleftarrow{R} \mathbb{Z}_N$$
$$g \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2, Z_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3 \xleftarrow{R} \mathbb{G}_{p_3},$$
$$L = ((N, \mathbb{G}, \mathbb{G}_T, e), g, g^{\alpha} X_2, X_3, g^{s} Y_2, Z_2),$$
$$T \xleftarrow{R} e(g, g)^{\alpha s}, T' \xleftarrow{R} \mathbb{G}_T$$

$\mathcal{A}$'s advantage to break the above assumption is:

$$Adv3_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(L, T) = 1] - \Pr[\mathcal{A}(L, T') = 1]|$$

2.5 $l-$SDH Assumption

Next, we will describe the $l-$SDH assumption [10] that is used to prove our traceability.

$l-$SDH Assumption. Assuming $\mathbb{G}$ is a bilinear group with prime order p and generator $g \in \mathbb{G}$, the $l-$SDH is depicted as: provided a $(l+1)$-vector $(g, g^a, g^{a^2}, ..., g^{a^l})$ to output the tuple $(c, g^{1/(a+c)}) \in \mathbb{Z}_p \times \mathbb{G}$. The attacker , $\mathcal{A}$, possesses an advantage at least ε if we have

$$\Pr[\mathcal{A}(g, g^a, g^{a^2}, ..., g^{a^l}) = (c, g^{1/(a+c)})] \geq \varepsilon$$

Definition 4. *Assume there exists no algorithm possesses the advantage at least ε to solve the $l-$SDH problem in time t, then we have $(l, t, \varepsilon)-$SDH assumption stands.*

2.6 Traceabilit

Now, we will propose the concrete definition of traceability, depicted as a security game between attacker $\mathcal{A}$ and challenger $\mathcal{C}$, for our traceable CP-ABE:

Setup. The Setup algorithm will be run by $\mathcal{C}$ to generate the public parameters PP, which are then sent to $\mathcal{A}$.

Key Query. $\mathcal{A}$ makes a number of q key queries associated with attribute sets $\{(id_1, S_1), (id_2, S_2), ..., (id_q, S_q)\}$.

Key Forgery. A decryption key SK^* will be given by $\mathcal{A}$, and it wins if $\mathbf{Trace}(T, SK^*) \neq \phi$ with $\mathbf{Trace}(T, SK^*) \notin \{id_1, id_2, ..., id_q\}$.

3. Traceable CP-ABE for the MAS

3.1 Our Construction

We propose our traceable CP-ABE construction by applying a simple encoding method which is adaptively secure over the composite order bilinear groups. Note that, its order of bilinear groups will be $N = p_1 p_2 p_3$. Additionally, we will employ the random members of subgroup $\mathbb{G}_{p_1}$ to encode the policy and attributes, and emply the random members of subgroup $\mathbb{G}_{p_3}$ to randomize the key and ciphertext.

Setup$(1^\lambda, U)$. It firstly runs $\mathcal{G}(1^\lambda)$, which denotes the group parameters generator, to obtain $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_\mathbb{T}, e)$ that will be used, in which $\mathbb{G}$ and $\mathbb{G}_T$ are the groups with order $N = p_1 p_2 p_3$. $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ denotes the bilinear map. $\mathbb{G}_{p_i}$ denotes the subgroup with order p_i, and $g \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$ denotes the generators of subgroup $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_3}$. Next, it chooses parameters $\alpha, a \in \mathbb{Z}_N$, $h \in \mathbb{G}_{p_1}$ randomly, and for every $i \in U$, it picks random $u_i \in \mathbb{Z}_N$. Finally, it sets the public parameters PP to be:

$$PP = (N, h, g, g^a, e(g,g)^\alpha, \{U_i = g^{u_i}\}_{i \in U})$$

In addition, it sets the master key MK to be:

$$MK = (\alpha, a, X_3).$$

Note that, the initial tracing table T is set to be ϕ denoting empty.

KeyGen(MK, id, S) : It picks a parameter $trc \in \mathbb{Z}_N^*$ randomly for tracing, then chooses parameters $t \in \mathbb{Z}_N$, $R, R_0, R_0' \in \mathbb{G}_{p_3}$ randomly , and for every $i \in S$, it picks $R_i \in \mathbb{G}_{p_3}$. Finally, it sets the user's secret key to be:

$$SK_{id,S} = (K = g^{\frac{\alpha}{a+trc}} h^t R,\ K' = trc, L = g^t R_0, L' = g^{at} R_0', \{K_i = U_i^{(a+trc)t} R_i\}_{i \in S})$$

Here, if $\gcd(a+trc, N) \neq 1$ or trc is used already , this algorithm will choose another $trc \in \mathbb{Z}_N^*$. Finally, the algorithm adds the pair (trc, id) into T for traceability.

Encrypt($PP, \mathcal{D}, M$) : Here, $\mathcal{D}$ denotes the set of minimal sets generated by the MAS. Let $\mathcal{D} = \{S_1, S_2, ..., S_m\}$, where $S_i \subset U, \forall\, i \in [m]$. Then it picks $s \in \mathbb{Z}_N$ and further picks $s_i \in \mathbb{Z}_N$ randomly for each $i \in [m]$. The ciphertext is set to be:

$$CT_{\mathcal{D}} = (\mathcal{D}, C = M \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{as},$$
$$\{C_{i,1} = h^s (\prod_{j \in S_i} U_j)^{s_i}, C_{i,2} = g^{s_i}\}_{i=1}^m)$$

Decrypt($CT_{\mathcal{D}}, SK_{id,S}$) : Let the ciphertext be $CT_{\mathcal{D}} = (\mathcal{D}, C, C_0, C_0', \{C_{i,1}, C_{i,2}\}_{i=1}^m)$ and the private key be $SK_{id,S} = (K,\ K', L, L', \{K_i\}_{i \in S})$. Assuming the attributes set S satisfies the MAS that is generated by $\mathcal{D}$, we have that there must be a minimal set in $\mathcal{D}$, which is the subset of S . Let $S_j \subset S$, then we compute

$$D = e(C_{j,1}, L^{K'} L')$$
$$E = e(C_0^{K'} C_0', K) \cdot e(C_{j,2}, \prod_{i \in S_j} K_i)$$

Finally, it computes $C \cdot D / E = M$.

Trace($T, SK_{id,S}$) The tracing algorithm is defined the same as that in [9] which takes the tracing table T and the secret key $SK_{id,S}$ as input. Next, it will search K' in the tracing table T , and once K' is found, it will output the corresponding id , otherwise output ϕ .

Correctness. In this part, we give the correctness validation as follows:

$$\begin{aligned}
&C \cdot D / E \\
&= M \cdot e(g,g)^{\alpha s} \cdot \frac{e(h^s (\prod_{i \in S_j} U_i)^{s_j}, (g^t R_0)^{trc} g^{at} R_0')}{e(g^{s \cdot trc} g^{as}, g^{\frac{\alpha}{a+trc}} h^t R) \cdot e(g^{s_j}, \prod_{i \in S_j} (U_i^{(a+trc)t} R_i))} \\
&= M \cdot e(g,g)^{\alpha s} \cdot \frac{e(g,h)^{(a+trc)ts} e(g^{(a+trc)t}, (\prod_{i \in S_j} U_i)^{s_j})}{e(g,g)^{\alpha s} \cdot e(g,h)^{(a+trc)ts} \cdot e(g^{s_j}, \prod_{i \in S_j} (U_i^{(a+trc)t}))} \\
&= M \cdot e(g,g)^{\alpha s} \cdot \frac{1}{e(g,g)^{\alpha s}} \\
&= M
\end{aligned}$$

3.2 Security Proof

Assuming there exists an attacker, $\mathcal{A}$, who carries out q key queries, then we will use $2q+3$ games between $\mathcal{A}$ and a challenger $\mathcal{C}$ to prove the security of our traceable CP-ABE construction. The orthogonal element of group $\mathbb{G}_{p_2}$ is used to construct the semi-functional key and ciphertext. Next, we construct the semi-functional key and ciphertext:

Semi-functional Key: The semi-functional key is divided into two styles: type 1 and type 2. We construct type 1 as follows. Suppose id is the user's identity with attributes set S. Then we pick parameters $d, b, b, z_i \in \mathbb{Z}_N$, $g_2 \in \mathbb{G}_{p_2}$, $R, R_0 \in \mathbb{G}_{p_3}$, and for each $i \in S$ pick $R_i \in \mathbb{G}_{p_3}$ randomly. At last, we set the key of type 1 as

$$SK_{id,S} = (K = g^{\frac{\alpha}{a+trc}} h^t R g_2^d,\ K' = trc, L = g^t R_0 g_2^b,$$
$$L' = g^{at} R_0' g_2^{b'}, \{K_i = U_i^{(a+trc)t} R_i g_2^{z_i}\}_{i \in S}$$

In addition, we set the key of type 2 as:

$$SK_{id,S} = (K = g^{\frac{\alpha}{a+trc}} h^t R g_2^d,\ K' = trc, L = g^t R_0,$$
$$L' = g^{at} R_0', \{K_i = U_i^{(a+trc)t} R_i\}_{i \in S}$$

Semi-functional Ciphertext: Suppose $\mathcal{D}$ is the basis for a monotone access structure Π. Let $\mathcal{D} = \{S_1, ..., S_m\}$, where for each $i \in [m]$ we have $S_i \subset U$. Next, we choose random parameters $c, c', c'' \in \mathbb{Z}_N$ with the restriction that $b \cdot c'' = d \cdot c$, then chooses $g_2 \in \mathbb{G}_{p_2}$, and for each $i \in [m]$ choose $s_i \in \mathbb{Z}_N$. Finally, wet set the semi-functional ciphertext as

$$CT_{\mathcal{D}} = (\mathcal{D}, C = M \cdot e(g,g)^{\alpha s}, C_0 = g^s g_2^c, C_0' = g^{as} g_2^{c'},$$
$$\{C_{i,1} = h^s (\prod_{j \in S_i} U_j)^{s_i} g_2^{c''}, C_{i,2} = g^{s_i}\}_{i=1}^m)$$

As we can see that if a legitimate semi-functional ciphertext is decrypted by the corresponding semi-functional key, it will generate an additional term $e(g_2, g_2)^{b'c''-c'd}$. Moreover, if we have $b'c'' - c'd = 0$, then the decryption will succeed, and in this situation we call it nominally semi-functional.

In the $2q+3$ consecutive games, $\mathbf{Game}_{Real}$ as the begining one is the actual security game defined in part 2.3 and the next one is $\mathbf{Game}_0$, in which each key is normal ,however, the ciphertext is semi-functional. For $k=1$ to q, we define:

$\mathbf{Game}_{k,1}$. The first $k-1$ keys are type 2, the key k is type 1, the others are normal. Additionally, the challenge ciphertext will be semi-functional.

$\mathbf{Game}_{k,2}$. The first k keys are type 2 with the rest ones normal. Additionally, the challenge ciphertext is also semi-functional.

For $\mathbf{Game}_{q,2}$, the keys will be type 2. Furthermore, for $\mathbf{Game}_{Final}$, which is the last one, the keys will be type 2, however, the challenge ciphertext becomes to a semi-functional encryption on some random message. Additionally, one point needs to be noticed is that $\mathcal{A}$'s

advantage in the game $\mathbf{Game}_{Final}$ will be 0 . Next, the indistinguishability of the games by using the following lemmas will be proved.

Lemma 1. *Assuming there is a polynomial time adversary,$\mathcal{A}$, satisfying* $Adv^{\mathcal{A}}_{\mathbf{Game}_{Real}} - Adv^{\mathcal{A}}_{\mathbf{Game}_0} = \varepsilon$ *. Then a polynomial time simulator, $\mathcal{B}$, will be constructed with advantage ε to break Assumption 1.*

Proof. We establish the simulator $\mathcal{B}$ to take the parameters, (g, X_3, T) of Assumption 1, then $\mathcal{B}$ simulates for either $\mathbf{Game}_{Real}$ or $\mathbf{Game}_0$ depending on T .

Setup: $\mathcal{B}$ first picks $\alpha, \beta, a \in \mathbb{Z}_N$ randomly. Next, for every $i \in U$, it picks $u_i \in \mathbb{Z}_N$ randomly and sets $U_i = g^{u_i}$, then it starts to interact with $\mathcal{A}$ by giving the public parameters as follows:

$$PP = (N, g, g^a, h = g^\beta, Y = e(g,g)^\alpha, \{U_i\}_{i \in U})$$

while the master key $MK = (\alpha, a, X_3)$ will be kept private to $\mathcal{B}$.

Key Query: $\mathcal{B}$ responds to $\mathcal{A}$'s queries by performing **KeyGen** algorithm, because he keeps MK .

Challenge: $\mathcal{B}$ will receive the challenge messages M_0, M_1 and challenge basis $\mathcal{D}^*$ from $\mathcal{A}$. Then $\mathcal{B}$ chooses $M_b \in \{M_0, M_1\}$ randomly. Let $\mathcal{D}^* = \{S_0, S_1, ..., S_m\}$, where each $S_i \subset U$. Next, for each $i \in [m]$, $\mathcal{B}$ chooses an exponent $s_i \in \mathbb{Z}_N$ random and further constructs the challenge ciphertext $CT_{\mathcal{D}^*}$ as follows:

$$CT_{\mathcal{D}^*} = (C = M_b \cdot e(g^\alpha, T), C_0 = T, C_0' = T^a, \{C_{i,1} = T^\beta (\prod_{j \in S_i} U_j)^{s_i}, C_{i,2} = g^{s_i}\}_{i=1}^m)$$

Finally, $CT_{\mathcal{D}^*}$ is sent to $\mathcal{A}$.

Assuming that $T \in \mathbb{G}_{p_1 p_2}$, then T can be written as $T = g^s g_2^c$ for some $s, c \in \mathbb{Z}_N$, so we have $C = M_b \cdot Y^s, C_0 = g^s g_2^c$, $C_0' = g^{as} g_2^{ac}, C_{i,1} = h^s (\prod_{j \in S_i} U_j)^{s_i} g_2^{\beta c}$ and $\{C_{i,2} = g^{s_i}\}_{i=1}^m$. Note that, we set $c' = ac$, $c'' = \beta c$ implicitly. Since the values of a, β *model* p_2 are independent of their values *model* p_1 according to Chinese Remainder Theorem, therefore, we have $CT_{\mathcal{D}^*}$ will be a semi-functional ciphertext distributed correctly. We can see that if $T \in \mathbb{G}_{p_1 p_2}$, $\mathcal{B}$ simulates $\mathbf{Game}_0$, else if $T \in \mathbb{G}_{p_1}$, $\mathcal{B}$ simulates $\mathbf{Game}_{Real}$. This completes our proof for **Lemma 1**.

Lemma 2. *Assuming there is a polynomial time adversary, $\mathcal{A}$,satisfying* $Adv^{\mathcal{A}}_{\mathbf{Game}_{k-1,2}} - Adv^{\mathcal{A}}_{\mathbf{Game}_{k,1}} = \varepsilon$ *. Then a polynomial time simulator, $\mathcal{B}$, will be constructed with advantage ε to break Assumption 2.*

Proof. We establish the simulator $\mathcal{B}$ to take the parameters, $(g, X_1 X_2, X_3, Y_2 Y_3, T)$, of Assumption 2, and $\mathcal{B}$ simulates either $\mathbf{Game}_{k-1,2}$ or $\mathbf{Game}_{k,1}$ depending on T .

Setup: $\mathcal{B}$ chooses parameters $\alpha, \beta, a \in \mathbb{Z}_N$ randomly. Next, for each $i \in U$, $\mathcal{B}$ chooses $u_i \in \mathbb{Z}_N$ and sets $U_i = g^{u_i}$, then it starts to interact with $\mathcal{A}$ by giving the public parameters as follows:

$$PP = (N, g, g^a, h = g^\beta, Y = e(g,g)^\alpha, \{U_i\}_{i \in U})$$

while the master key $MK = (\alpha, a, X_3)$ will be kept private to $\mathcal{B}$.

Key Query: For constructing the semi-functional keys of type 2, $\mathcal{B}$ picks $t \in \mathbb{Z}_N$, $trc \in \mathbb{Z}_N^*$, elements R_0, R_0', R_i of $\mathbb{G}_{p_3}$ randomly, and then constructs the key:

$$K = g^{\frac{\alpha}{a+trc}} h^t (Y_2 Y_3)^t,\ K' = trc, L = g^t R_0, L' = g^{at} R_0', \{K_i = U_i^{(a+trc)t} R_i\}_{i \in S}$$

Note that, the key is semi-functional of type 2 distributed correctly. Additionally, to construct the rest $q-k$ keys that are normal, $\mathcal{B}$ can merely perform **KeyGen** algorithm because he keeps MK.

For key k, $\mathcal{B}$ implicitly makes the $\mathbb{G}_{p_1}$ part of T to be g^t, then chooses random R, R_0, R_0', R_i of $\mathbb{G}_{p_3}$, $trc \in \mathbb{Z}_N^*$, and further sets the key to be:

$$K = g^{\frac{\alpha}{a+trc}} T^\beta R,\ K' = trc, L = TR_0, L' = T^a R_0', \{K_i = T^{(a+trc)u_i} R_i\}_{i \in S}$$

Assuming $T \in \mathbb{G}_{p_1 p_3}$, it will be a normal key, and else assuming $T \in \mathbb{G}$, it becomes a semi-functional key of type 1. Additionally, if let g_2^b be the $\mathbb{G}_{p_2}$ part of T, then we have $d = \beta b \ \ model\ \ p_2$, $b' = ab \ \ model\ \ p_2$ and $z_i = (a + trc) b u_i$.

Challenge: $\mathcal{B}$ will two challenge messages M_0, M_1 and a challenge basis $\mathcal{D}^*$ from $\mathcal{A}$. Then $\mathcal{B}$ chooses random $M_b \in \{M_0, M_1\}$. Let $\mathcal{D}^* = \{S_0, S_1, ..., S_m\}$, where each $S_i \subset U$. Next, for each $i \in [m]$, $\mathcal{B}$ chooses random $s_i \in \mathbb{Z}_N$ and constructs the challenge ciphertext $CT_{\mathcal{D}^*}$ as follows:

$$CT_{\mathcal{D}^*} = (C = M_b \cdot e(g^\alpha, X_1 X_2), C_0 = X_1 X_2, C_0' = (X_1 X_2)^a,$$
$$\{C_{i,1} = (X_1 X_2)^\beta (\prod\nolimits_{j \in S_i} U_j)^{s_i}, C_{i,2} = g^{s_i}\}_{i=1}^m)$$

Now suppose that $T \in \mathbb{G}$, and let the $\mathbb{G}_{p_1 p_2}$ part of T be $g^s g_2^c$. Therefore, we implicitly set $c' = ac$, $c'' = \beta c$. Additionally, we know that the kth semi-functional key and ciphertext are distributed correctly except for the fact that the exponent $c' = ac \ \ model\ \ p_2$ in L' part of the ciphertext is correlated with $a \ \ model\ \ p_2$ in the K_0 part of the key, and that the exponent $c'' = \beta c \ model\ \ p_2$ in $C_{i,1}$ part of the ciphertext is correlated with $\beta \ \ model\ \ p_2$ in the K part of the key. Therefore, if the correct semi-functional ciphertext is decrypted by the corresponding semi-functional key of type 1, a valid message M will be obtained.

Thus, if $T \in \mathbb{G}$, we have that $\mathcal{B}$ simulates $\mathbf{Game}_{k,1}$, and else if $T \in \mathbb{G}_{p_1 p_3}$, $\mathcal{B}$ simulates $\mathbf{Game}_{k-1,2}$. This gives complete proof for **Lemma 2**.

Lemma 3. *Assuming there is a polynomial time adversary, $\mathcal{A}$,satisfying $Adv^{\mathcal{A}}_{\mathbf{Game}_{k,1}} - Adv^{\mathcal{A}}_{\mathbf{Game}_{k,2}} = \varepsilon$. Then a polynomial time simulator, $\mathcal{B}$, will be constructed with advantage ε to break Assumption 2.*

Proof. We establish the simulator $\mathcal{B}$ to take the parameters, $(g, X_1X_2, X_3, Y_2Y_3, T)$ of Assumption 2, and $\mathcal{B}$ simulates either $\mathbf{Game}_{k,1}$ or $\mathbf{Game}_{k,2}$ depending on T.

Setup: $\mathcal{B}$ picks $\alpha, \beta, a \in \mathbb{Z}_N$ randomly. Next, for every $i \in U$, it picks $u_i \in \mathbb{Z}_N$ and sets $U_i = g^{u_i}$, then it begins to interact with $\mathcal{A}$ by giving the public parameters as follows:

$$PP = (N, g, g^a, h = g^\beta, Y = e(g,g)^\alpha, \{U_i\}_{i \in U}),$$

while the master key $MK = (\alpha, a, X_3)$ will be kept private to $\mathcal{B}$.

Key Query: We construct the first $k-1$ key using the similar method as that in **Lemma 2**. For the kth key query, $\mathcal{B}$ also processes it using the similar method, however, additionally adding a term $(Y_2Y_3)^h$ to K part as follows by choosing a random $h \in \mathbb{Z}_N$:

$$K = g^{\frac{\alpha}{a+trc}} T^\beta R (Y_2Y_3)^h,\ K' = trc,\ L = TR_0,\ L' = T^a R_0', \{K_i = T^{(a+trc)u_i} R_i\}_{i \in S}$$

It must be noted that the adding term randomizes the $\mathbb{G}_{p_2}$ part of K, therefore, it is not nominally semi-functional any more.

If $T \in \mathbb{G}$, we have that it is a semi-functional key of type 1 distributed correctly, so $\mathcal{B}$ simulates $\mathbf{Game}_{k,1}$, and similarly if $T \in \mathbb{G}_{p_1p_3}$, it becomes semi-functional key of type 2, so it simulates $\mathbf{Game}_{k,2}$. This completes our proof for **Lemma 3**.

Lemma 4. *Assuming there is a polynomial time adversary, $\mathcal{A}$,satisfying $Adv^{\mathcal{A}}_{\mathbf{Game}_{q,2}} - Adv^{\mathcal{A}}_{\mathbf{Game}_{Final}} = \varepsilon$. Then a polynomial time simulator, $\mathcal{B}$, will be constructed with advantage ε to break Assumption 3.*

Proof. We establish the simulator $\mathcal{B}$ to take the parameters, $(g, X_3, g^\alpha X_2, g^s Y_2, Z_2, T)$ of Assumption 3, and $\mathcal{B}$ simulates either $\mathbf{Game}_{q,2}$ or $\mathbf{Game}_{Final}$ depending on T.

Setup: $\mathcal{B}$ picks $\alpha, \beta, a \in \mathbb{Z}_N$ randomly. Next, for every $i \in U$, it picks $u_i \in \mathbb{Z}_N$ randomly and sets $U_i = g^{u_i}$, then it begins to interact with $\mathcal{A}$ by giving the public parameters as follows:

$$PP = (N, g, g^a, h = g^\beta, Y = e(g, g^\alpha X_2)^\alpha = e(g,g)^\alpha, \{U_i\}_{i \in U}),$$

while the master key $MK = (\alpha, a, X_3)$ will be kept private to $\mathcal{B}$.

Key Query: To generate the semi-functional key of type 2, $\mathcal{B}$ first chooses random $t \in \mathbb{Z}_N$, $trc \in \mathbb{Z}_N^*$, random elements R_0, R_0', R_i of $\mathbb{G}_{p_3}$, and sets the key as:

$$K = (g^{\alpha} X_2)^{\frac{1}{a+trc}} (g^{\beta})^t Z_2^t R = g^{\frac{\alpha}{a+trc}} h^t X_2^{\frac{1}{a+trc}} Z_2^t R,$$
$$K' = trc, L = g^t R_0, L' = g^{at} R_0', \{K_i = U_i^{(a+trc)t} R_i\}_{i \in S}$$

We can see that the above key is distributed correctly.

Challenge: $\mathcal{B}$ will receive two challenge messages M_0, M_1 and a challenge basis $\mathcal{D}^*$ from $\mathcal{B}$. Then $\mathcal{B}$ chooses a random $M_b \in \{M_0, M_1\}$. Let $\mathcal{D}^* = \{S_0, S_1, ..., S_m\}$, where each $S_i \subset U$. Next, for each $i \in [m]$, $\mathcal{B}$ chooses $s_i \in \mathbb{Z}_N$ randomly and constructs the challenge ciphertext $CT_{\mathcal{D}^*}$ as follows:

$$CT_{\mathcal{D}^*} = (C = M_b \cdot T, C_0 = g^s Y_2, C_0' = (g^s Y_2)^a,$$
$$\{C_{i,1} = (g^s Y_2)^{\beta} (\prod_{j \in S_i} U_j)^{s_i}, C_{i,2} = g^{s_i}\}_{i=1}^m)$$

Assuming $T = e(g,g)^{\alpha s}$, then $CT_{\mathcal{D}^*}$ is precisely semi-functional ciphertext and $\mathcal{B}$ simulates $\mathbf{Game}_{q,2}$, else assuming T is random in $\mathbb{G}_T$, $\mathcal{B}$ simulates $\mathbf{Game}_{Final}$. This completes the proof of **Lemma 4**.

Theorem 2. *Suppose Assumptions 1, 2 and 3 hold, then we have that our traceable CP-ABE construction proposed above for the* MAS *is adaptively secure in the standard model.*

Proof. Suppose Assumptions 1, 2 and 3 hold, then we can say $\mathbf{Game}_{Real}$ is indistinguishable from $\mathbf{Game}_{Final}$ as shown by the above four lemmas. Moreover, the challenge message M_b is hidden by a random element of $\mathbb{G}_T$ in the game. Thus, $\mathcal{A}$ has no non-negligible advantage to break our traceable CP-ABE construction proposed above for MAS.

3.3 Traceability

This part will provide the specific traceability proof for our proposed scheme above based on two assumptions, namely the $l-$SDH assumption and Assumption 3. Note that, this proof is similar with that in [9].

Theorem 3: *Suppose the* $l-$SDH *assumption and Assumption 3 hold, then we have that our traceable CP-ABE has the traceability provided* $q \le l$.

Assume $l = q+1$ without loss of generality, then a polynomial time simulator, $\mathcal{B}$,will be constructed to break either $l-$SDH assumption or Assumption 3 as follows:

- $\mathcal{B}$ takes a case of Assumption 3 as $P_{A3} = (N, \mathbb{G}, \mathbb{G}_T, e, \tilde{g}, X_1 X_2, X_3, Y_2 Y_3, T)$, and if $b = 1$, $T \in \mathbb{G}_{p_1 p_3}$, otherwise, $T \in \mathbb{G}$.
- $\mathcal{B}$ takes a case of $l-$SDH assumption as $P_{SDH} = (N, \mathbb{G}, \mathbb{G}_T, e, \tilde{g}, \tilde{g}^a, ..., \tilde{g}^{a^l})$.

Here, $\mathcal{B}$'s goal is to break at least of one of the two assumptions. So $\mathcal{B}$ firstly chooses a random bit $\Gamma \in \{0,1\}$, and if $\Gamma = 0$, it takes P_{A3} as input, then picks $\tilde{a} \in \mathbb{Z}_N^*$ and sets

$A_i = \tilde{g}^{\tilde{a}^i} = \tilde{g}^{a^i}$. Otherwise, it takes P_{SDH} as input, then sets $A_i = \tilde{g}^{a^i}$ and chooses a generator $X_3 \in \mathbb{G}_{p_3}$.

$\mathcal{B}$ takes $(N, \mathbb{G}, \mathbb{G}_T, e, X_3, \{A_i\}_{i=0}^{l})$ as input, then it begins the interaction with $\mathcal{A}$ as follows:

- **Setup**. $\mathcal{B}$ chooses q different values $c_1, c_2, ..., c_q \in \mathbb{Z}_N^*$ uniform at random. Next, $\mathcal{B}$ defines a polynomial function $f(y) = \prod_{i=1}^{q}(y + c_i)$ and expand it as $f(y) = \sum_{i=0}^{q} \alpha_i y^i$ where $\alpha_0, \alpha_1, ..., \alpha_q \in \mathbb{Z}_N$. Then $\mathcal{B}$ computes g and g^a as follows:

$$g = \prod_{i=0}^{q}(A_i)^{\alpha_i} = \tilde{g}^{f(a)}, g^a = \prod_{i=1}^{q+1}(A_i)^{\alpha_{i-1}} = \tilde{g}^{f(a)\cdot a}$$

Next, $\mathcal{B}$ picks parameters $\alpha, \beta \in \mathbb{Z}_N$ randomly, and for every attribute $i \in U$, it picks $u_i \in \mathbb{Z}_N$ randomly. Finally, the public parameters are set as:
$PP = (N, h = g^{\beta}, g, g^a, e(g,g)^{\alpha}, \{U_i = g^{u_i}\}_{i \in U})$.

- **Key Query**. $\mathcal{B}$ makes key queries for (id_i, S_i) to $\mathcal{B}$ with $i \le q$. Let $f_i(y) = f(y) \Big/ (y + c_i) = \prod_{j=1, j \neq i}^{q}(y + c_i)$, then $\mathcal{B}$ expands $f_i(y)$ to get $f_i(y) = \sum_{j=0}^{q-1} \beta_j y^j$ and computes

$$\sigma_i = \prod_{j=0}^{q-1}(A_j)^{\beta_j} = \tilde{g}^{f_i(a)} = g^{1/(a+c_i)}$$

Next, $\mathcal{B}$ picks random parameters $t \in \mathbb{Z}_N, R, R_0, R_0' \in \mathbb{G}_{p_3}$, for every attribute $x \in S_i$, it picks $R_x \in \mathbb{G}_{p_3}$ randomly. Finally, it sets the key SK_{id,S_i} as

$$K = (\sigma_i)^{\alpha} h^t R = g^{\alpha/(a+c_i)} h^t R,\ K' = c_i, L = g^t R_0,$$
$$L' = g^{at} R_0', \{K_x = (g^a \cdot g^{c_i})^{u_x t} R_x = U_x^{(a+c_i)t} R_i\}_{i \in S_i}$$

Finally, $\mathcal{B}$ adds the pair (id_i, c_i) into the tracing table T.

- **Key Forgery**. If $\mathcal{A}$ does not win the game, $\mathcal{B}$ will pick a bit $\beta' \in \{0,1\}$ and tuple $(c_r, w_r) \in \mathbb{Z}_{p_1} \times \mathbb{G}_{p_1}$ randomly, which are used as the guess for Assumption 3 and $l-$SDH problem. Otherwise, $\mathcal{B}$ will makes use of the long division method to write the function $f(y)$ to be $f(y) = \eta(y)(y + K') + \eta_{-1}$ with the polynomial $\eta(y) = \sum_{i=0}^{q-1} \eta_i y^i$ and $\eta_{-1} \in \mathbb{Z}_N^*$. Next, $\mathcal{B}$ will compute $\gcd(\gamma_{-1}, N)$.

(1) Assuming $\gcd(\gamma_{-1}, N) \neq 1$.

If $\mathcal{B}$ takes P_{SDH} as input, it will pick a bit $\beta' \in \{0,1\}$ and tuple $(c_r, w_r) \in \mathbb{Z}_{p_1} \times \mathbb{G}_{p_1}$ randomly, which are used as the guess for Assumption 3 and $l-$SDH problem.

If $\mathcal{B}$ takes P_{A3} as input, it will pick a random pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times \mathbb{G}_{p_1}$ as the guess for $l-$SDH problem, and continues to determine β' as follows:

$\mathcal{B}$ obtains $(n,n')\in\mathbb{Z}_N$ from the value of $\gcd(\gamma_{-1},N)$ satisfying $n\cdot n'=N$ and $(n,n')\in\{(p_1,p_2p_3),(p_2p_3,p_1),(p_2,p_1p_3),(p_1p_3,p_2),(p_3,p_1p_2),(p_1p_2,p_3)\}$.

- If $\tilde{g}^n=1$ and $(Y_2Y_3)^{n'}=1$, $\mathcal{B}$ obtains $n=p_1$, otherwise obtains $n'=p_1$. Then $\mathcal{B}$ computes $e(T^{p_1},X_1X_2)$, and if its value equals to 1, $\mathcal{B}$ sets $\beta'=1$, otherwise sets $\beta'=0$.
- Otherwise, if $X_3^n=1$ and $(X_1X_2)^{n'}=1$, $\mathcal{B}$ obtains $n=p_3$, otherwise obtains $n'=p_3$. Then $\mathcal{B}$ computes $e(T^{p_3},Y_2Y_3)$, and if its value equals to 1, $\mathcal{B}$ sets $\beta'=1$, otherwise sets $\beta'=0$.
- Otherwise, if $X_3^n=1$, $\mathcal{B}$ obtains $n'=p_2$, otherwise if $X_3^{n'}=1$, it obtains $n=p_2$. Then $\mathcal{B}$ computes $T^{p_1p_3}$, and if its value equals to 1, $\mathcal{B}$ sets $\beta'=1$, otherwise sets $\beta'=0$.

(2) Assuming $\gcd(\gamma_{-1},N)=1$.

If $\mathcal{B}$ takes P_{A3} as input, it picks a bit $\beta'\in\{0,1\}$ and tuple $(c_r,w_r)\in\mathbb{Z}_{p_1}\times\mathbb{G}_{p_1}$ randomly, which are used as the guess for Assumption 3 and $l-\mathrm{SDH}$ problem.

If $\mathcal{B}$ takes P_{SDH} as input, it picks a bit $\beta'\in\{0,1\}$ randomly as the guess for Assumption 3 problem, and continues to determine $(c_r,w_r)\in\mathbb{Z}_{p_1}\times\mathbb{G}_{p_1}$ as follows:

Let $L=g^tL_2L_3$ where $t\in\mathbb{Z}_N$, $L_2\in\mathbb{G}_{p_2}$, $L_3\in\mathbb{G}_{p_3}$. Additionally, we have $L'=g^{at}L_2'L_3'$ and $K=g^{\alpha/(a+K')}h^tK_2K_3$ where $K_2\in\mathbb{G}_{p_2}$, L_2', L_3', $K_3\in\mathbb{G}_{p_3}$. Next, $\mathcal{B}$ sets:

$$\sigma=((K/L^{\beta})^{p_2p_3})^{(p_2p_3\alpha)^{-1}}=\tilde{g}^{1/(a+K')}=\tilde{g}^{\eta(a)}\tilde{g}^{\eta_{-1}/(a+K')}$$

$$w_r=(\sigma\cdot\prod_{i=0}^{q-1}A_i^{-\eta_i})^{1/\eta_{-1}}=\tilde{g}^{1/(a+K')},c_r=K'\ model\ p_1$$

Note that (c_r,w_r) is the solution for the $l-\mathrm{SDH}$ problem.

4. Analysis

Next, the specific scheme analysis and experimental demonstration will be implemented Firstly, we describe all the symbols that will be used. S denotes a user's attributes set and $|S|$ denotes the size; l denotes the rows of a span program $(\mathbb{A},\rho)$; $|\mathcal{D}|$ denotes the number of minimal sets for a MAS; m denotes the number of matching attributes during decryption; T_p denotes the pairing computations; $E_{\mathbb{G}}$ and $E_{\mathbb{G}_T}$ respectively denote the exponent operations in $\mathbb{G}$ (or $\mathbb{G}_{p_1}$) and $\mathbb{G}_T$; $|\mathbb{G}_{p_1}|,|\mathbb{G}_{p_1p_3}|,|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the element size in the group $\mathbb{G}_{p_1},\mathbb{G}_{p_1}\times\mathbb{G}_{p_3},\mathbb{G}$ and $\mathbb{G}_T$ respectively. $|\mathbb{Z}_{p_1}|$ and $|\mathbb{Z}_N|$ denote the element size in $\mathbb{Z}_{p_1}$ and $\mathbb{Z}_N$ respectively.

4.1 Scheme analysis

Table 1. Troperty comparisons with other schemes

Scheme	Group	Order	Security	Assumption	Access Policy	Authority Accountability	Large Universe
Liu[9]	Composite	$p_1p_2p_3$	Adaptive (standard)	Assumption 1,2,3	LSSS	No	No
Ning[11]	Prime	p	Selective (standard)	q-type	LSSS	No	Yes
Zhang[13]	Composite	$p_1p_2p_3$	Adaptive (random)	Assumption 1,2,3	LSSS	Yes	Yes
Ours	Composite	$p_1p_2p_3$	Adaptive (standard)	Assumption 1,2,3	Any MAS	No	No

Table 2. Comparisons of communication and computation cost

Scheme	PK Size	CT Size	SK Size	KeyGen Cost	Encryption Cost	Decryption Cost
Liu[9]	$(n+3)\lvert\mathbb{G}_{p_1}\rvert + \lvert\mathbb{G}_T\rvert$	$(2l+2)\lvert\mathbb{G}_{p_1}\rvert + \lvert\mathbb{G}_T\rvert$	$(\lvert S\rvert+3)\lvert\mathbb{G}_{p_1,p_3}\rvert + \lvert\mathbb{Z}_N\rvert$	$(\lvert S\rvert+4)E_{\mathbb{G}}$	$(3l+2)E_{\mathbb{G}} + E_{\mathbb{G}_T}$	$(m+1)E_{\mathbb{G}} + mE_{\mathbb{G}_T} + (2m+1)T_p$
Ning[11]	$6\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$(3l+2)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$(2\lvert S\rvert+3)\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_N\rvert$	$(3\lvert S\rvert+5)E_{\mathbb{G}}$	$(5l+2)E_{\mathbb{G}} + E_{\mathbb{G}_T}$	$(m+1)E_{\mathbb{G}} + mE_{\mathbb{G}_T} + (3m+1)T_p$
Zhang[13]	$4\lvert\mathbb{G}_{p_1}\rvert + \lvert\mathbb{G}_T\rvert$	$(2l+3)\lvert\mathbb{G}_{p_1}\rvert + \lvert\mathbb{G}_T\rvert$	$(\lvert S\rvert+3)\lvert\mathbb{G}_{p_1,p_3}\rvert + \lvert\mathbb{Z}_{p_1}\rvert + \lvert\mathbb{Z}_N\rvert$	$(\lvert S\rvert+10)E_{\mathbb{G}} + (\lvert S\rvert+4)T_p$	$(3l+3)E_{\mathbb{G}} + E_{\mathbb{G}_T}$	$(2m+3)E_{\mathbb{G}} + mE_{\mathbb{G}_T} + (2m+3)T_p$
Ours	$(n+3)\lvert\mathbb{G}_{p_1}\rvert + \lvert\mathbb{G}_T\rvert$	$(2\lvert\mathcal{D}\rvert+2)\lvert\mathbb{G}_{p_1}\rvert + \lvert\mathbb{G}_T\rvert$	$(\lvert S\rvert+3)\lvert\mathbb{G}_{p_1,p_3}\rvert + \lvert\mathbb{Z}_N\rvert$	$(\lvert S\rvert+4)E_{\mathbb{G}}$	$(3\lvert\mathcal{D}\rvert+2)E_{\mathbb{G}} + E_{\mathbb{G}_T}$	$2E_{\mathbb{G}} + 3T_p$

In this section, some comparisons are made between our scheme and several previous schemes to show our characteristics. From **Table 1**, we can see that while our scheme does not support authority accountability and large universe, it is proved adaptively secure in the standard model which is only achieved by Liu et al's traceable CP-ABE that our scheme base on.

From **Table 2**, we can see that Zhang et al.' scheme needs more computation for generating the key in order to achieve authority accountability. Ning et al.'s scheme is constructed over prime order groups, which has high efficiency, however, just as is described above, the scheme is proved only selectively secure. Our scheme and Liu et al.'s scheme have the same public key size, user's secret key size and key generation cost. However, the ciphertext size and encryption cost in Liu et al.s scheme are linear with l, the size of a LSSS, while in our scheme they are linear with $|\mathcal{D}|$, the size of minimal authorized sets in a MAS.

4.2 Experimental verification

The experimental environment is 64 bit Ubuntu 14.04 operating system with Intel Core i7-3770 CPU (3.4 GHz) and memory 4G. The experimental code is modified and written based on PBC-0.5.14 [20] and cpabe-0.11 [21], and it uses the 160 bit elliptic curve group in hypersingular curve based on 512 bit finite field. The experimental result is the average value of that runs 20 times.

In this section, our proposed scheme is verified and compared with Liu's scheme [9] and Zhang's scheme [13] which are also constructed based on the composite order bilinear groups. We mainly consider the pairing and exponential operations in the groups $\mathbb{G}$ and $\mathbb{G}_T$. In the composite order bilinear groups, the time to run a pairing operation is about $1.26\,s$, the exponential operation in the group $\mathbb{G}$ is about $0.53\,s$, and the exponential operation in the group $\mathbb{G}_T$ is about $0.18\,s$. The specific computation time is shown in **Table 3**.

Table 3. Computation time

operation	time(s)	Liu[9]			Zhang[13]			Ours		
		key generation	encryption	decryption	key generation	encryption	decryption	key generation	encryption	decryption
pairing	1.26	0	0	$2m+1$	$k+5$	0	$2m+3$	0	0	3
exponential in $\mathbb{G}$	0.53	$k+4$	$3l+2$	$m+1$	$k+8$	$3l+3$	$2m+3$	$k+4$	$3\lvert\mathcal{D}\rvert+2$	2
exponential in $\mathbb{G}_T$	0.18	0	1	m	0	1	$m+1$	0	0	0
computation time		$0.53k+2.12$	$1.59l+1.24$	$3.23m+1.79$	$1.79k+10.54$	$1.59l+1.77$	$3.76m+5.55$	$0.53k+2.12$	$1.59\lvert\mathcal{D}\rvert+1.06$	4.84

Assuming that the number of users' attributes and attributes matched during decryption are between 5 and 50.

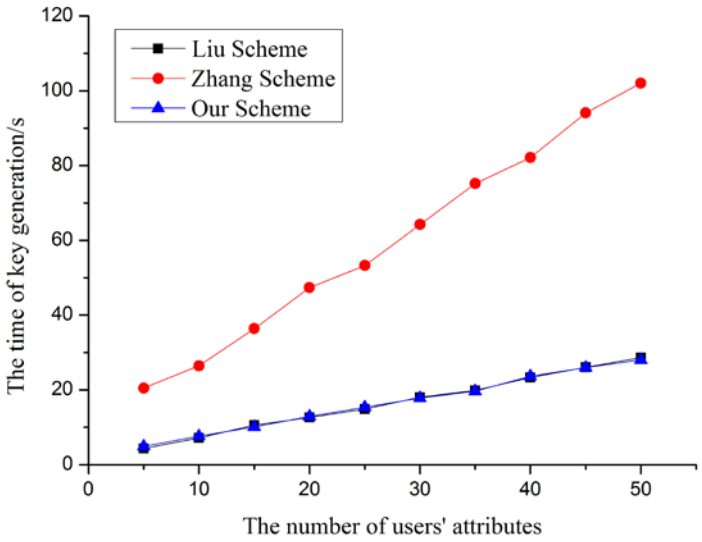

Fig. 2. The time of key generation

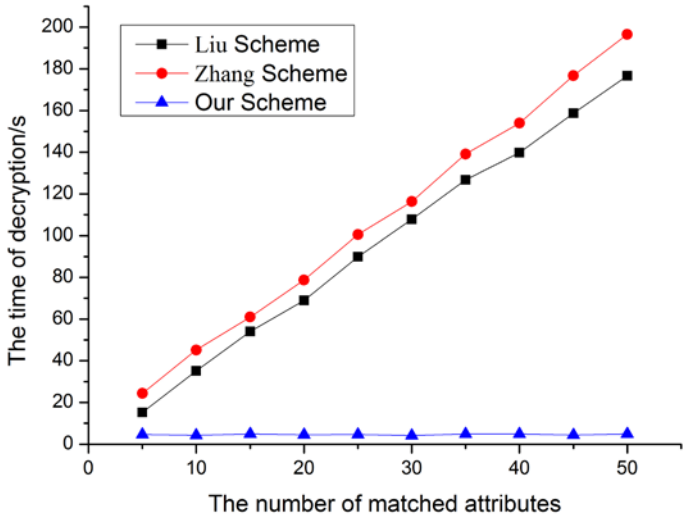

Fig. 3. The time of decryption

As shown in **Table 3** and **Fig. 2**, our scheme and Liu scheme [9] have the same key generation cost, which increases linearly with the number of users' attributes. For Zhang scheme [13], it requires zero knowledge proof of user and authorization attribute which increases communication cost, and its key generation cost is much higher than that of tour scheme and Liu scheme [9]. For decryption computation, as shown in **Table 3** and **Fig. 3**, the decryption time of Liu scheme [9] is $(3.23m+1.79)s$ and Zhang scheme [13] is $(3.76m+5.55)s$. The decryption cost of two schemes increases linearly with the number of matched attributes during decryption. However, the decryption cost of our scheme only needs three pairing operations and two exponential computations in the group $\mathbb{G}$, and the computation cost is a constant value 4.84. For encryption computation, as shown in **Table 3**, the encryption cost of Liu scheme [9] and Zhang scheme [13] increases linearly with the number of rows l of LSSS matrix representing monotonic access structure, while our scheme uses the set of minimum authorized subset to represent monotonic access structure, therefore, the encryption cost increases linearly with the set size of minimum authorized subset.

It must be noted that, there is not any obvious correlation between the size of minimal sets and corresponding access structure. Suppose that $1<t<n$, so $|\mathcal{D}|$ is defined as $|\mathcal{D}|=n!/((n-t)!t!)$. It is obvious $|\mathcal{D}|$ is greater than n, however, there is a LSSS, whose size is $l=n$, to achieve the (t,n)-threshold access. Additionally, there exist some MAS for which $|\mathcal{D}|$ is a constant value, however, the size of LSSS achieving MAS is polynomial with the number of attributes in the access structure. Take a simple example with n attributes, if we simply use the AND-gate, then $|\mathcal{D}|$ equals to 1, however, the LSSS size equals to n, namely $l=n$. Next, we take several non-trivial examples. If $\mathbb{S}_0=\{A_1=\{s_1,...,s_{\lceil n/2\rceil}\}, A_2=\{s_{\lceil n/2\rceil+1},...,s_n\}\}$ is the collection of minimal sets for a MAS, $\mathbb{S}$, with attributes $s_1,s_2,...,s_n$, then we have $|\mathcal{D}|=2$ and the size l of LSSS achieving $\mathbb{S}$ is at least $\mathcal{O}(n)$.

Similarly, if $\mathbb{S}_0 = \{A_1 = \{s_1, ..., s_{\lceil n/3 \rceil}\}, A_2 = \{s_{\lceil n/3 \rceil+1}, ..., s_{\lceil 2n/3 \rceil}\}, A_3 = \{s_{\lceil 2n/3 \rceil+1}, ..., s_n\}\}$, then we have $|\mathcal{D}| = 3$ and l is also at least $\mathcal{O}(n)$. Therefore, our scheme has shorter ciphertext under these circumstances.

The most important is that our decryption needs only three pairing operations and two exponent operations in $\mathbb{G}$, which are both constant in our scheme, while they are linear with the matching attributes m in the other three schemes.

References

[1] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Proc. of International Conference on Theory and Applications of Cryptographic Techniques*, Heidelberg, Berlin, Germany: Springer, pp. 457-473, 2005. Article (CrossRef Link)

[2] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Proc. of Workshop on the Theory and Application of Cryptographic Techniques*, Heidelberg, Berlin, Germany: Springer, pp. 47-53, 1984. Article (CrossRef Link)

[3] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. of the 13th ACM Conference on Computer and Communications Security*, Alexandria, USA, pp. 89-98, 2006. Article (CrossRef Link)

[4] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *Proc. of IEEE Symposium on Security & Privacy*, Berkeley, USA, pp. 321-334, 2007. Article (CrossRef Link)

[5] V. Goyal, A. Jain A, O. Pandey O and A. Sahai, "Bounded Ciphertext Policy Attribute Based Encryption," in *Proc. of International Colloquium on Automata, Languages, and Programming*, Heidelberg, Berlin, Germany: Springer, pp. 579-591, 2008. Article (CrossRef Link)

[6] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. of the 2007 ACM Conference on Computer and Communications Security*, Alexandria, Virginia, USA, pp. 456-465, 2007. Article (CrossRef Link)

[7] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," in *Proc. of International Workshop on Public Key Cryptography*, Heidelberg, Berlin, Germany: Springer, pp. 53-70, 2011. Article (CrossRef Link)

[8] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," in *Proc. of International Conference on Theory & Applications of Cryptographic Techniques*, Berlin, Germany: Springer, pp. 62-91, 2010. Article (CrossRef Link)

[9] Z. Liu, Z. Cao and D.S. Wong, "White-Box Traceable Ciphertext-Policy Attribute-Based Encryption Supporting Any Monotone Access Structures," *IEEE Transactions on Information Forensics & Security*, vol. 8, no. 1, pp. 76-88, 2013. Article (CrossRef Link)

[10] B. Dan and X. Boyen, "Short Signatures Without Random Oracles," in *Proc. of Advances in Cryptology - EUROCRYPT 2004*, Berlin, Germany: Springer, pp. 56-73, 2004. Article (CrossRef Link)

[11] J. Ning, Z. Cao, X. Dong, L. Wei and X. Lin, "Large Universe Ciphertext-Policy Attribute-Based Encryption with White-Box Traceability," in *Proc. of Computer Security - ESORICS 2014*, Berlin, Germany: Springer, pp. 55-72, 2014. Article (CrossRef Link)

[12] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. of ACM Sigsac Conference on Computer & Communications Security*, Berlin, Germany, ACM, pp. 463-474, 2013. Article (CrossRef Link)

[13] Y. Zhang, J. Li, D. Zheng, X. Chen and L. Hui, "Accountable Large-Universe Attribute-Based Encryption Supporting Any Monotone Access Structures," in *Proc. of Australasian Conference on Information Security and Privacy*, Berlin, Germany: Springer, pp. 509-524, 2016. Article (CrossRef Link)

[14] V. Odelu, A. K. Das , Y. S. Rao , S. Kumari, M. K. Khan and K. K. R. Choo, "Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment," *Computer Standards & Interfaces*, vol. 54, pp. 3-9, 2016. Article (CrossRef Link)
[15] V, Odelu, A. K. Das, M, Khurram Khan, K. K. R. Choo and M. Jo, "Expressive CP-ABE Scheme for Mobile Devices in IoT satisfying Constant-size Keys and Ciphertexts," *IEEE Access*, vol. 5, pp. 3273-3283, 2017. Article (CrossRef Link)
[16] V. Odelu and A. K. Das, "Design of a new CP‐ABE with constant‐size secret keys for lightweight devices using elliptic curve cryptography," *Security & Communication Networks*, vol. 9, no. 17, pp. 4048-4059, 2016. Atricle(CrossRef Link)
[17] T. Pandit and R. Barua, "Efficient Fully Secure Attribute-Based Encryption Schemes for General Access Structures," *Provable Security*, Heidelberg, Berlin: Springer, pp. 193-214, 2013.
[18] B. Dan, E. J. Goh and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," in *Proc. of International Conference on Theory of Cryptography*, Heidelberg, Berlin: Springer, pp. 325-341, 2005. Atricle(CrossRef Link)
[19] A. Lewko and B. Waters, "New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts," in *Proc. of International Conference on Theory of Cryptography*, Heidelberg, Berlin: Springer, pp. 455-479, 2010. Atricle(CrossRef Link)
[20] B. Lynn, "The pairing-based cryptography (PBC) library," 2006, [Online]: http://crypto.stanford.edu/pbc.
[21] J. Bethencourt, A. Sahai and B. Waters, "Advanced crypto software collection: the cpabetoolkit," 2011,[Online]:http://acsc.cs.utexas.edu/cpabe.

Guangbo Wang was born in Shandong, he received the PhD in Zhengzhou Information Science and Technology Institute. He is now an information security engineer in Beijing Science and Technology Information Research Center. His research interests include information security and cryptography.(Email:691759571@qq.com).

Feng Li was born in Shanxi, he received the Master's degree in Shandong University. He is now an information security senior engineer in Beijing Science and Technology Information Research Center. His research interests include information security and cryptography.(Email: lyx_wj@sina.com).

PengCheng Wang was born in Shandong, he received the Master's degree in Electronic Engineering Institute. He is now an information security senior engineer in Beijing Science and Technology Information Research Center. His research interests include information security and cryptography.(Email: 928822191@qq.com).

Yixiao Hu was born in Sichun, he received the Master's degree in Zhengzhou Information Science and Technology Institute. He is now an information security senior engineer in Beijing Science and Technology Information Research Center. His research interests include information security and cryptography.(Email: fox14901648@163.com).