**SCIENCE CHINA**
Information Sciences

• **Supplementary File** •

# Traceable and Revocable CP-ABE with shorter ciphertexts

Jianting NING [1], Zhenfu CAO[2*], Xiaolei DONG[2] & Lifei WEI[3]

[1] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[2]*Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai* 200062*, China;*
[3]*College of Information Technology, Shanghai Ocean University, Shanghai* 201306*, China*

## Appendix A Black-box Traceable and Revocable CP-ABE

### Appendix A.1 Definition

A Black-box Traceable and Revocable system (BTR-CPABE system) is a CP-ABE system where a decryption black-box can be traced to the corresponding malicious users who built it and can revoke the traced malicious users. We enhance the conventional (non-traceable and non-revocable ) CP-ABE system by assigning and identifying users with unique indices, adding the functionalities of traceability and revocability to it. In particular, following the notation of CP-ABE system introduced in [4], a BTR-CPABE system consists of five algorithms as follows:

• $\texttt{Setup}(\lambda, \mathcal{U}, \mathcal{N}) \rightarrow (pp, msk)$. On input a security parameter $\lambda$, the attribute universe description $\mathcal{U}$, the number of users $\mathcal{N}$ in the system, it outputs the public parameters $pp$ and a master secret key $msk$.

• $\texttt{KeyGen}(pp, msk, S) \rightarrow sk_{n_i,S}$. On input $pp$, $msk$ and a set of attributes $S$, it outputs a private key $sk_{n_i,S}$, which is assigned and identified by a unique index $n_i \in \{1, ..., \mathcal{N}\}$.

• $\texttt{Encrypt}(pp, \mathbb{A}, M, RL) \rightarrow ct$. On input $pp$, an access structure $\mathbb{A}$ over the universe of attributes, a message $M$ and a revocation list $RL \subseteq \{1, ..., \mathcal{N}\}$, it outputs a ciphertext $ct$. The revocation list $RL$ is used to store the indexes of revoked users and to ensure that the resulting ciphertext cannot be decrypted by any secret key which is in the revocation list even though the associated attribute set of the key satisfies the ciphertext's policy.

• $\texttt{Decrypt}(pp, sk_{n_i,S}, ct) \rightarrow M$ or $\perp$. On input $pp$, a secret key $sk_{n_i,S}$, and a ciphertext $ct$, if $n_i \in \{1, ..., \mathcal{N}\} \backslash RL$ and $S$ satisfies $ct$'s access policy, it outputs $M$. Otherwise, it output $\perp$.

• $\texttt{Trace}^{\mathcal{D}}(pp, S_{\mathcal{D}}, RL_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{N}_T$: The tracing algorithm takes $pp$, a non-empty attribute set $S_{\mathcal{D}}$, a revocation list $RL_{\mathcal{D}}$ and a probability value (lower-bound) $\epsilon$ [1)]. It is an oracle algorithm interacts with a key-like decryption black-box $\mathcal{D}$. It runs in a polynomial time in $1^{\lambda}$ and $1/\epsilon$, and outputs an index set $\mathbb{N}_T \subseteq \{1, ..., \mathcal{N}\}$ of malicious user(s). Note that in our setting, we treat $\mathcal{D}$ as a probabilistic circuit that takes as input a ciphertext $ct$ and returns a message $M$ or $\perp$. Such a decryption black-box does not need to be perfect, since we only require it to decrypt successfully with non-negligible probability.

### Appendix A.2 Message-hiding Security

The message-hiding security is a typical semantic security similar to that of conventional CP-ABE system [4], excepting every key query is companied with a unique index. Similar to [5], to capture the security that an adversary can choose keys to corrupt adaptively, we allow an adversary to specify the index (which is originally assigned by the $\texttt{KeyGen}$ algorithm) to a decryption key when he makes a key query. Note that to guarantee that each user/key can be identified by an index uniquely, an adversary can adaptively ask for a decryption key corresponding to $(n_i, S_{n_i})$ for $i \in \{1, ..., q\}$, where $n_i \in \{1, ..., \mathcal{N}\}, q \leqslant \mathcal{N}$. Also note that for any two pairs $(n_i, S_{n_i})$ and $(n_j, S_{n_j})$ where $n_i \neq n_j$ for $\forall i \neq j, i, j \in \{1, ..., q\}$, we do not require $S_{n_i} \neq S_{n_j}$. The message-hiding security is described by a security game $Game_{MH}$ between an adversary $\mathcal{A}$ and a challenger. The phases of the game are as follows:

* Corresponding author (email: zfcao@sei.ecnu.edu.cn)

1) Note that $\epsilon$ is the lower-bound of a key-like decryption black-box's decryption ability, and it has to be polynomially related to the security parameter.

- **Setup** : The challenger runs the $\texttt{Setup}(\lambda, \mathcal{U}, \mathcal{N})$ and sends the public parameters $pp$ to $\mathcal{A}$.
- **Query Phase 1** : For $i = 1$ to $q_1$, $\mathcal{A}$ submits $(n_i, S_{n_i})$, and the challenger responds with $sk_{n_i, S_{n_i}}$.
- **Challenge** : $\mathcal{A}$ submits two equal length messages $M_0, M_1$, an access policy $\mathbb{A}^*$ and a revocation list $RL^*$. $\mathbb{A}^*$ cannot be satisfied by any of the queried attribute sets $S_{n_1}, ..., S_{n_{q_1}}$. The challenge flips a random coin $\beta \in \{0, 1\}$, gives an encryption of $M_\beta$ under $\mathbb{A}^*$ to $\mathcal{A}$.
- **Query Phase 2** : For $i = q_1 + 1$ to $q$, $\mathcal{A}$ submits $(n_i, S_{n_i})$ with the restriction that none of these queried attribute sets satisfy $\mathbb{A}^*$, and the challenger responds with $sk_{n_i, S_{n_i}}$.
- **Guess** : $\mathcal{A}$ outputs a guess $\beta' \in \{0, 1\}$ for $\beta$.

$\mathcal{A}$'s advantage is defined as $Adv = \Pr[\beta' = \beta] - \frac{1}{2}$.

**Definition 1.** A $\mathcal{N}$-user BTR-CPABE system is message-hiding secure if there exists no probabilistic polynomial-time (PPT) adversary has non-negligible advantage in $Game_{MH}$.

### Appendix A.3 Black-box Traceability

The black-box traceability definition is described by a security game $Game_{BT}$ between an adversary $\mathcal{A}$ and a challenger. The phases of the game are as follows:

- **Setup** : The challenger runs the $\texttt{Setup}(\lambda, \mathcal{U}, \mathcal{N})$ and sends the public parameters $pp$ to $\mathcal{A}$.
- **Key Query** : For $i = 1$ to $q$, $\mathcal{A}$ submits $(n_i, S_{n_i})$, and the challenger responds with $sk_{n_i, S_{n_i}}$.
- **(Key-like) Decryption Black-box Generation** : $\mathcal{A}$ outputs a decryption black-box $\mathcal{D}$ associated with a non-empty attribute set $S_{\mathcal{D}}$ and a probability value $\epsilon$.
- **Trace** : The challenger runs $\texttt{Trace}^{\mathcal{D}}(pp, S_{\mathcal{D}}, RL_{\mathcal{D}}, \epsilon)$ to get a set $\mathbb{N}_T \subseteq \{1, ..., \mathcal{N}\}$ of malicious user.

Let $\mathbb{N}_{\mathcal{D}} = \{n_i | 1 \leqslant i \leqslant q\}$ be the index set of corrupted keys. We say $\mathcal{A}$ wins the above game if the following conditions hold: (1) $\mathcal{D}$ is a useful key-like decryption black-box. That is, it holds that $\Pr[\mathcal{D}(\texttt{Encrypt}(pp, \mathbb{A}_{\mathcal{D}}, M, RL_{\mathcal{D}})) = M] \geqslant \epsilon$ for any access policy $\mathbb{A}_{\mathcal{D}}$ that is satisfied by $S_{\mathcal{D}}$, where the probability is taken over the random coins of $\mathcal{D}$ and the random choices of message $M$; (2) $((n_i \in RL_{\mathcal{D}})\ OR\ (S_{\mathcal{D}} \nsubseteq S_{n_i})\ for\ \forall n_i \in \mathbb{N}_T)$, or $\mathbb{N}_T \nsubseteq \mathbb{N}_{\mathcal{D}}$, or $\mathbb{N}_T = \emptyset$.

**Definition 2.** A $\mathcal{N}$-user BTR-CPABE system is traceable against key-like decryption black-box if there exists no PPT adversary has non-negligible advantage in $Game_{BT}$.

## Appendix B Background

### Appendix B.1 Notation

We define $[l] = \{1, 2, ..., l\}, [l_1, l_2] = \{l_1, l_1 + 1, ..., l_2\}$ for $l \in \mathbb{N}$. By PPT we denote probabilistic polynomial-time.

### Appendix B.2 Access Policy, Linear Secret-Sharing Schemes and Composite Order Bilinear Groups

As of previous work, we use linear secret-sharing schemes (LSSS) to realize monotonic access structures which specify the access policies associated with ciphertexts. As of previous work, we construct our system in composite order bilinear groups. We refer the interested readers to [4] for the formal definitions of access structures, LSSS and composite order bilinear groups.

### Appendix B.3 Assumptions

The message-hiding security of our E-CPABE system will rely on four assumptions (the Assumption 1, the General Subgroup Decision Assumption, the 3-Party Diffie-Hellman Assumption in a Subgroup, the Source Group $q$-Parallel BDHE Assumption in a Subgroup in [4]), which are used in [4] to achieve full security of the CP-ABE scheme in [4]. We refer to [4] for the details of these assumptions.

External Diffie-Hellman (XDH) Assumption in a Subgroup: Given a group generator $\mathcal{G}$, define the following distribution: $GD = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}, a, b \xleftarrow{R} \mathbb{Z}_N,\ g \xleftarrow{R} G_{p_1}, g_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3},\ D = (GD, g, g_2, g_3, g^a, g^b),\ T_0 = g^{ab}, T_1 \xleftarrow{R} G_{p_1}$. An algorithm $\mathcal{A}$'s advantage in breaking this assumption is: $Adv^{XDH}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies the XDH Assumption in a Subgroup if $Adv^{XDH}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of $\lambda$ for any polynomial time algorithm $\mathcal{A}$.

## Appendix C Enhanced CP-ABE

Following the routes of [1–3,5], instead of constructing BTR-CPABE directly, we define a simpler primitive named Enhanced CP-ABE (E-CPABE) firstly, then we show how BTR-CPABE can be transformed from E-CPABE.

## Appendix C.1 Definition

An E-CPABE system consists of the following algorithms.

• $\texttt{Setup}_E(\lambda,\mathcal{U},\mathcal{N}) \rightarrow (pp, msk)$. On input a security parameter $\lambda$, the attribute universe description $\mathcal{U}$, the numbers of users $\mathcal{N}$ in the system, it outputs the public parameters $pp$, a master secret key $msk$.

• $\texttt{KeyGen}_E(pp, msk, S) \rightarrow sk_{n_i,S}$. On input $pp$, $msk$ and a set of attributes $S$, it outputs a private key $sk_{n_i,S}$, which is assigned and identified by a unique index $n_i \in [\mathcal{N}]$.

• $\texttt{Encrypt}_E(pp, \mathbb{A}, n_j, M, RL) \rightarrow ct$. On input $pp$, an access structure $\mathbb{A}$ over the universe of attributes, an index $n_j \in [\mathcal{N}+1]$, a message $M$ and a revocation list $RL$, it outputs a ciphertext $ct$.

• $\texttt{Decrypt}_E(pp, sk_{n_i,S}, ct) \rightarrow M$ or $\perp$. On input $pp$, a secret key $sk_{n_i,S}$, and a ciphertext $ct$ encrypted with index $n_j$, if $(n_i \in [\mathcal{N}]\backslash RL) \wedge (S\ satisfies\ ct's\ access\ policy) \wedge (n_i \geqslant n_j)$, it outputs the message $M$. Otherwise, it output $\perp$.

## Appendix C.2 Message-hiding Security

The message-hiding security is described by a security game between an adversary $\mathcal{A}$ and a challenger:

• **Setup** : The challenger runs the $\texttt{Setup}_E(\lambda,\mathcal{U},\mathcal{N})$ algorithm and sends public parameters $pp$ to $\mathcal{A}$.

• **Query Phase 1** : For $i = 1$ to $q_1$, $\mathcal{A}$ submits $(n_i, S_{n_i})$, and the challenger responds with $sk_{n_i,S_{n_i}}$.

• **Challenge** : $\mathcal{A}$ submits two equal length messages $M_0, M_1$, an access policy $\mathbb{A}^*$, a revocation list $RL^*$. The challenge flips a random coin $\beta \in \{0,1\}$ and gives $ct \leftarrow \texttt{Encrypt}_E(pp, \mathbb{A}^*, n_j, M_\beta, RL^*)$ to $\mathcal{A}$.

• **Query Phase 2** : For $i = q_1 + 1$ to $q$, $\mathcal{A}$ submits $(n_i, S_{n_i})$, and the challenger responds with $sk_{n_i,S_{n_i}}$.

• **Guess** : $\mathcal{A}$ outputs a guess $\beta' \in \{0,1\}$ for $\beta$.

We define game $Game^E_{MH_1}$ as follows. We let the challenger give $ct \leftarrow \texttt{Encrypt}_E(pp, \mathbb{A}^*, 1, M_\beta, RL^*)$ to $\mathcal{A}$ during the **Challenge** phase. And $\mathcal{A}$ wins the game if $\beta' = \beta$ with the restriction that none of the queried attribute sets $S_{n_1}, ..., S_{n_q}$ satisfy $\mathbb{A}^*$. $\mathcal{A}$'s advantage is defined to be $Adv_1 = \Pr[\beta' = \beta] - \frac{1}{2}$ in this game. And we define game $Game^E_{MH_{\mathcal{N}+1}}$ as follows. We let the challenger give $ct \leftarrow \texttt{Encrypt}_E(pp, \mathbb{A}^*, \mathcal{N}+1, M_\beta, RL^*)$ to $\mathcal{A}$ during the **Challenge** phase. And $\mathcal{A}$ wins the game if $\beta' = \beta$. $\mathcal{A}$'s advantage is defined to be $Adv_{\mathcal{N}+1} = \Pr[\beta' = \beta] - \frac{1}{2}$ in this game.

**Definition 3.** A $\mathcal{N}$-user E-CPABE is message-hiding secure if there exists no PPT adversary has non-negligible advantage in $Game^E_{MH_1}$ and $Game^E_{MH_{\mathcal{N}+1}}$.

## Appendix C.3 Index-hiding Security

The index-hiding security against key-like decryption black-box is to guarantee that there has no adversary can distinguish between $\texttt{Encrypt}_E(pp, \mathbb{A}_{S^*}, n_j, M, RL^*)$ and $\texttt{Encrypt}_E(pp, \mathbb{A}_{S^*}, n_j + 1, M, RL^*)$ for any non-empty attribute set $S^* \subseteq \mathcal{U}$ without a secret key $sk_{n_j,S_{n_j}}$, where $\mathbb{A}_{S^*} = \wedge_{x \in S^*} x$ is defined to be the strictest access policy and $S_{n_j} \supseteq S^*$. It is described by a security game $Game^E_{IH}$ between an adversary $\mathcal{A}$ and a challenger. The game takes as input a parameter $n_j \in [\mathcal{N}]$ which is given to both $\mathcal{A}$ and the challenger. The phases of the game are as follows:

• **Setup** : The challenger runs the $\texttt{Setup}_E(\lambda,\mathcal{U},\mathcal{N})$ algorithm and sends public parameters $pp$ to $\mathcal{A}$.

• **Key Query** : For $i = 1$ to $q$, $\mathcal{A}$ submits $(n_i, S_{n_i})$, and the challenger responds with $sk_{n_i,S_{n_i}}$.

• **Challenge** : $\mathcal{A}$ submits a message $M$, an access policy $\mathbb{A}^*$ and a revocation list $RL^*$. The challenge flips a random coin $\beta \in \{0,1\}$ and gives $ct \leftarrow \texttt{Encrypt}_E(pp, \mathbb{A}^*, n_j + \beta, M, RL^*)$ to $\mathcal{A}$.

• **Guess** : $\mathcal{A}$ outputs a guess $\beta' \in \{0,1\}$ for $\beta$.

We define $\mathcal{A}$ wins the game if $\beta' = \beta$ with the restriction that none of the queried pairs $\{(n_1, S_{n_1}), ..., (n_q, S_{n_q})\}$ satisfy $((n_i \in [\mathcal{N}]\backslash RL^*) \wedge (S_{n_i} \supseteq S^*) \wedge (n_i = n_j))$ for any $i \in [q]$. $\mathcal{A}$'s advantage is defined as $Adv_{n_j} = \Pr[\beta' = \beta] - \frac{1}{2}$.

**Definition 4.** A $\mathcal{N}$-user E-CPABE is index-hiding secure against key-like decryption black-box if there exists no PPT adversary has non-negligible advantage $Adv_{n_j}$ for any $n_j \in [\mathcal{N}]$ in $Game^E_{IH}$.

## Appendix C.4 Transform from E-CPABE to BTR-CPABE

We show how a BTR-CPABE system can be transformed from an E-CPABE system following the routes of [1–3, 5]. We denote an E-CPABE system as $\Sigma_e$, then a BTR-CPABE system can be transformed from $\Sigma_e$ by the following two steps providing that $\Sigma_e$ is message-hiding secure and index-hiding secure.

Step 1: Set $n_j$ of $\texttt{Encrypt}_E(pp, \mathbb{A}, n_j, M, RL)$ equal to 1, i.e., $\texttt{Encrypt}_E(pp, \mathbb{A}, n_j, M, RL) = \texttt{Encrypt}_E(pp, \mathbb{A}, 1, M, RL)$; Step 2: Add a $\texttt{Trace}$ algorithm to $\Sigma_e$ defined as follows.

• $\texttt{Trace}^{\mathcal{D}}(pp, S_{\mathcal{D}}, RL_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{N}_T \subseteq [\mathcal{N}]$: The tracing algorithm takes as input $pp$, a non-empty attribute set $S_{\mathcal{D}}$, a revocation list $RL_{\mathcal{D}}$ and a probability value $\epsilon$. Given a decryption black-box $\mathcal{D}$ associated with the non-empty attribute set $S_{\mathcal{D}}$, it works as follows:

1. For $n = 1$ to $\mathcal{N}+1$, do as follows: (1) Repeat the following steps $8\lambda(\mathcal{N}/\epsilon)^2$ times: Firstly, randomly sample message $M$ from the message space. Then, let $ct \leftarrow \texttt{Encrypt}_E(pp, \mathbb{A}_{S_{\mathcal{D}}}, n, M, RL_{\mathcal{D}})$, where $\mathbb{A}_{S_{\mathcal{D}}}$ is the strictest access policy of $S_{\mathcal{D}}$. Next, Call oracle $\mathcal{D}$ on input $ct$ and compare the output of $\mathcal{D}$ with $M$; (2) Let $f_n$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
2. Let $\mathbb{N}_T$ be the set of all $n \in [\mathcal{N}]$ for which $f_n - f_{n+1} \geqslant \epsilon/(4\mathcal{N})$.
3. Output the set $\mathbb{N}_T$ as the malicious users.

**Theorem 1.** If $\Sigma_e$ is message-hiding and index-hiding secure against key-like decryption black-box, the modified $\Sigma_e$ (denote by $\Sigma_{btr}$) after the transformation is traceable BTR-CPABE against key-like decryption black-box.

*Proof.* We use a similar proof method from [1, 3, 5]. Note that if we always set the parameter $n$ of the algorithm $\texttt{Encrypt}_E(pp, \mathbb{A}, n, M, RL)$ equal to 1, the functions of E-CPABE are identical to that of BTR-CPABE. Hence, $Game_{MH}$ for $\Sigma_{btr}$ is the same to $Game^E_{MH_1}$ for $\Sigma_e$. It implies that if $\Sigma_e$ is adaptively (resp. selective) message-hiding in $Game^E_{MH_1}$, then $\Sigma_{btr}$ is adaptively (resp. selective) message-hiding in $Game_{MH}$.

In the proof sketch below, we show that if $\Sigma_e$ is adaptively (resp. selective) index-hiding secure against key-like decryption black-box and adaptively (resp. selective) message-hiding secure in $Game^E_{MH_{\mathcal{N}+1}}$, the probability of winning the traceability game defined in Section Appendix A.3 is negligible. That is, $\mathbb{N}_T \nsubseteq \mathbb{N}_{\mathcal{D}}$, or $S_{\mathcal{D}} \nsubseteq S_{u_i}$ *for* $\forall u_i \in \mathbb{N}_T$, or $\mathbb{N}_T = \emptyset$, where $\mathcal{D}$ is the key-like decryption black-box generated by an adversary. Let $p_n = \Pr[\mathcal{D}(\texttt{Encrypt}_E(pp, \mathbb{A}_{\mathcal{D}}, n, M, RL) = M]$, where the probability is over the random coins of $\mathcal{D}$ and the random choice of $M$. Following the fact that $\mathcal{D}$ is an $\epsilon$-userful decryption black-box and $\Sigma_e$ is adaptively (resp. selective) message-hiding secure in $Game^E_{MH_{\mathcal{N}+1}}$, we have that $p_1 \geqslant \epsilon$ and $p_{\mathcal{N}+1}$ is negligible. Then there must exist some $n \in [\mathcal{N}]$ such that $p_n - p_{n+1} \geqslant \epsilon/(2\mathcal{N})$. By the Chernoff bound it follows that with overwhelming probability, $f_n - f_{n+1} \geqslant \epsilon/(4\mathcal{N})$. Hence, the set $\mathbb{N}_T$ is non-empty. For any $n \in \mathbb{N}_T$ such that $f_n - f_{n+1} \geqslant \epsilon/(4\mathcal{N})$ we know, by Chernoff, that with overwhelming probability $p_n - p_{n+1} \geqslant \epsilon/(8\mathcal{N})$. Thus, $u \subseteq \mathbb{N}_{\mathcal{D}}$ and $S_{\mathcal{D}} \subseteq S_u$. Otherwise, $\mathcal{D}$ is able to distinguish $\texttt{Encrypt}_E(pp, \mathbb{A}_{\mathcal{D}}, n, M, RL)$ from $\texttt{Encrypt}_E(pp, \mathbb{A}_{\mathcal{D}}, n+1, M, RL)$. Therefore, we have $S_n$ *satisfies* $\mathbb{A}_{\mathcal{D}}$ *for* $\forall n \in \mathbb{N}_T$ and $\mathbb{N}_T \subseteq \mathbb{N}_{\mathcal{D}}$.

## Appendix C.5 Message-hiding Security

**Proof of message-hiding security in $Game_{MH_1}$:**

Since the structures of the CP-ABE part of the E-CPABE are similar to that of the CP-ABE in [4], the proof of message-hiding security in $Game_{MH_1}$ is also similar to that of [4]. For simplicity, we will reduce the proof of message-hiding security in $Game_{MH_1}$ of the E-CPABE to that of the CP-ABE in [4]. By $\Sigma_{cpabe}$, $\Sigma_{ecpabe}$ we denote the CP-ABE in [4] and the E-CPABE respectively.

**Lemma 1.** [4] If Assumption 1, the general subgroup decision assumption, the 3-party Diffie-Hellman assumption in a subgroup, and the source group q-parallel BDHE assumption in a subgroup hold, $\Sigma_{cpabe}$ (in [4]) is fully secure.

**Lemma 2.** If $\Sigma_{cpabe}$ is fully secure, then no polynomial time adversary can achieve a non-negligible advantage in winning $Game_{MH_1}$ for our E-CPABE system $\Sigma_{ecpabe}$.

*Proof.* Suppose there exists a PPT adversary $\mathcal{A}$ achieves a non-negligible advantage $Adv_{\mathcal{A}}\Sigma_{ecpabe}$ in adaptively breaking $\Sigma_{ecpabe}$. We construct a PPT algorithm $\mathcal{B}$ that achieves a non-negligible advantage $Adv_{\mathcal{B}}\Sigma_{cpabe}$ in adaptively breaking $\Sigma_{cpabe}$, which equals to $Adv_{\mathcal{A}}\Sigma_{ecpabe}$.

• **Setup** : $\Sigma_{cpabe}$ gives $\mathcal{B}$ the public parameter $pp_{\Sigma_{cpabe}} = (GD, g, g^a, g^b, e(g,g)^\alpha, \{H_k = g^{h_k}\}_{k\in\mathcal{U}})$ . $\mathcal{B}$ randomly chooses $\{\alpha_i, r_i, f_i\}_{i\in[m]}, \{b_j\}_{j\in[m]} \in \mathbb{Z}_N$, and sends $\mathcal{A}$ the new public parameter $pp$ as follows: $(GD, g, u = g^a, v = g^b, \{E_i = e(g,g)^\alpha e(g,g)^{\alpha'_i}, G_i = g^{r_i}, F_i = g^{f_i}\}_{i\in[m]}, \{B_j = g^{b_j}\}_{j\in[m]}, \{H_k = g^{h_k}\}_{k\in\mathcal{U}})$, where $\{\alpha_i\}_{i\in m} \in \mathbb{Z}_N$ are implicitly chosen such that $\{\alpha + \alpha'_i \equiv \alpha_i \ mod\ p_1\}_{i\in m}$.

• **Phase 1**: $\mathcal{A}$ submits $((i,j), S_{i,j})$ to $\mathcal{B}$ to query a decryption key. $\mathcal{B}$ submits $S_{i,j}$ to $\Sigma_{cpabe}$ and gets a decryption key $\hat{sk}_{S_{i,j}} = \langle \hat{K} = g^\alpha g^{at} g^{bs'} R, \hat{K}' = g^{s'} R', \hat{K}'' = g^t R'', \{\hat{K}_k = H_k^t R_k\}_{k\in S_{i,j}}\rangle$. $\mathcal{B}$ randomly chooses $R''' \in G_{p_3}$ and sends $\mathcal{A}$ the decryption key $sk_{(i,j),S_{i,j}}$ as follows: $\langle K_{i,j} = \hat{K} g^{\alpha'_i} g^{r_i b_j}, K'_{i,j} = \hat{K}', K''_{i,j} = \hat{K}'', K'''_{i,j} = (\hat{K}'')^{f_i} R''', \{K_{i,j,k} = \hat{K}_k\}_{k\in S_{i,j}}\rangle$.

• **Challenge**: $\mathcal{A}$ submits two equal length messages $M_0, M_1$, an LSSS matrix $(A^*, \rho^*)$ and a revocation list $RL^*$ to $\mathcal{B}$, let $RL^*_{i'}, RL^*_{j'}$ be the sets of revoked row index and column index. $\mathcal{B}$ submits $(M_0, M_1, (A^*, \rho^*))$ to $\Sigma_{cpabe}$ and gets the challenge ciphertext $\hat{ct} = \langle (A^*, \rho^*), \hat{C}_0 = M_b e(g,g)^{\alpha\hat{s}}, \hat{C} = g^{\hat{s}}, \hat{C}' = g^{b\hat{s}}, \{\hat{C}_k = g^{aA^*_k \cdot \hat{\vec{\vartheta}}} H^{-\hat{\theta}_k}_{\rho^*(k)}, \hat{C}'_k = g^{\hat{\theta}_k}\}_{k\in[l]}\rangle$. $\mathcal{B}$ chooses random $\vec{\vartheta}' = (s', \vartheta'_2, ..., \vartheta'_n), \{\sigma_d\}_{d\in[2]}, \{\gamma_d\}_{d\in[6]}$,
$\{\delta'_i\}_{i\in[m]}, \{\delta''_i\}_{i\in[m]}, \{\eta_i\}_{i\in[m]}, \{\mu_j\}_{j\in[y-1]}, \{\theta'_k\}_{k\in[l]}, \{\tau_{i,d}\}_{i\in[x-1],d\in[4]} \in \mathbb{Z}_N$ under constraints that $\gamma_2\gamma_3 - \gamma_1\gamma_4 \neq 0$, $\gamma_1\gamma_6 - \gamma_2\gamma_5 = 0$ and $(\gamma_1 + \gamma_5)\gamma_4 - (\gamma_2 + \gamma_6)\gamma_3 = 0$. Note that $x = 1, y = 1$ in $Game_{MH_1}$.
For each row $i \in [m]$, it creates row ciphertexts $(C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}, C_{i,6}, C_{i,7})$ as follows:

• if $(i = x) \wedge (i \in RL_{i'})$: $C_{i,1} = G_i^{(\gamma_3+\gamma_5)\delta_{i,1}} \hat{C}^{\frac{r_i(\gamma_3+\gamma_5)}{(\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2}}, C_{i,2} = G_i^{(\gamma_4+\gamma_6)\delta_{i,1}} \hat{C}^{\frac{r_i(\gamma_4+\gamma_6)}{(\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2}}$,
$C_{i,3} = g^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_{i,1}} \hat{C}, C_{i,4} = u^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_{i,1}} F_i^{\eta_i} u^{s'}$,
$C_{i,5} = v^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_{i,1}} \hat{C}, C_{i,6} = g^{\eta_i}, C_{i,7} = M \cdot E_i^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_{i,1}} \hat{C}_0 e(g^{\alpha'_i}, \hat{C})$.

• if $(i = x) \wedge (i \notin RL_{i'})$: $C_{i,1} = G_i^{\gamma_1\delta_{i,2}} \hat{C}^{\frac{r_i\gamma_1}{\sigma_1\gamma_1+\sigma_2\gamma_2}}, C_{i,2} = G_i^{\gamma_2\delta_{i,2}} \hat{C}^{\frac{r_i\gamma_2}{\sigma_1\gamma_1+\sigma_2\gamma_2}}, C_{i,3} = g^{(\sigma_1\gamma_1+\sigma_2\gamma_2)\delta_{i,2}} \hat{C}$,
$C_{i,4} = u^{(\sigma_1\gamma_1+\sigma_2\gamma_2)\delta_{i,2}} F_i^{\eta_i} u^{s'}, C_{i,5} = v^{(\sigma_1\gamma_1+\sigma_2\gamma_2)\delta_{i,2}} \hat{C}', C_{i,6} = g^{\eta_i}, C_{i,7} = M \cdot E_i^{(\sigma_1\gamma_1+\sigma_2\gamma_2)\delta_{i,2}} \hat{C}_0 e(g^{\alpha'_i}, \hat{C})$.

• if $(i > x) \wedge (i \in RL_{i'})$: $C_{i,1} = G_i^{\gamma_3\delta_{i,3}} \hat{C}^{\frac{r_i\gamma_3}{\gamma_3\sigma_1+\gamma_4\sigma_2}}, C_{i,2} = G_i^{\gamma_4\delta_{i,3}} \hat{C}^{\frac{r_i\gamma_4}{\gamma_3\sigma_1+\gamma_4\sigma_2}}, C_{i,3} = g^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_{i,3}} \hat{C}$,
$C_{i,4} = u^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_{i,3}} F_i^{\eta_i} u^{s'}, C_{i,5} = v^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_{i,3}} \hat{C}', C_{i,6} = g^{\eta_i}, C_{i,7} = M \cdot E_i^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_{i,3}} \hat{C}_0 e(g^{\alpha'_i}, \hat{C})$.

• if $(i > x) \wedge (i \notin RL_{i'})$: $C_{i,1} = G_i^{(\gamma_1+\gamma_5)\delta_{i,4}} \hat{C}^{\frac{r_i(\gamma_1+\gamma_5)}{(\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2}}, C_{i,2} = G_i^{\gamma_4\delta_{i,4}} \hat{C}^{\frac{r_i\gamma_4}{(\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2}}$,
$C_{i,3} = g^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_{i,4}} \hat{C}, C_{i,4} = u^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_{i,4}} F_i^{\eta_i} u^{s'}$,
$C_{i,5} = v^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_{i,4}} \hat{C}', C_{i,6} = g^{\eta_i}, C_{i,7} = M \cdot E_i^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_{i,4}} \hat{C}_0 e(g^{\alpha'_i}, \hat{C})$.
For each column $j \in [m]$, it creates column ciphertexts $(C_{j,1}, C_{j,2})$ as follows: if $(j \geqslant y) \wedge (j \in RL_{j'})$: $C_{j,1} = B_j^{\sigma_1} g^{\gamma_2\mu_j}, C_{j,2} = B_j^{\sigma_2} g^{-\gamma_1\mu_j}$; if $(j \geqslant y) \wedge (j \notin RL_{j'})$: $C_{j,1} = B_j^{\sigma_1}, C_{j,2} = B_j^{\sigma_2}$.

For each $k \in [l]$, it sets policy ciphertexts $(D_k, D'_k)$ as follows: $D_k = u^{A_k \cdot \vec{\vartheta}'} H^{-\theta'_k}_{\rho(k)} / \hat{C}_k, D'_k = g^{\theta'_k} / \hat{C}'_k$.

Note that $\{\delta_i\}_{i\in[m]}, \{\theta_k\}_{k\in[l]}$ and $\vec{\vartheta} = (s, \vartheta_2, ..., \vartheta_n)$ are implicitly chosen such that $\{\delta_{i,1} + \frac{\hat{s}}{(\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2} \equiv \delta_i \bmod p_1\}_{i\in[m]\setminus RL_{i'}}, \{\delta_{i,2}+\frac{\hat{s}}{\sigma_1\gamma_1+\sigma_2\gamma_2} \equiv \delta_i \bmod p_1\}_{i\in[m]\setminus RL_{i'}}, \{\delta_{i,3}+\frac{\hat{s}}{\sigma_1\gamma_3+\sigma_2\gamma_4} \equiv \delta_i \bmod p_1\}_{i\in RL_{i'}}, \{\delta_{i,4}+\frac{\hat{s}}{(\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2} \equiv \delta_i \bmod p_1\}_{i\in RL_{i'}}, \{\theta'_k - \hat{\theta}_k \equiv \theta_k \bmod p_1\}_{k\in[l]}, s' - \hat{s} \equiv s \bmod p_1, \{\vartheta'_d - \hat{\vartheta}_d \equiv \vartheta_d \bmod p_1\}_{d\in\{2,3,...,n\}}$. $\mathcal{B}$ returns $ct$ to $\mathcal{A}$.

- **Phase 2**: This phase is the same with Phase 1.
- **Guess**: $\mathcal{A}$ outputs his guess $\beta'$, and gives it to $\mathcal{B}$. $\mathcal{B}$ sends $\beta'$ to $\Sigma_{cpabe}$.

Since the distributions of the public parameter, decryption keys and challenge ciphertext in the above game are the same as that in the real system, we have $Adv_{\mathcal{B}}\Sigma_{cpabe} = Adv_{\mathcal{A}}\Sigma_{ecpabe}$.

**Theorem 2.** If Assumption 1, the general subgroup decision assumption, the 3-party Diffie-Hellman assumption in a subgroup, and the source group q-parallel BDHE assumption in a subgroup hold, no polynomial time adversary can achieve a non-negligible advantage in winning $Game_{MH_1}$.

*Proof.* It follows directly from Lemma 1 and Lemma 2.

**Proof of message-hiding security in $Game_{MH_{\mathcal{N}+1}}$:**

**Theorem 3.** No PPT $\mathcal{A}$ can achieve a non-negligible advantage in winning $Game_{MH_{\mathcal{N}+1}}$ for $\Sigma_{ecpabe}$ .

*Proof.* Since an encryption to index $\mathcal{N}+1$ contains no information about the message, the argument for message-hiding in $Game_{MH_{\mathcal{N}+1}}$ is straightforward. The simulator simply runs $\texttt{Setup}_E$ and $\texttt{KeyGen}_E$ and encrypts message $M_\beta$ under $(A^*, \rho^*)$, $RL^*$ and index $(m+1, 1)$. Obviously, $C_{i,7} = E_i^{\tau_{i,4}}$ contains no information about the message. Thus, the bit $\beta$ is perfectly hidden and $Adv_{\mathcal{A}}\Sigma_{ecpabe} = 0$ in $Game_{MH_{\mathcal{N}+1}}$.

## Appendix C.6 Index-hiding Security

**Theorem 4.** If the 3-party Diffie-Hellman assumption and the XDH assumption in a subgroup hold, no PPT adversary can achieve a non-negligible advantage in winning $Game_{IH}$.

*Proof.* This theorem follows from the following Lemma 3 and Lemma 4 immediately.

**Lemma 3.** If the 3-party Diffie-Hellman assumption in a subgroup holds, no PPT adversary can achieve a non-negligible advantage in distinguishing between an encryption to $(x, y)$ and $(x, y+1)$ in $Game_{IH}$.

*Proof.* In $Game_{IH}$ with index $(x, y)$ and challenge (access policy, revocation list) tuple $((A^*, \rho^*), RL^*)$, the restriction is that the adversary $\mathcal{A}$ does not query a secret key for (index, attribute set) tuple $((i, j), S_{i,j})$ such that $(i, j) = (x, y) \wedge (S_{i,j}\ satisfies\ (A^*, \rho^*)) \wedge ((i, j) \in [m, m] \setminus RL^*)$. $\mathcal{A}$ eventually behaves in one of two different ways under the above restriction: **Case 1:** In Phase 1 and Phase 2, $\mathcal{A}$ will not query a secret key with index $(x, y)$ for attribute set $S_{(x,y)}$; **Case 2:** In Phase 1 and Phase 2, $\mathcal{A}$ queries a secret key with index $(x, y)$ for attribute set $S_{(x,y)}$ with the restriction that $S_{(x,y)}$ does not satisfy the corresponding strictest access $\mathbb{A}_{S^*}$, where $\mathbb{A}_{S^*}$ is submitted by $\mathcal{A}$ during Challenge phase.

Suppose there exists a PPT adversary $\mathcal{A}$ achieves a non-negligible advantage $Adv_{\mathcal{A}}$ in winning $Game_{IH}$. We construct a PPT algorithm $\mathcal{B}$ that achieves a non-negligible advantage $Adv_{\mathcal{B}}$ to solve a 3-Party Diffie-Hellman problem instance in a subgroup. On input $(GD = (N, G, G_T, e), g, g_2, g_3, g^a, g^b, g^c, T)$, the goal of $\mathcal{B}$ is to determine $T = g^{abc}$ or $T$ is a random element in $G_{p_1}$, where $a, b, c \in \mathbb{Z}_N$, $G$ is a bilinear group of order $N = p_1p_2p_3$, $g, g_2, g_3$ are generators of $G_{p_1}, G_{p_2}, G_{p_3}$ respectively.

- **Setup** : $\mathcal{B}$ first randomly chooses an attribute $\hat{h} \in \mathcal{U}$ to guess whether $\hat{h}$ will be in the challenge attribute set $S^*$. It then randomly chooses $\{\alpha_i\}_{i\in[m]}, \{r_i, f'_i\}_{i\in[m]\setminus\{x\}}, \{b_j\}_{j\in[m]\setminus\{y\}}, \{h_k\}_{k\in\mathcal{U}\setminus\{\hat{h}\}} \in \mathbb{Z}_N, r'_x, f_x, b'_y, h'_{\hat{h}}, \hat{u}, \hat{v} \in \mathbb{Z}_N$. It sets $(GD, g, u = (g^c)^{\hat{u}}, v = g^{\hat{v}}, \{E_i = e(g, g)^{\alpha_i}\}_{i\in[m]}, \{G_i = g^{r_i}, F_i = (g^c)^{f'_i}\}_{i\in[m]\setminus\{x\}}, G_x = (g^b)^{r'_x}, F_x = g^{f_x}, \{B_j = g^{b_j}\}_{j\in[m]\setminus\{y\}}, B_y = (g^c)^{b'_y}, \{H_k = g^{h_k}\}_{k\in\mathcal{U}\setminus\{\hat{h}\}}, H_{\hat{h}} = (g^c)^{h'_{\hat{h}}})$ as the public parameter $pp$ and sends it to $\mathcal{A}$. Note that $r_x, b_y, h_{\hat{h}}$ and $\{f_i\}_{i\in[m,m]\setminus\{x\}}$ are implicitly chosen such that $br'_x \equiv r_x \bmod p_1, b'_y/c \equiv b_y \bmod p_1, h'_{\hat{h}}/c \equiv h_{\hat{h}} \bmod p_1, \{cf'_i \equiv f_i \bmod p_1\}_{i\in[m]\setminus\{x\}}$.
- **KeyQuery** : To respond to the key query of $\mathcal{A}$ for $((i, j), S_{(i,j)})$,
- if $(i, j) \neq (x, y)$: $\mathcal{B}$ randomly chooses $t_{i,j}, s_{i,j} \in \mathbb{Z}_N$ and $R, R', R'', R''', \{R_k\}_{k\in S_{i,j}} \in G_{p_3}$. The private key $sk_{(x,y),S_{(x,y)}}$ is set as follows: $\langle K_{i,j}, K'_{i,j} = g^{s_{i,j}}R', K''_{i,j} = g^{t_{i,j}}R'', K'''_{i,j} = F_i^{t_{i,j}}R''', \{K_{i,j,k} = H_k^{t_{i,j}}R_k\}_{k\in S_{i,j}}\rangle$, where $K_{i,j}$ is given by $K_{i,j} = g^{\alpha_i}g^{r_ib_j}u^{t_{i,j}}v^{s_{i,j}}R$ : if $i \neq x, j \neq y$; $K_{i,j} = g^{\alpha_i}(g^b)^{r'_ib_j}u^{t_{i,j}}v^{s_{i,j}}R$ : if $i = x, j \neq y$; $K_{i,j} = g^{\alpha_i}(g^c)^{r_ib'_j}u^{t_{i,j}}v^{s_{i,j}}R$ : if $i \neq x, j = y$.
- if $(i, j) = (x, y)$: it implies $\mathcal{A}$ behaves in **Case 2**. If $\hat{h} \in S_{i,j}$, $\mathcal{B}$ aborts and outputs a random $\beta' \in \{0, 1\}$ to the challenger. Otherwise, $\mathcal{B}$ randomly chooses $t'_{x,y} \in \mathbb{Z}_N$ and sets $t_{x,y}$ by implicitly setting $t'_{x,y} - br'_xb'_y/\hat{u} \equiv t_{x,y} \bmod p_1$. It then randomly chooses $s_{x,y} \in \mathbb{Z}_N$ and $R, R', R'', R''', \{R_k\}_{k\in S_{i,j}} \in G_{p_3}$. The private key $sk_{(x,y),S_{(x,y)}}$ is set as follows: $\langle K_{x,y} = g^{\alpha_x}u^{t'_{x,y}}v^{s_{x,y}}R, K'_{x,y} = g^{s_{x,y}}R', K''_{x,y} = g^{t'_{x,y}}(g^b)^{-r'_xb'_y/\hat{u}}R'', K'''_{x,y} = (g^{t'_{x,y}}(g^b)^{-r'_xb'_y/\hat{u}})^{f_x}R''', \{K_{i,j,k} = (g^{t'_{x,y}}(g^b)^{-r'_xb'_y/\hat{u}})^{h_x}R_k\}_{k\in S_{i,j}}\rangle$
- **Challenge** : $\mathcal{A}$ submits a message $M$, an attribute set $S^*$, and a revocation list $RL^*$. Let $RL^*_{i'}, RL^*_{j'}$ be the sets of revoked row index and column index, respectively. If $\hat{h} \notin S^*$, $\mathcal{B}$ aborts and outputs a random $\beta' \in \{0, 1\}$ to the challenger. Otherwise, $\mathcal{B}$ constructs the LSSS matrix $(A^*, \rho^*)$ ( with size $l \times n$) for $\mathbb{A}_{S^*}$. Note that the attribute set $S^* \setminus \{\hat{h}\}$ does not satisfy $\mathbb{A}_{S^*}$. $\mathcal{B}$ chooses a vector $\vec{\vartheta} \in \mathbb{Z}_N^n$ that is orthogonal to all of the rows $A^*_k$ of $A^*$ such that $\rho(k) \in S^* \setminus \{\hat{h}\}$ [2)] and has first entry equal to 1. It then randomly chooses $s' \in \mathbb{Z}_N$ and $\vec{\vartheta}'$ with the first entry equal to

2) Such a vector must exist since $S^* \setminus \{\hat{h}\}$ fails to satisfy $(A^*, \rho^*)$, and it is efficiently computable.

zero, $\sigma_1, \sigma_2, \{\gamma_d\}_{d\in[6]}, \{\delta_i\}_{i\in[x-1]}, \delta'_x, \{\delta_i\}_{i\in[x+1,m]}, \{\eta'_i\}_{i\in[x-1]}$,
$\eta_x, \{\eta'_i\}_{i\in[x+1,m]}, \{\mu_j\}_{j\in[y-1]}, \{\theta'_k\}_{\forall k\in[l]\ s.t.\ \rho(k)=\hat{h}}, \{\theta_k\}_{k\in[l]\ s.t.\ \rho(k)\neq\hat{h}}, \{\tau_{i,d}\}_{i\in[x-1],d\in[4]}, \{b'_j\}_{j\in[m]\setminus\{y\}} \in \mathbb{Z}_N$ under constraints that $\gamma_2\gamma_3 - \gamma_1\gamma_4 \neq 0$, $\gamma_1\gamma_6 - \gamma_2\gamma_5 = 0$, $(\gamma_1+\gamma_5)\gamma_4 - (\gamma_2+\gamma_6)\gamma_3 = 0$ and $\sigma_1\gamma_3 + \sigma_2\gamma_4 = 0$.
For each row $i \in [m]$, the algorithm creates row ciphertexts $(C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}, C_{i,6}, C_{i,7})$ as follows:

- if $i < x$: $C_{i,1} = g^{\tau_{i,1}}, C_{i,2} = g^{\tau_{i,2}}, C_{i,3} = g^{\tau_{i,3}}, C_{i,4} = u^{\tau_{i,3}} F_i^{\eta'_i} u^{s'}, C_{i,5} = v^{\tau_{i,3}}$,
$C_{i,6} = g^{\eta'_i}(g^a)^{\frac{\hat{u}(\gamma_2\gamma_3-\gamma_1\gamma_4)\delta'_x}{b_y f'_i}}, C_{i,7} = E_i^{\tau_{i,4}}$.
- if $(i = x) \wedge (i \in RL_{i'})$: $C_{i,1} = g^{r'_x(\gamma_3+\gamma_5)\delta'_x}, C_{i,2} = g^{r'_x(\gamma_4+\gamma_6)\delta'_x}, C_{i,3} = (g^b)^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta'_x/b_y}$,
$C_{i,4} = (g^b)^{\hat{u}((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta'_x/b_y} F_x^{\eta_x} u^{s'}, C_{i,5} = (C_{i,3})^{\hat{v}}, C_{i,6} = g^{\eta_x}, C_{i,7} = M\cdot e(g^b, g)^{\alpha_x((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta'_x/b_y}$.
- if $(i = x) \wedge (i \notin RL_{i'})$: $C_{i,1} = g^{r'_x\gamma_1\delta'_x}, C_{i,2} = g^{r'_x\gamma_2\delta'_x}, C_{i,3} = (g^b)^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta'_x/b_y}$,
$C_{i,4} = (g^b)^{\hat{u}(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta'_x/b_y} F_x^{\eta_x} u^{s'}, C_{i,5} = (C_{i,3})^{\hat{v}}, C_{i,6} = g^{\eta_x}, C_{i,7} = M\cdot e(g^b, g)^{\alpha_x(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta'_x/b_y}$.
- if $(i > x) \wedge (i \in RL_{i'})$: $C_{i,1} = (g^b)^{r_i\gamma_3\delta_i}, C_{i,2} = (g^b)^{r_i\gamma_4\delta_i}, C_{i,3} = (g^b)^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}, C_{i,4} = F_i^{\eta_i} u^s$,
$C_{i,5} = (C_{i,3})^{\hat{v}}, C_{i,6} = g^{\eta'_i}(g^b)^{\frac{-\hat{u}(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}{f'_i}}, C_{i,7} = M\cdot e(g^b, g)^{\alpha_i(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}$.
- if $(i > x) \wedge (i \notin RL_{i'})$: $C_{i,1} = (g^b)^{r_i(\gamma_1+\gamma_5)\delta_i}, C_{i,2} = (g^b)^{r_i(\gamma_2+\gamma_6)\delta_i}, C_{i,3} = (g^b)^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i}$,
$C_{i,4} = F_i^{\eta'_i} u^{s'}, C_{i,5} = (C_{i,3})^{\hat{v}}, C_{i,6} = g^{\eta'_i}(g^b)^{\frac{-\hat{u}((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i}{f'_i}}, C_{i,7} = M\cdot e(g^b, g)^{\alpha_i((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i}$.

For each column $j \in [m]$, the algorithm creates column ciphertexts $(C_{j,1}, C_{j,2})$ as follows:

- if $(j < y) \wedge (j \in RL_{j'})$: $C_{j,1} = g^{b'_j(\gamma_2+\gamma_4)/b_y} g^{b_j\sigma_1} g^{\gamma_4\mu_j} g^{\gamma_2\mu_j}, C_{j,2} = g^{-b'_j(\gamma_1+\gamma_3)/b_y} g^{b_j\sigma_2} g^{-\gamma_3\mu_j} g^{-\gamma_1\mu_j}$.
- if $(j < y) \wedge (j \notin RL_{j'})$: $C_{j,1} = g^{b'_j\gamma_4/b_y} g^{b_j\sigma_1} g^{\gamma_4\mu_j}, C_{j,2} = g^{-b'_j\gamma_3/b_y} g^{b_j\sigma_1} g^{-\gamma_3\mu_j}$.
- if $(j = y) \wedge (j \in RL_{j'})$: $C_{j,1} = (T)^{\gamma_4}(g^c)^{b'_y\sigma_1}(g^c)^{\gamma_2\mu_j}, C_{j,2} = (T)^{-\gamma_3}(g^c)^{b'_y\sigma_2}(g^c)^{-\gamma_1\mu_j}$.
- if $(j = y) \wedge (j \notin RL_{j'})$: $C_{j,1} = (T)^{\gamma_4}(g^c)^{b'_y\sigma_1}, C_{j,2} = (T)^{-\gamma_3}(g^c)^{b'_y\sigma_2}$.
- if $(j > y) \wedge (j \in RL_{j'})$: $C_{j,1} = g^{b'_j\gamma_4/b_y} g^{b_j\sigma_1}(g^c)^{\gamma_2\mu_j}, C_{j,2} = g^{-b'_j\gamma_3/b_y} g^{b_j\sigma_2}(g^c)^{-\gamma_1\mu_j}$.
- if $(j > y) \wedge (j \notin RL_{j'})$: $C_{j,1} = g^{b'_j\gamma_4/b_y} g^{b_j\sigma_1}, C_{j,2} = g^{-b'_j\gamma_3/b_y} g^{b_j\sigma_2}$.

For each $k \in [l]$, the algorithm sets policy ciphertexts $(D_k, D'_k)$ as follows: if $\rho(k) \neq \hat{h}$: $D_k = u^{A^*_k\cdot\vec{\vartheta}'} H_{\rho(k)}^{-\theta_k}, D'_k = g^{\theta_k}$; if $\rho(k) = \hat{h}$: $D_k = u^{s'A^*_k\cdot\bar{\vec{\vartheta}}} u^{A^*_k\cdot\vec{\vartheta}'}(g^c)^{-h_{\hat{h}}\theta'_k}, D'_k = g^{\theta'_k}(g^a)^{-\hat{u}(\gamma_2\gamma_3-\gamma_1\gamma_4)\delta'_x(A^*_k\bar{\vec{\vartheta}})/(b_y h_{\hat{h}})}$.
$\{b_j\}_{j\in[m]\setminus\{y\}}, s, \{\eta_i\}_{i\in[m]\setminus\{x\}}, \vec{\vartheta}, \{\theta_k\}_{k\in[l]\ s.t.\ \rho(k)=\hat{h}}$ are implicitly chosen such that $\{b'_j \equiv abb_j\ mod\ p_1\}_{j\in[m]\setminus\{y\}}$, $\forall i \in [x-1] : \eta'_i + \frac{\hat{u}a(\gamma_2\gamma_3-\gamma_1\gamma_4)\delta'_x}{b_y f'_i} \equiv \eta_i\ mod\ p_1$, $\forall i \in [x+1,m] \wedge (i \in RL_{i'}) : \eta'_i - \frac{\hat{u}b(\sigma_1\gamma_3+\sigma_2\gamma_4)\delta_i}{f'_i} \equiv \eta_i\ mod\ p_1$, $\forall i \in [x+1,m] \wedge (i \notin RL_{i'}) : \eta'_i - \frac{\hat{u}b((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i}{f'_i} \equiv \eta_i\ mod\ p_1$, $\forall j \in [y+1,m] : \vec{\vartheta} = s\bar{\vec{\vartheta}} + \vec{\vartheta}'$, $\forall k \in [l]\ s.t.\ \rho(k) = \hat{h} : \theta'_k - a\hat{u}(\gamma_2\gamma_3-\gamma_1\gamma_4)\delta'_x(A^*_k\bar{\vec{\vartheta}})/(b_y h_{\hat{h}}) \equiv \theta_k\ mod\ p_1$. If $T = g^{abc}$, then $ct$ is an encryption to the index $(x, y)$; and if $T$ is random, then $ct$ is an encryption to the index $(x, y+1)$.

- **Guess** : $\mathcal{A}$ sends a guess $\beta' \in \{0,1\}$ to $\mathcal{B}$. $\mathcal{B}$ sends this $\beta'$ to the challenger.

Note that the distributions of the public parameter, decryption keys and challenge ciphertext are same as the real system when $\mathcal{B}$ does not abort. Also note that since $S^* \neq \varnothing$ and if $\mathcal{A}$ behaves in **Case 2** then $S_{(x,y)}$ must satisfy $S^*\backslash S_{(x,y)} \neq \varnothing$, $\mathcal{B}$ does not abort happens with probability at least $1/|\mathcal{U}|$. Therefore, the advantage of $\mathcal{B}$ in breaking the 3-Party Diffie-Hellman problem will be at least $Adv_{\mathcal{A}}/|\mathcal{U}|$.

**Lemma 4.** If the 3-party Diffie-Hellman assumption and the XDH assumption in a subgroup hold, no PPT adversary can achieve a non-negligible advantage in distinguishing between an encryption to $(x, m)$ and $(x+1, 1)$ in $Game_{IH}$.

*Proof.* We refer to rows with ciphertexts generated with random exponents as "less-than" rows, rows with ciphertexts involved with exponents $\gamma_3+\gamma_5, \gamma_4+\gamma_6, \gamma_1, \gamma_2$ as "target" rows, and rows with ciphertexts involved with $\gamma_3, \gamma_4, \gamma_1+\gamma_5, \gamma_2+\gamma_6$ as "greater-than" rows. And by "Encrypt to column $y$" we denote that column ciphertexts $(C_{j,1}, C_{j,2})$ for all $j \geqslant y$. To prove this lemma, we define the following hybrid experiments: $\Lambda_1$ : Encrypt to column $m$, row $x$ is the target row, row $i+1$ is the greater-than row; $\Lambda_2$ : Encrypt to column $m+1$, row $x$ is the target row, row $x+1$ is the greater-than row; $\Lambda_3$ : Encrypt to column $m+1$, row $x$ is the less-than row, row $x+1$ is the greater-than row; $\Lambda_4$ : Encrypt to column 1, row $x$ is the less-than row, row $x+1$ is the greater-than row; $\Lambda_5$ : Encrypt to column 1, row $x$ is the less-than row, row $x+1$ is the target row.

**Claim 1.** If the 3-party Diffie-Hellman assumption in a subgroup holds, no PPT adversary can achieve a non-negligible advantage in distinguishing between experiments $\Lambda_1$ and $\Lambda_2$.

*Proof.* The proof is almost identical to that of Lemma 3.

**Claim 2.** If the 3-party Diffie-Hellman assumption in a subgroup holds, no PPT adversary can achieve a non-negligible advantage in distinguishing between experiments $\Lambda_2$ and $\Lambda_3$.

*Proof.* Suppose there exists a PPT adversary $\mathcal{A}$ achieves a non-negligible advantage in distinguish between experiments $\Lambda_2$ and $\Lambda_3$. We construct a PPT algorithm $\mathcal{B}$ that achieves a non-negligible advantage in solving the 3-party Diffie-Hellman assumption in a subgroup. On input $(GD = (N, G, G_T, e), g, g_2, g_3, g^a$,
$g^b, g^c, T)$, the goal of $\mathcal{B}$ is to determine $T = g^{abc}$ or $T$ is a random element in $G_{p_1}$.

- **Setup**: $\mathcal{B}$ randomly chooses $\{\alpha_i, r_i\}_{i\in[m]\setminus\{x\}}, \alpha_x = ac, r_x = a, \{f_i\}_{i\in[m]}, \{b_j\}_{j\in[m]}, \hat{u}, \hat{v} \in \mathbb{Z}_N, \{h_k\}_{k\in\mathcal{U}} \in \mathbb{Z}_N$. It sets $(GD, g, u = g^{\hat{u}}, v = g^{\hat{v}}, \{E_i = e(g,g)^{\alpha_i}, G_i = g^{r_i}\}_{i\in[m]\setminus\{x\}}, E_i = e(g^a, g^c), G_i = g^a, \{F_i = g^{f_i}\}_{i\in[m]}, \{B_j = g^{b_j}(g^c)^{-1}\}_{j\in[m]}, \{H_k = g^{h_k}\}_{k\in\mathcal{U}})$ as $pp$ and sends it to $\mathcal{A}$.

• **Key Query**: To respond to $\mathcal{A}$'s query for $((i,j),S_{i,j})$, $\mathcal{B}$ randomly chooses $t_{i,j},s_{i,j}\in\mathbb{Z}_N$ and $R,R',R'',R''',\{R_k\}_{k\in S}\in G_{p_3}$. The private key $sk_{(i,j),S}$ is set as follows: $\langle K_{i,j},K'_{i,j}=g^{s_{i,j}}R',K''_{i,j}=g^{t_{i,j}}R'',K'''_{i,j}=F_i^{t_{i,j}}R''',\{K_{i,j,k}=H_k^{t_{i,j}}R_k\}_{k\in S_{i,j}}\rangle$, where $K_{i,j}=g^{\alpha_i}g^{r_ib_j}(g^c)^{-r_i}u^{t_{i,j}}v^{s_{i,j}}R:if\ i\neq x$; $K_{i,j}=(g^a)^{b_j}u^{t_{i,j}}v^{s_{i,j}}R:if\ i=x$.

• **Challenge**: $\mathcal{A}$ submits a message $M$, an attribute set $S^*$ and a revocation list $RL^*$. Let $RL^*_{i'},RL^*_{j'}$ be the sets of revoked row index and column index, respectively. $\mathcal{B}$ constructs the LSSS matrix $(A^*,\rho^*)$ for $\mathbb{A}^*$. For $A^*$ an $l\times n$ matrix, $\mathcal{B}$ randomly chooses $\vec{\vartheta}=(s,\vartheta_2,...,\vartheta_n)$. $\mathcal{B}$ then randomly chooses $\{\sigma_d\}_{d\in[2]},\{\gamma_d\}_{d\in[6]},\{\delta_i\}_{i\in[m]},\{\eta_i\}_{i\in[m]},\{\mu_j\}_{j\in[m]},\{\theta_k\}_{k\in[l]}$, $\{\tau_{i,d}\}_{i\in[x-1],d\in[4]}\in\mathbb{Z}_N$ under constraints that $\gamma_2\gamma_3-\gamma_1\gamma_4\neq 0$, $\gamma_1\gamma_6-\gamma_2\gamma_5=0$, $(\gamma_1+\gamma_5)\gamma_4-(\gamma_2+\gamma_6)\gamma_3=0$. The ciphertext $ct$ is set as follows:

For each row $i\in[m]$, the algorithm creates row ciphertexts $(C_{i,1},C_{i,2},C_{i,3},C_{i,4},C_{i,5},C_{i,6},C_{i,7})$ as follows:

• if $i<x$: $C_{i,1}=g^{\tau_{i,1}},C_{i,2}=g^{\tau_{i,2}},C_{i,3}=g^{\tau_{i,3}},C_{i,4}=u^{\tau_{i,3}}F_i^{\eta_i}u^s,C_{i,5}=v^{\tau_{i,3}},C_{i,6}=g^{\eta_i},C_{i,7}=E_i^{\tau_{i,4}}$.

• if $(i=x)\wedge(i\in RL_{i'})$:

$C_{i,1}=(g^a)^{(\gamma_3+\gamma_5)\delta_i},C_{i,2}=(g^a)^{(\gamma_4+\gamma_6)\delta_i},C_{i,3}=(g^b)^{((\gamma_3+\gamma_5)\gamma_2-(\gamma_4+\gamma_6)\gamma_1)\delta_i}g^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}$,
$C_{i,4}=(g^b)^{\hat{u}((\gamma_3+\gamma_5)\gamma_2-(\gamma_4+\gamma_6)\gamma_1)\delta_i}u^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}F_i^{\eta_i}u^s$,
$C_{i,5}=(g^b)^{\hat{v}((\gamma_3+\gamma_5)\gamma_2-(\gamma_4+\gamma_6)\gamma_1)\delta_i}v^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i},C_{i,6}=g^{\eta_i}$,
$C_{i,7}=M\cdot e(g,T)^{((\gamma_3+\gamma_5)\gamma_2-(\gamma_4+\gamma_6)\gamma_1)\delta_i}e(g^a,g^c)^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}$.

• if $(i=x)\wedge(i\notin RL_{i'})$: $C_{i,1}=(g^a)^{\gamma_1\delta_i},C_{i,2}=(g^a)^{\gamma_2\delta_i},C_{i,3}=(g^b)^{(\gamma_3\gamma_2-\gamma_4\gamma_1)\delta_i}g^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$,
$C_{i,4}=(g^b)^{\hat{u}(\gamma_3\gamma_2-\gamma_4\gamma_1)\delta_i}u^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}F_i^{\eta_i}u^s,C_{i,5}=(g^b)^{\hat{v}(\gamma_3\gamma_2-\gamma_4\gamma_1)\delta_i}v^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$,
$C_{i,6}=g^{\eta_i},C_{i,7}=Me(g,T)^{(\gamma_3\gamma_2-\gamma_4\gamma_1)\delta_i}e(g^a,g^c)^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$.

• if $(i>x)\wedge(i\in RL_{i'})$: $C_{i,1}=G_i^{\gamma_3\delta_i},C_{i,2}=G_i^{\gamma_4\delta_i},C_{i,3}=g^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i},C_{i,4}=u^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}F_i^{\eta_i}u^s$,
$C_{i,5}=v^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i},C_{i,6}=g^{\eta_i},C_{i,7}=M\cdot E_i^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}$.

• if $(i>x)\wedge(i\notin RL_{i'})$: $C_{i,1}=G_i^{(\gamma_1+\gamma_5)\delta_i},C_{i,2}=G_i^{(\gamma_2+\gamma_6)\delta_i},C_{i,3}=g^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i},C_{i,6}=g^{\eta_i}$,
$C_{i,4}=u^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i}F_i^{\eta_i}u^s,C_{i,5}=v^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i},C_{i,7}=M\cdot E_i^{((\gamma_1+\gamma_5)\sigma_1+(\gamma_2+\gamma_6)\sigma_2)\delta_i}$.

For each column $j\in[m]$, the algorithm creates column ciphertexts $(C_{j,1},C_{j,2})$ as follows: if $(j<y)\wedge(j\in RL_{j'})$: $C_{j,1}=g^{b_j\sigma_1}(g^c)^{-\sigma_1}g^{\gamma_4\mu_j}g^{\gamma_2\mu_j},C_{j,2}=g^{b_j\sigma_2}(g^c)^{-\sigma_2}g^{-\gamma_3\mu_j}g^{-\gamma_1\mu_j}$; if $(j<y)\wedge(j\notin RL_{j'})$: $C_{j,1}=g^{b_j\sigma_1}(g^c)^{-\sigma_1}g^{\gamma_4\mu_j},C_{j,2}=g^{b_j\sigma_2}(g^c)^{-\sigma_2}g^{-\gamma_3\mu_j}$.

For each $k\in[l]$, it sets policy ciphertexts $(D_k,D'_k)$ as follows: $D_k=u^{A_k\cdot\vec{\vartheta}}H_{\rho(k)}^{-\theta_k},D'_k=g^{\theta_k}$. $\mathcal{B}$ outputs $ct$.

• **Guess**: $\mathcal{A}$ sends a guess $\beta'\in\{0,1\}$ to $\mathcal{B}$. $\mathcal{B}$ sends this $\beta'$ to the challenger.

Note that $ct$ corresponding to row $x$ indicates the target low when $T=g^{abc}$, and $ct$ corresponding to row $x$ indicates the less-than row when $T$ is random. Thus, the advantage of $\mathcal{B}$ in the reduction is straightforwardly taken from the advantage of $\mathcal{A}$.

**Claim 3.** If the 3-party Diffie-Hellman assumption in a subgroup holds, no PPT adversary can achieve a non-negligible advantage in distinguishing between experiments $\Lambda_3$ and $\Lambda_4$.

*Proof.* We define hybrid experiments between experiments $\Lambda_3$ and $\Lambda_4$ to prove this claim. By $\Lambda_{3,m+1}(=\Lambda_3),\Lambda_{3,m},...,\Lambda_{3,1}(=\Lambda_4)$ we denote the hybrid games. In experiment $\Lambda_{3,\xi}$, it encrypts to column $\xi$. As in the proof of Lemma 3, it is sufficient to prove the indistinguishability of experiments $\Lambda_{3,\xi}$ and $\Lambda_{3,\xi+1}$ for $1\leqslant\xi\leqslant m$. The proof of this indistinguishability is almost identical to that of Lemma 3.

**Claim 4.** If the XDH assumption holds, no PPT adversary can achieve a non-negligible advantage in distinguishing between experiments $\Lambda_4$ and $\Lambda_5$.

*Proof.* Suppose there exists a PPT adversary $\mathcal{A}$ achieves a non-negligible advantage in distinguish between experiments $\Lambda_4$ and $\Lambda_5$. We construct a PPT algorithm $\mathcal{B}$ that achieves a non-negligible advantage in solving the XDH assumption. On input $(GD=(N,G,G_T,e),g,g_2,g_3,g^a,g^b,T)$, the goal of $\mathcal{B}$ is to determine $T=g^{ab}$ or $T$ is a random element in $G_{p_1}$.

• **Setup**: $\mathcal{B}$ chooses random $\{\alpha_i,r_i,f_i\}_{i\in[m]},\{b_j\}_{j\in[m]},\hat{u},\hat{v},\{h_k\}_{k\in\mathcal{U}}\in\mathbb{Z}_N$. It sets $(GD,g,u=g^{\hat{u}},v=g^{\hat{v}},\{E_i=e(g,g)^{\alpha_i},G_i=g^{r_i}\}_{i\in[m]},\{F_i=g^{f_i}\}_{i\in[m]},\{B_j=g^{b_j}\}_{j\in[m]},\{H_k=g^{h_k}\}_{k\in\mathcal{U}})$ as $pp$, sends it to $\mathcal{A}$.

• **Key Query**: To respond to $\mathcal{A}$'s query for $((i,j),S_{i,j})$, $\mathcal{B}$ randomly chooses $t_{i,j},s_{i,j}\in\mathbb{Z}_N$ and $R,R',R'',R''',\{R_k\}_{k\in S}\in G_{p_3}$. The private key $sk_{(i,j),S}$ is set as follows: $\langle K_{i,j}=g^{\alpha_i}g^{r_ib_j}u^{t_{i,j}}v^{s_{i,j}}R,K'_{i,j}=g^{s_{i,j}}R',K''_{i,j}=g^{t_{i,j}}R'',K'''_{i,j}=F_i^{t_{i,j}}R''',\{K_{i,j,k}=H_k^{t_{i,j}}R_k\}_{k\in S_{i,j}}\rangle$.

• **Challenge**: $\mathcal{A}$ submits a message $M$, an attribute set $S^*$ and a revocation list $RL^*$. Let $RL^*_{i'},RL^*_{j'}$ be the sets of revoked row index and column index, respectively. $\mathcal{B}$ constructs the LSSS matrix $(A^*,\rho^*)$ for $\mathbb{A}^*$. For $A^*$ an $l\times n$ matrix, $\mathcal{B}$ randomly chooses $\vec{\vartheta}=(s,\vartheta_2,...,\vartheta_n)$. $\mathcal{B}$ then randomly chooses $\{\sigma_d\}_{d\in[2]},\{\gamma_d\}_{d\in[4]},\{\delta_i\}_{i\in[m]},\{\eta_i\}_{i\in[m]},\{\mu_j\}_{j\in[m]},\{\theta_k\}_{k\in[l]}$, $\{\tau_{i,d}\}_{i\in[x-1],d\in[4]}\in\mathbb{Z}_N$ under constraints that $\gamma_2\gamma_3-\gamma_1\gamma_4\neq 0$, $\gamma_1\gamma_6-\gamma_2\gamma_5=0$, $(\gamma_1+\gamma_5)\gamma_4-(\gamma_2+\gamma_6)\gamma_3=0$. The ciphertext $ct$ is set as follows:

For each row $i\in[m]$, the algorithm creates row ciphertexts $(C_{i,1},C_{i,2},C_{i,3},C_{i,4},C_{i,5},C_{i,6},C_{i,7})$ as follows:

• if $i\leqslant x$: $C_{i,1}=g^{\tau_{i,1}},C_{i,2}=g^{\tau_{i,2}},C_{i,3}=g^{\tau_{i,3}},C_{i,4}=u^{\tau_{i,3}}F_i^{\eta_i}u^s,C_{i,5}=v^{\tau_{i,3}},C_{i,6}=g^{\eta_i},C_{i,7}=E_i^{\tau_{i,4}}$.

• if $(i=x+1)\wedge(i\in RL_{i'})$:

$C_{i,1}=T^{r_i\gamma_1\delta_i}(g^b)^{r_i(\gamma_3+\gamma_5)\delta_i},C_{i,2}=T^{r_i\gamma_2\delta_i}(g^b)^{r_i(\gamma_4+\gamma_6)\delta_i},C_{i,3}=T^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}(g^b)^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}$,
$C_{i,4}=(C_{i,3})^{\hat{u}}F_i^{\eta_i}u^s,C_{i,5}=(C_{i,3})^{\hat{v}},C_{i,6}=g^{\eta_i},C_{i,7}=Me(g,T)^{\alpha_i(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}\cdot e(g^b,g)^{\alpha_i((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}$.

• if $(i=x+1)\wedge(i\notin RL_{i'})$:

$C_{i,1}=T^{r_i\gamma_3\delta_i}(g^b)^{r_i\gamma_1\delta_i},C_{i,2}=T^{r_i\gamma_4\delta_i}(g^b)^{r_i\gamma_2\delta_i},C_{i,3}=T^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}(g^b)^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$,
$C_{i,4}=(C_{i,3})^{\hat{u}}F_i^{\eta_i}u^s,C_{i,5}=(C_{i,3})^{\hat{v}},C_{i,6}=g^{\eta_i},C_{i,7}=Me(g,T)^{\alpha_i(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i}e(g^b,g)^{\alpha_i(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$.

• if $(i>x+1)\wedge(i\in RL_{i'})$:

$C_{i,1} = (g^a)^{r_i\gamma_1\delta_i} g^{r_i(\gamma_3+\gamma_5)\delta_i}, C_{i,2} = (g^a)^{r_i\gamma_2\delta_i} g^{r_i(\gamma_4+\gamma_6)\delta_i}, C_{i,3} = (g^a)^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i} g^{((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}$,
$C_{i,4} = (C_{i,3})^{\hat{u}} F_i^{\eta_i} u^s, C_{i,5} = (C_{i,3})^{\hat{v}}, C_{i,6} = g^{\eta_i}, C_{i,7} = Me(g,g^a)^{\alpha_i(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i} e(g,g)^{\alpha_i((\gamma_3+\gamma_5)\sigma_1+(\gamma_4+\gamma_6)\sigma_2)\delta_i}$.

- if $(i > x) \wedge (i \notin RL_{i'})$: $C_{i,1} = (g^a)^{r_i\gamma_3\delta_i} g^{r_i\gamma_1\delta_i}, C_{i,2} = (g^a)^{r_i\gamma_4\delta_i} g^{r_i\gamma_2\delta_i}, C_{i,3} = (g^a)^{(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i} g^{(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$,
$C_{i,4} = (C_{i,3})^{\hat{u}} F_i^{\eta_i} u^s, C_{i,5} = (C_{i,3})^{\hat{v}}, C_{i,6} = g^{\eta_i}, C_{i,7} = Me(g,g^a)^{\alpha_i(\gamma_3\sigma_1+\gamma_4\sigma_2)\delta_i} e(g,g)^{\alpha_i(\gamma_1\sigma_1+\gamma_2\sigma_2)\delta_i}$.

For each column $j \in [m]$, the algorithm creates column ciphertexts $(C_{j,1}, C_{j,2})$ as follows: if $(j \geqslant y) \wedge (j \in RL_{j'})$: $C_{j,1} = B_j^{\sigma_1} g^{\gamma_2\mu_j}, C_{j,2} = B_j^{\sigma_2} g^{-\gamma_1\mu_j}$; if $(j \geqslant y) \wedge (j \notin RL_{j'})$: $C_{j,1} = B_j^{\sigma_1}, C_{j,2} = B_j^{\sigma_2}$.
$\mathcal{B}$ outputs $ct = \langle \{C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}, C_{i,6}, C_{i,7}\}_{i\in[m]}, \{C_{j,1}, C_{j,2}\}_{j\in[m]}, \{D_k, D'_k\}_{k\in[l]}, (A,\rho)\rangle$.

- **Guess**: $\mathcal{A}$ outputs a guess $\beta' \in \{0,1\}$, and $\mathcal{B}$ outputs the same value to the XDH challenger.

Note that $ct$ corresponds to $\Lambda_4$ when $T = g^{ab}$, and $ct$ corresponds to $\Lambda_5$ when $T$ is random. Thus, the advantage of $\mathcal{B}$ in the reduction is straightforwardly taken from the advantage of $\mathcal{A}$.

## References


1 Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *Advances in Cryptology-EUROCRYPT 2006*, pages 573–592. Springer, 2006.

2 Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 211–220. ACM, 2006.

3 Sanjam Garg, Abishek Kumarasubramanian, Amit Sahai, and Brent Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 121–130. ACM, 2010.

4 Allison Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *Advances in Cryptology–CRYPTO 2012*, pages 180–198. Springer, 2012.

5 Zhen Liu, Zhenfu Cao, and Duncan S Wong. Blackbox traceable cp-abe: how to catch people leaking their keys by selling decryption devices on ebay. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 475–486. ACM, 2013.