# Attribute Based Encryption: Traitor Tracing, Revocation and Fully Security on Prime Order Groups

Xiaoyi Li, Kaitai Liang, Zhen Liu, and Duncan S. Wong

Security and Data Sciences, ASTRI, Hong Kong SAR, China
School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, U.K.
xiaoyili@astri.org, k.liang@mmu.ac.uk, liu.zhen@my.cityu.edu.hk, duncanwong@astri.org

**Abstract.** A Ciphertext-Policy Attribute-Based Encryption (CP-ABE) allows users to specify the access policies without having to know the identities of users. In this paper, we contribute by proposing an ABE scheme which enables revoking corrupted users. Given a key-like blackbox, our system can identify at least one of the users whose key must have been used to construct the blackbox and can revoke the key from the system. This paper extends the work of Liu and Wong to achieve traitor revocability. We construct an Augmented Revocable CP-ABE (AugR-CP-ABE) scheme, and describe its security by message-hiding and index-hiding games. Then we prove that an AugR-CP-ABE scheme with message-hiding and index-hiding properties can be transferred to a secure Revocable CP-ABE with fully collusion-resistant blackbox traceability. In the proof for index-hiding, we divide the adversary's behaviors in two ways and build direct reductions that use adversary to solve the D3DH problem. Our scheme achieves the sub-linear overhead of $O(\sqrt{N})$, where $N$ is the number of users in the system. This scheme is highly expressive and can take any monotonic access structures as ciphertext policies.
Compared with the scheme in [16], our scheme achieves the same efficiency level, obtaining the sub-linear overhead of $O(\sqrt{N})$, where $N$ is the number of users in the system. This scheme is highly expressive and can take any monotonic access structures as ciphertext policies.

**Keywords**: Traitor Tracing, Revocation, Ciphertext-policy Attribute Based Encryption, Prime Order Groups

## 1 Introduction

Attribute-Based Encryption (ABE) system is first introduced by Sahai and Waters [22], which is based on users' roles and does not have to know their identities in the system. In an Attribute-Based Encryption (CP-ABE) system, each user possesses a set of attributes and a private key generated based on his/her attributes. The encrypting party will define an *access policy* over role-based/descriptive *attributes* to encrypt a message without having to know the identities of the targeted receivers. As a result, only the user who owns the appropriate attributes which satisfy the access policy are able to decrypt the ciphertext. Intuitively, Alice, for example, is to encrypt a message under "(Development Department AND (Manager OR Engineer))", which is an *access policy* defined over descriptive *attributes*, so that only those receivers who have their decryption keys associated with the attributes which satisfy this policy can decrypt it correctly.

Among the CP-ABE schemes recently proposed, [3,4,7,23,10,19,8,11,20], progress has been made with regard to the schemes' security, access policy expressivity, and efficiency. While the schemes with practical security and expressivity (i.e. full security against adaptive adversaries in the standard model and high expressivity of supporting any monotone access structures) have been proposed in [10,19,11], the traceability of traitors which intentionally expose their decryption keys has been becoming an important concern related to the applicability of CP-ABE. Assume in a communication system, the sender wants to assure that only those users who have paid for the service can access the content. This concern can be solved by encrypting the content and only receivers who own the legitimate keys can decrypt the content correctly. If we build such a system with ABE, however, due to the nature of

CP-ABE, the attributes (and the corresponding decryption privilege) are generally *shared* by multiple users. As a result, a malicious user, with his attributes shared with multiple other users, might have an intention to leak the corresponding decryption key or some decryption privilege in the form of a decryption blackbox/device in which the decryption key is embedded, for example, for financial gain or for some other incentives, as he only has little risk of getting caught.

Recently a handful of traceable CP-ABE schemes have been proposed in [14,13,5]. In the whitebox traceable CP-ABE schemes, given a well-formed decryption key as input, a tracing algorithm can find out the malicious user who leaked or sold well-formed decryption keys. Liu et al. [14] proposed such a whitebox traceable CP-ABE scheme that can deter users from these malicious behaviors. As malicious users invent a decryption blackbox/device which keeps the embedded decrypt keys and algorithms hidden, Liu et al.[13] proved that the blackbox traceable CP-ABE scheme supports fully collusion-resistant blackbox traceable in the standard model, where *fully collusion-resistant blackbox traceability* means that the number of colluding users in constructing a decryption blackbox/device is not limited and can be arbitrary. This scheme is fully secure in the standard model and highly expressive (i.e. supporting any monotonic access structures).

It should be observed that a tracing system is not designed to protect the encrypted content. It is used to distinct the compromised users from other legitimate users, which means the corrupted user/key is still remained in the system and an effective blackbox is likely to be produced with these corrupted keys in the wild market. The exposed compromised users need leave or be removed from the system to avoid incurring more losses. When any of these happens, the corresponding user keys should be revoked. We added the revocability in the scheme so that we can remove the compromised keys as needed. We focus on achieving direct revocation in traceable CP-ABE system. In a direct revocation mechanism, it does not need any periodic key updates and it does not affect any non-revoked users either. A system-wide revocation list could be made public and revocation could be taken into effect promptly as the revocation list could be updated immediately once a key is revoked. Specifically, we generate $Q'_i$, which is a part of ciphertext, with a non-revoked index list $\bar{R}$. When decrypting, we first recover $\bar{\boldsymbol{K}}_{i,j}$ which has a common item $h\prod_{j'\in\bar{R}_i} h_{j'}$ with $Q'_i$ if they share a consistent revocation list $R$. Then $\bar{\boldsymbol{K}}_{i,j}$ is used in the following decryption process. To avoid a further loss, the revocation list should be updated timely once corrupted users are found. For the security proof for message-hiding, we re-construct the Semi-functional Keys by replacing $h$ with $hh_j$, which can realize revocability, and adding the random item $\bar{\boldsymbol{K}}_{i,j,j'}$ accordingly. As a contrast, the random items for Semi-functional Ciphertexts remain the same, which is irrelevant to the revocability. For the security proof for index-hiding, we have two ways for adversary to take and add more sub-cases in **Case II** which make the security proof a non-trivial work. In this paper, We continue our work on prime order groups as an extension for [16].

### 1.1 Our results

It has been shown (e.g. in [6,9]) that the constructions on composite order groups will result in significant loss of efficiency and the security will rely on some non-standard assumptions (e.g. the Subgroup Decision Assumptions) and an additional assumption that the group order is hard to factor. The previous work in [16] achieves better security than the scheme in [13], which is constructed on composite order groups. In this paper, we add the revocability in [16] and prove it highly expressive and full secure in the standard model. On the efficiency aspect, this new scheme achieves the same efficient level as in [16], i.e. the overhead for the fully collusion-resistant blackbox traceability is in $O(\sqrt{N})$, $N$ is the number of users in a system.

Table 1 compares this new scheme with the previous work on blackbox traceable CP-ABE [13] and the traceable CP-ABE on prime order group but without revocability [16]. We only change the size of

| | Ciphertext Size | Private Key Size | Public Key Size | Pairing Computation in Decryption | On Prime Order Groups | Revocation | Order of the Groups |
|---|---|---|---|---|---|---|---|
| [13] | $2l + 17\sqrt{N}$ | $\|S\| + 4$ | $\|\mathcal{U}\| + 3 + 4\sqrt{N}$ | $2\|I\| + 10$ | $\times$ | $\times$ | $p_1 p_2 p_3$ |
| [16] | $6l + 3 + 46\sqrt{N}$ | $6\|S\| + 12$ | $24\|\mathcal{U}\| + 22 + 14\sqrt{N}$ | $6\|I\| + 30$ | $\surd$ | $\times$ | $p$ |
| this paper | $6l + 3 + 46\sqrt{N}$ | $6\|S\| + 9 + 3\sqrt{N}$ | $24\|\mathcal{U}\| + 22 + 23\sqrt{N}$ | $6\|I\| + 30$ | $\surd$ | $\surd$ | $p$ |

[1] Let $l$ be the size of an access policy, $|S|$ the size of the attribute set of a private key, $|\mathcal{U}|$ the size of the attribute universe, and $|I|$ the number of attributes in a decryption key that satisfies a ciphertext's access policy.

**Table 1.** Features and Efficiency Comparison

keypair as we need add revocation items in the key. Both the ciphertext and the pairing computation in decryption are kept unchanged. This implies both this new scheme and [16] have better security than the scheme in [13], although all of them are fully secure in the standard model and have overhead in $O(\sqrt{N})$.

*Related Work.* In the literature, several revocation mechanisms have been proposed in the context of CP-ABE. In [21], Sahai et al. proposed an *indirect* revocation mechanism, which requires an authority to periodically broadcast a key update information so that only the non-revoked users can update their keys and continue to decrypt messages. In [1], Attrapadung and Imai proposed a *direct* revocation mechanism, which allows a revocation list to be specified directly during encryption so that the resulting ciphertext cannot be decrypted by any decryption key which is in the revocation list even though the associated attribute set of the key satisfies the ciphertext policy. For ABE scheme, in [13] Liu et al. defined a 'functional' CP-ABE that has the same functionality as the conventional CP-ABE (i.e. having all the appealing properties of the conventional CP-ABE), except that each user is assigned and identified by a unique index, which will enable the traceability of traitors. Liu et al. also defined the security and the fully collusion-resistant blackbox traceability for such a 'functional' CP-ABE. Furthermore, Liu et al. defined a new primitive called Augmented CP-ABE (AugCP-ABE) and formalized its security using message-hiding and index-hiding games. Then Liu et al. proved that *an AugCP-ABE scheme with message-hiding and index-hiding properties can be directly transferred to a secure CP-ABE with fully collusion-resistant blackbox traceability.* With such a framework, Liu et al. obtained a fully secure and fully collusion-resistant blackbox traceable CP-ABE scheme by constructing an AugCP-ABE scheme with message-hiding and index-hiding properties. In [16], Liu et al. obtain a prime order construction and it will be tempting to bring the revocation into [16] as a practical enhancement and implementation. In this paper, we leverage the revocation idea from [15].

*Outline.* In this paper, we follow the same framework in [16]. In particular, in Section 2, we propose a definition for CP-ABE supporting key-like blackbox traceability and direct revocation. the definition is 'functional', namely each decryption key is uniquely indexed by $k \in \{1, \ldots, N\}$ and given a key-like decryption blackbox, the tracing algorithm *Trace* can return the index $k$ of a decryption key which has been used for building the decryption blackbox. In our direct revocation definition, the *Encrypt* algorithm takes a revocation list $R \subseteq \{1, \ldots, N\}$ as an additional input so that a message encrypted under the (revocation list, access policy) pair $(R, \mathbb{A})$ would only allow users whose (index, attribute set) pair $(k, S)$ satisfies $(k \in [N] \setminus R)$ $\mathsf{AND}$ ($S$ satisfies $\mathbb{A}$) to decrypt. In Section 3, we revisit the definitions and security models of Augmented Revocable CP-ABE (AugR-CP-ABE for short) from [15]. We refers to the 'functional' CP-ABE in Section 2 as Revocable CP-ABE (R-CP-ABE for short), then extend the R-CP-ABE to AugR-CP-ABE, which will lastly be transformed to a key-like blackbox *traceable* R-CP-ABE. More specifically, we define the encryption algorithm of AugR-CP-ABE as $\mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M, R, \mathbb{A}, \bar{k})$ which takes one more parameter $\bar{k} \in \{1, \ldots, N+1\}$ than the original one in R-CP-ABE. This also changes the decryption criteria in AugR-CP-ABE in such a way that an encrypted message can be recovered using a decryption key $\mathsf{SK}_{k,S}$, which is identified by index

$k \in \{1, \ldots, N\}$ and associated with an attribute set $S$, only if $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A}) \wedge (k \geq \bar{k})$. In Section 4 we propose our AugR-CP-ABE construction on prime order groups and prove that our AugR-CP-ABE construction is message-hiding and index-hiding in the standard model. As a result, we obtain a fully secure and fully collusion-resistant blackbox traceable R-CP-ABE scheme on prime order groups.

To construct the AugR-CP-ABE, we continue our work in [16] and leverage the revocation idea from [15]. In particular, besides achieving the important features for practicality, such as revocation, high expressivity and efficiency, the construction is proved secure and traceable in the standard model.

## 2 Revocable CP-ABE and Blackbox Traceability

We follow the definition in [15]. Given a positive integer $n$, our Revocable Ciphertext-Policy Attribute-Based Encryption (R-CP-ABE) system consists of four algorithms:

$\mathsf{Setup}(\lambda, \mathcal{U}, N) \rightarrow (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter $\lambda$, the attribute universe $\mathcal{U}$, and the number of users $N$ in the system, then runs in polynomial time in $\lambda$, and outputs the public parameter $\mathsf{PP}$ and a master secret key $\mathsf{MSK}$.

$\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, S) \rightarrow \mathsf{SK}_{k,S}$. The algorithm takes as input the public parameter $\mathsf{PP}$, the master secret key $\mathsf{MSK}$, and an attribute set $S$, and outputs a private decryption key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in [N]$.

$\mathsf{Encrypt}(\mathsf{PP}, M, R, \mathbb{A}) \rightarrow CT_{R,\mathbb{A}}$. The algorithm takes as input the public parameter $\mathsf{PP}$, a message $M$, a revocation list $R \subseteq [N]$, and an access policy $\mathbb{A}$ over $\mathcal{U}$, and outputs a ciphertext $CT_{R,\mathbb{A}}$ such that only users whose indices are not revoked by $R$ and attributes satisfy $\mathbb{A}$ can recover $M$. $R$ and $\mathbb{A}$ are implicitly included in $CT_{R,\mathbb{A}}$.

$\mathsf{Decrypt}(\mathsf{PP}, CT_{R,\mathbb{A}}, \mathsf{SK}_{k,S}) \rightarrow M$ or $\perp$. The algorithm takes as input the public parameter $\mathsf{PP}$, a ciphertext $CT_{R,\mathbb{A}}$, and a private key $\mathsf{SK}_{k,S}$. If $(k \in [N] \setminus R)$ $\mathsf{AND}$ ($S$ satisfies $\mathbb{A}$), the algorithm outputs a message $M$, otherwise it outputs $\perp$ indicating the failure of decryption.

**Correctness.** For any attribute set $S \subseteq \mathcal{U}$, index $k \in [N]$, revocation list $R \subseteq [N]$, access policy $\mathbb{A}$ over $\mathcal{U}$, and message $M$, suppose $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(\lambda, \mathcal{U}, N)$, $\mathsf{SK}_{k,S} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, S)$, $CT_{R,\mathbb{A}} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, M, R, \mathbb{A})$. If $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A})$, then $\mathsf{Decrypt}(\mathsf{PP}, CT_{R,\mathbb{A}}, \mathsf{SK}_{k,S}) = M$.

**Security.** Now we define the security of a R-CP-ABE system using a $\mathsf{message\text{-}hiding\ game}$.

$\mathsf{Game}_{\mathsf{MH}}$. The $\mathsf{Message\text{-}hiding\ game}$ is defined between a challenger and an adversary $\mathcal{A}$ as follows:

**Setup.** The challenger runs $\mathsf{Setup}(\lambda, \mathcal{U}, N)$ and gives the public parameter $\mathsf{PP}$ to $\mathcal{A}$.

**Phase 1.** For $i = 1$ to $Q_1$, $\mathcal{A}$ adaptively submits (index, attribute set) pair $(k_i, S_{k_i})$, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.

**Challenge.** $\mathcal{A}$ submits two equal-length messages $M_0, M_1$ and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0, 1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*)$ to $\mathcal{A}$.

**Phase 2.** For $i = Q_1 + 1$ to $Q$, $\mathcal{A}$ adaptively submits $(k_i, S_{k_i})$, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.

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

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried $\{(k_t, S_{k_t})\}_{i=1}^{Q}$ can satisfy $(k_t \in [N] \setminus R^*)$ $\mathsf{AND}$ ($S_{k_t}$ satisfies $\mathbb{A}^*$). The advantage of $\mathcal{A}$ is defined as $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

**Definition 1.** *An $N$-user R-CP-ABE system is secure if for all polynomial-time adversaries $\mathcal{A}$ the advantage* $\mathsf{MHAdv}_{\mathcal{A}}$ *is negligible in $\lambda$.*

The $\mathsf{message\text{-}hiding\ game}$ is a typical semantic security game and is based on that for conventional CP-ABE [10,11], where the revocation list $R$ is always empty. It is clear that such a CP-ABE system [10,11] has the following properties: fully collusion-resistant security, meaning that several users should not be able to decrypt a message that none of them are individually granted to access, fine-grained access control on encrypted data, and efficient one-to-many encryption.

It is worth noticing that, as pointed in [13], in the definition of the game: (1) the adversary is allowed to specify the index of the private key when it makes key queries for the attribute sets of its choice, i.e., for $t = 1$ to $Q$, the adversary submits (index, attribute set) pair $(k_t, S_{k_t})$ to query a private key for attribute set $S_{k_t}$, where $Q \leq N$, $k_t \in [N]$, and $k_t \neq k_{t'}$ $\forall 1 \leq t \neq t' \leq Q$ (this is to guarantee that each user/key can be *uniquely* identified by an index); and (2) for $k_t \neq k_{t'}$ we do not require $S_{k_t} \neq S_{k_{t'}}$, i.e., different users/keys may have the same attribute set. We remark that these two points apply to the rest of the paper.

### 2.1 Blackbox Traceability

Now we define the traceability against key-like decryption blackbox. A key-like decryption blackbox $\mathcal{D}$ can be viewed as a probabilistic circuit that takes as input a ciphertext $CT_{R,\mathbb{A}}$ and outputs a message $M$ or $\perp$, and such a decryption blackbox does not need to be perfect, namely, we only require it to be able to decrypt with non-negligible success probability. In particular, a key-like decryption blackbox $\mathcal{D}$ is described by a (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$ and a non-negligible probability value $\epsilon$ (i.e. $0 \leq \epsilon \leq 1$ is polynomially related to $\lambda$), and advertised that for any ciphertext generated under the (revocation list, access policy) pair $(R, \mathbb{A})$, if $((S_{\mathcal{D}}$ satisfies $\mathbb{A})$ $\mathsf{AND}$ $([N] \setminus R) \cap ([N] \setminus R_D) \neq \emptyset)$ can be satisfied by $S_{\mathcal{D}}$ and $R_{\mathcal{D}}$, this blackbox $\mathcal{D}$ can decrypt the corresponding ciphertext with probability at least $\epsilon$. Specifically, once a blackbox is found being able to decrypt ciphertext, we can regard it as a key-like decryption blackbox with the corresponding (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$, and the ciphertext is related to the pair $(R, \mathbb{A})$ which satisfies $((S_{\mathcal{D}}$ satisfies $\mathbb{A})$ $\mathsf{AND}$ $([N] \setminus R) \cap ([N] \setminus R_D) \neq \emptyset)$. If we set the revocation list $R$ and $R_{\mathcal{D}}$ as empty, we can get the same definition for key-like decryption blackbox as shown in [13].

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq [N]$. *This is an oracle algorithm that interacts with a key-like decryption blackbox $\mathcal{D}$. By given the public parameter $\mathsf{PP}$, a revocation list $R_{\mathcal{D}}$, a non-empty attribute set $S_{\mathcal{D}}$, and a probability value (lower-bound) $\epsilon$, the algorithm runs in time polynomial in $\lambda$ and $1/\epsilon$, and outputs an index set $\mathbb{K}_T \subseteq [N]$ which identifies the set of malicious users. Note that $\epsilon$ has to be polynomially related to $\lambda$.*

The following $\mathsf{Tracing\ Game}$ captures the notion of **fully collusion-resistant traceability**. In the game, the adversary targets to build a decryption blackbox $\mathcal{D}$ that functions as a private decryption key with the pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$ (as the name of key-like decryption blackbox implies) which can decrypt ciphertexts under some (revocation list, access policy) pairs $(R, \mathbb{A})$. The tracing algorithm, on the other side, is designed to extract the index of at least one of the malicious users whose decryption keys have been used for constructing $\mathcal{D}$.

$\mathsf{Game}_{\mathsf{TR}}$. The Tracing Game is defined between a challenger and an adversary $\mathcal{A}$ as follows:

**Setup.** The challenger runs $\mathsf{Setup}(\lambda, \mathcal{U}, N)$ and gives the public parameter $\mathsf{PP}$ to $\mathcal{A}$.

**Key Query.** For $i = 1$ to $Q$, $\mathcal{A}$ adaptively submits $(k_i, S_{k_i})$, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.

**(Key-like) Decryption Blackbox Generation.** $\mathcal{A}$ outputs a decryption blackbox $\mathcal{D}$ associated with a (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$, $S_{\mathcal{D}} \subseteq \mathcal{U}, R_{\mathcal{D}} \subseteq [N]$ and a non-negligible probability (lower-bound) value $\epsilon$.

**Tracing.** The challenger runs $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon)$ to obtain an index set $\mathbb{K}_T \subseteq [N]$.

Let $\mathbb{K}_{\mathcal{D}} = \{k_i | 1 \leq i \leq Q\}$ be the index set of keys corrupted by the adversary. We say that the adversary $\mathcal{A}$ wins the game if the following conditions hold:

1. For any (revocation list, access policy) pair $(R, \mathbb{A})$ which satisfied $((S_{\mathcal{D}}$ satisfies $\mathbb{A})$ $\mathsf{AND}$ $([N] \setminus R) \cap ([N] \setminus R_D) \neq \emptyset)$, we have
$$\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathsf{PP}, M, R, \mathbb{A})) = M] \geq \epsilon,$$
where the probability is taken over the random choices of message $M$ and the random coins of $\mathcal{D}$. A decryption blackbox satisfying this condition is said to be a *useful key-like decryption blackbox.*
2. $\mathbb{K}_T = \emptyset$, or $\mathbb{K}_T \not\subseteq \mathbb{K}_{\mathcal{D}}$, or $\big((k_t \in R_{\mathcal{D}})$ $\mathsf{OR}$ $(S_{\mathcal{D}} \not\subseteq S_{k_t})$ $\forall k_t \in \mathbb{K}_T\big)$.

We denote by $\mathsf{TRAdv}_{\mathcal{A}}$ the probability that adversary $\mathcal{A}$ wins this game.

**Definition 2.** *An $N$-user Blackbox Traceable CP-ABE system is traceable if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible in $\lambda$.*

## 3 Augmented R-CP-ABE Definitions

### 3.1 Definitions and Security Models

An Augmented R-CP-ABE (AugR-CP-ABE) system consists of the following four algorithms:

$\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N) \rightarrow (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter $\lambda$, the attribute universe $\mathcal{U}$, and the number of users $N$ in the system, then runs in polynomial time in $\lambda$, and outputs the public parameter $\mathsf{PP}$ and a master secret key $\mathsf{MSK}$.

$\mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S) \rightarrow \mathsf{SK}_{k,S}$. The algorithm takes as input $\mathsf{PP}$, $\mathsf{MSK}$, and an attribute set $S$, and outputs a private key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in [N]$.

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}, \bar{k}) \rightarrow CT_{R,\mathbb{A}}$. The algorithm takes as input $\mathsf{PP}$, a message $M$, a revocation list $R \subseteq [N]$, an access policy $\mathbb{A}$ over $\mathcal{U}$, and an index $\bar{k} \in [N+1]$, and outputs a ciphertext $CT_{R,\mathbb{A}}$. $\mathbb{A}$ **is included in $CT_{R,\mathbb{A}}$, but the value of $\bar{k}$ is not**.

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT_{R,\mathbb{A}}, \mathsf{SK}_{k,S}) \rightarrow M$ or $\perp$. The algorithm takes as input $\mathsf{PP}$, a ciphertext $CT_{R,\mathbb{A}}$, and a private key $\mathsf{SK}_{k,S}$. If $(k \in [N] \setminus R)$ $\mathsf{AND}$ ($S$ satisfies $\mathbb{A}$), the algorithm outputs a message $M$, otherwise it outputs $\perp$ indicating the failure of decryption.

**Correctness.** For any attribute set $S \subseteq \mathcal{U}$, index $k \in [N]$, revocation list $R \subseteq [N]$, access policy $\mathbb{A}$ over $\mathcal{U}$, encryption index $\bar{k} \in [N+1]$, and message $M$, suppose $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N)$, $\mathsf{SK}_{k,S} \leftarrow \mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S)$, $CT_{R,\mathbb{A}} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}, \bar{k})$. If $(k \in [N] \setminus R) \wedge (S$ satisfies $\mathbb{A}) \wedge (k \geq \bar{k})$ then $\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT_{R,\mathbb{A}}, \mathsf{SK}_{k,S}) = M$.

**Security.** The security of AugR-CP-ABE is defined by the following three games, where the first two are for message-hiding, and the third one is for the index-hiding property.

In the first two **message-hiding games** between a challenger and an adversary $\mathcal{A}$, $\bar{k} = 1$ (the first game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) or $\bar{k} = N+1$ (the second game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$).

**Setup.** The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N)$ and gives the public parameter $\mathsf{PP}$ to $\mathcal{A}$.

**Phase 1.** For $t = 1$ to $Q_1$, $\mathcal{A}$ adaptively submits (index, attribute set) pair $(k_t, S_{k_t})$, and the challenger responds with $\mathsf{SK}_{k_t, S_{k_t}}$, which corresponds to attribute set $S_{k_t}$ and is assigned index $k_t$.

**Challenge.** $\mathcal{A}$ submits two equal-length messages $M_0, M_1$ and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*, \bar{k})$ to $\mathcal{A}$.

**Phase 2.** For $t = Q_1 + 1$ to $Q$, $\mathcal{A}$ adaptively submits (index, attribute set) pair $(k_t, S_{k_t})$, and the challenger responds with $\mathsf{SK}_{k_t,S_{k_t}}$, which corresponds to attribute set $S_{k_t}$ and is assigned index $k_t$.
**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for $b$.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$. In the Challenge phase the challenger sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*, 1)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried $\{(k_t, S_{k_t})\}_{i=1}^{Q}$ can satisfy $(k \in [N] \setminus R^*)$ $\mathsf{AND}$ $(S_{k_t}$ satisfies $\mathbb{A}^*)$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$. In the Challenge phase the challenger sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*, N+1)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $b' = b$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MH}^{\mathsf{A}}_{N+1}\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

**Definition 3.** *A $N$-user Augmented R-CP-ABE system is message-hiding if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$ the advantages $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$ and $\mathsf{MH}^{\mathsf{A}}_{N+1}\mathsf{Adv}_{\mathcal{A}}$ are negligible in $\lambda$.*

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. In the third game, **index-hiding game**, for any non-empty attribute set $S^* \subseteq \mathcal{U}$, we define **the strictest access policy** as $\mathbb{A}_{S^*} = \bigwedge_{x \in S^*} x$, and require that an adversary cannot distinguish between an encryption using $(\mathbb{A}_{S^*}, R^*, \bar{k})$ and $(\mathbb{A}_{S^*}, R^*, \bar{k}+1)$ without a private decryption key $\mathsf{SK}_{\bar{k},S_{\bar{k}}}$ such that $(\bar{k} \in [N] \setminus R^*) \wedge (S_{\bar{k}} \supseteq S^*)$. The game takes as input a parameter $\bar{k} \in [N]$ which is given to both the challenger and the adversary $\mathcal{A}$. The game proceeds as follows:

**Setup.** The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N)$ and gives the public parameter $\mathsf{PP}$ to $\mathcal{A}$.
**Key Query.** For $t = 1$ to $Q$, $\mathcal{A}$ adaptively submits (index, attribute set) pair $(k_t, S_{k_t})$, and the challenger responds with $\mathsf{SK}_{k_t,S_{k_t}}$, which corresponds to attribute set $S_{k_t}$ and is assigned index $k_t$.
**Challenge.** $\mathcal{A}$ submits a message $M$ and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, \mathbb{A}^*, \bar{k} + b)$ to $\mathcal{A}$.
**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for $b$.

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried pairs $\{(k_t, S_{k_t})\}_{i=1}^{Q}$ can satisfy $(k_t = \bar{k}) \wedge (k_t \in [N] \setminus R^*) \wedge (S_{k_t}$ satisfies $\mathbb{A}_{S^*})$, i.e. $(k_t = \bar{k}) \wedge (k_t \in [N] \setminus R^*) \wedge (S_{k_t} \supseteq S^*)$. The advantage of $\mathcal{A}$ is defined as $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] = |\Pr[b' = b] - \frac{1}{2}|$.

**Definition 4.** *A $N$-user Augmented R-CP-ABE system is index-hiding if for all PPT adversaries $\mathcal{A}$ the advantages $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}]$ for $\bar{k} = 1, \ldots, N$ are negligible in $\lambda$.*

## 3.2 The Reduction of Traceable R-CP-ABE to Augmented R-CP-ABE

We now show that an AugR-CP-ABE with message-hiding and index-hiding implies a secure and traceable R-CP-ABE. Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}_{\mathsf{A}}, \mathsf{Decrypt}_{\mathsf{A}})$ be an AugR-CP-ABE with message-hiding and index-hiding, define $\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}) = \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}, 1)$, then $\Sigma = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}, \mathsf{Decrypt}_{\mathsf{A}})$ is a R-CP-ABE derived from $\Sigma_{\mathsf{A}}$. In the following, we show that if $\Sigma_{\mathsf{A}}$ is message-hiding and index-hiding, then $\Sigma$ is secure. Furthermore, we propose a tracing algorithm $\mathsf{Trace}$ for $\Sigma$ and show that if $\Sigma_{\mathsf{A}}$ is message-hiding and index-hiding, then $\Sigma$ (equipped with $\mathsf{Trace}$) is traceable.

### 3.2.1 R-CP-ABE Security

**Theorem 1.** *If $\Sigma_{\mathsf{A}}$ is an AugR-CP-ABE with message-hiding and index-hiding properties, then $\Sigma$ is a secure and traceable R-CP-ABE.*

*Proof.* Note that $\Sigma$ is a special case of $\Sigma_{\mathsf{A}}$ where the encryption algorithm always sets $\bar{k} = 1$. Hence, $\mathsf{Game}_{\mathsf{MH}}$ for $\Sigma$ is identical to $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ for $\Sigma_{\mathsf{A}}$, which implies that $\mathsf{MHAdv}_{\mathcal{A}}$ for $\Sigma$ in $\mathsf{Game}_{\mathsf{MH}}$ is equal to $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$ for $\Sigma_{\mathsf{A}}$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$, i.e., if $\Sigma_{\mathsf{A}}$ is message-hiding (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$), then $\Sigma$ is secure.

### 3.2.2 R-CP-ABE Traceability

Now we show that if $\Sigma_{\mathsf{A}}$ is message-hiding (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$) and index-hiding, $\Sigma$ is traceable. As shown in [13], with the following $\mathsf{Trace}$ algorithm [13], $\Sigma$ achieves fully collusion-resistant blackbox traceability against key-like decryption blackbox.

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon) \to \mathbb{K}_T \subseteq [N]$: Given a key-like decryption blackbox $\mathcal{D}$ associated with a non-empty attribute set $S_{\mathcal{D}}$ and probability $\epsilon > 0$, the tracing algorithm works as follows:

1. For $\bar{k} = 1$ to $N+1$, do the following:
   (a) The algorithm repeats the following $8\lambda(N/\epsilon)^2$ times:
      i. Sample $M$ from the message space at random.
      ii. Let $CT_{R,\mathbb{A}_{S_{\mathcal{D}}}} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}_{S_{\mathcal{D}}}, \bar{k})$, where $\mathbb{A}_{S_{\mathcal{D}}}$ is the strictest access policy of $S_{\mathcal{D}}$.
      iii. Query oracle $\mathcal{D}$ on input $CT_{R,\mathbb{A}_{S_{\mathcal{D}}}}$, and compare the output of $\mathcal{D}$ with $M$.
   (b) Let $\hat{p}_{\bar{k}}$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
2. Let $\mathbb{K}_T$ be the set of all $\bar{k} \in [N]$ for which $\hat{p}_{\bar{k}} - \hat{p}_{\bar{k}+1} \geq \epsilon/(4N)$. Then output $\mathbb{K}_T$ as the index set of the private keys of malicious users.

**Theorem 2.** *If $\Sigma_{\mathsf{A}}$ is message-hiding and index-hiding, then $\Sigma$ is traceable using the* $\mathsf{Trace}$ *algorithm against key-like decryption blackbox.*

*Proof.* In the proof sketch below, we show that if the key-like decryption blackbox output by the adversary is a useful one then the traced $\mathbb{K}_T$ will satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge \big(\exists k_t \in \mathbb{K}_T \ s.t.( \ k_{k_t} \in [N] \setminus R_{\mathcal{D}}) \wedge (S_{\mathcal{D}} \subseteq S_{k_t})\big)$ with overwhelming probability, which implies that the adversary can win the game $\mathsf{Game}_{\mathsf{TR}}$ only with negligible probability, i.e., $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible.

Let $\mathcal{D}$ be the key-like decryption blackbox output by the adversary, and $(R_{\mathcal{D}}, S_{\mathcal{D}})$ be the (revocation list, attribute set) pair which can be used to describe $\mathcal{D}$. Define

$$p_{\bar{k}} = \Pr[\mathcal{D}(\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}_{S_{\mathcal{D}}}, \bar{k})) = M],$$

where the probability is taken over the random choice of message $M$ and the random coins of $\mathcal{D}$. We have that $p_1 \geq \epsilon$ and $p_{N+1}$ is negligible. The former follows the fact that $\mathcal{D}$ is a useful key-like decryption blackbox, and the later follows that $\Sigma_{\mathsf{A}}$ is message-hiding (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$). Then there must exist some $\bar{k} \in [N]$ such that $p_{\bar{k}} - p_{\bar{k}+1} \geq \epsilon/(2N)$. By the Chernoff bound it follows that with overwhelming probability, $\hat{p}_{\bar{k}} - \hat{p}_{\bar{k}+1} \geq \epsilon/(4N)$. Hence, we have $\mathbb{K}_T \neq \emptyset$.

For any $k_t \in \mathbb{K}_T$ (i.e., $\hat{p}_{k_t} - \hat{p}_{k_t+1} \geq \frac{\epsilon}{4N}$), we know, by Chernoff, that with overwhelming probability $p_{k_t} - p_{k_t+1} \geq \epsilon/(8N)$. Clearly $(k_t \in \mathbb{K}_{\mathcal{D}}) \wedge (k_t \in [N] \setminus R_{\mathcal{D}}) \wedge (S_{\mathcal{D}} \subseteq S_{k_t})$ since otherwise, $\mathcal{D}$ can be directly used to win the index-hiding game for $\Sigma_A$. Hence, we have $(\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge \big((k_t \in [N] \setminus R_{\mathcal{D}}) \wedge (S_{\mathcal{D}} \subseteq S_{k_t}) \ \forall k_t \in \mathbb{K}_T\big)$.

## 4 An Augmented R-CP-ABE Construction on Prime Order Groups

Now we construct an AugR-CP-ABE scheme on prime order groups, and prove that this AugR-CP-ABE scheme is message-hiding and index-hiding in the standard model. Combined with the results in Section 3.2, we obtain a R-CP-ABE scheme that is fully collusion-resistant blackbox traceable in the standard model, fully secure in the standard model, and on prime order groups.

### 4.1 Preliminaries

Before proposing our AugR-CP-ABE construction , we first review some preliminaries.

**Bilinear Groups.** Let $\mathcal{G}$ be a group generator, which takes a security parameter $\lambda$ and outputs $(p, \mathbb{G}, \mathbb{G}_T, e)$ where $p$ is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $p$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that: (1) (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_p, e(g^a, h^b) = e(g, h)^{ab}$, (2) (Non-Degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order $p$ in $\mathbb{G}_T$. We refer to $\mathbb{G}$ as the *source group* and $\mathbb{G}_T$ as the *target group*. We assume that group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map $e$ are efficiently computable, and the description of $\mathbb{G}$ and $\mathbb{G}_T$ includes a generator of $\mathbb{G}$ and $\mathbb{G}_T$ respectively.

**Complexity Assumptions.** We will base the message-hiding property of our AugR-CP-ABE scheme on the Decisional Linear Assumption (DLIN), the Decisional 3-Party Diffie-Hellman Assumption (D3DH) and the Source Group $q$-Parallel BDHE Assumption, and will base the index-hiding property of our AugR-CP-ABE scheme on the DLIN assumption and the D3DH assumption. Note that the DLIN assumption and the D3DH assumption are standard and generally accepted assumptions, and the Source Group $q$-Parallel BDHE Assumption is introduced and proved by Lewko and Waters in [12]. Please refer to Appendix A for the details of the three assumptions.

**Dual Pairing Vector Spaces.** Our construction will use dual pairing vector spaces, a tool introduced by Okamoto and Takashima [17,18,19] and developed by Lewko [9] and Lewko and Waters [12]. Let $\boldsymbol{v} = (v_1, \ldots, v_n)$ be a vector over $\mathbb{Z}_p$, the notation $g^{\boldsymbol{v}}$ denotes a tuple of group elements as $g^{\boldsymbol{v}} := (g^{v_1}, \ldots, g^{v_n})$. Furthermore, for any $a \in \mathbb{Z}_p$ and $\boldsymbol{v} = (v_1, \ldots, v_n), \boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_p^n$, define

$$(g^{\boldsymbol{v}})^a := g^{a\boldsymbol{v}} = (g^{av_1}, \ldots, g^{av_n}), \quad g^{\boldsymbol{v}} g^{\boldsymbol{w}} := g^{\boldsymbol{v}+\boldsymbol{w}} = (g^{v_1+w_1}, \ldots, g^{v_n+w_n}),$$

and define a bilinear map $e_n$ on $n$-tuples of $\mathbb{G}$ as $e_n(g^{\boldsymbol{v}}, g^{\boldsymbol{w}}) := \prod_{i=1}^{n} e(g^{v_i}, g^{w_i}) = e(g, g)^{(\boldsymbol{v}\cdot\boldsymbol{w})}$, where the dot/inner product $\boldsymbol{v} \cdot \boldsymbol{w}$ is computed modulo $p$.

For a fixed (constant) dimension $n$, we say two bases $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ and $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*)$ of $\mathbb{Z}_p^n$ are "dual orthonormal" when

$$\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* \equiv 0 (\text{mod} p) \ \forall 1 \leq i \neq j \leq n, \quad \boldsymbol{b}_i \cdot \boldsymbol{b}_i^* \equiv \psi (\text{mod} p) \ \forall 1 \leq i \leq n,$$

where $\psi$ is a non-zero element of $\mathbb{Z}_p$. (This is a slight abuse of the terminology "orthonormal", since $\psi$ is not constrained to be 1.) For a generator $g \in \mathbb{G}$, we note that $e_n(g^{\boldsymbol{b}_i}, g^{\boldsymbol{b}_j^*}) = 1$ whenever $i \neq j$, where 1 here denotes the identity element in $\mathbb{G}_T$. Let $Dual(\mathbb{Z}_p^n, \psi)$ denote the set of pairs of dual orthonormal bases of dimension $n$ with dot products $\boldsymbol{b}_i \cdot \boldsymbol{b}_i^* = \psi$, and $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n, \psi)$ denote choosing a random pair of bases from this set. As our AugR-CP-ABE construction will use dual pairing vector spaces, the security proof will use a lemma and a Subspace Assumption, which are introduced and proved by Lewko and Waters [12], in the setting of dual pairing vector spaces. Please refer to Appendix A.1 for the details of this lemma and the Subspace Assumption. Here we would like to stress that *the Subspace Assumption is implied by DLIN assumption.*

To construct our AugR-CP-ABE scheme, we further define a new notation. In particular, for any $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$, $\boldsymbol{v'} = (v'_1, \ldots, v'_{n'}) \in \mathbb{Z}_p^{n'}$, we define

$$(g^{\boldsymbol{v}})^{\boldsymbol{v'}} := ((g^{\boldsymbol{v}})^{v'_1}, \ldots, (g^{\boldsymbol{v}})^{v'_{n'}}) = (g^{v'_1 v_1}, \ldots, g^{v'_1 v_n}, \ldots, g^{v'_{n'} v_1}, \ldots, g^{v'_{n'} v_n}) \in \mathbb{G}^{nn'}.$$

Note that for any $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{Z}_p^n, \boldsymbol{v'}, \boldsymbol{w'} \in \mathbb{Z}_p^{n'}$, we have

$$\begin{aligned} e_{nn'}((g^{\boldsymbol{v}})^{\boldsymbol{v'}}, (g^{\boldsymbol{w}})^{\boldsymbol{w'}}) &= \prod_{j=1}^{n'} \prod_{i=1}^{n} e(g^{v'_j v_i}, g^{w'_j w_i}) = \prod_{j=1}^{n'} (\prod_{i=1}^{n} e(g^{v_i}, g^{w_i}))^{v'_j w'_j} \\ =& \big(e_n(g^{\boldsymbol{v}}, g^{\boldsymbol{w}})\big)^{(\boldsymbol{v'}\cdot\boldsymbol{w'})} = \big(e(g,g)^{(\boldsymbol{v}\cdot\boldsymbol{w})}\big)^{(\boldsymbol{v'}\cdot\boldsymbol{w'})} = e(g,g)^{(\boldsymbol{v}\cdot\boldsymbol{w})(\boldsymbol{v'}\cdot\boldsymbol{w'})} = e_{nn'}((g^{\boldsymbol{v'}})^{\boldsymbol{v}}, (g^{\boldsymbol{w'}})^{\boldsymbol{w}}). \end{aligned}$$

**Linear Secret-Sharing Schemes (LSSS).** As of previous work, we use linear secret-sharing schemes (LSSS) to express the access policies. An LSSS is a share-generating matrix $A$ whose rows are labeled by attributes via a function $\rho$. An attribute set $S$ satisfies the LSSS access matrix $(A, \rho)$ if the rows labeled by the attributes in $S$ have the *linear reconstruction* property, namely, there exist constants $\{\omega_i | \rho(i) \in S\}$ such that, for any valid shares $\{\lambda_i\}$ of a secret $s$, we have $\sum_{\rho(i)\in S} \omega_i \lambda_i = s$. The formal definitions of access structures and LSSS can be found in Appendix C.

**Notations.** Suppose the number of users $N$ in the system equals $n^2$ for some $n$ [1], so we use $[n, n]$ instead of $[N]$ in the following content. We arrange the users in a $n \times n$ matrix and uniquely assign a tuple $(i, j)$ where $1 \le i, j \le n$, to each user. A user at position $(i, j)$ of the matrix has index $k = (i-1) * n + j$. For simplicity, we directly use $(i, j)$ as the index where $(i, j) \ge (\bar{i}, \bar{j})$ means that $((i > \bar{i}) \vee (i = \bar{i} \wedge j \ge \bar{j}))$. The use of pairwise notation $(i, j)$ is purely a notational convenience, as $k = (i-1) * n + j$ defines a bijection between $\{(i, j) | 1 \le i, j \le n\}$ and $\{1, \dots, N\}$. We conflate the notation and consider the attribute universe to be $[\mathcal{U}] = \{1, 2 \dots, \mathcal{U}\}$, so $\mathcal{U}$ serves both as a description of the attribute universe and as a count of the total number of attributes. Given a bilinear group order $p$, one can randomly choose $r_x, r_y, r_z \in \mathbb{Z}_p$, and set $\boldsymbol{\chi}_1 = (r_x, 0, r_z)$, $\boldsymbol{\chi}_2 = (0, r_y, r_z)$, $\boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. Let $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ be the subspace spanned by $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, i.e. $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} = \{\nu_1 \boldsymbol{\chi}_1 + \nu_2 \boldsymbol{\chi}_2 | \nu_1, \nu_2 \in \mathbb{Z}_p\}$. We can see that $\boldsymbol{\chi}_3$ is orthogonal to the subspace $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and that $\mathbb{Z}_p^3 = span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\} = \{\nu_1 \boldsymbol{\chi}_1 + \nu_2 \boldsymbol{\chi}_2 + \nu_3 \boldsymbol{\chi}_3 | \nu_1, \nu_2, \nu_3 \in \mathbb{Z}_p\}$. For any $\boldsymbol{v} \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, we have $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) = 0$, and for random $\boldsymbol{v} \in \mathbb{Z}_p^3$, $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) \ne 0$ happens with overwhelming probability.

### 4.2 AugR-CP-ABE Construction

$\mathsf{Setup_A}(\lambda, \mathcal{U}, N = n^2) \to (\mathsf{PP}, \mathsf{MSK})$. The algorithm chooses a bilinear group $\mathbb{G}$ of order $p$ and two generators $g, h \in \mathbb{G}$. It randomly chooses $\{h_j \in \mathbb{Z}_p\}_{j\in[n]}$, $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*) \in Dual(\mathbb{Z}_p^3, \psi)$ and $(\mathbb{B}_1, \mathbb{B}_1^*), \dots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*) \in Dual(\mathbb{Z}_p^6, \psi)$. We let $\boldsymbol{b}_j, \boldsymbol{b}_j^* (1 \le j \le 3)$ denote the basis vectors belonging to $(\mathbb{B}, \mathbb{B}^*)$, $\boldsymbol{b}_{0,j}, \boldsymbol{b}_{0,j}^* (1 \le j \le 3)$ denote the basis vectors belonging to $(\mathbb{B}_0, \mathbb{B}_0^*)$, and $\boldsymbol{b}_{x,j}, \boldsymbol{b}_{x,j}^* (1 \le j \le 6)$ denote the basis vectors belonging to $(\mathbb{B}_x, \mathbb{B}_x^*)$ for each $x \in [\mathcal{U}]$. The algorithm also chooses random exponents

$$\alpha_1, \alpha_2 \in \mathbb{Z}_p, \quad \{r_i, z_i, \ \alpha_{i,1}, \alpha_{i,2} \in \mathbb{Z}_p\}_{i\in[n]}, \quad \{c_{j,1}, c_{j,2}, \ y_j, h_j \in \mathbb{Z}_p\}_{j\in[n]}.$$

The public parameter $\mathsf{PP}$ and the master secret key $\mathsf{MSK}$ are set to

$$\begin{aligned}\mathsf{PP} = \Big( & (p, \mathbb{G}, \mathbb{G}_T, e), \ g, h, g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, \ \{h_j\}_{j\in[n]}, h^{\boldsymbol{b}_1}, h^{\boldsymbol{b}_2}, \ \{h_j^{\boldsymbol{b}_1}, h_j^{\boldsymbol{b}_2}\}_{j\in[n]}, \\ & h^{\boldsymbol{b}_{0,1}}, h^{\boldsymbol{b}_{0,2}}, \ \{h^{\boldsymbol{b}_{x,1}}, h^{\boldsymbol{b}_{x,2}}, h^{\boldsymbol{b}_{x,3}}, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]}, \\ & F_1 = e(g, h)^{\psi\alpha_1}, \ F_2 = e(g, h)^{\psi\alpha_2}, \ \ \{F_{1,j} = e(g, h_j)^{\psi\alpha_1}, F_{2,j} = e(g, h_j)^{\psi\alpha_2}\}_{j\in[n]}, \\ & \{E_{i,1} = e(g, g)^{\psi\alpha_{i,1}}, \ E_{i,2} = e(g, g)^{\psi\alpha_{i,2}}\}_{i\in[n]}, \\ & \{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}\}_{i\in[n]}, \ \ \{\boldsymbol{H}_j = g^{c_{j,1}\boldsymbol{b}_1^* + c_{j,2}\boldsymbol{b}_2^*}, \ \ \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]}\Big).\end{aligned}$$

$$\mathsf{MSK} = \Big(\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \ \boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,2}^*, \ \{\boldsymbol{b}_{x,1}^*, \boldsymbol{b}_{x,2}^*, \boldsymbol{b}_{x,3}^*, \boldsymbol{b}_{x,4}^*\}_{x\in[\mathcal{U}]}, \alpha_1, \alpha_2, \ \{r_i, z_i, \ \alpha_{i,1}, \alpha_{i,2}\}_{i\in[n]}, \ \{c_{j,1}, c_{j,2}\}_{j\in[n]}\Big).$$

In addition, a counter $ctr = 0$ is implicitly included in $\mathsf{MSK}$.

$\mathsf{KeyGen_A}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{(i,j),S}$. The algorithm first sets $ctr = ctr + 1$ and computes the corresponding index in the form of $(i, j)$ where $1 \le i, j \le n$ and $(i-1) * n + j = ctr$. Then it randomly

[1] If the number of users is not a square, we add some "dummy" users to pad to the next square.

chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, and outputs a private key

$$\begin{aligned}\mathsf{SK}_{(i,j),S} = \langle\ (i,j), S,\ & \boldsymbol{K}_{i,j} = g^{(\alpha_{i,1}+r_i c_{j,1})\boldsymbol{b}_1^* + (\alpha_{i,2}+r_i c_{j,2})\boldsymbol{b}_2^*} (h h_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}, \\ & \boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i}, \\ & \{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}\}_{j' \in [n] \setminus \{j\}}, \\ & \boldsymbol{K}_{i,j,0} = g^{\delta_{i,j,1}\boldsymbol{b}_{0,1}^* + \delta_{i,j,2}\boldsymbol{b}_{0,2}^*},\ \ \{\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{b}_{x,1}^* + \boldsymbol{b}_{x,2}^*) + \sigma_{i,j,2}(\boldsymbol{b}_{x,3}^* + \boldsymbol{b}_{x,4}^*)}\}_{x \in S}\ \rangle.\end{aligned}$$

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A} = (A, \rho), (\bar{i}, \bar{j})) \to CT_{R,(A,\rho)}$. $R \subseteq [n,n]$ is a revocation list. $A$ is an $l \times m$ LSSS matrix and $\rho$ maps each row $A_k$ of $A$ to an attribute $\rho(k) \in [\mathcal{U}]$. The encryption is for recipients whose (index, attributes set) pair $\big((i,j), S_{(i,j)}\big)$ satisfy $\big((i,j) \in [n,n] \backslash R\big) \wedge \big(S_{(i,j)}\ satisfies\ (A,\rho)\big) \wedge \big((i,j) \geq (\bar{i}, \bar{j})\big)$. Let $\bar{R} = [n,n] \setminus R$ and for $i \in [n], \bar{R}_i = \{j' | (i,j') \in \bar{R}\}$, that is, $\bar{R}$ is the non-revoked index list, and $\bar{R}_i$ is the set of non-revoked column index on the $i$-th row. The algorithm first chooses random

$$\kappa,\ \tau,\ s_1, \ldots, s_n,\ t_1, \ldots, t_n \in \mathbb{Z}_p, \quad \boldsymbol{v}_c,\ \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n \in \mathbb{Z}_p^3,$$
$$\xi_{1,1}, \xi_{1,2}, \ldots, \xi_{l,1}, \xi_{l,2} \in \mathbb{Z}_p, \quad \boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{Z}_p^m.$$

It also chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. Then it randomly chooses

$$\boldsymbol{v}_i \in \mathbb{Z}_p^3\ for\ i = 1, \ldots, \bar{i}, \quad \boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ for\ i = \bar{i}+1, \ldots, n.$$

Let $\pi_1$ and $\pi_2$ be the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ respectively. The algorithm creates a ciphertext $\langle R, (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows:

1. For each row $i \in [n]$:
   - if $i < \bar{i}$: choose random $\hat{s}_i \in \mathbb{Z}_p$, then set

$$\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa},$$
$$\boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = (h \prod_{j' \in \bar{R}_i} h_{j'})^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad T_i = e(g,g)^{\hat{s}_i}.$$

   - if $i \geq \bar{i}$: set

$$\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa},$$
$$\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = (h \prod_{j' \in \bar{R}_i} h_{j'})^{\tau s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},$$
$$T_i = M \frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}}{({F_1}'{F_2}')^{\tau s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} F_1^{\pi_1} F_2^{\pi_2}},$$

     where ${F_1}' = F_1 \prod_{j' \in \bar{R}_i} F_{1,j'}$ and ${F_2}' = F_2 \prod_{j' \in \bar{R}_i} F_{2,j'}$ respectively.
2. For each column $j \in [n]$:
   - if $j < \bar{j}$: choose random $\mu_j \in \mathbb{Z}_p$, then set $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.
   - if $j \geq \bar{j}$: set $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau \boldsymbol{v}_c} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.
3. $\boldsymbol{P}_0 = h^{\pi_1 \boldsymbol{b}_{0,1} + \pi_2 \boldsymbol{b}_{0,2}},\ \ \{\boldsymbol{P}_k = h^{(A_k \cdot \boldsymbol{u}_1 + \xi_{k,1})\boldsymbol{b}_{\rho(k),1} - \xi_{k,1}\boldsymbol{b}_{\rho(k),2} + (A_k \cdot \boldsymbol{u}_2 + \xi_{k,2})\boldsymbol{b}_{\rho(k),3} - \xi_{k,2}\boldsymbol{b}_{\rho(k),4}}\}_{k \in [l]}$.

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT_{R,(A,\rho)}, \mathsf{SK}_{(i,j),S}) \to M$ or $\perp$. The algorithm parses $CT_{R,(A,\rho)}$ and $\mathsf{SK}_{(i,j),S}$ to $\langle R, (A,\rho), (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ and $\langle\ (i,j), S,\ \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \backslash \{j\}}, \boldsymbol{K}_{i,j,0},\ \{\boldsymbol{K}_{i,j,x}\}_{x \in S}\ \rangle$ respectively. If $(i,j) \in R$ or $S$ does not satisfy $(A,\rho)$, the algorithm outputs $\perp$, otherwise it

1. Computes constants $\{\omega_k \in \mathbb{Z}_p | \rho(k) \in S\}$ such that $\sum_{\rho(k)\in S} \omega_k A_k = (1, 0, \ldots, 0)$, then computes
$$D_P = e_3(\boldsymbol{K}_{i,j,0}, \boldsymbol{P}_0) \prod_{\rho(k)\in S} e_6(\boldsymbol{K}_{i,j,\rho(k)}, \boldsymbol{P}_k)^{\omega_k}.$$
2. Since $(i, j) \in \bar{R}(= [n, n] \setminus R)$ implies $j \in \bar{R}_i$, the algorithm can compute
$$\begin{aligned}\bar{\boldsymbol{K}}_{i,j} &= \boldsymbol{K}_{i,j} \cdot (\prod_{j'\in\bar{R}_i\setminus\{j\}} \bar{\boldsymbol{K}}_{i,j,j'}) \\ &= g^{(\alpha_{i,1}+r_ic_{j,1})\boldsymbol{b}_1^*+(\alpha_{i,2}+r_ic_{j,2})\boldsymbol{b}_2^*}(hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*} \\ &\qquad \cdot (\prod_{j'\in\bar{R}_i\setminus\{j\}} h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}) \\ &= g^{(\alpha_{i,1}+r_ic_{j,1})\boldsymbol{b}_1^*+(\alpha_{i,2}+r_ic_{j,2})\boldsymbol{b}_2^*}(h \prod_{j'\in\bar{R}_i} h_{j'})^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}.\end{aligned}$$
Note that if $(i, j) \in R$ (implying $j \notin \bar{R}_i$), the algorithm cannot produce such a $\bar{\boldsymbol{K}}_{i,j}$. The algorithm then computes
$$D_I = \frac{e_3(\bar{\boldsymbol{K}}_{i,j}, \boldsymbol{Q}_i) \cdot e_3(\boldsymbol{K}''_{i,j}, \boldsymbol{Q}''_i) \cdot e_9(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{K}'_{i,j}, \boldsymbol{Q}'_i) \cdot e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)}.$$
3. Computes $M = T_i/(D_P \cdot D_I)$ as the output message. Assume the ciphertext is generated from message $M'$ and index $(\bar{i}, \bar{j})$, it can be verified that only when $(i > \bar{i})$ or $(i = \bar{i} \wedge j \geq \bar{j})$, $M = M'$ will hold. This follows from the facts that for $i > \bar{i}$, we have $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) = 0$ (since $\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$), and for $i = \bar{i}$, we have that $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) \neq 0$ happens with overwhelming probability (since $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$).

**Correctness.** Assume the ciphertext is generated from revocation list $R$, message $M'$ and index $(\bar{i}, \bar{j})$. For $i \geq \bar{i}$ we have

$$\begin{aligned}&\frac{e_3(\bar{\boldsymbol{K}}_{i,j}, \boldsymbol{Q}_i) \cdot e_3(\boldsymbol{K}''_{i,j}, \boldsymbol{Q}''_i)}{e_3(\boldsymbol{K}'_{i,j}, \boldsymbol{Q}'_i)} \\ =&\frac{e(g,g)^{\psi(\alpha_{i,1}+r_ic_{j,1}+\alpha_{i,2}+r_ic_{j,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} e(h\prod_{j'\in\bar{R}} h_{j'}, g)^{\psi(\sigma_{i,j,1}+\delta_{i,j,1}+\sigma_{i,j,2}+\delta_{i,j,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{e(g, h\prod_{j'\in\bar{R}} h_{j'})^{\psi(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1}+\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} e(g,h)^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\pi_1\psi+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\pi_2\psi}} \\ =&\frac{e(g,g)^{\psi(\alpha_{i,1}+r_ic_{j,1}+\alpha_{i,2}+r_ic_{j,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{e(g, h\prod_{j'\in\bar{R}} h_{j'})^{\psi(\alpha_1+\alpha_2)\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} e(g,h)^{\psi(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\pi_1+\psi(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\pi_2}} \\ =&\frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} \cdot e(g,g)^{\psi r_i(c_{j,1}+c_{j,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{(F'_1F'_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} F_1^{\pi_1} F_2^{\pi_2} \cdot e(g,h)^{\psi(\sigma_{i,j,1}+\delta_{i,j,1})\pi_1+\psi(\sigma_{i,j,2}+\delta_{i,j,2})\pi_2}},\end{aligned}$$

where $F_1{}' = F_1 \prod_{j'\in\bar{R}_i} F_{1,j'}$ and $F_2{}' = F_2 \prod_{j'\in\bar{R}_i} F_{2,j'}$ respectively.

If $i \geq \bar{i} \wedge j \geq \bar{j}$: we have

$$\frac{e_9(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)} = \frac{e_9((\boldsymbol{G}_i)^{\kappa s_i \boldsymbol{v}_i}, (\boldsymbol{Y}_j)^{\boldsymbol{w}_j})}{e_9((\boldsymbol{G}_i)^{s_i\boldsymbol{v}_i}, (\boldsymbol{H}_j)^{\tau\boldsymbol{v}_c}(\boldsymbol{Y}_j)^{\kappa\boldsymbol{w}_j})} = \frac{1}{e_3(\boldsymbol{G}_i, \boldsymbol{H}_j)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}} = \frac{1}{e(g,g)^{\psi r_i(c_{j,1}+c_{j,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}.$$

If $i > \bar{i} \wedge j < \bar{j}$: note that for $i > \bar{i}$, we have $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) = 0$ (since $\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$), then we have

$$\frac{e_9(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)} = \frac{e_9((\boldsymbol{G}_i)^{\kappa s_i \boldsymbol{v}_i}, (\boldsymbol{Y}_j)^{\boldsymbol{w}_j})}{e_9((\boldsymbol{G}_i)^{s_i \boldsymbol{v}_i}, (\boldsymbol{H}_j)^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j})} = \frac{1}{e_3(\boldsymbol{G}_i, \boldsymbol{H}_j)^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c) + \tau s_i \mu_j (\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3)}}$$
$$= \frac{1}{e(g,g)^{\psi r_i (c_{j,1} + c_{j,2}) \tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}}.$$

If $i = \bar{i} \wedge j < \bar{j}$: note that for $i = \bar{i}$, we have that $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) \neq 0$ happens with overwhelming probability (since $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$), then we have

$$\frac{e_9(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)} = \frac{e_9((\boldsymbol{G}_i)^{\kappa s_i \boldsymbol{v}_i}, (\boldsymbol{Y}_j)^{\boldsymbol{w}_j})}{e_9((\boldsymbol{G}_i)^{s_i \boldsymbol{v}_i}, (\boldsymbol{H}_j)^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j})} = \frac{1}{e_3(\boldsymbol{G}_i, \boldsymbol{H}_j)^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c) + \tau s_i \mu_j (\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3)}}$$
$$= \frac{1}{e(g,g)^{\psi r_i (c_{j,1} + c_{j,2}) \tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \cdot e(g,g)^{\psi r_i (c_{j,1} + c_{j,2}) \tau s_i \mu_j (\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3)}}.$$

Note that

$$\begin{aligned} D_P =& e_3(\boldsymbol{K}_{i,j,0}, \boldsymbol{P}_0) \prod_{\rho(k) \in S} e_6(\boldsymbol{K}_{i,j,\rho(k)}, \boldsymbol{P}_k)^{\omega_k} \\ =& e_3(\boldsymbol{K}_{i,j,0}, \boldsymbol{P}_0) \prod_{\rho(k) \in S} \left( e(g^{\sigma_{i,j,1}}, h^{A_k \cdot \boldsymbol{u}_1}) e(g^{\sigma_{i,j,2}}, h^{A_k \cdot \boldsymbol{u}_2}) \right)^{\psi \omega_k} \\ =& e(g,h)^{\psi(\delta_{i,j,1}\pi_1 + \delta_{i,j,2}\pi_2)} e(g,h)^{\psi(\sigma_{i,j,1}\pi_1 + \sigma_{i,j,2}\pi_2)}. \end{aligned}$$

Thus from the values of $T_i, D_P$ and $D_I$, for $M = T_i/(D_P \cdot D_I)$ we have that: (1) if $(i > \bar{i}) \vee (i = \bar{i} \wedge j \geq \bar{j})$, then $M = M'$; (2) if $i = \bar{i} \wedge j < \bar{j}$, then $M = M' \cdot e(g,g)^{\psi r_i (c_{j,1} + c_{j,2}) \tau s_i \mu_j (\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3)}$; (3) if $i < \bar{i}$, then $M$ has no relation with $M'$.

### 4.3 Security of The AugR-CP-ABE Construction

The following Theorem 3 and Theorem 4 show that our AugR-CP-ABE construction is message-hiding, and Theorem 5 shows that our AugR-CP-ABE construction is index-hiding.

**Theorem 3.** *Suppose the DLIN assumption, the D3DH assumption, and the source group q-parallel BDHE assumption hold. Then no PPT adversary can win* $\mathsf{Game}_{\mathsf{MH}_1}^{\mathsf{A}}$ *with non-negligible advantage.*

*Proof.* We begin by defining our various types of semi-functional keys and ciphertexts. The semi-functional space in the exponent will correspond to the span of $\boldsymbol{b}_3, \boldsymbol{b}_3^*$, the span of $\boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,3}^*$ and the span of each $\boldsymbol{b}_{x,5}, \boldsymbol{b}_{x,6}, \boldsymbol{b}_{x,5}^*, \boldsymbol{b}_{x,6}^*$.

**Semi-functional Keys.** To produce a semi-functional key for an attribute set $S$, one first calls the normal key generation algorithm to produce a normal key consisting of $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}$, $\boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S}$ with index $(i,j)$. One then chooses random value $\gamma$. The semi-functional key is

$$\boldsymbol{K}_{i,j}(hh_j)^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}'_{i,j} g^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}''_{i,j} g^{z_i \gamma \boldsymbol{b}_3^*}, \{\bar{\boldsymbol{K}}_{i,j,j'} h_{j'}^{\gamma \boldsymbol{b}_3^*}\}_{j' \in [n] \setminus \{j\}},\ \boldsymbol{K}_{i,j,0},\ \{\boldsymbol{K}_{i,j,x}\}_{x \in S}.$$

**Semi-functional Ciphertexts.** To produce a semi-functional ciphertext for an LSSS matrix $(A, \rho)$ of size $l \times m$, one first calls the normal encryption algorithm to produce a normal ciphertext consisting of $\langle R, (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n, (\boldsymbol{P}_k)_{k=0}^l \rangle$. One then chooses random values $\pi_3, \xi_{k,3} (1 \leq k \leq l) \in \mathbb{Z}_p$ and a random vector $\boldsymbol{u}_3 \in \mathbb{Z}_p^m$ with first entry equal to $\pi_3$. The semi-functional ciphertext is:

$$\langle R, (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i h^{\pi_3 \boldsymbol{b}_3}, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ \boldsymbol{P}_0 h^{\pi_3 \boldsymbol{b}_{0,3}}, (\boldsymbol{P}_k h^{(A_k \cdot \boldsymbol{u}_3 + \xi_{k,3}) \boldsymbol{b}_{\rho(k),5} - \xi_{k,3} \boldsymbol{b}_{\rho(k),6}})_{k=1}^l \rangle.$$

Our proof is obtained via a hybrid argument over a sequence of games:

$\mathsf{Game}_{real}$: The real message-hiding game (i.e. $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) as defined in the Section 3.1.

$\mathsf{Game}_t$ $(0 \leq t \leq Q)$: Let $Q$ denote the total number of key queries that the attacker makes. For each $t$ from 0 to $Q$, we define $\mathsf{Game}_t$ as follows: In $\mathsf{Game}_t$, the ciphertext given to the attacker is semi-functional, as are the first $t$ keys. The remaining keys are normal.

$\mathsf{Game}_{final}$: In this game, all of the keys given to the attacker are semi-functional, and the ciphertext given to the attacker is a semi-functional encryption of a *random message*.

The outer structure of our hybrid argument will progress as shown in Fig. 1. First, we transition from $\mathsf{Game}_{real}$ to $\mathsf{Game}_0$, then to $\mathsf{Game}_1$, next to $\mathsf{Game}_2$, and so on. We ultimately arrive at $\mathsf{Game}_Q$, where the ciphertext and all of the keys given to the attacker are semi-functional. We then transition to $\mathsf{Game}_{final}$, which is defined to be like $\mathsf{Game}_Q$, except that the ciphertext given to the attacker is a semi-functional encryption of a random message. This will complete our proof, since any attacker has a zero advantage in this final game.

The transitions from $\mathsf{Game}_{real}$ to $\mathsf{Game}_0$ and from $\mathsf{Game}_Q$ to $\mathsf{Game}_{final}$ are relatively easy, and can be accomplished directly via computational assumptions. The transitions from $\mathsf{Game}_{t-1}$ to $\mathsf{Game}_t$ require more intricate arguments. For these steps, we will need to treat $\mathsf{Phase\ 1}$ key requests (before the challenge ciphertext) and $\mathsf{Phase\ 2}$ key requests (after the challenge ciphertext) differently. We will also need to define two additional types of semi-functional keys:

**Nominal Semi-functional Keys.** To produce a nominal semi-functional key for an attribute set $S$, one first calls the normal key generation algorithm to produce a normal key consisting of $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S}$ with index $(i, j)$. One then chooses random values $\sigma_{i,j,3}, \delta_{i,j,3} \in \mathbb{Z}_p$. The nominal semi-functional key is: $\boldsymbol{K}_{i,j}(hh_j)^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3}$, $\boldsymbol{K}'_{i,j}g^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3}$, $\boldsymbol{K}''_{i,j}g^{z_i(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3}$, $\{\bar{\boldsymbol{K}}_{i,j,j'}h_{j'}{}^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3}\}_{j' \in [n] \setminus \{j\}}, \boldsymbol{K}_{i,j,0}g^{\delta_{i,j,3}\boldsymbol{b}^*_{0,3}}$, $\{\boldsymbol{K}_{i,j,x}g^{\sigma_{i,j,3}(\boldsymbol{b}^*_{x,5}+\boldsymbol{b}^*_{x,6})}\}_{x \in S}$. We note that a nominal semi-functional key still correctly decrypts a semi-functional ciphertext.

**Temporary Semi-functional Keys.** A temporary semi-functional key is similar to a nominal semi-functional key, except that the semi-functional component attached to $\boldsymbol{K}'_{i,j}$ will now be randomized (this will prevent correct decryption of a semi-functional ciphertext) and $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}''_{i,j}$ and $\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}$ change accordingly. More formally, to produce a temporary semi-functional key for an attribute set $S$, one first calls the normal key generation algorithm to produce a normal key consisting of $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}, \boldsymbol{K}_{i,j,0}$, $\{\boldsymbol{K}_{i,j,x}\}_{x \in S}$ with index $(i, j)$. One then chooses random values $\sigma_{i,j,3}, \delta_{i,j,3}, \gamma \in \mathbb{Z}_p$. The temporary semi-functional key is formed as:

$$\boldsymbol{K}_{i,j}(hh_j)^{\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}'_{i,j}g^{\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}''_{i,j}g^{z_i\gamma\boldsymbol{b}^*_3},\ \{\bar{\boldsymbol{K}}_{i,j,j'}h_{j'}^{\gamma\boldsymbol{b}^*_3}\}_{j' \in [n] \setminus \{j\}},\ \boldsymbol{K}_{i,j,0}g^{\delta_{i,j,3}\boldsymbol{b}^*_{0,3}},\ \{\boldsymbol{K}_{i,j,x}g^{\sigma_{i,j,3}(\boldsymbol{b}^*_{x,5}+\boldsymbol{b}^*_{x,6})}\}_{x \in S}.$$

For each $t$ from 1 to $Q$, we define the following additional games:

$\mathsf{Game}^N_t$: This is like $\mathsf{Game}_t$, except that the $t^{th}$ key given to the attacker is a nominal semi-functional key. The first $t-1$ keys are still semi-functional in the original sense, while the remaining keys are normal.

$\mathsf{Game}^T_t$: This is like $\mathsf{Game}_t$, except that the $t^{th}$ key given to the attacker is a temporary semi-functional key. The first $t-1$ keys are still semi-functional in the original sense, while the remaining keys are normal.

In order to transition from $\mathsf{Game}_{t-1}$ to $\mathsf{Game}_t$ in our hybrid argument, we will transition first from $\mathsf{Game}_{t-1}$ to $\mathsf{Game}^N_t$, then to $\mathsf{Game}^T_t$, and finally to $\mathsf{Game}_t$. The transition from $\mathsf{Game}^N_t$ to $\mathsf{Game}^T_t$ will require different computational assumptions for Phase 1 and Phase 2 queries (As shown in Fig. 1, we use two lemmas based on different assumptions to obtain the transition).

As shown in Fig. 1, we use a series of lemmas, i.e. Lemmas 4, 5, 6, 7, 8, and 9, to prove the transitions. The details of these lemmas and their proofs can be found in Appendix B.1.

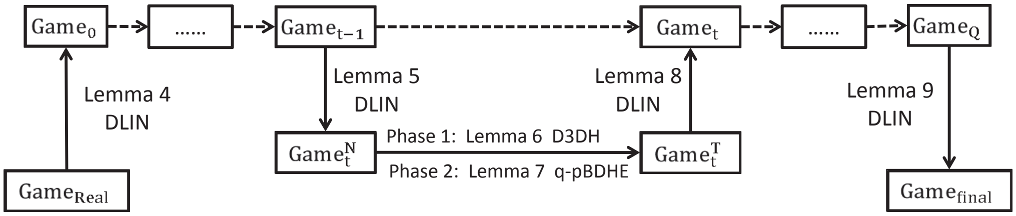

**Fig. 1.** Lemmas 4, 5, 8, and 9 rely on the subspace assumption (w.r.t. Definition 5), which is implied by DLIN assumption, Lemma 6 relies on the D3DH assumption, and Lemma 7 relies on the source group $q$-parallel BDHE assumption.

**Theorem 4.** *No PPT adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$ *with non-negligible advantage.*

*Proof.* The argument for security of $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$ is very straightforward since an encryption to index $N+1 = (n+1,1)$ contains no information about the message. The simulator simply runs actual $\mathsf{Setup}_{\mathsf{A}}$ and $\mathsf{KeyGen}_{\mathsf{A}}$ algorithms and encrypts the message $M_b$ by the challenge access policy $\mathbb{A}$ and index $(n+1,1)$. Since for all $i = 1$ to $n$, the values of $T_i$ contain no information about the message, the bit $b$ is perfectly hidden and $\mathsf{MH}^{\mathsf{A}}_{N+1}\mathsf{Adv}_{\mathcal{A}} = 0$.

**Theorem 5.** *Suppose that the D3DH assumption and the DLIN assumption hold. Then no PPT adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

*Proof.* Theorem 5 follows Lemma 1 and Lemma 2 below.

**Lemma 1.** *Suppose that the D3DH assumption holds. Then for $\bar{j} < n$ no PPT adversary can distinguish between an encryption to $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}+1)$ in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

*Proof.* In $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$, the adversary $\mathcal{A}$ will eventually behave in one of two different ways:

**Case I:** In Key Query phase, $\mathcal{A}$ will not submit $((\bar{i}, \bar{j}), S_{(\bar{i},\bar{j})})$ for some attribute set $S_{(\bar{i},\bar{j})}$ to query the corresponding private key. In Challenge phase, $\mathcal{A}$ submits a message $M$ and a non-empty attribute set $S^*$. There is not any restriction on $S^*$.

**Case II:** In Key Query phase, $\mathcal{A}$ will submit $((\bar{i}, \bar{j}), S_{(\bar{i},\bar{j})})$ for some attribute set $S_{(\bar{i},\bar{j})}$ to query the corresponding private key. In Challenge phase, $\mathcal{A}$ submits a message $M$ and a non-empty attribute set $S^*$ with the restriction that the corresponding strictest access policy $\mathbb{A}_{S^*}$ is not satisfied by $S_{(\bar{i},\bar{j})}$. **Case II** has the following sub-cases:

1. $(\bar{i}, \bar{j}) \notin [n,n] \setminus R^*$, $S_{(\bar{i},\bar{j})}$ satisfies $\mathbb{A}^*$.
2. $(\bar{i}, \bar{j}) \notin [n,n] \setminus R^*$, $S_{(\bar{i},\bar{j})}$ does not satisfy $\mathbb{A}^*$.
3. $(\bar{i}, \bar{j}) \in [n,n] \setminus R^*$, $S_{(\bar{i},\bar{j})}$ does not satisfy $\mathbb{A}^*$.

We flip a random coin $c \in \{0,1\}$ as our guess on which case that $\mathcal{A}$ is in. In particular, if $c = 0$, we guess that $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**. In this case, it follows the restriction in the index-hiding game for Augmented Broadcast Encryption (AugBE) in [6], where the adversay does not query the key with index $(\bar{i}, \bar{j})$ or $(\bar{i}, \bar{j})$ is not in the receiver list $[n,n] \setminus R^*$. If $c = 1$, we guess that $\mathcal{A}$ is in **Case I**, **Case II.2** or **Case II.3**. As of the fully secure CP-ABE schemes in [10,19,11,12,13], we assume that the size of attribute universe (i.e. $|\mathcal{U}|$) is polynomial in the security parameter $\lambda$, so that a degradation of $O(1/|\mathcal{U}|)$ in the security reduction is acceptable. The proof details of Lemma 1 can be found in Appendix B.2.

**Lemma 2.** *Suppose the D3DH assumption and the DLIN assumption hold. Then for any $1 \le \bar{i} \le n$ no PPT adversary can distinguish between an encryption to $(\bar{i}, n)$ and $(\bar{i}+1, 1)$ in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

*Proof.* The proof of this lemma follows from a series of lemmas that establish the indistinguishability of the following games, where "less-than row" implies the corresponding $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$ and $T_i$ is a random element (i.e. $T_i = e(g,g)^{\hat{s}_i}$), "target row" implies the corresponding $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$ and $T_i$ is well-formed, and "greater-than row" implies the corresponding $\boldsymbol{v}_i$ is randomly chosen from $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and $T_i$ is well-formed.

- $H_1$: Encrypt to column $n$, row $\bar{i}$ is the target row, row $\bar{i}+1$ is the greater-than row.
- $H_2$: Encrypt to column $n+1$, row $\bar{i}$ is the target row, row $\bar{i}+1$ is the greater-than row.
- $H_3$: Encrypt to column $n+1$, row $\bar{i}$ is the less-than row, row $\bar{i}+1$ is the greater-than row (no target row).
- $H_4$: Encrypt to column 1, row $\bar{i}$ is the less-than row, row $\bar{i}+1$ is the greater-than row (no target row).
- $H_5$: Encrypt to column 1, row $\bar{i}$ is the less-than row, row $\bar{i}+1$ is the target row.

It can be observed that game $H_1$ corresponds to the encryption being done to $(\bar{i}, n)$ and game $H_5$ corresponds to encryption to $(\bar{i}+1, 1)$. As shown in Fig. 2, we use a series of lemmas, i.e. Lemmas 10, 11, 12, and 13, to prove the indistinguishability of the games $H_1$ and $H_5$. The details of these lemmas and their proofs can be found in Appendix B.3.

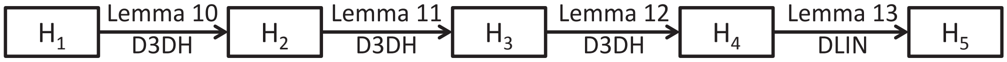

**Fig. 2.** Lemmas 10, 11, and 12 rely on the D3DH assumption, and Lemma 13 relies on the DLIN assumption.

## 5 Conclusion

In this paper, we proposed a new Augmented R-CP-ABE construction on prime order groups, and proved its message-hiding and index-hiding properties in the standard model. This CP-ABE achieves full security in the standard model on prime order groups. Our contributions are (1)adding the revocation list, and (2)proving its full security with revocability. We follow the proof method in [16] for message-hiding, and build two direct reductions for the proof for index-hiding. The scheme is a fully collusion-resistant blackbox traceable R-CP-ABE scheme. It achieves the most efficient level to date, with the overhead in $O(\sqrt{N})$ only.

## References

1. N. Attrapadung and H. Imai. Conjunctive broadcast and attribute-based encryption. In *Pairing*, pages 248–265, 2009.
2. A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
3. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
4. L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. In *ACM Conference on Computer and Communications Security*, pages 456–465, 2007.
5. H. Deng, Q. Wu, B. Qin, J. Mao, X. Liu, L. Zhang, and W. Shi. Who is touching my cloud. In *ESORICS, Part I*, pages 362–379, 2014.
6. S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In *ACM Conference on Computer and Communications Security*, pages 121–130, 2010.

7. V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.
8. J. Herranz, F. Laguillaumie, and C. Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In *Public Key Cryptography*, pages 19–34, 2010.
9. A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.
10. 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 *EUROCRYPT*, pages 62–91, 2010.
11. A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pages 180–198, 2012.
12. A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. *IACR Cryptology ePrint Archive*, 2012:326, 2012.
13. Z. Liu, Z. Cao, and D. S. Wong. Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay. In *ACM Conference on Computer and Communications Security*, pages 475–486, 2013.
14. 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 and Security*, 8(1):76–88, 2013.
15. Z. Liu and D. S. Wong. Practical ciphertext-policy attribute-based encryption: Traitor tracing, revocation, and large universe. In T. Malkin, V. Kolesnikov, A. B. Lewko, and M. Polychronakis, editors, *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers*, volume 9092 of *Lecture Notes in Computer Science*, pages 127–146. Springer, 2015.
16. Z. Liu and D. S. Wong. Traceable CP-ABE on prime order groups: Fully secure and fully collusion-resistant blackbox traceable. In S. Qing, E. Okamoto, K. Kim, and D. Liu, editors, *Information and Communications Security - 17th International Conference, ICICS 2015, Beijing, China, December 9-11, 2015, Revised Selected Papers*, volume 9543 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2015.
17. T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.
18. T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.
19. T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.
20. Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM Conference on Computer and Communications Security*, pages 463–474, 2013.
21. A. Sahai, H. Seyalioglu, and B. Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *CRYPTO*, pages 199–217, 2012.
22. A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.
23. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography*, pages 53–70, 2011.

## A Assumptions

**The Decisional Linear Assumption (DLIN)** Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}, \quad g, f, v \xleftarrow{R} \mathbb{G}, \quad c_1, c_2 \xleftarrow{R} \mathbb{Z}_p,\\
&D := ((p, \mathbb{G}, \mathbb{G}_T, e), g, f, v, f^{c_1}, v^{c_2}),\\
&T_0 = g^{c_1+c_2}, T_1 \xleftarrow{R} \mathbb{G}.
\end{aligned}$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking this assumption to be:

$$Adv^{DL}_{\mathcal{G},\mathcal{A}} := |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1)] = 1|.$$

We say that $\mathcal{G}$ satisfies the DLIN Assumption if $Adv^{DL}_{\mathcal{G},\mathcal{A}}$ is a negligible function of the security parameter $\lambda$ for any PPT algorithm $\mathcal{A}$.

**The Decisional 3-Party Diffie-Hellman Assumption (D3DH)** Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}, \quad g \xleftarrow{R} \mathbb{G}, \quad x, y, z \xleftarrow{R} \mathbb{Z}_p,\\
&D := ((p, \mathbb{G}, \mathbb{G}_T, e), g, g^x, g^y, g^z),\\
&T_0 = g^{xyz}, T_1 \xleftarrow{R} \mathbb{G}.
\end{aligned}$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking this assumption to be:

$$Adv_{\mathcal{G},\mathcal{A}}^{D3DH} := |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1)] = 1|.$$

We say that $\mathcal{G}$ satisfies the D3DH Assumption if $Adv_{\mathcal{G},\mathcal{A}}^{D3DH}$ is a negligible function of the security parameter $\lambda$ for any PPT algorithm $\mathcal{A}$.

**The Source Group $q$-Parallel BDHE Assumption [12]** Given a group generator $\mathcal{G}$ and a positive integer $q$, define the following distribution:

$$\begin{aligned}
&(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}, \quad g \xleftarrow{R} \mathbb{G}, \quad c, d, f, b_1, \ldots, b_q \xleftarrow{R} \mathbb{Z}_p,\\
&D = \big((p, \mathbb{G}, \mathbb{G}_T, e),\ g, g^f, g^{df},\ g^c, g^{c^2}, \ldots, g^{c^q},\ , g^{c^{q+2}}, \ldots, g^{c^{2q}},\\
&\qquad g^{c^i/b_j}\ \forall i \in \{1, \ldots, 2q\} \setminus \{q+1\}, j \in \{1, \ldots, q\},\\
&\qquad g^{dfb_j}\ \forall j \in \{1, \ldots, q\},\\
&\qquad g^{dfc^i b_{j'}/b_j}\ \forall i \in \{1, \ldots, q\}, j, j' \in \{1, \ldots, q\}\ s.t.\ j \neq j'\big),\\
&T_0 = g^{dc^{q+1}}, T_1 \xleftarrow{R} \mathbb{G}.
\end{aligned}$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking this assumption to be:

$$Adv_{\mathcal{G},\mathcal{A}}^{qPB} := |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1)] = 1|.$$

We say that $\mathcal{G}$ satisfies the Source Group $q$-Parallel BDHE Assumption if $Adv_{\mathcal{G},\mathcal{A}}^{qPB}$ is a negligible function of the security parameter $\lambda$ for any PPT algorithm $\mathcal{A}$.

### A.1 Assumptions for Dual Pairing Vector Spaces

Let $(\mathbb{B}, \mathbb{B}^*)$ denote a pair of dual orthonormal bases over $\mathbb{Z}_p^n$, $A \in \mathbb{Z}_p^{m \times m}$ be an invertible matrix for some $m \leq n$, and $S_m \subseteq \{1, \ldots, n\}$ be a subset of size $m$. Then new dual orthonormal bases $(\mathbb{B}_A, \mathbb{B}_A^*)$ are defined as follows. Let $B_m$ denote the $n \times m$ matrix over $\mathbb{Z}_p$ whose columns are the vectors $\boldsymbol{b}_i \in \mathbb{B}$ such that $i \in S_m$. Then $B_m A$ is also an $n \times m$ matrix. $\mathbb{B}_A$ is formed by retaining all of the vectors $\boldsymbol{b}_i \in \mathbb{B}$ for $i \notin S_m$ and exchanging the $\boldsymbol{b}_i$ for $i \in S_m$ with the columns of $B_m A$. To define $\mathbb{B}_A^*$, similarly let $B_m^*$ denote the $n \times m$ matrix over $\mathbb{Z}_p$ whose columns are the vectors $\boldsymbol{b}_i^* \in \mathbb{B}^*$ such that $i \in S_m$. Then $B_m^*(A^{-1})^t$ is also an $n \times m$ matrix, where $(A^{-1})^t$ denotes the transpose of $A^{-1}$. $\mathbb{B}_A^*$ is formed by retaining all of the vectors $\boldsymbol{b}_i^* \in \mathbb{B}^*$ for $i \notin S_m$ and exchanging the $\boldsymbol{b}_i^*$ for $i \in S_m$ with the columns of $\mathbb{B}_m^*(A^{-1})^t$. We have

**Lemma 3.** *[9] For any fixed positive integers $m \leq n$, any fixed invertible $A \in \mathbb{Z}_p^{m \times m}$ and set $S_m \subseteq \{1, \ldots, n\}$ of size $m$, if $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n, \psi)$, then $(\mathbb{B}_A, \mathbb{B}_A^*)$ is also distributed as a random sample from $Dual(\mathbb{Z}_p^n, \psi)$. In particular, the distribution of $(\mathbb{B}_A, \mathbb{B}_A^*)$ is independent of $A$.*

The "Subspace Assumption" is introduced by Lewko [9], and is generalized by Lewko and Waters [12]. In particular, let the parameter $m$ denote the number of bases, and each basis pair has its own dimension $n_i$ and its own parameter $k_i$ where $k_i$ is a positive integer such that $k_i \leq \frac{n_i}{3}$. The following statement of the subspace assumption is implied by DLIN assumption, and is proved by Lewko and Waters [12, Appendix A]. Note that this reduction (i.e., *the Subspace Assumption is implied by DLIN assumption*) holds for any valid choices of the parameters $m, n_i, k_i$. We refer to [12] for more details of the following statement of the subspace assumption.

The $m$ dual orthonormal bases pairs will be denoted by $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_m, \mathbb{B}_m^*)$. For each $i$ from 1 to $m$, the basis vectors comprising $(\mathbb{B}_i, \mathbb{B}_i^*)$ will be denoted by $\boldsymbol{b}_{i,1}, \ldots, \boldsymbol{b}_{i,n_i}$ and $\boldsymbol{b}_{i,1}^*, \ldots, \boldsymbol{b}_{i,n_i}^*$ respectively.

**Definition 5.** *(The Subspace Assumption [12]) Given a group generator $\mathcal{G}$, define the following distribution:*

$$
\begin{aligned}
(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}, \quad g \xleftarrow{R} \mathbb{G}, \quad \psi, \eta, \beta, \tau_1, \tau_2, \tau_3, \mu_1, \mu_2, \mu_3 \xleftarrow{R} \mathbb{Z}_p,& \\
(\mathbb{B}_1, \mathbb{B}_1^*) \xleftarrow{R} \mathit{Dual}(\mathbb{Z}_p^{n_1}, \psi), \ldots, (\mathbb{B}_m, \mathbb{B}_m^*) \xleftarrow{R} \mathit{Dual}(\mathbb{Z}_p^{n_m}, \psi),& \\
\forall i \in \{1, \ldots, m\} : \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad& \\
\boldsymbol{U}_{i,1} := g^{\mu_1 \boldsymbol{b}_{i,1} + \mu_2 \boldsymbol{b}_{i,k_i+1} + \mu_3 \boldsymbol{b}_{i,2k_i+1}}, \boldsymbol{U}_{i,2} := g^{\mu_1 \boldsymbol{b}_{i,2} + \mu_2 \boldsymbol{b}_{i,k_i+2} + \mu_3 \boldsymbol{b}_{i,2k_i+2}},& \\
\ldots, \boldsymbol{U}_{i,k_i} := g^{\mu_1 \boldsymbol{b}_{i,k_i} + \mu_2 \boldsymbol{b}_{i,2k_i} + \mu_3 \boldsymbol{b}_{i,3k_i}},& \\
\boldsymbol{V}_{i,1} := g^{\tau_1 \eta \boldsymbol{b}_{i,1}^* + \tau_2 \beta \boldsymbol{b}_{i,k_i+1}^*}, \boldsymbol{V}_{i,2} := g^{\tau_1 \eta \boldsymbol{b}_{i,2}^* + \tau_2 \beta \boldsymbol{b}_{i,k_i+2}^*},& \\
\ldots, \boldsymbol{V}_{i,k_i} := g^{\tau_1 \eta \boldsymbol{b}_{i,k_i}^* + \tau_2 \beta \boldsymbol{b}_{i,2k_i}^*},& \\
\boldsymbol{W}_{i,1} := g^{\tau_1 \eta \boldsymbol{b}_{i,1}^* + \tau_2 \beta \boldsymbol{b}_{i,k_i+1}^* + \tau_3 \boldsymbol{b}_{i,2k_i+1}^*}, \boldsymbol{W}_{i,2} := g^{\tau_1 \eta \boldsymbol{b}_{i,2}^* + \tau_2 \beta \boldsymbol{b}_{i,k_i+2}^* + \tau_3 \boldsymbol{b}_{i,2k_i+2}^*},& \\
\ldots, \boldsymbol{W}_{i,k_i} := g^{\tau_1 \eta \boldsymbol{b}_{i,k_i}^* + \tau_2 \beta \boldsymbol{b}_{i,2k_i}^* + \tau_3 \boldsymbol{b}_{i,3k_i}^*},& \\
D := \big((p, \mathbb{G}, \mathbb{G}_T, e), g, \ \{g^{\boldsymbol{b}_{i,1}}, g^{\boldsymbol{b}_{i,2}}, \ldots, g^{\boldsymbol{b}_{i,2k_i}}, \ \ g^{\boldsymbol{b}_{i,3k_i+1}}, \ldots, g^{\boldsymbol{b}_{i,n_i}},& \\
g^{\eta \boldsymbol{b}_{i,1}^*}, \ldots, g^{\eta \boldsymbol{b}_{i,k_i}^*}, \ \ g^{\beta \boldsymbol{b}_{i,k_i+1}^*}, \ldots, g^{\beta \boldsymbol{b}_{i,2k_i}^*}, \ \ g^{\boldsymbol{b}_{i,2k_i+1}^*}, \ldots, g^{\boldsymbol{b}_{i,n_i}^*},& \\
\boldsymbol{U}_{i,1}, \boldsymbol{U}_{i,2}, \ldots, \boldsymbol{U}_{i,k_i}\}_{i=1}^m, \ \mu_3\big).&
\end{aligned}
$$

*We assume that for any PPT adversary $\mathcal{A}$ (with output in $\{0,1\}$),*

$$
\mathit{Adv}_{\mathcal{G},\mathcal{A}} := |\Pr[\mathcal{A}(D, \{\boldsymbol{V}_{i,1}, \ldots, \boldsymbol{V}_{i,k_i}\}_{i=1}^m) = 1] - \Pr[\mathcal{A}(D, \{\boldsymbol{W}_{i,1}, \ldots, \boldsymbol{W}_{i,k_i}\}_{i=1}^m) = 1]|
$$

*is negligible in the security parameter $\lambda$.*

# B Proofs

## B.1 Proof of Theorem 3

**Lemma 4.** *Under the subspace assumption, no PPT attacker can achieve a non-negligible difference in advantage between $\mathsf{Game}_{real}$ and $\mathsf{Game}_0$.*

*Proof.* Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between $\mathsf{Game}_{real}$ and $\mathsf{Game}_0$, we will create a PPT algorithm $\mathcal{B}$ to break the subspace assumption. We will employ the subspace assumption with parameters $m = \mathcal{U} + 2$, $n_i = 3, k_i = 1$ for two values of $i$, and $n_i = 6, k_i = 2$ for the rest of the values of $i$. In order to reconcile the notation of the assumption with the notation of our construction as conveniently as possible, we will denote the bases involved in the assumption by

$(\mathbb{D},\mathbb{D}^*),(\mathbb{D}_0,\mathbb{D}_0^*)\in Dual(Z_p^3,\psi)$ and $(\mathbb{D}_1,\mathbb{D}_1^*),\ldots,(\mathbb{D}_{\mathcal{U}},\mathbb{D}_{\mathcal{U}}^*)\in Dual(Z_p^6,\psi)$. $\mathcal{B}$ is given (we will ignore the $\boldsymbol{U}$ terms and $\mu_3$ because they will not be needed):

$$\begin{aligned}\mathbb{G},p,g,\ g^{\boldsymbol{d}_1},g^{\boldsymbol{d}_2},\ g^{\boldsymbol{d}_{0,1}},g^{\boldsymbol{d}_{0,2}},\ \{g^{\boldsymbol{d}_{x,1}},g^{\boldsymbol{d}_{x,2}},g^{\boldsymbol{d}_{x,3}},g^{\boldsymbol{d}_{x,4}}\}_{x\in[\mathcal{U}]},\\ g^{\eta\boldsymbol{d}_1^*},g^{\beta\boldsymbol{d}_2^*},g^{\boldsymbol{d}_3^*},\ g^{\eta\boldsymbol{d}_{0,1}^*},g^{\beta\boldsymbol{d}_{0,2}^*},g^{\boldsymbol{d}_{0,3}^*},\ \{g^{\eta\boldsymbol{d}_{x,1}^*},g^{\eta\boldsymbol{d}_{x,2}^*},g^{\beta\boldsymbol{d}_{x,3}^*},g^{\beta\boldsymbol{d}_{x,4}^*},g^{\boldsymbol{d}_{x,5}^*},g^{\boldsymbol{d}_{x,6}^*}\}_{x\in[\mathcal{U}]},\\ \boldsymbol{T}_1,\ \boldsymbol{T}_{0,1},\ \{\boldsymbol{T}_{x,1},\boldsymbol{T}_{x,2}\}_{x\in[\mathcal{U}]}.\end{aligned}$$

The exponents of the unknown terms $\boldsymbol{T}_1,\boldsymbol{T}_{0,1}$ are distributed either as $\tau_1\eta\boldsymbol{d}_1^*+\tau_2\beta\boldsymbol{d}_2^*$ and $\tau_1\eta\boldsymbol{d}_{0,1}^*+\tau_2\beta\boldsymbol{d}_{0,2}^*$ respectively, or as $\tau_1\eta\boldsymbol{d}_1^*+\tau_2\beta\boldsymbol{d}_2^*+\tau_3\boldsymbol{d}_3^*$ and $\tau_1\eta\boldsymbol{d}_{0,1}^*+\tau_2\beta\boldsymbol{d}_{0,2}^*+\tau_3\boldsymbol{d}_{0,3}^*$ respectively. Similarly, the exponents of the unknown terms $\boldsymbol{T}_{x,1},\boldsymbol{T}_{x,2}$ are distributed either as $\tau_1\eta\boldsymbol{d}_{x,1}^*+\tau_2\beta\boldsymbol{d}_{x,3}^*$ and $\tau_1\eta\boldsymbol{d}_{x,2}^*+\tau_2\beta\boldsymbol{d}_{x,4}^*$ respectively, or as $\tau_1\eta\boldsymbol{d}_{x,1}^*+\tau_2\beta\boldsymbol{d}_{x,3}^*+\tau_3\boldsymbol{d}_{x,5}^*$ and $\tau_1\eta\boldsymbol{d}_{x,2}^*+\tau_2\beta\boldsymbol{d}_{x,4}^*+\tau_3\boldsymbol{d}_{x,6}^*$ respectively. It is $\mathcal{B}$'s task to determine if these $\tau_3$ contributions are present or not.

**Setup.** B implicitly sets the bases for the construction as:

$$\begin{aligned}&\boldsymbol{b}_1=\eta\boldsymbol{d}_1^*,\quad \boldsymbol{b}_2=\beta\boldsymbol{d}_2^*,\quad \boldsymbol{b}_3=\boldsymbol{d}_3^*,\quad \boldsymbol{b}_1^*=\eta^{-1}\boldsymbol{d}_1,\quad \boldsymbol{b}_2^*=\beta^{-1}\boldsymbol{d}_2,\quad \boldsymbol{b}_3^*=\boldsymbol{d}_3,\\ &\boldsymbol{b}_{0,1}=\eta\boldsymbol{d}_{0,1}^*,\boldsymbol{b}_{0,2}=\beta\boldsymbol{d}_{0,2}^*,\boldsymbol{b}_{0,3}=\boldsymbol{d}_{0,3}^*,\boldsymbol{b}_{0,1}^*=\eta^{-1}\boldsymbol{d}_{0,1},\boldsymbol{b}_{0,2}^*=\beta^{-1}\boldsymbol{d}_{0,2},\boldsymbol{b}_{0,3}^*=\boldsymbol{d}_{0,3},\end{aligned}$$

$$\begin{aligned}&\boldsymbol{b}_{x,1}=\eta\boldsymbol{d}_{x,1}^*,\quad \boldsymbol{b}_{x,2}=\eta\boldsymbol{d}_{x,2}^*,\quad \boldsymbol{b}_{x,3}=\beta\boldsymbol{d}_{x,3}^*,\quad \boldsymbol{b}_{x,4}=\beta\boldsymbol{d}_{x,4}^*,\quad \boldsymbol{b}_5=\boldsymbol{d}_5^*,\boldsymbol{b}_6=\boldsymbol{d}_6^*\ \forall x\in[\mathcal{U}],\\ &\boldsymbol{b}_{x,1}^*=\eta^{-1}\boldsymbol{d}_{x,1},\boldsymbol{b}_{x,2}^*=\eta^{-1}\boldsymbol{d}_{x,2},\boldsymbol{b}_{x,3}^*=\beta^{-1}\boldsymbol{d}_{x,3},\boldsymbol{b}_{x,4}^*=\beta^{-1}\boldsymbol{d}_{x,4},\boldsymbol{b}_5^*=\boldsymbol{d}_5,\boldsymbol{b}_6^*=\boldsymbol{d}_6\ \forall x\in[\mathcal{U}].\end{aligned}$$

We note that these are properly distributed because $(\mathbb{D},\mathbb{D}^*),(\mathbb{D}_0,\mathbb{D}_0^*)$, etc. are randomly chosen (up to sharing the same $\psi$ value).
$\mathcal{B}$ chooses random exponents

$$\theta,\ \alpha_1',\alpha_2'\in\mathbb{Z}_p,\ \ \{r_i,\ z_i,\ \alpha_{i,1}',\alpha_{i,2}'\ \in\mathbb{Z}_p\}_{i\in[n]},\ \ \{c_{j,1}',c_{j,2}',\ y_j,\ \upsilon_j\in\mathbb{Z}_p\}_{j\in[n]}.$$

Then $\mathcal{B}$ gives to $\mathcal{A}$ the following public parameter:

$$\begin{aligned}\Big(\ &g,h=g^\theta,\ \{h_j\}_{j\in[n]},\ g^{\boldsymbol{b}_1},g^{\boldsymbol{b}_2},h^{\boldsymbol{b}_1}=(g^{\boldsymbol{b}_1})^\theta,h^{\boldsymbol{b}_2}=(g^{\boldsymbol{b}_2})^\theta,\\ &\{h_j^{\boldsymbol{b}_1}=(g^{\boldsymbol{b}_1})^{\upsilon_j},h_j^{\boldsymbol{b}_2}=(g^{\boldsymbol{b}_2})^{\upsilon_j}\}_{j\in[n]},\ \ h^{\boldsymbol{b}_{0,1}}=(g^{\boldsymbol{b}_{0,1}})^\theta,h^{\boldsymbol{b}_{0,2}}=(g^{\boldsymbol{b}_{0,2}})^\theta,\\ &\{h^{\boldsymbol{b}_{x,1}}=(g^{\boldsymbol{b}_{x,1}})^\theta,\ldots,h^{\boldsymbol{b}_{x,4}}=(g^{\boldsymbol{b}_{x,4}})^\theta\}_{x\in\mathcal{U}},\ \ F_1=e_3(g^{\boldsymbol{d}_1},g^{\eta\boldsymbol{d}_1^*})^{\theta\alpha_1'},\ F_2=e_3(g^{\boldsymbol{d}_2},g^{\beta\boldsymbol{d}_2^*})^{\theta\alpha_2'},\\ &\{F_{1,j}=e_3(g^{\boldsymbol{d}_1},g^{\eta\boldsymbol{d}_1^*})^{\upsilon_j\alpha_1'},F_{2,j}=e_3(g^{\boldsymbol{d}_2},g^{\eta\boldsymbol{d}_2^*})^{\upsilon_j\alpha_2'}\}_{j\in[n]},\\ &\{\boldsymbol{G}_i=g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \boldsymbol{Z}_i=g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \ E_{i,1}=e_3(g^{\boldsymbol{d}_1},g^{\eta\boldsymbol{d}_1^*})^{\alpha_{i,1}'},E_{i,2}=e_3(g^{\boldsymbol{d}_2},g^{\beta\boldsymbol{d}_2^*})^{\alpha_{i,2}'}\}_{i\in[n]},\\ &\{\boldsymbol{H}_j=(g^{\boldsymbol{d}_1})^{c_{j,1}'}(g^{\boldsymbol{d}_2})^{c_{j,2}'},\ \boldsymbol{Y}_j=(\boldsymbol{H}_j)^{y_j}\}_{j\in[n]}\ \Big).\end{aligned}$$

Note that $\mathcal{B}$ implicitly sets

$$\alpha_1=\eta\alpha_1',\ \alpha_2=\beta\alpha_2',\ \ \{\alpha_{i,1}=\eta\alpha_{i,1}',\ \alpha_{i,2}=\beta\alpha_{i,2}'\}_{i\in[n]},\ \ \{c_{j,1}=\eta c_{j,1}',\ c_{j,2}=\beta c_{j,2}'\}_{j\in[n]}.$$

**Phase 1.** To respond to a query for $((i,j),S_{(i,j)})$, $\mathcal{B}$ produces a normal key as follows. It randomly chooses $\sigma_{i,j,1}',\sigma_{i,j,2}',\delta_{i,j,1}',\delta_{i,j,2}'\in\mathbb{Z}_p$, and outputs a private key $\mathsf{SK}_{(i,j),S_{(i,j)}}=\langle(i,j),S_{(i,j)},\boldsymbol{K}_{i,j},\boldsymbol{K}_{i,j}',$

$\boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\rangle$ as:

$$\begin{aligned}
\boldsymbol{K}_{i,j} =& g^{(\alpha_{i,1}+r_i c_{j,1})\boldsymbol{b}_1^*+(\alpha_{i,2}+r_i c_{j,2})\boldsymbol{b}_2^*}(hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*} \\
=& (g^{\boldsymbol{d}_1})^{\alpha'_{i,1}+r_i c'_{j,1}+(\theta+\upsilon_j)(\sigma'_{i,j,1}+\delta'_{i,j,1})}(g^{\boldsymbol{d}_2})^{\alpha'_{i,2}+r_i c'_{j,2}+(\theta+\upsilon_j)(\sigma'_{i,j,2}+\delta'_{i,j,2})}, \\
\boldsymbol{K}'_{i,j} =& (g^{\boldsymbol{d}_1})^{\alpha'_1+\sigma'_{i,j,1}+\delta'_{i,j,1}}(g^{\boldsymbol{d}_2})^{\alpha'_2+\sigma'_{i,j,2}+\delta'_{i,j,2}}, \ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i}, \\
\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}} =& \{h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}\}_{j'\in[n]\setminus\{j\}}, \\
=& \{g^{\boldsymbol{d}_1\upsilon_{j'}(\sigma'_{i,j,1}+\delta'_{i,j,1})}g^{\boldsymbol{d}_2\upsilon_{j'}(\sigma'_{i,j,2}+\delta'_{i,j,2})}\}_{j'\in[n]\setminus\{j\}}. \\
\boldsymbol{K}_{i,j,0} =& (g^{\boldsymbol{d}_{0,1}})^{\delta'_{i,j,1}}(g^{\boldsymbol{d}_{0,2}})^{\delta'_{i,j,2}}, \\
\boldsymbol{K}_{i,j,x} =& (g^{\boldsymbol{d}_{x,1}})^{\sigma'_{i,j,1}}(g^{\boldsymbol{d}_{x,2}})^{\sigma'_{i,j,1}}(g^{\boldsymbol{d}_{x,3}})^{\sigma'_{i,j,2}}(g^{\boldsymbol{d}_{x,4}})^{\sigma'_{i,j,2}} \ \forall x \in S_{(i,j)}.
\end{aligned}$$

Note that $\mathcal{B}$ implicitly sets

$$\sigma_{i,j,1} = \eta\sigma'_{i,j,1}, \ \sigma_{i,j,2} = \beta\sigma'_{i,j,2}, \ \ \delta_{i,j,1} = \eta\delta'_{i,j,1}, \ \delta_{i,j,2} = \beta\delta'_{i,j,2}.$$

**Challenge.** $\mathcal{A}$ submits to $\mathcal{B}$ a revocation list $R$, an LSSS matrix $(A,\rho)$ of size $l \times m$ and two equal length messages $M_0, M_1$, $\mathcal{B}$ produces the challenge ciphertext for index $(\bar{i}=1, \bar{j}=1)$ as follows. $\mathcal{B}$ first chooses random

$$\begin{aligned}
\kappa, \ \tau, \quad s_1,\ldots,s_n, \quad t_1,\ldots,t_n \ &\in \mathbb{Z}_p, \\
\boldsymbol{v}_c \ \in \mathbb{Z}_p^3, \quad \boldsymbol{w}_1,\ldots,\boldsymbol{w}_n \ &\in \mathbb{Z}_p^3, \\
\xi'_{1,1},\xi'_{1,2},\ldots,\xi'_{l,1},\xi'_{l,2} \ \in \mathbb{Z}_p, \quad \boldsymbol{u}'_1,\boldsymbol{u}'_2 \ &\in \mathbb{Z}_p^m,
\end{aligned}$$

where the first entries of $\boldsymbol{u}'_1$ and $\boldsymbol{u}'_2$ are equal to 0. It also chooses a random vector $\boldsymbol{u} \in \mathbb{Z}_p$ with first entry equal to 1, and chooses random exponents $\xi'_{1,3},\ldots,\xi'_{l,3} \in \mathbb{Z}_p$. $\mathcal{B}$ implicitly sets

$$\begin{aligned}
&\pi_1 = \tau_1, \ \pi_2 = \tau_2, \\
&\boldsymbol{u}_1 = \tau_1\boldsymbol{u} + \boldsymbol{u}'_1, \ \boldsymbol{u}_2 = \tau_2\boldsymbol{u} + \boldsymbol{u}'_2, \\
&\xi_{k,1} = \xi'_{k,3}\tau_1 + \xi'_{k,1}, \ \xi_{k,2} = \xi'_{k,3}\tau_2 + \xi'_{k,2} \ \forall k \in [l].
\end{aligned}$$

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$, then it chooses random $\boldsymbol{v}_1 \in \mathbb{Z}_p^3, \boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} \ for \ i = 2,\ldots,n$.
$\mathcal{B}$ chooses a random $b \in \{0,1\}$, then creates a ciphertext $\langle R, (A,\rho), \ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n, \ (\boldsymbol{P}_k)_{k=0}^l\rangle$ as follows (note that $\bar{i}=1, \bar{j}=1$):

1. For each $i \in [n]$: it sets

$$\begin{aligned}
&\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa}, \\
&\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = (h\prod_{j'\in\bar{R}_i} h_{j'})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}\boldsymbol{Z}_i^{t_i}\boldsymbol{T}_1^{\theta}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \\
&T_i = M_b\frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{({F_1}'{F_2}')^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}e_3(g^{\boldsymbol{d}_1},\boldsymbol{T}_1)^{\theta\alpha'_1}e_3(g^{\boldsymbol{d}_2},\boldsymbol{T}_1)^{\theta\alpha'_2}},
\end{aligned}$$

   where ${F_1}' = F_1\prod_{j'\in\bar{R}_i}F_{1,j'} = e_3(g^{\boldsymbol{d}_1},g^{\eta\boldsymbol{d}_1^*})^{\theta\alpha'_1}\prod_{j'\in\bar{R}_i}e_3(g^{\boldsymbol{d}_1},g^{\eta\boldsymbol{d}_1^*})^{\upsilon_j\alpha'_1}$ and ${F_2}' = F_2\prod_{j'\in\bar{R}_i}F_{2,j'} = e_3(g^{\boldsymbol{d}_2},g^{\eta\boldsymbol{d}_2^*})^{\theta\alpha'_2}\prod_{j'\in\bar{R}_i}e_3(g^{\boldsymbol{d}_2},g^{\eta\boldsymbol{d}_2^*})^{\upsilon_j\alpha'_2}$ respectively.

2. For each $j \in [n]$: it sets $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau \boldsymbol{v}_c} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j}, \;\; \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.
3. $$\begin{aligned} \boldsymbol{P}_0 =& \boldsymbol{T}_{0,1}^{\theta}, \\ \boldsymbol{P}_k =& \big( (\boldsymbol{T}_{\rho(k),1})^{A_k \cdot \boldsymbol{u} + \xi'_{k,3}} (T_{\rho(k),2})^{-\xi'_{k,3}} \\ & (g^{\eta \boldsymbol{d}^*_{\rho(k),1}})^{A_k \cdot \boldsymbol{u}'_1 + \xi'_{k,1}} (g^{\eta \boldsymbol{d}^*_{\rho(k),2}})^{-\xi'_{k,1}} (g^{\beta \boldsymbol{d}^*_{\rho(k),3}})^{A_k \cdot \boldsymbol{u}'_2 + \xi'_{k,2}} (g^{\beta \boldsymbol{d}^*_{\rho(k),4}})^{-\xi'_{k,2}} \big)^{\theta} \; \forall k \in [l]. \end{aligned}$$

**Phase 2.** Same with Phase 1.

If the exponents of the $\boldsymbol{T}$ terms *do not* include the $\tau_3$ terms, then $\boldsymbol{Q}'_i$ and $\boldsymbol{P}_0$ are in their normal forms, and the exponent vector of $\boldsymbol{P}_k$ is

$$\begin{aligned} & (A_k \cdot \tau_1 \boldsymbol{u} + A_k \cdot \boldsymbol{u}'_1 + \tau_1 \xi'_{k,3} + \xi'_{k,1}) \eta \boldsymbol{d}^*_{\rho(k),1} + (-\tau_1 \xi'_{k,3} - \xi'_{k,1}) \eta \boldsymbol{d}^*_{\rho(k),2} \\ & + (A_k \cdot \tau_2 \boldsymbol{u} + A_k \cdot \boldsymbol{u}'_2 + \tau_2 \xi'_{k,3} + \xi'_{k,2}) \beta \boldsymbol{d}^*_{\rho(k),3} + (-\tau_2 \xi'_{k,3} - \xi'_{k,2}) \beta \boldsymbol{d}^*_{\rho(k),4} \\ =& (A_k \cdot \boldsymbol{u}_1 + \xi_{k,1}) \boldsymbol{b}_{\rho(k),1} - \xi_{k,1} \boldsymbol{b}_{\rho(k),2} + (A_k \cdot \boldsymbol{u}_2 + \xi_{k,2}) \boldsymbol{b}_{\rho(k),3} - \xi_{k,2} \boldsymbol{b}_{\rho(k),4}. \end{aligned}$$

Thus we have a properly distributed normal ciphertext in this case.

If the exponents of the $T$ terms *do* include the $\tau_3$ terms, then $\boldsymbol{Q}'_i$ and $\boldsymbol{P}_0$ are in their semi-functional forms with $\pi_3 = \tau_3$, and the exponent vector of $\boldsymbol{P}_k$ is

$$\begin{aligned} & (A_k \cdot \boldsymbol{u}_1 + \xi_{k,1}) \boldsymbol{b}_{\rho(k),1} - \xi_{k,1} \boldsymbol{b}_{\rho(k),2} + (A_k \cdot \boldsymbol{u}_2 + \xi_{k,2}) \boldsymbol{b}_{\rho(k),3} - \xi_{k,2} \boldsymbol{b}_{\rho(k),4} \\ & + (A_k \cdot \tau_3 \boldsymbol{u} + \tau_3 \xi'_{k,3}) \boldsymbol{b}_{\rho(k),5} - \tau_3 \xi'_{k,3} \boldsymbol{b}_{\rho(k),6}. \end{aligned}$$

This is a properly distributed semi-functional ciphertext with $\boldsymbol{u}_3 = \tau_3 \boldsymbol{u}$ and $\xi_{k,3} = \tau_3 \xi'_{k,3}$. (Note that these values are distributed randomly and independently from $\boldsymbol{u}_1, \boldsymbol{u}_2, \xi_{k,1}, \xi_{k,2}$.)

Thus, when the $\tau_3$ terms are absent, $\mathcal{B}$ properly simulates $\mathsf{Game}_{real}$, and when the $\tau_3$ terms are present, $\mathcal{B}$ properly simulates $\mathsf{Game}_0$. As a result, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference in advantage between these games to gain a non-negligible advantage against the subspace assumption.

**Lemma 5.** *Under the subspace assumption, no PPT attacker can achieve a non-negligible difference in advantage between* $\mathsf{Game}_{t-1}$ *and* $\mathsf{Game}_t^N$ *for any* $t$ *from* 1 *to* $Q$.

*Proof.* Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between $\mathsf{Game}_{t-1}$ and $\mathsf{Game}_t^N$, we will create a PPT algorithm $\mathcal{B}$ to break the subspace assumption. We will employ the subspace assumption with parameters $m = \mathcal{U} + 2$, $n_i = 3, k_i = 1$ for two values of $i$, and $n_i = 6, k_i = 2$ for the rest of the values of $i$. In order to reconcile the notation of the assumption with the notation of our construction as conveniently as possible, we will denote the bases involved in the assumption by $(\mathbb{B}, \mathbb{B}^*), (\mathbb{B}_0, \mathbb{B}_0^*) \in Dual(Z_p^3, \psi)$ and $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*) \in Dual(Z_p^6, \psi)$. B is given (we will ignore $\mu_3$ because it will not be needed):

$$\begin{aligned} \mathbb{G}, p, g, \; g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, \; g^{\boldsymbol{b}_{0,1}}, g^{\boldsymbol{b}_{0,2}}, \; \{g^{\boldsymbol{b}_{x,1}}, g^{\boldsymbol{b}_{x,2}}, g^{\boldsymbol{b}_{x,3}}, g^{\boldsymbol{b}_{x,4}}\}_{x \in [\mathcal{U}]}, \\ g^{\eta \boldsymbol{b}_1^*}, g^{\beta \boldsymbol{b}_2^*}, g^{\boldsymbol{b}_3^*}, \; g^{\eta \boldsymbol{b}_{0,1}^*}, g^{\beta \boldsymbol{b}_{0,2}^*}, g^{\boldsymbol{b}_{0,3}^*}, \; \{g^{\eta \boldsymbol{b}_{x,1}^*}, g^{\eta \boldsymbol{b}_{x,2}^*}, g^{\beta \boldsymbol{b}_{x,3}^*}, g^{\beta \boldsymbol{b}_{x,4}^*}, g^{\boldsymbol{b}_{x,5}^*}, g^{\boldsymbol{b}_{x,6}^*}\}_{x \in [\mathcal{U}]}, \\ \boldsymbol{U}_1 = g^{\mu_1 \boldsymbol{b}_1 + \mu_2 \boldsymbol{b}_2 + \mu_3 \boldsymbol{b}_3}, \; \boldsymbol{U}_{0,1} = g^{\mu_1 \boldsymbol{b}_{0,1} + \mu_2 \boldsymbol{b}_{0,2} + \mu_3 \boldsymbol{b}_{0,3}}, \\ \{\boldsymbol{U}_{x,1} = g^{\mu_1 \boldsymbol{b}_{x,1} + \mu_2 \boldsymbol{b}_{x,3} + \mu_3 \boldsymbol{b}_{x,5}}, \; \boldsymbol{U}_{x,2} = g^{\mu_1 \boldsymbol{b}_{x,2} + \mu_2 \boldsymbol{b}_{x,4} + \mu_3 \boldsymbol{b}_{x,6}}\}_{x \in [\mathcal{U}]}, \\ \boldsymbol{T}_1, \; \boldsymbol{T}_{0,1}, \; \{\boldsymbol{T}_{x,1}, \boldsymbol{T}_{x,2}\}_{x \in [\mathcal{U}]}. \end{aligned}$$

The exponents of the unknown terms $\boldsymbol{T}_1, \boldsymbol{T}_{0,1}$ are distributed either as $\tau_1 \eta \boldsymbol{b}_1^* + \tau_2 \beta \boldsymbol{b}_2^*$ and $\tau_1 \eta \boldsymbol{b}_{0,1}^* + \tau_2 \beta \boldsymbol{b}_{0,2}^*$ respectively, or as $\tau_1 \eta \boldsymbol{b}_1^* + \tau_2 \beta \boldsymbol{b}_2^* + \tau_3 \boldsymbol{b}_3^*$ and $\tau_1 \eta \boldsymbol{b}_{0,1}^* + \tau_2 \beta \boldsymbol{b}_{0,2}^* + \tau_3 \boldsymbol{b}_{0,3}^*$ respectively. Similarly, the exponents of the unknown terms $\boldsymbol{T}_{x,1}, \boldsymbol{T}_{x,2}$ are distributed either as $\tau_1 \eta \boldsymbol{b}_{x,1}^* + \tau_2 \beta \boldsymbol{b}_{x,3}^*$ and $\tau_1 \eta \boldsymbol{b}_{x,2}^* + \tau_2 \beta \boldsymbol{b}_{x,4}^*$ respectively, or as $\tau_1 \eta \boldsymbol{b}_{x,1}^* + \tau_2 \beta \boldsymbol{b}_{x,3}^* + \tau_3 \boldsymbol{b}_{x,5}^*$ and $\tau_1 \eta \boldsymbol{b}_{x,2}^* + \tau_2 \beta \boldsymbol{b}_{x,4}^* + \tau_3 \boldsymbol{b}_{x,6}^*$ respectively. It is $\mathcal{B}$'s task to determine if these $\tau_3$ contributions are present or not.

**Setup.** B implicitly sets $(\mathbb{B}, \mathbb{B}^*), (\mathbb{B}_0, \mathbb{B}_0^*), \{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ as the bases for the construction.

$\mathcal{B}$ chooses random exponents

$$\theta,\ \alpha_1', \alpha_2' \in \mathbb{Z}_p,\quad \{r_i,\ z_i,\ \alpha_{i,1}', \alpha_{i,2}'\ \in \mathbb{Z}_p\}_{i\in[n]},\quad \{c_{j,1}', c_{j,2}',\ y_j, \upsilon_j\ \in \mathbb{Z}_p\}_{j\in[n]}.$$

Then $\mathcal{B}$ gives to $\mathcal{A}$ the following public parameter:

$$\Big(\ g, h = g^{\theta}, \{h_j\}_{j\in[n]},\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\theta}, h^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\theta},$$
$$\{h_j^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\upsilon_j}, h_j^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\upsilon_j}\}_{j\in[n]},\quad h^{\boldsymbol{b}_{0,1}} = (g^{\boldsymbol{b}_{0,1}})^{\theta}, h^{\boldsymbol{b}_{0,2}} = (g^{\boldsymbol{b}_{0,2}})^{\theta},$$
$$\{h^{\boldsymbol{b}_{x,1}} = (g^{\boldsymbol{b}_{x,1}})^{\theta}, \ldots, h^{\boldsymbol{b}_{x,4}} = (g^{\boldsymbol{b}_{x,4}})^{\theta}\}_{x\in[\mathcal{U}]},\quad F_1 = e_3(g^{\boldsymbol{b}_1}, g^{\eta\boldsymbol{b}_1^*})^{\theta\alpha_1'},\ F_2 = e_3(g^{\boldsymbol{b}_2}, g^{\beta\boldsymbol{b}_2^*})^{\theta\alpha_2'},$$
$$\{F_{1,j} = e_3(g^{\boldsymbol{b}_1}, g^{\eta\boldsymbol{b}_1^*})^{\upsilon_j\alpha_1'}, F_{2,j} = e_3(g^{\boldsymbol{b}_2}, g^{\eta\boldsymbol{b}_2^*})^{\upsilon_j\alpha_2'}\}_{j\in[n]},$$
$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad E_{i,1} = e_3(g^{\boldsymbol{b}_1}, g^{\eta\boldsymbol{b}_1^*})^{\alpha_{i,1}'}, E_{i,2} = e_3(g^{\boldsymbol{b}_2}, g^{\beta\boldsymbol{b}_2^*})^{\alpha_{i,2}'}\}_{i\in[n]},$$
$$\{\boldsymbol{H}_j = (g^{\eta\boldsymbol{b}_1^*})^{c_{j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{c_{j,2}'},\ \boldsymbol{Y}_j = (\boldsymbol{H}_j)^{y_j}\}_{j\in[n]}\ \Big).$$

Note that $\mathcal{B}$ implicitly sets

$$\alpha_1 = \eta\alpha_1',\ \alpha_2 = \beta\alpha_2',\quad \{\alpha_{i,1} = \eta\alpha_{i,1}',\ \alpha_{i,2} = \beta\alpha_{i,2}'\}_{i\in[n]},\quad \{c_{j,1} = \eta c_{j,1}',\ c_{j,2} = \beta c_{j,2}'\}_{j\in[n]}.$$

**Phase 1.** To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ acts as follows.

- If it is in the first $t-1$ key queries, $\mathcal{B}$ generates a semi-functional key as follow. $\mathcal{B}$ randomly chooses $\delta_{i,j,1}', \delta_{i,j,2}', \sigma_{i,j,1}', \sigma_{i,j,2}', \gamma \in \mathbb{Z}_p$, and outputs a private key $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}_{i,j}', \boldsymbol{K}_{i,j}'', \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ as:

$$\begin{aligned}\boldsymbol{K}_{i,j} =& (g^{\eta\boldsymbol{b}_1^*})^{\alpha_{i,1}'+r_ic_{j,1}'+(\theta+\upsilon_j)(\sigma_{i,j,1}'+\delta_{i,j,1}')}(g^{\beta\boldsymbol{b}_2^*})^{\alpha_{i,2}'+r_ic_{j,2}'+(\theta+\upsilon_j)(\sigma_{i,j,2}'+\delta_{i,j,2}')}g^{(\theta+\upsilon_j)\gamma\boldsymbol{b}_3^*},\\ \boldsymbol{K}_{i,j}' =& (g^{\eta\boldsymbol{b}_1^*})^{\alpha_1'+\sigma_{i,j,1}'+\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{\alpha_2'+\sigma_{i,j,2}'+\delta_{i,j,2}'}g^{\gamma\boldsymbol{b}_3^*},\ \boldsymbol{K}_{i,j}'' = (\boldsymbol{K}_{i,j}')^{z_i},\\ \{\bar{\boldsymbol{K}}_{i,j,j'} =& (g^{\eta\boldsymbol{b}_1^*})^{\upsilon_{j'}(\sigma_{i,j,1}'+\delta_{i,j,1}')}(g^{\beta\boldsymbol{b}_2^*})^{\upsilon_{j'}(\sigma_{i,j,2}'+\delta_{i,j,2}')}g^{(\theta+\upsilon_{j'})\gamma\boldsymbol{b}_3^*}\}_{j'\in[n]\setminus\{j\}},\\ \boldsymbol{K}_{i,j,0} =& (g^{\eta\boldsymbol{b}_{0,1}^*})^{\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_{0,2}^*})^{\delta_{i,j,2}'},\\ \boldsymbol{K}_{i,j,x} =& (g^{\eta\boldsymbol{b}_{x,1}^*})^{\sigma_{i,j,1}'}(g^{\eta\boldsymbol{b}_{x,2}^*})^{\sigma_{i,j,1}'}(g^{\beta\boldsymbol{b}_{x,3}^*})^{\sigma_{i,j,2}'}(g^{\beta\boldsymbol{b}_{x,4}^*})^{\sigma_{i,j,2}'}\ \forall x \in S_{(i,j)}.\end{aligned}$$

  Note that this is a properly distributed semi-functional key with implicitly setting

$$\sigma_{i,j,1} = \eta\sigma_{i,j,1}',\ \sigma_{i,j,2} = \beta\sigma_{i,j,2}',\quad \delta_{i,j,1} = \eta\delta_{i,j,1}',\ \delta_{i,j,2} = \beta\delta_{i,j,2}'.$$

- If it is the $t^{th}$ key query: $\mathcal{B}$ randomly chooses $\delta_{i,j,1}', \delta_{i,j,2}', \delta_{i,j,3}' \in \mathbb{Z}_p$, and outputs a private key $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}_{i,j}', \boldsymbol{K}_{i,j}'', \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}})$ as:

$$\begin{aligned}\boldsymbol{K}_{i,j} =& (g^{\eta\boldsymbol{b}_1^*})^{\alpha_{i,1}'+r_ic_{j,1}'+(\theta+\upsilon_j)\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{\alpha_{i,2}'+r_ic_{j,2}'+(\theta+\upsilon_j)\delta_{i,j,2}'}\boldsymbol{T}_1^{(\theta+\upsilon_j)}\boldsymbol{T}_1^{(\theta+\upsilon_j)\delta_{i,j,3}'},\\ \boldsymbol{K}_{i,j}' =& (g^{\eta\boldsymbol{b}_1^*})^{\alpha_1'+\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{\alpha_2'+\delta_{i,j,2}'}\boldsymbol{T}_1(\boldsymbol{T}_1)^{\delta_{i,j,3}'},\ \boldsymbol{K}_{i,j}'' = (\boldsymbol{K}_{i,j}')^{z_i},\\ \{\bar{\boldsymbol{K}}_{i,j,j'} =& (g^{\eta\boldsymbol{b}_1^*})^{\upsilon_{j'}\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{\upsilon_{j'}\delta_{i,j,2}'}\boldsymbol{T}_1^{\upsilon_{j'}}\boldsymbol{T}_1^{\upsilon_{j'}\delta_{i,j,3}'}\}_{j'\in[n]\setminus\{j\}},\\ \boldsymbol{K}_{i,j,0} =& (g^{\eta\boldsymbol{b}_{0,1}^*})^{\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_{0,2}^*})^{\delta_{i,j,2}'}\boldsymbol{T}_{0,1}^{\delta_{i,j,3}'},\\ \boldsymbol{K}_{i,j,x} =& \boldsymbol{T}_{x,1}\boldsymbol{T}_{x,2}\ \forall x \in S_{(i,j)}.\end{aligned}$$

  Note that $\mathcal{B}$ implicitly sets

$$\sigma_{i,j,1} = \eta\tau_1,\ \sigma_{i,j,2} = \beta\tau_2,\quad \delta_{i,j,1} = \eta(\delta_{i,j,1}' + \delta_{i,j,3}'\tau_1),\ \delta_{i,j,2} = \beta(\delta_{i,j,2}' + \delta_{i,j,3}'\tau_2).$$

If the exponents of the $T$ terms *do not* include the $\tau_3$ terms, then this is a properly distributed normal key. If they *do* include the $\tau_3$ terms, then this is a properly distributed nominal semi-functional key with $\sigma_{i,j,3} = \tau_3$ and $\delta_{i,j,3} = \delta'_{i,j,3}\tau_3$. (Note that these values are distributed randomly and independently from $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2}$.)

- If it is in the $\{t+1, \dots, Q\}$ key queries: $\mathcal{B}$ generates a normal key as follows. $\mathcal{B}$ randomly chooses $\delta'_{i,j,1}, \delta'_{i,j,2}, \sigma'_{i,j,1}, \sigma'_{i,j,2} \in \mathbb{Z}_p$, and outputs a private key $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S_{(i,j)}}\ \rangle$ as:

$$\begin{aligned}
\boldsymbol{K}_{i,j} =& (g^{\eta \boldsymbol{b}_1^*})^{\alpha'_{i,1} + r_i c'_{j,1} + (\theta + \upsilon_j)(\sigma'_{i,j,1} + \delta'_{i,j,1})} (g^{\beta \boldsymbol{b}_2^*})^{\alpha'_{i,2} + r_i c'_{j,2} + (\theta + \upsilon_j)(\sigma'_{i,j,2} + \delta'_{i,j,2})},\\
\boldsymbol{K}'_{i,j} =& (g^{\eta \boldsymbol{b}_1^*})^{\alpha'_1 + \sigma'_{i,j,1} + \delta'_{i,j,1}} (g^{\beta \boldsymbol{b}_2^*})^{\alpha'_2 + \sigma'_{i,j,2} + \delta'_{i,j,2}},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},\\
\{\bar{\boldsymbol{K}}_{i,j,j'} =& (g^{\eta \boldsymbol{b}_1^*})^{\upsilon_{j'}(\sigma'_{i,j,1} + \delta'_{i,j,1})} (g^{\beta \boldsymbol{b}_2^*})^{\upsilon_{j'}(\sigma'_{i,j,2} + \delta'_{i,j,2})}\}_{j' \in [n] \setminus \{j\}},\\
\boldsymbol{K}_{i,j,0} =& (g^{\eta \boldsymbol{b}_{0,1}^*})^{\delta'_{i,j,1}} (g^{\beta \boldsymbol{b}_{0,2}^*})^{\delta'_{i,j,2}},\\
\boldsymbol{K}_{i,j,x} =& (g^{\eta \boldsymbol{b}_{x,1}^*})^{\sigma'_{i,j,1}} (g^{\eta \boldsymbol{b}_{x,2}^*})^{\sigma'_{i,j,1}} (g^{\beta \boldsymbol{b}_{x,3}^*})^{\sigma'_{i,j,2}} (g^{\beta \boldsymbol{b}_{x,4}^*})^{\sigma'_{i,j,2}}\ \forall x \in S_{(i,j)}.
\end{aligned}$$

Note that this is a properly distributed normal key with implicitly setting

$$\sigma_{i,j,1} = \eta \sigma'_{i,j,1},\ \sigma_{i,j,2} = \beta \sigma'_{i,j,2},\ \ \delta_{i,j,1} = \eta \delta'_{i,j,1},\ \delta_{i,j,2} = \beta \delta'_{i,j,2}.$$

**Challenge.** $\mathcal{A}$ submits to $\mathcal{B}$ a revocation list $R$, an LSSS matrix $(A, \rho)$ of size $l \times m$ and two equal length messages $M_0, M_1$, $\mathcal{B}$ produces a semi-functional ciphertext for index $(\bar{i} = 1, \bar{j} = 1)$ as follows. $\mathcal{B}$ first chooses random

$$\begin{aligned}
\kappa,\ \tau,\quad s_1, \dots, s_n,\quad t_1, \dots, t_n\ &\in \mathbb{Z}_p,\\
\boldsymbol{v}_c\ \in \mathbb{Z}_p^3,\quad \boldsymbol{w}_1, \dots, \boldsymbol{w}_n\ &\in \mathbb{Z}_p^3,\\
\xi'_{1,1}, \xi'_{1,2}, \dots, \xi'_{l,1}, \xi'_{l,2}\ \in \mathbb{Z}_p,\quad \boldsymbol{u}'_1, \boldsymbol{u}'_2\ &\in \mathbb{Z}_p^m,
\end{aligned}$$

where the first entries of $\boldsymbol{u}'_1$ and $\boldsymbol{u}'_2$ are equal to 0. It also chooses a random vector $\boldsymbol{u} \in \mathbb{Z}_p^m$ with first entry equal to 1, and chooses random exponents $\xi'_{1,3}, \dots, \xi'_{l,3} \in \mathbb{Z}_p$. $\mathcal{B}$ implicitly sets

$$\begin{aligned}
&\pi_1 = \mu_1,\ \pi_2 = \mu_2,\ \pi_3 = \mu_3,\\
&\boldsymbol{u}_1 = \mu_1 \boldsymbol{u} + \boldsymbol{u}'_1,\ \boldsymbol{u}_2 = \mu_2 \boldsymbol{u} + \boldsymbol{u}'_2,\ \boldsymbol{u}_3 = \mu_3 \boldsymbol{u},\\
&\xi_{k,1} = \xi'_{k,1} + \xi'_{k,3}\mu_1,\ \xi_{k,2} = \xi'_{k,2} + \xi'_{k,3}\mu_2,\ \xi_{k,3} = \xi'_{k,3}\mu_3\ \forall k \in [l].
\end{aligned}$$

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$, then it chooses random $\boldsymbol{v}_1 \in \mathbb{Z}_p^3, span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ for\ i = 2, \dots, n$.
$\mathcal{B}$ chooses a random $b \in \{0,1\}$, then creates a ciphertext $\langle R, (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows (note that $\bar{i} = 1, \bar{j} = 1$):

1. For each $i \in [n]$: it sets

$$\begin{aligned}
&\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i \boldsymbol{v}_i},\quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa},\\
&\boldsymbol{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1 + \boldsymbol{b}_2)},\quad \boldsymbol{Q}'_i = (h \prod_{j' \in \bar{R}_i} h_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1 + \boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} U_1^{\theta},\quad \boldsymbol{Q}''_i = g^{t_i (\boldsymbol{b}_1 + \boldsymbol{b}_2)},\\
&T_i = M_b \frac{(E_{i,1} E_{i,2})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}}{(F'_1 F'_2)^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} e_3(U_1, g^{\eta \boldsymbol{b}_1^*})^{\theta \alpha'_1} e_3(U_1, g^{\eta \boldsymbol{b}_2^*})^{\theta \alpha'_2}},
\end{aligned}$$

where ${F_1}' = F_1 \prod_{j' \in \bar{R}_i} F_{1,j'} = e_3(g^{\boldsymbol{b}_1}, g^{\eta \boldsymbol{b}_1^*})^{\theta \alpha_1'} \prod_{j' \in \bar{R}_i} e_3(g^{\boldsymbol{b}_1}, g^{\eta \boldsymbol{b}_1^*})^{\upsilon_{j'} \alpha_1'}$ and ${F_2}' = F_2 \prod_{j' \in \bar{R}_i} F_{2,j'} = e_3(g^{\boldsymbol{b}_2}, g^{\eta \boldsymbol{b}_2^*})^{\theta \alpha_2'} \prod_{j' \in \bar{R}_i} e_3(g^{\boldsymbol{b}_2}, g^{\eta \boldsymbol{b}_2^*})^{\upsilon_{j'} \alpha_2'}$ respectively.

2. For each $j \in [n]$: it sets $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau \boldsymbol{v}_c} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j}$, $\boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.
3.

$$\begin{aligned} \boldsymbol{P}_0 =& \boldsymbol{U}_{0,1}^{\theta}, \\ \boldsymbol{P}_k =& \big((g^{\boldsymbol{b}_{\rho(k),1}})^{A_k \cdot \boldsymbol{u}_1' + \xi_{k,1}'} (g^{\boldsymbol{b}_{\rho(k),2}})^{-\xi_{k,1}'} \\ & (g^{\boldsymbol{b}_{\rho(k),3}})^{A_k \cdot \boldsymbol{u}_2' + \xi_{k,2}'} (g^{\boldsymbol{b}_{\rho(k),4}})^{-\xi_{k,2}'} \boldsymbol{U}_{\rho(k),1}^{A_k \cdot \boldsymbol{u} + \xi_{k,3}'} \boldsymbol{U}_{\rho(k),2}^{-\xi_{k,3}'} \big)^{\theta} \ \forall k \in [l]. \end{aligned}$$

**Phase 2.** Same with Phase 1.

Thus, when the $\tau_3$ terms are absent, $\mathcal{B}$ properly simulates $\mathsf{Game}_{t-1}$, and when the $\tau_3$ terms are present, $\mathcal{B}$ properly simulates $\mathsf{Game}_t^N$. As a result, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference in advantage between these games to gain a non-negligible advantage against the subspace assumption.

**Lemma 6.** *Under the D3DH assumption, no PPT attacker can achieve a non-negligible difference in advantage between* $\mathsf{Game}_t^N$ *and* $\mathsf{Game}_t^T$ *for any* $t$ *from* 1 *to* $Q_1$ *(recall these are all the Phase 1 queries).*

*Proof.* Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between $\mathsf{Game}_t^N$ and $\mathsf{Game}_t^T$ from some $t$ between 1 and $Q_1$, we will create a PPT algorithm $\mathcal{B}$ to break the D3DH assumption. $\mathcal{B}$ is given $g, g^x, g^y, g^z, T$, where $T$ is either $g^{xyz}$ or a random element of $\mathbb{G}$. $\mathcal{B}$ will simulate either $\mathsf{Game}_t^N$ or $\mathsf{Game}_t^T$ with $\mathcal{A}$ depending on the nature of $T$.

**Setup.** $\mathcal{B}$ chooses random dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$, $(\mathbb{D}_0, \mathbb{D}_0^*)$ of dimension 3 and $(\mathbb{D}_x, \mathbb{D}_x^*)$ of dimension 6, all with the same value of $\psi$. It then implicitly sets $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{B}_0, \mathbb{B}_0^*)$ as follows:

$$\begin{array}{l} \boldsymbol{b}_1 = \boldsymbol{d}_1, \quad \boldsymbol{b}_2 = \boldsymbol{d}_2, \quad \boldsymbol{b}_3 = (xy)^{-1} \boldsymbol{d}_3, \quad \boldsymbol{b}_1^* = \boldsymbol{d}_1^*, \quad \boldsymbol{b}_2^* = \boldsymbol{d}_2^*, \quad \boldsymbol{b}_3^* = (xy) \boldsymbol{d}_3^*, \\ \boldsymbol{b}_{0,1} = \boldsymbol{d}_{0,1}, \boldsymbol{b}_{0,2} = \boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3} = (xy)^{-1} \boldsymbol{d}_{0,3}, \boldsymbol{b}_{0,1}^* = \boldsymbol{d}_{0,1}^*, \boldsymbol{b}_{0,2}^* = \boldsymbol{d}_{0,2}^*, \boldsymbol{b}_{0,3}^* = (xy) \boldsymbol{d}_{0,3}^*. \end{array}$$

We note $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{B}_0, \mathbb{B}_0^*)$ are properly distributed.

$\mathcal{B}$ sets the *normal* portions of $(\mathbb{B}_1, \mathbb{B}_1^*), \dots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ as follows:

$$\begin{array}{l} \boldsymbol{b}_{x,1} = \boldsymbol{d}_{x,1}, \boldsymbol{b}_{x,2} = \boldsymbol{d}_{x,2}, \boldsymbol{b}_{x,3} = \boldsymbol{d}_{x,3}, \boldsymbol{b}_{x,4} = \boldsymbol{d}_{x,4} \ \forall x \in [\mathcal{U}], \\ \boldsymbol{b}_{x,1}^* = \boldsymbol{d}_{x,1}^*, \boldsymbol{b}_{x,2}^* = \boldsymbol{d}_{x,2}^*, \boldsymbol{b}_{x,3}^* = \boldsymbol{d}_{x,3}^*, \boldsymbol{b}_{x,4}^* = \boldsymbol{d}_{x,4}^* \ \forall x \in [\mathcal{U}]. \end{array}$$

The semi-functional portions of these bases will be set later (at which point we may verify that all of $(\mathbb{B}_1, \mathbb{B}_1^*), \dots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ are properly distributed).

$\mathcal{B}$ chooses $\theta, \alpha_1, \alpha_2, r_i, \alpha_{i,1}, \alpha_{i,2}, z_i (i \in [n]), c_{j,1}, c_{j,2}, y_j, \upsilon_j (j \in [n]) \in \mathbb{Z}_p$ randomly. We observe that $\mathcal{B}$ can now produce the public parameter (with $h = g^{\theta}, \{h_j = g^{\upsilon_j}\}_{j \in [n]}$), and also know the master secret key (enabling it to create normal keys). It gives the public parameter to $\mathcal{A}$.

**Phase 1.** To create the first $t - 1$ semi-functional keys in response to $\mathcal{A}$'s key requests, $\mathcal{B}$ first creates a normal key, then chooses a random exponent $\gamma' \in \mathbb{Z}_p$, and multiples $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}$ and $\boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}$ by $g^{(\theta + \upsilon_j) \gamma' \boldsymbol{d}_3^*}, g^{\gamma' \boldsymbol{d}_3^*}$ and $g^{z_i \gamma' \boldsymbol{d}_3^*}, g^{\upsilon_{j'} \gamma' \boldsymbol{d}_3^*}$ respectively. We are using here that $\mathcal{B}$ does not need to know $g^{\boldsymbol{b}_3^*}$ precisely in order to create well-distributed semi-functional keys – it suffices for $\mathcal{B}$ to know $g^{c \boldsymbol{b}_3^*}$ for some (non-zero) $c \in \mathbb{Z}_p$.

$\mathcal{A}$ requests the $t^{th}$ key for some pair $((i_t, j_t), S_{(i_t,j_t)})$ where $S_{(i_t,j_t)} \subseteq [\mathcal{U}]$. At this point, B implicitly defines the semi-functional parts of the bases $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ as follows (note that these have not been involved in the game before this):

$$\begin{aligned}&\boldsymbol{b}_{x,5} = x^{-1}\boldsymbol{d}_{x,5},\, \boldsymbol{b}_{x,6} = \boldsymbol{d}_{x,6},\, \boldsymbol{b}^*_{x,5} = x\boldsymbol{d}^*_{x,5},\, \boldsymbol{b}^*_{x,6} = \boldsymbol{d}^*_{x,6}\ \forall x \notin S_{(i_t,j_t)},\\ &\boldsymbol{b}_{x,5} = \boldsymbol{d}_{x,5}, \qquad \boldsymbol{b}_{x,6} = \boldsymbol{d}_{x,6},\, \boldsymbol{b}^*_{x,5} = \boldsymbol{d}^*_{x,5}, \quad \boldsymbol{b}^*_{x,6} = \boldsymbol{d}^*_{x,6}\ \forall x \in S_{(i_t,j_t)}.\end{aligned}$$

We observe that all of $(\mathbb{B}, \mathbb{B}^*), (\mathbb{B}_0, \mathbb{B}_0^*), (\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ are properly distributed, and their distribution is independent of $x, y$, and $S_{(i_t,j_t)}$ (the involvement of $x, y$, and $S_{(i_t,j_t)}$ is only present in $\mathcal{B}$'s view and is information-theoretically hidden from $\mathcal{A}$, see [12, Lemma 11]).

To create the $t^{th}$ key, $\mathcal{B}$ chooses random exponents $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2}, \delta'_{i,j,3} \in \mathbb{Z}_p$, then forms the key as

$$\begin{aligned}\boldsymbol{K}_{i,j} =& (g^{\boldsymbol{d}_1^*})^{\alpha_{i,1}+r_i c_{j,1}+(\theta+\upsilon_j)(\sigma_{i,j,1}+\delta_{i,j,1})}(g^{\boldsymbol{d}_2^*})^{\alpha_{i,2}+r_i c_{j,2}+(\theta+\upsilon_j)(\sigma_{i,j,2}+\delta_{i,j,2})}T^{(\theta+\upsilon_j)\boldsymbol{d}_3^*}g^{(\theta+\upsilon_j)\delta'_{i,j,3}\boldsymbol{d}_3^*},\\ \boldsymbol{K}'_{i,j} =& (g^{\boldsymbol{d}_1^*})^{\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1}}(g^{\boldsymbol{d}_2^*})^{\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2}}T^{\boldsymbol{d}_3^*}g^{\delta'_{i,j,3}\boldsymbol{d}_3^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},\\ \{\bar{\boldsymbol{K}}_{i,j,j'} =& (g^{\boldsymbol{d}_1^*})^{\upsilon_{j'}(\sigma_{i,j,1}+\delta_{i,j,1})}(g^{\boldsymbol{d}_2^*})^{\upsilon_{j'}(\sigma_{i,j,2}+\delta_{i,j,2})}T^{\upsilon_{j'}\boldsymbol{d}_3^*}g^{\upsilon_{j'}\delta'_{i,j,3}\boldsymbol{d}_3^*}\}_{j'\in[n]\setminus\{j\}},\\ \boldsymbol{K}_{i,j,0} =& (g^{\boldsymbol{d}^*_{0,1}})^{\delta_{i,j,1}}(g^{\boldsymbol{d}^*_{0,2}})^{\delta_{i,j,2}}g^{\delta'_{i,j,3}\boldsymbol{d}^*_{0,3}},\\ \boldsymbol{K}_{i,j,x} =& (g^{\boldsymbol{d}^*_{x,1}})^{\sigma_{i,j,1}}(g^{\boldsymbol{d}^*_{x,2}})^{\sigma_{i,j,1}}(g^{\boldsymbol{d}^*_{x,3}})^{\sigma_{i,j,2}}(g^{\boldsymbol{d}^*_{x,4}})^{\sigma_{i,j,2}}(g^z)^{\boldsymbol{d}^*_{x,5}+\boldsymbol{d}^*_{x,6}}\ \forall x \in S_{(i_t,j_t)}.\end{aligned}$$

If $T = g^{xyz}$, this is a properly distributed nominal semi-functional key with $\sigma_{i,j,3} = z, \delta_{i,j,3} = (xy)^{-1}\delta'_{i,j,3}$. Otherwise, this is a properly distributed temporary semi-functional key.

**Challenge.** At some *later* point, $\mathcal{A}$ submits $\mathcal{B}$ a revocation list $R$, an LSSS matrix $(A, \rho)$ of size $l \times m$ and two equal length messages $M_0, M_1$, $\mathcal{B}$ produces a semi-functional ciphertext for index $(\bar{i} = 1, \bar{j} = 1)$ as follows.

Note that $S_{(i_t,j_t)}$ does not satisfy $(A, \rho)$, $\mathcal{B}$ first computes a vector $\boldsymbol{w} \in \mathbb{Z}_p^m$ that has first entry equal to 1 and is orthogonal to all of the rows $A_k$ of $A$ such that $\rho(k) \in S_{(i_t,j_t)}$ (such a vector must exist since $S_{(i_t,j_t)}$ fails to satisfy $(A, \rho)$, and it is efficiently computable). $\mathcal{B}$ also chooses a random vector $\boldsymbol{u}'_3 \in \mathbb{Z}_p^m$ subject to the constraint that the first entry is zero. It implicitly sets $\pi_3 = xy$ and sets $\boldsymbol{u}_3 = xy\boldsymbol{w} + x\boldsymbol{u}'_3$. We note that $\pi_3$ is random because all of the dual orthonormal bases are distributed independently of $x, y$, and $\boldsymbol{u}_3$ is distributed as a random vector with first entry equal to $\pi_3$. $\mathcal{B}$ also chooses random values $\xi_{k,3} \in \mathbb{Z}_p$ for all $k$ such that $\rho(k) \in S_{(i_t,j_t)}$ and random values $\xi'_{k,3} \in \mathbb{Z}_p$ for all $k$ such that $\rho(k) \notin S_{(i_t,j_t)}$. For values of $k$ such that $\rho(k) \notin S_{(i_t,j_t)}$, it implicitly sets $\xi_{k,3} = x\xi'_{k,3}$. $\mathcal{B}$ can then produce the semi-functional components of the ciphertext as it can compute:

$$\begin{aligned}g^{\pi_3\boldsymbol{b}_3} &= g^{\boldsymbol{d}_3},\ g^{\pi_3\boldsymbol{b}_{0,3}} = g^{\boldsymbol{d}_{0,3}},\\ g^{(A_k\cdot\boldsymbol{u}_3+\xi_{k,3})\boldsymbol{b}_{\rho(k),5}-\xi_{k,3}\boldsymbol{b}_{\rho(k),6}} &= (g^y)^{(A_k\cdot\boldsymbol{w})\boldsymbol{d}_{\rho(k),5}}g^{(A_k\cdot\boldsymbol{u}'_3+\xi'_{k,3})\boldsymbol{d}_{\rho(k),5}}(g^x)^{-\xi'_{k,3}\boldsymbol{d}_{\rho(k),6}}\ \forall k\ s.t.\ \rho(k) \notin S_{(i_t,j_t)},\\ g^{(A_k\cdot\boldsymbol{u}_3+\xi_{k,3})\boldsymbol{b}_{\rho(k),5}-\xi_{k,3}\boldsymbol{b}_{\rho(k),6}} &= (g^x)^{(A_k\cdot\boldsymbol{u}'_3)\boldsymbol{d}_{\rho(k),5}}g^{\xi_{k,3}\boldsymbol{d}_{\rho(k),5}-\xi_{k,3}\boldsymbol{d}_{\rho(k),6}}\ \forall k\ s.t.\ \rho(k) \in S_{(i_t,j_t)}.\end{aligned}$$

Here we have used the fact that $A_k \cdot \boldsymbol{w} \equiv 0 \bmod p$ to avoid needing to produce a multiple of $g^{xy\boldsymbol{d}_{\rho(k),5}}$ for $k$ such that $\rho(k) \in S_{(i_t,j_t)}$.

Note that $h = g^\theta$ and $\mathcal{B}$ knows the value of $\theta$, $\mathcal{B}$ can produce the semi-functional components using the value of $\theta$ and the above values. Then it multiplies these semi-functional components by the normal components to form the semi-functional ciphertext, which is given to $\mathcal{A}$.

**Phase 2.** $\mathcal{B}$ can respond to $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

If $T = g^{xyz}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_t^N$, and if $T$ is a random group element, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_t^T$. Thus, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference in advantage between these games to gain a non-negligible advantage against the D3DH assumption.

**Lemma 7.** *Under the source group q-parallel BDHE assumption, no PPT attacker can achieve a non-negligible difference in advantage between* $\mathsf{Game}_t^N$ *and* $\mathsf{Game}_t^T$ *for a* $t > Q_1$ *using a revocation list* $R \subseteq [N]$*, and an access matrix* $(A, \rho)$ *of size* $l \times m$ *where* $l, m \leq q$*.*

*Proof.* Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between $\mathsf{Game}_t^N$ and $\mathsf{Game}_t^T$ for some $t$ such that $Q_1 < t \leq Q$ using an access matrix with dimensions $\leq q$, we will create a PPT algorithm $\mathcal{B}$ to break the source group $q$-parallel BDHE assumption. $\mathcal{B}$ is given: $g, g^f, g^{df}, g^{c^i}\ \forall i \in [2q] \setminus \{q+1\}, g^{c^i/b_j}\ \forall i \in [2q] \setminus \{q+1\}, j \in [q], g^{dfb_j}\ \forall j \in [q], g^{dfc^i b_{j'}/b_j}\ \forall i \in [q], j, j' \in [q], j \neq j'$, and $T$, where $T$ is either equal to $g^{dc^{q+1}}$ or is a random element of $\mathbb{G}$. $\mathcal{B}$ will simulate either $\mathsf{Game}_t^N$ or $\mathsf{Game}_t^T$ with $\mathcal{A}$ depending on the nature of $T$.

**Setup.** $\mathcal{B}$ chooses random dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$, $(\mathbb{D}_0, \mathbb{D}_0^*)$ of dimension 3 and $(\mathbb{D}_x, \mathbb{D}_x^*)$ of dimension 6, all with the same value of $\psi$. It then implicitly sets $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{B}_0, \mathbb{B}_0^*)$ as follows:

$$\boldsymbol{b}_1 = \boldsymbol{d}_1, \quad \boldsymbol{b}_2 = \boldsymbol{d}_2, \quad \boldsymbol{b}_3 = (cd)^{-1}\boldsymbol{d}_3, \quad \boldsymbol{b}_1^* = \boldsymbol{d}_1^*, \quad \boldsymbol{b}_2^* = \boldsymbol{d}_2^*, \quad \boldsymbol{b}_3^* = (cd)\boldsymbol{d}_3^*,$$
$$\boldsymbol{b}_{0,1} = \boldsymbol{d}_{0,1}, \boldsymbol{b}_{0,2} = \boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3} = (c)^{-1}\boldsymbol{d}_{0,3}, \boldsymbol{b}_{0,1}^* = \boldsymbol{d}_{0,1}^*, \boldsymbol{b}_{0,2}^* = \boldsymbol{d}_{0,2}^*, \boldsymbol{b}_{0,3}^* = (c)\boldsymbol{d}_{0,3}^*.$$

We note $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{B}_0, \mathbb{B}_0^*)$ are properly distributed.

$\mathcal{B}$ sets the *normal* portions of $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ as follows:

$$\boldsymbol{b}_{x,1} = \boldsymbol{d}_{x,1}, \boldsymbol{b}_{x,2} = \boldsymbol{d}_{x,2}, \boldsymbol{b}_{x,3} = \boldsymbol{d}_{x,3}, \boldsymbol{b}_{x,4} = \boldsymbol{d}_{x,4}\ \forall x \in [\mathcal{U}],$$
$$\boldsymbol{b}_{x,1}^* = \boldsymbol{d}_{x,1}^*, \boldsymbol{b}_{x,2}^* = \boldsymbol{d}_{x,2}^*, \boldsymbol{b}_{x,3}^* = \boldsymbol{d}_{x,3}^*, \boldsymbol{b}_{x,4}^* = \boldsymbol{d}_{x,4}^*\ \forall x \in [\mathcal{U}].$$

The semi-functional portions of these bases will be set later (at which point we may verify that all of $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ are properly distributed).

$\mathcal{B}$ chooses $\theta, \alpha_1, \alpha_2, r_i, z_i, \alpha_{i,1}, \alpha_{i,2} (i \in [n]), c_{j,1}, c_{j,2}, y_j, \upsilon_j (j \in [n]) \in \mathbb{Z}_p$ randomly. We observe that $\mathcal{B}$ can now produce the public parameter (with $h = g^\theta, \{h_j = g^{\upsilon_j}\}_{j \in [n]}$), and also know the master secret key (enabling it to create normal keys). It gives the public parameter to $\mathcal{A}$.

**Phase 1.** To create the first $Q_1$ semi-functional keys in response to $\mathcal{A}$'s key requests, $\mathcal{B}$ first creates a normal key, then chooses a random exponent $\gamma' \in \mathbb{Z}_p$, and multiples $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}_{i,j}'$, $\boldsymbol{K}_{i,j}''$ and $\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}$ by $g^{(\theta + \upsilon_j)\gamma' \boldsymbol{d}_3^*}, g^{\gamma' \boldsymbol{d}_3^*}, g^{z_i \gamma' \boldsymbol{d}_3^*}$ and $g^{\upsilon_{j'} \gamma' \boldsymbol{d}_3^*}$ respectively. As in the proof of the previous lemma, we note here that $\mathcal{B}$ does not need to know $g^{\boldsymbol{b}_3^*}$ precisely in order to create well-distributed semi-functional keys.

**Challenge.** Before requesting the $t^{th}$ key, $\mathcal{A}$ will request the challenge ciphertext for some revocation list $R \subseteq [N]$ and access matrix $(A, \rho)$ of size $l \times m$, where both $l, m \leq q$. For each attribute $x \in [\mathcal{U}]$, we let $J_x$ denote the set of indices $k \in [l]$ such that $\rho(k) = x$. For each attribute $x \in [\mathcal{U}]$, $\mathcal{B}$ chooses a random value $\eta_x' \in \mathbb{Z}_p$ and defines a value $\eta_x$ by

$$\eta_x = \eta_x' + \sum_{k \in J_x} cA_{k,1}/b_k + \cdots + c^m A_{k,m}/b_k.$$

At this point, $\mathcal{B}$ implicitly sets the semi-functional portions of the bases $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ as follows (note that these have played no role in the game before this point):

$$\boldsymbol{b}_{x,5} = \boldsymbol{d}_{x,5}, \boldsymbol{b}_{x,6} = \eta_x^{-1}\boldsymbol{d}_{x,6}, \ \boldsymbol{b}_{x,5}^* = \boldsymbol{d}_{x,5}^*, \boldsymbol{b}_{x,6}^* = \eta_x \boldsymbol{d}_{x,6}^*\ \forall x \in [\mathcal{U}].$$

We observe that all of $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ are properly distributed.

$\mathcal{B}$ produces a semi-functional ciphertext for index $(\bar{i} = 1, \bar{j} = 1)$ as follows.

To create the challenge ciphertext, $\mathcal{B}$ first creates a normal ciphertext using the normal encryption algorithm. To create the semi-functional components, it implicitly sets $\pi_3 = cdf$. It also chooses random values $u'_2, \dots, u'_m \in \mathbb{Z}_p$ and random values $\xi'_{k,3} \in \mathbb{Z}_p$ for each $k \in [l]$. It implicitly sets $\boldsymbol{u}_3 = (cdf, dfc^2 + u'_2, \dots, dfc^m + u'_m)$. [2] This is distributed as a random vector with first entry equal to $\pi_3$. For each $k \in [l]$, $\mathcal{B}$ implicitly sets $\xi_{k,3} = -dfb_k\eta_{\rho(k)} + \xi'_{k,3}\eta_{\rho(k)}$. These are distributed as uniformly random elements because each $\xi'_{k,3}$ is random and $\eta_{\rho(k)} \neq 0$ (with all but negligible probability). We observe:

$$\begin{aligned} A_k \cdot \boldsymbol{u}_3 + \xi_{k,3} =& df(cA_{k,1} + c^2 A_{k,2} + \dots, c^m A_{k,m}) + A_{k,2}u'_2 + \dots + A_{k,m}u'_m \\ & - dfb_k(\eta'_{\rho(k)} + \sum_{k' \in J_{\rho(k)}} cA_{k',1}/b_{k'} + \dots + c^m A_{k',m}/b_{k'}) + \xi'_{k,3}\eta_{\rho(k)}. \end{aligned}$$

By definition, $k \in J_{\rho(k)}$, so we have some cancelation here:

$$\begin{aligned} A_k \cdot \boldsymbol{u}_3 + \xi_{k,3} =& A_{k,2}u'_2 + \dots + A_{k,m}u'_m \\ & - dfb_k(\eta'_{\rho(k)} + \sum_{k' \in J_{\rho(k)} \setminus \{k\}} cA_{k',1}/b_{k'} + \dots + c^m A_{k',m}/b_{k'}) + \xi'_{k,3}\eta_{\rho(k)}. \end{aligned}$$

We now see that $\mathcal{B}$ can compute $g^{A_k \cdot \boldsymbol{u}_3 + \xi_{k,3}}$ using the terms it is given in the assumption, enabling it to produce $g^{(A_k \cdot \boldsymbol{u}_3 + \xi_{k,3})\boldsymbol{b}_{\rho(k),5}} = g^{(A_k \cdot \boldsymbol{u}_3 + \xi_{k,3})\boldsymbol{d}_{\rho(k),5}}$. We also see that

$$-\xi_{k,3}\boldsymbol{b}_{\rho(k),6} = -\xi_{k,3}\eta^{-1}_{\rho(k)}\boldsymbol{d}_{\rho(k),6} = (dfb_k - \xi'_{k,3})\boldsymbol{d}_{\rho(k),6},$$

so $\mathcal{B}$ can also produce $g^{-\xi_{k,3}\boldsymbol{b}_{\rho(k),6}}$. In this way, $\mathcal{B}$ can produce the semi-functional component of $\boldsymbol{P}_k$ for each $k \in [l]$ with the proper distribution, as $h = g^\theta$ and $\mathcal{B}$ knows the value of $\theta$.
$\mathcal{B}$ also produces the semi-functional components of $\boldsymbol{Q}'_i$ and $\boldsymbol{P}_0$ as it can compute:

$$g^{\pi_3 \boldsymbol{b}_3} = (g^f)^{\boldsymbol{d}_3}, \quad g^{\pi_3 \boldsymbol{b}_{0,3}} = (g^{df})^{\boldsymbol{d}_{0,3}}.$$

It gives the resulting properly distributed semi-functional ciphertext to $\mathcal{A}$.

**Phase 2.** To create the $Q_1^{th}, \dots, (t-1)^{th}$ semi-functional keys in response to $\mathcal{A}$'s key requests, $\mathcal{B}$ first creates a normal key, then chooses a random exponent $\gamma' \in \mathbb{Z}_p$, and multiples $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}'_{i,j}$, $\boldsymbol{K}''_{i,j}$ and $\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}$ by $g^{(\theta + \upsilon_j)\gamma' \boldsymbol{d}^*_3}, g^{\gamma' \boldsymbol{d}^*_3}, g^{z_i \gamma' \boldsymbol{d}^*_3}$ and $g^{\upsilon_{j'}\gamma' \boldsymbol{d}^*_3}$ respectively. As in the proof of the previous lemma, we note here that $\mathcal{B}$ does not need to know $g^{\boldsymbol{b}^*_3}$ precisely in order to create well-distributed semi-functional keys.

$\mathcal{A}$ requests the $t^{th}$ key for some pair $((i_t, j_t), S_{(i_t,j_t)})$ where $S_{(i_t,j_t)} \subseteq [\mathcal{U}]$. $\mathcal{B}$ can create the normal parts of the key using the normal key generation algorithm. To create the semi-functional parts, $\mathcal{B}$ proceeds as follows. Since $S_{(i_t,j_t)}$ does not satisfy $(A, \rho)$, $\mathcal{B}$ can (efficiently) compute a vector $\boldsymbol{w} = (w_1, \dots, w_m) \in \mathbb{Z}_p^m$ such that its first entry is non-zero and $\boldsymbol{w}$ is orthogonal (modulo $p$) to all rows $A_k$ of $A$ such that $\rho(k) \in S_{(i_t,j_t)}$. We may assume the first entry of $\boldsymbol{w}$ is randomized. $\mathcal{B}$ implicitly sets $\sigma_{i,j,3} = w_1 c^q + \dots + w_m c^{q-m+1}$, which is properly distributed because $w_1$ is random (and $c$ is non-zero with all but negligible probability). $\mathcal{B}$ also chooses a random value $\delta'_{i,j,3}$ and implicitly sets $\delta_{i,j,3} = -w_2 c^{q-1} - \dots - w_m c^{q-m+1} + fc^{-1}\delta'_{i,j,3}$. This is properly distributed because $\delta'_{i,j,3}$ is random (and $fc^{-1}$ is non-zero with all but negligible probability).

We observe that

$$(\sigma_{i,j,3} + \delta_{i,j,3})\boldsymbol{b}^*_3 = (w_1 dc^{q+1} + df\delta'_{i,j,3})\boldsymbol{d}^*_3.$$

[2] Note that this is assuming that $m \geq 2$. For the case of $m = 1$, we will set $\boldsymbol{u}_3 = (cdf)$, $\sigma_{i,j,3} = w_1 c^q$, and $\delta_{i,j,3} = fc^{-1}\delta'_{i,j,3}$, and it can be verified that the following proof follows as well.

$\mathcal{B}$ forms the semi-functional part of $\boldsymbol{K}'_{i,j}$ as: $T^{w_1\boldsymbol{d}_3^*}(g^{df})^{\delta'_{i,j,3}\boldsymbol{d}_3^*}$. If $T = g^{dc^{q+1}}$, this is equal to $g^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}_3^*}$, as required for a nominal semi-functional key. Otherwise, this exponent is distributed as a random multiple of $\boldsymbol{b}_3^*$, as required for a temporary semi-functional key. $\mathcal{B}$ forms the semi-functional parts of $\boldsymbol{K}_{i,j}, \boldsymbol{K}''_{i,j}$ and $\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\backslash\{j\}}$ as $(T^{w_1\boldsymbol{d}_3^*}(g^{df})^{\delta'_{i,j,3}\boldsymbol{d}_3^*})^{(\theta+\upsilon_j)}$, $(T^{w_1\boldsymbol{d}_3^*}(g^{df})^{\delta'_{i,j,3}\boldsymbol{d}_3^*})^{z_{i_t}}$ and $(T^{w_1\boldsymbol{d}_3^*}(g^{df})^{\delta'_{i,j,3}\boldsymbol{d}_3^*})^{\upsilon_{j'}}$ respectively. We also have

$$\delta_{i,j,3}\boldsymbol{b}_{0,3}^* = (-w_2c^q - \cdots - w_mc^{q-m+2} + f\delta'_{i,j,3})\boldsymbol{d}_{0,3}^*,$$

enabling $\mathcal{B}$ to produce $g^{\delta_{i,j,3}\boldsymbol{b}_{0,3}^*}$ using the terms given in the assumption.
Now, $\mathcal{B}$ can also produce $g^{\sigma_{i,j,3}}$, and hence can compute $g^{\sigma_{i,j,3}\boldsymbol{b}_{x,5}^*} = g^{\sigma_{i,j,3}\boldsymbol{d}_{x,5}^*}$ for each $x \in S_{i_t,j_t}$. We observe

$$\begin{aligned}\sigma_{i,j,3}\boldsymbol{b}_{x,6}^* &= \sigma_{i,j,3}\eta_x\boldsymbol{d}_{x,6}^*, \text{ and}\\ \sigma_{i,j,3}\eta_x &= (w_1c^q + \cdots + w_mc^{q-m+1})(\eta'_x + \sum_{k\in J_x} cA_{k,1}/b_k + \cdots + c^mA_{k,m}/b_k).\end{aligned}$$

For each $k \in J_x$, we have $\rho(k) = x$. So for $x \in S_{(i_t,j_t)}$, we have $A_k \cdot \boldsymbol{w} = 0$ modulo $p$ for every $k \in J_x$. Thus, all of the terms involving $c^{q+1}$ cancel, and we are left with terms that can be created in the exponent from the group elements given in the assumption (note that $m \le q$, so $2q$ is an upper bound on the powers of $c$ involved here). This shows that $\mathcal{B}$ can create $g^{\sigma_{i,j,3}\boldsymbol{b}_{x,6}^*}$ for all $x \in S_{(i_t,j_t)}$, and hence can produce properly distributed semi-functional components for each $\boldsymbol{K}_{i,j,x}$ of the $t^{th}$ key.
$\mathcal{B}$ can respond to the rest of $\mathcal{A}$'s key requests by producing normal keys via the normal key generation algorithm.

If $T = g^{dc^{q+1}}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_t^N$, and if $T$ is distributed randomly, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_t^T$. Thus, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference in advantage between these games to achieve a non-negligible advantage against the source group $q$-parallel BDHE assumption.

**Lemma 8.** *Under the subspace assumption, no PPT attacker can achieve a non-negligible difference in advantage between* $\mathsf{Game}_t^T$ *and* $\mathsf{Game}_t$ *for any* $t$ *from* $1$ *to* $Q$.

*Proof.* This proof is almost identical to the proof of Lemma 5, except that $\mathcal{B}$ adds an additional terms of $g^{(\theta+\upsilon_j)\gamma\boldsymbol{b}_3^*}$, $g^{\gamma\boldsymbol{b}_3^*}, g^{z_i\gamma\boldsymbol{b}_3^*}$ and $g^{\upsilon_{j'}\gamma\boldsymbol{b}_3^*}$ to $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}$ and $\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\backslash\{j\}}$ respectively for the $t^{th}$ key (where it chooses $\gamma \in \mathbb{Z}_p$ randomly). This ensures that when the $\tau_3$ terms are not present, the $t^{th}$ key will be a properly distributed semi-functional key.

**Lemma 9.** *Under the subspace assumption, no PPT attacker can achieve a non-negligible difference in advantage between* $\mathsf{Game}_Q$ *and* $\mathsf{Game}_{final}$.

*Proof.* Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between $\mathsf{Game}_Q$ and $\mathsf{Game}_{final}$, we will create a PPT algorithm $\mathcal{B}$ to break the subspace assumption. We will employ the subspace assumption with parameters $m = \mathcal{U}+2$, $n_i = 3, k_i = 1$ for two values of $i$, and $n_i = 6, k_i = 2$ for the rest of the values of $i$. To coincide with our notation for the construction, we will denote the bases involved in the assumption by $(\mathbb{B}, \mathbb{B}^*), (\mathbb{B}_0, \mathbb{B}_0^*) \in Dual(Z_p^3, \psi)$ and $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*) \in$

$Dual(Z_p^6, \psi)$. B is given (we will ignore $\mu_3$ and $\boldsymbol{T}_{0,1}, \{\boldsymbol{T}_{x,1}, \boldsymbol{T}_{x,2}\}_{x\in[\mathcal{U}]}$ because they do not be needed):

$$\begin{aligned}
&\mathbb{G}, p, g,\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2},\ g^{\boldsymbol{b}_{0,1}}, g^{\boldsymbol{b}_{0,2}},\ \{g^{\boldsymbol{b}_{x,1}}, g^{\boldsymbol{b}_{x,2}}, g^{\boldsymbol{b}_{x,3}}, g^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]},\\
&g^{\eta\boldsymbol{b}_1^*}, g^{\beta\boldsymbol{b}_2^*}, g^{\boldsymbol{b}_3^*},\ g^{\eta\boldsymbol{b}_{0,1}^*}, g^{\beta\boldsymbol{b}_{0,2}^*}, g^{\boldsymbol{b}_{0,3}^*},\ \{g^{\eta\boldsymbol{b}_{x,1}^*}, g^{\eta\boldsymbol{b}_{x,2}^*}, g^{\beta\boldsymbol{b}_{x,3}^*}, g^{\beta\boldsymbol{b}_{x,4}^*}, g^{\boldsymbol{b}_{x,5}^*}, g^{\boldsymbol{b}_{x,6}^*}\}_{x\in[\mathcal{U}]},\\
&\boldsymbol{U}_1 = g^{\mu_1\boldsymbol{b}_1+\mu_2\boldsymbol{b}_2+\mu_3\boldsymbol{b}_3},\ \boldsymbol{U}_{0,1} = g^{\mu_1\boldsymbol{b}_{0,1}+\mu_2\boldsymbol{b}_{0,2}+\mu_3\boldsymbol{b}_{0,3}},\\
&\{\boldsymbol{U}_{x,1} = g^{\mu_1\boldsymbol{b}_{x,1}+\mu_2\boldsymbol{b}_{x,3}+\mu_3\boldsymbol{b}_{x,5}},\ \boldsymbol{U}_{x,2} = g^{\mu_1\boldsymbol{b}_{x,2}+\mu_2\boldsymbol{b}_{x,4}+\mu_3\boldsymbol{b}_{x,6}}\}_{x\in[\mathcal{U}]},\\
&\boldsymbol{T}_1.
\end{aligned}$$

The exponent of the unknown term $\boldsymbol{T}_1$ is distributed either as $\tau_1\eta\boldsymbol{b}_1^*+\tau_2\beta\boldsymbol{b}_2^*$, or as $\tau_1\eta\boldsymbol{b}_1^*+\tau_2\beta\boldsymbol{b}_2^*+\tau_3\boldsymbol{b}_3^*$. It is $\mathcal{B}$'s task to determine if this $\tau_3$ contribution is present or not.

**Setup.** $\mathcal{B}$ sets $(\mathbb{B}, \mathbb{B}^*), (\mathbb{B}_0, \mathbb{B}_0^*), \{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ as the bases for the construction. $\mathcal{B}$ chooses random exponents

$$\theta, \alpha_1', \alpha_2' \ \in \mathbb{Z}_p,\quad \{r_i,\ z_i,\ \alpha_{i,1}', \alpha_{i,2}' \ \in \mathbb{Z}_p\}_{i\in[n]},\quad \{c_{j,1}', c_{j,2}', y_j, \upsilon_j \ \in \mathbb{Z}_p\}_{j\in[n]}.$$

Then $\mathcal{B}$ gives to $\mathcal{A}$ the following public parameter:

$$\begin{aligned}
\Big(\ & g, h = g^\theta,\ \{h_j\}_{j\in[n]},\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^\theta, h^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^\theta,\\
& \{h_j^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\upsilon_j}, h_j^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\upsilon_j}\}_{j\in[n]},\quad h^{\boldsymbol{b}_{0,1}} = (g^{\boldsymbol{b}_{0,1}})^\theta, h^{\boldsymbol{b}_{0,2}} = (g^{\boldsymbol{b}_{0,2}})^\theta,\\
& \{h^{\boldsymbol{b}_{x,1}} = (g^{\boldsymbol{b}_{x,1}})^\theta, \ldots, h^{\boldsymbol{b}_{x,4}} = (g^{\boldsymbol{b}_{x,4}})^\theta\}_{x\in[\mathcal{U}]},\quad F_1 = e_3(g^{\boldsymbol{b}_1}, T_1)^\theta,\ F_2 = e_3(g^{\boldsymbol{b}_2}, T_1)^\theta,\\
& \{F_{1,j} = e_3(g^{\boldsymbol{b}_1}, T_1)^{\upsilon_j}, F_{2,j} = e_3(g^{\boldsymbol{b}_2}, T_1)^{\upsilon_j}\}_{j\in[n]},\\
& \{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\
& E_{i,1} = e_3(g^{\boldsymbol{b}_1}, T_1^\theta)e_3(g^{\boldsymbol{b}_1}, g^{\eta\boldsymbol{b}_1^*})^{\alpha_{i,1}'}, E_{i,2} = e_3(g^{\boldsymbol{b}_2}, T_1^\theta)e_3(g^{\boldsymbol{b}_2}, g^{\beta\boldsymbol{b}_2^*})^{\alpha_{i,2}'}\}_{i\in[n]},\\
& \{\boldsymbol{H}_j = (g^{\eta\boldsymbol{b}_1^*})^{c_{j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{c_{j,2}'},\ \boldsymbol{Y}_j = (\boldsymbol{H}_j)^{y_j}\}_{j\in[n]}\ \Big).
\end{aligned}$$

Note that $\mathcal{B}$ implicitly sets

$$\begin{aligned}
&\alpha_1 = \eta\tau_1, \alpha_2 = \beta\tau_2,\\
&\{\alpha_{i,1} = \eta\big((\theta+\upsilon_j)\tau_1 + \alpha_{i,1}'\big),\ \alpha_{i,2} = \beta\big((\theta+\upsilon_j)\tau_2 + \alpha_{i,2}'\big)\}_{i,j\in[n]},\quad \{c_{j,1} = \eta c_{j,1}',\ c_{j,2} = \beta c_{j,2}'\}_{j\in[n]}.
\end{aligned}$$

**Phase 1.** To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ generates a semi-functional key as follow. $\mathcal{B}$ randomly chooses $\delta_{i,j,1}', \delta_{i,j,2}', \sigma_{i,j,1}', \sigma_{i,j,2}', \gamma' \in \mathbb{Z}_p$, and outputs a private key $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}_{i,j}', \boldsymbol{K}_{i,j}'', \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ as:

$$\begin{aligned}
\boldsymbol{K}_{i,j} =& \boldsymbol{T}_1^{(\theta+\upsilon_j)}(g^{\eta\boldsymbol{b}_1^*})^{\alpha_{i,1}'+r_ic_{j,1}'+(\theta+\upsilon_j)(\sigma_{i,j,1}'+\delta_{i,j,1}')}(g^{\beta\boldsymbol{b}_2^*})^{\alpha_{i,2}'+r_ic_{j,2}'+(\theta+\upsilon_j)(\sigma_{i,j,2}'+\delta_{i,j,2}')}g^{(\theta+\upsilon_j)\gamma'\boldsymbol{b}_3^*},\\
\boldsymbol{K}_{i,j}' =& \boldsymbol{T}_1(g^{\eta\boldsymbol{b}_1^*})^{\sigma_{i,j,1}'+\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_2^*})^{\sigma_{i,j,2}'+\delta_{i,j,2}'}g^{\gamma'\boldsymbol{b}_3^*},\ \boldsymbol{K}_{i,j}'' = (\boldsymbol{K}_{i,j}')^{z_i},\\
\{\bar{\boldsymbol{K}}_{i,j,j'} =& \boldsymbol{T}_1^{\upsilon_{j'}}(g^{\eta\boldsymbol{b}_1^*})^{\upsilon_{j'}(\sigma_{i,j,1}'+\delta_{i,j,1}')}(g^{\beta\boldsymbol{b}_2^*})^{\upsilon_{j'}(\sigma_{i,j,2}'+\delta_{i,j,2}')}g^{\upsilon_{j'}\gamma'\boldsymbol{b}_3^*}\}_{j'\in[n]\setminus\{j\}},\\
\boldsymbol{K}_{i,j,0} =& (g^{\eta\boldsymbol{b}_{0,1}^*})^{\delta_{i,j,1}'}(g^{\beta\boldsymbol{b}_{0,2}^*})^{\delta_{i,j,2}'},\\
\boldsymbol{K}_{i,j,x} =& (g^{\eta\boldsymbol{b}_{x,1}^*})^{\sigma_{i,j,1}'}(g^{\eta\boldsymbol{b}_{x,2}^*})^{\sigma_{i,j,1}'}(g^{\beta\boldsymbol{b}_{x,3}^*})^{\sigma_{i,j,2}'}(g^{\beta\boldsymbol{b}_{x,4}^*})^{\sigma_{i,j,2}'}\ \forall x \in S_{(i,j)}.
\end{aligned}$$

Note that this is a properly distributed semi-functional key with implicitly setting

$$\sigma_{i,j,1} = \eta\sigma_{i,j,1}',\ \sigma_{i,j,2} = \beta\sigma_{i,j,2}',\quad \delta_{i,j,1} = \eta\delta_{i,j,1}',\ \delta_{i,j,2} = \beta\delta_{i,j,2}'.$$

We note that the multiple of $\boldsymbol{b}_3^*$ appearing in the exponent of $\boldsymbol{K}'_{i,j}(\boldsymbol{K}_{i,j}, \boldsymbol{K}''_{i,j}, \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j' \in [n] \setminus \{j\}}, resp.)$ is either equal to $\gamma'$ ( $\gamma'$, $z_i\gamma'$, $\gamma'$, resp.) or $\gamma' + \tau_3$ ($\gamma' + \tau_3$, $z_i(\gamma' + \tau_3)$, $\gamma' + \tau_3$, resp.), depending on the nature of $\boldsymbol{T}_1$. Either way, this is a properly distributed semi-functional key (whose distribution is independent of $\tau_3$ even if it is present).

**Challenge.** $\mathcal{A}$ submits $\mathcal{B}$ a revocation list $R$, an LSSS matrix $(A, \rho)$ of size $l \times m$ and two equal length messages $M_0, M_1$. To create the semi-functional ciphertext $\mathcal{B}$ can use the same procedure employed in the proof of Lemma 5 to use the $\boldsymbol{U}$ terms to provide the semi-functional components. We repeat the description of this procedure below for the reader's convenience. The only difference here comes in computing the blinding factor for $\boldsymbol{T}_i$.

$\mathcal{B}$ produces a semi-functional ciphertext for index $(\bar{i} = 1, \bar{j} = 1)$ as follows.

$\mathcal{B}$ first chooses random

$$\begin{aligned} \kappa,\ \tau,\quad s_1, \dots, s_n,\quad t_1, \dots, t_n\ &\in \mathbb{Z}_p, \\ \boldsymbol{v}_c\ \in \mathbb{Z}_p^3,\quad \boldsymbol{w}_1, \dots, \boldsymbol{w}_n\ &\in \mathbb{Z}_p^3, \\ \xi'_{1,1}, \xi'_{1,2}, \dots, \xi'_{l,1}, \xi'_{l,2}\ \in \mathbb{Z}_p,\quad \boldsymbol{u}'_1, \boldsymbol{u}'_2\ &\in \mathbb{Z}_p^m, \end{aligned}$$

where the first entries of $\boldsymbol{u}'_1$ and $\boldsymbol{u}'_2$ are equal to 0. It also chooses a random vector $\boldsymbol{u} \in \mathbb{Z}_p^m$ with first entry equal to 1, and chooses random exponents $\xi'_{1,3}, \dots, \xi'_{l,3} \in \mathbb{Z}_p$. $\mathcal{B}$ implicitly sets

$$\begin{aligned} &\pi_1 = \mu_1,\ \pi_2 = \mu_2,\ \pi_3 = \mu_3, \\ &\boldsymbol{u}_1 = \mu_1 \boldsymbol{u} + \boldsymbol{u}'_1,\ \boldsymbol{u}_2 = \mu_2 \boldsymbol{u} + \boldsymbol{u}'_2,\ \boldsymbol{u}_3 = \mu_3 \boldsymbol{u}, \\ &\xi_{k,1} = \xi'_{k,3}\mu_1 + \xi'_{k,1},\ \xi_{k,2} = \xi'_{k,3}\mu_2 + \xi'_{k,2},\ \xi_{k,3} = \xi'_{k,3}\mu_3\ \forall k \in [l]. \end{aligned}$$

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$, then it chooses random $\boldsymbol{v}_1 \in \mathbb{Z}_p^3, \boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ for\ i = 2, \dots, n$.

$\mathcal{B}$ chooses a random $b \in \{0, 1\}$, then creates a ciphertext $\langle R, (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows (note that $\bar{i} = 1, \bar{j} = 1$):

1. For each $i \in [n]$: set

$$\begin{aligned} &\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i \boldsymbol{v}_i},\quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa}, \\ &\boldsymbol{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1 + \boldsymbol{b}_2)},\quad \boldsymbol{Q}'_i = (h \prod_{j' \in \bar{R}_i} h_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1 + \boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} \boldsymbol{U}_1^{\theta},\quad \boldsymbol{Q}''_i = g^{t_i (\boldsymbol{b}_1 + \boldsymbol{b}_2)}, \\ &T_i = M_b \frac{(E_{i,1} E_{i,2})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}}{(F'_1 F'_2)^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} e_3(\boldsymbol{U}_1, \boldsymbol{T}_1)^{\theta}}, \end{aligned}$$

where ${F_1}' = F_1 \prod_{j' \in \bar{R}_i} F_{1,j'} = e_3(g^{\boldsymbol{b}_1}, \boldsymbol{T_1})^{\theta\alpha'_1} \prod_{j' \in \bar{R}_i} e_3(g^{\boldsymbol{b}_1}, \boldsymbol{T_1})^{\upsilon_{j'}\alpha'_1}$ and ${F_2}' = F_2 \prod_{j' \in \bar{R}_i} F_{2,j'} = e_3(g^{\boldsymbol{b}_2}, \boldsymbol{T_1})^{\theta\alpha'_2} \prod_{j' \in \bar{R}_i} e_3(g^{\boldsymbol{b}_2}, \boldsymbol{T_1})^{\upsilon_{j'}\alpha'_2}$ respectively.

2. For each $j \in [n]$: set $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau \boldsymbol{v}_c} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j},\quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.

3. Set

$$\begin{aligned} \boldsymbol{P}_0 =& \boldsymbol{U}_{0,1}^{\theta}, \\ \boldsymbol{P}_k =& \big((g^{\boldsymbol{b}_{\rho(k),1}})^{A_k \cdot \boldsymbol{u}'_1 + \xi'_{k,1}} (g^{\boldsymbol{b}_{\rho(k),2}})^{-\xi'_{k,1}} \cdot (g^{\boldsymbol{b}_{\rho(k),3}})^{A_k \cdot \boldsymbol{u}'_2 + \xi'_{k,2}} (g^{\boldsymbol{b}_{\rho(k),4}})^{-\xi'_{k,2}} \boldsymbol{U}_{\rho(k),1}^{A_k \cdot \boldsymbol{u} + \xi'_{k,3}} \boldsymbol{U}_{\rho(k),2}^{-\xi'_{k,3}}\big)^{\theta}\ \forall k \in [l]. \end{aligned}$$

If the exponent of $\boldsymbol{T}_1$ is equal to $\tau_1 \eta \boldsymbol{b}_1^* + \tau \beta \boldsymbol{b}_2^*$ then we have

$$e_3(\boldsymbol{U}_1, \boldsymbol{T}_1)^{\theta} = e(g, g^{\theta})^{\psi(\tau_1 \eta \mu_1 + \tau_2 \beta \mu_2)} = e(g, h)^{\psi(\alpha_1 \pi_1 + \alpha_2 \pi_2)} = F_1^{\pi_1} F_2^{\pi_2},$$

and hence we have a properly distributed semi-functional encryption of $M_b$, as required in $\mathsf{Game}_Q$. If instead the exponent of $T_1$ is equal to $\tau_1\eta\boldsymbol{b}_1^* + \tau\beta\boldsymbol{b}_2^* + \tau_3\boldsymbol{b}_3^*$, then we have

$$e_3(\boldsymbol{U}_1, \boldsymbol{T}_1)^\theta = e(g, g^\theta)^{\psi(\tau_1\eta\mu_1+\tau_2\beta\mu_2+\tau_3\mu_3)} = e(g,h)^{\psi(\alpha_1\pi_1+\alpha_2\pi_2+\tau_3\mu_3)} = F_1^{\pi_1}F_2^{\pi_2}e(g,h)^{\tau_3\mu_3}.$$

Since $\tau_3$ is random (and independent of the semi-functional keys and the rest of the ciphertext), this blinding factor is distributed as a freshly random group element of $\mathbb{G}_T$. Therefore the ciphertext is distributed as a semi-functional encryption of a random message, as required in $\mathsf{Game}_{final}$.

**Phase 2.** Same with Phase 1.

Thus, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference in advantage between these games to achieve a non-negligible advantage against the subspace assumption.

### B.2 Proof of Lemma 1

*Proof.* Suppose there exists a PPT adversary $\mathcal{A}$ that breaks the Index Hiding Game with advantage $\epsilon$. We build a simulator $\mathcal{B}$ to solve a D3DH problem instance. $\mathcal{B}$ flips a random $coin \in \{0,1\}$, if $c = 0$, $\mathcal{B}$ interacts with $\mathcal{A}$ in **Case A** as guessing "$\mathcal{A}$ is not in **Case II.3**", otherwise $\mathcal{B}$ interacts with $\mathcal{A}$ in **Case B** as guessing "$\mathcal{A}$ is not in **Case II.1**".

**Case A**: $\mathcal{B}$ receives the D3DH challenge from the challenger as $((p, \mathbb{G}, \mathbb{G}_T, e), g, A = g^a, B = g^b, C = g^c, T)$, and it is expected to guess if $T$ is $g^{abc}$ or if it is random. In this case, the simulator guess the challenge value $\tilde{c}$ and generates the public parameters correctly. In case the value of the $\tilde{c}$ does not match the value later provided by the adversary then the simulation aborts. Since the simulator will successfully guess the right value of $\tilde{c}$ with probability at least $\frac{1}{2}$, the simulation will work with probability at least $\frac{1}{2}$.

**Setup.** Firstly, $\mathcal{B}$ randomly chooses a value for $\tilde{c}$ to guess that $\tilde{c} = 0$ (regardless of whether $\mathcal{A}$ behaves in **Case I** or **Caes II.1**) and $\tilde{c} = 1$ if $\mathcal{A}$ behaves in **Case II.1** or **Case II.2**.

$\mathcal{B}$ chooses random dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$, $(\mathbb{D}_0, \mathbb{D}_0^*)$ of dimension 3 and $(\mathbb{D}_x, \mathbb{D}_x^*)$ of dimension 6, all with the same value of $\psi$. It then implicitly sets $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ and $\{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ as follows:

$$\begin{array}{llllll}\boldsymbol{b}_1 = \boldsymbol{d}_1, & \boldsymbol{b}_2 = \boldsymbol{d}_2, & \boldsymbol{b}_3 = \boldsymbol{d}_3, & \boldsymbol{b}_1^* = \boldsymbol{d}_1^*, & \boldsymbol{b}_2^* = \boldsymbol{d}_2^*, & \boldsymbol{b}_3^* = \boldsymbol{d}_3^*,\end{array}$$
$$\boldsymbol{b}_{0,1} = (c)^{-1}\boldsymbol{d}_{0,1}, \boldsymbol{b}_{0,2} = (c)^{-1}\boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3} = \boldsymbol{d}_{0,3}, \boldsymbol{b}_{0,1}^* = (c)\boldsymbol{d}_{0,1}^*, \boldsymbol{b}_{0,2}^* = (c)\boldsymbol{d}_{0,2}^*, \boldsymbol{b}_{0,3}^* = \boldsymbol{d}_{0,3}^*.$$

$$\boldsymbol{b}_{x,1} = \boldsymbol{d}_{x,1}, \ldots, \boldsymbol{b}_{x,6} = \boldsymbol{d}_{x,6}, \boldsymbol{b}_{x,1}^* = \boldsymbol{d}_{x,1}^*, \ldots, \boldsymbol{b}_{x,6}^* = \boldsymbol{d}_{x,6}^* \ \forall x \in [\mathcal{U}].$$

We note $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ and $\{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ are properly distributed.

$\mathcal{B}$ chooses random exponents

$$\theta, \alpha_1, \alpha_2 \in \mathbb{Z}_p,\ \{\alpha_{i,1}, \alpha_{i,2},\ z_i' \in \mathbb{Z}_p\}_{i\in[n]},\quad \{r_i \in \mathbb{Z}_p\}_{i\in[n]\setminus\{\bar{i}\}},\quad \{c_{j,1}, c_{j,2},\ y_j, \upsilon_j \in \mathbb{Z}_p\}_{j\in[n]\setminus\{\bar{j}\}},$$
$$r_{\bar{i}}',\ c_{\bar{j},1}',\ c_{\bar{j},1}',\ y_{\bar{j}}', \upsilon_{\bar{j}}' \in \mathbb{Z}_p.$$

$\mathcal{B}$ gives $\mathcal{A}$ the following public parameter $\mathsf{PP}$:

$$\Big(\ g, h = g^\theta,\ g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}, \{h_j = g^{\upsilon_j'}\}_{i\in[n]\setminus\{\bar{i}\}}, h_{\bar{j}} = g^{\upsilon_{\bar{j}}'}, h^{\boldsymbol{b}_1} = g^{\theta\boldsymbol{d}_1}, h^{\boldsymbol{b}_2} = g^{\theta\boldsymbol{d}_2},$$
$$\{h_j^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\upsilon_j'},\quad h_j^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\upsilon_j'}\}_{i\in[n]\setminus\{\bar{i}\}},\ \{h_{\bar{j}}^{\boldsymbol{b}_1} = (C^{\boldsymbol{b}_1})^{\upsilon_{\bar{j}}'},\quad h_{\bar{j}}^{\boldsymbol{b}_2} = (C^{\boldsymbol{b}_2})^{\upsilon_{\bar{j}}'}\},$$
$$h^{\boldsymbol{b}_{0,1}} = g^{\theta\boldsymbol{d}_{0,1}}, h^{\boldsymbol{b}_{0,2}} = g^{\theta\boldsymbol{d}_{0,2}}, \{h^{\boldsymbol{b}_{x,1}} = g^{\theta\boldsymbol{d}_{x,1}}, \ldots, h^{\boldsymbol{b}_{x,4}} = g^{\theta'\boldsymbol{d}_{x,4}}\}_{x\in[\mathcal{U}]},$$
$$F_1 = e(g,h)^{\psi\alpha_1},\ F_2 = e(g,h)^{\psi\alpha_2},\quad \{F_{1,j} = e(g,h_j)^{\psi\alpha_1}, F_{2,j} = e(g,h_j)^{\psi\alpha_2}\}_{j\in[n]},$$
$$\{E_{i,1} = e(g,g)^{\psi\alpha_{i,1}},\ E_{i,2} = e(g,g)^{\psi\alpha_{i,2}}\}_{i\in[n]},$$
$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{d}_1+\boldsymbol{d}_2)}\}_{i\in[n]\setminus\{\bar{i}\}},\quad \boldsymbol{G}_{\bar{i}} = B^{r_{\bar{i}}'(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \{\boldsymbol{Z}_i = g^{z_i(\boldsymbol{d}_1+\boldsymbol{d}_2)}\}_{i\in[n]},$$
$$\{\boldsymbol{H}_j = g^{c_{j,1}\boldsymbol{d}_1^*+c_{j,2}\boldsymbol{d}_2^*},\ \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]\setminus\{\bar{j}\}},\quad \boldsymbol{H}_{\bar{j}} = C^{c_{\bar{j},1}'\boldsymbol{d}_1^*+c_{\bar{j},2}'\boldsymbol{d}_2^*},\ \boldsymbol{Y}_{\bar{j}} = (g^{c_{\bar{j},1}'\boldsymbol{d}_1^*+c_{\bar{j},2}'\boldsymbol{d}_2^*})^{y_{\bar{j}}'}\ \Big).$$

Note that $\mathcal{B}$ implicitly chooses $r_{\bar{i}},\ c_{\bar{j},1},\ c_{\bar{j},2}, \upsilon'_{\bar{j}}$ and $y_{\bar{j}} \in \mathbb{Z}_p$ such that

$$br'_{\bar{i}} \equiv r_{\bar{i}} \bmod p,\ \ cc'_{\bar{j},1} \equiv c_{\bar{j},1} \bmod p,\ \ cc'_{\bar{j},2} \equiv c_{\bar{j},2} \bmod p,\ \ c\upsilon'_{\bar{j}} \equiv \upsilon_{\bar{j}} \bmod p,\ \ y'_{\bar{j}}/c = y_{\bar{j}} \bmod p.$$

**Key Query.** To respond to a query for $((i,j), S_{(i,j)})$,

- if $(i,j) \neq (\bar{i},\bar{j})$: $\mathcal{B}$ randomly chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, then creates a private key $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j},\ \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ where

$$\boldsymbol{K}_{i,j} = \begin{cases} g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} g^{r_i c_{j,1}\boldsymbol{d}_1^*+r_i c_{j,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}, & : i \neq \bar{i}, j \neq \bar{j} \\ g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} B^{r'_i c_{j,1}\boldsymbol{d}_1^*+r'_i c_{j,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}, & : i = \bar{i}, j \neq \bar{j} \\ g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} C^{r_i c'_{j,1}\boldsymbol{d}_1^*+r_i c'_{j,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}, & : i \neq \bar{i}, j = \bar{j} \end{cases}$$

$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*},$$

$$\boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_{\bar{i}}},$$

$$\{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}\}_{j'\in[n]\setminus\{j\}}, \qquad : i \neq \bar{i}, j \neq \bar{j}$$

$$\boldsymbol{K}_{i,j,0} = C^{\delta_{i,j,1}\boldsymbol{d}_{0,1}^*+\delta_{i,j,2}\boldsymbol{d}_{0,2}^*},$$

$$\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}.$$

- if $(i,j) = (\bar{i},\bar{j})$: it means that $\mathcal{A}$ behaves in **Case II.1** or **Case II.2**. $\mathcal{B}$ chooses random $\sigma'_{i,j,1}, \sigma'_{i,j,2} \in \mathbb{Z}_p$ and sets the value of $\sigma_{i,j,1}, \sigma_{i,j,2}$ by implicitly setting $\sigma'_{i,j,1} - br'_i c'_{j,1}/(\theta' + \upsilon'_j) \equiv \sigma_{i,j,1} \bmod p$, $\sigma'_{i,j,2} - br'_i c'_{j,2}/(\theta' + \upsilon'_j) \equiv \sigma_{i,j,2} \bmod p$. In addition $\mathcal{B}$ randomly chooses $\delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$. $\mathcal{B}$ creates a private key $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j},\ \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ where

$$\boldsymbol{K}_{i,j} = g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma'_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma'_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*},$$

$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma'_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma'_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*} (B^{c'_{\bar{j},1}\boldsymbol{d}_1^*+c'_{\bar{j},2}\boldsymbol{d}_2^*})^{-r'_{\bar{i}}/(\theta'+\upsilon'_j)},\ \ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_{\bar{i}}},$$

$$\{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma'_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma'_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}\}_{j'\in[n]\setminus\{j\}},$$

$$\boldsymbol{K}_{i,j,0} = C^{\delta_{i,j,1}\boldsymbol{d}_{0,1}^*+\delta_{i,j,2}\boldsymbol{d}_{0,2}^*},$$

$$\boldsymbol{K}_{i,j,x} = g^{\sigma'_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma'_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)} (B^{-r'_{\bar{i}}/(\theta'+\upsilon'_j)})^{c'_{j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+c'_{j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}\ \forall x \in S_{(i,j)}.$$

**Challenge.** $\mathcal{A}$ submits a revocation list $R^*$, a message $M$ and an attribute set $S^*$, $\mathcal{B}$ sets $\bar{R}^* = [n,n] \setminus R^*$ and constructs the LSSS matrix $(A,\rho)$ for $\mathbb{A}_{S^*}$. Let $l \times m$ be the size of $A$.

- if $(\bar{i},\bar{j}) \in \bar{R}^* \wedge \tilde{c} = 1$: $\mathcal{A}$ is in **Case II.3**. $\mathcal{B}$ returns $b \in \{0,1\}$ to its challenger, then aborts.
- if $(\bar{i},\bar{j}) \in \bar{R}^* \wedge \tilde{c} = 0$: $\mathcal{B}$ continues the following interaction.
- if $(\bar{i},\bar{j}) \notin \bar{R}^* \wedge \tilde{c} = 1$: $\mathcal{B}$ continues the following interaction.
- if $(\bar{i},\bar{j}) \notin \bar{R}^* \wedge \tilde{c} = 0$: $\mathcal{B}$ sets $\tilde{c} = 1$ and generates the private key $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}$, $\{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$. Then $\mathcal{B}$ continues the following interaction.

$\mathcal{B}$ chooses random

$$\begin{aligned} \tau',\quad s_1,\ldots,s_{\bar{i}-1}, s'_{\bar{i}}, s_{\bar{i}+1},\ldots,s_n,\quad t_1,\ldots,t_{\bar{i}-1}, t_{\bar{i}}, t_{\bar{i}+1},\ldots,t_n &\in \mathbb{Z}_p, \\ \boldsymbol{w}_1,\ldots,\boldsymbol{w}_{\bar{j}-1}, \boldsymbol{w}'_{\bar{j}}, \boldsymbol{w}'_{\bar{j}+1},\ldots,\boldsymbol{w}'_n &\in \mathbb{Z}_p^3, \\ \xi_{1,1},\xi_{1,2},\ldots,\xi_{l,1},\xi_{l,2} \in \mathbb{Z}_p,\quad \pi_1,\pi_2 \in \mathbb{Z}_p,\quad \boldsymbol{u}_1,\boldsymbol{u}_2 &\in \mathbb{Z}_p^m, \end{aligned}$$

where the first entries of $\boldsymbol{u}'_1, \boldsymbol{u}'_2$ are equal to zero.

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$, then it chooses random

$$\boldsymbol{v}_i \in \mathbb{Z}_p^3 \ \mathit{for}\ i = 1, \ldots, \bar{i},$$
$$\boldsymbol{v}_i \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} \ \mathit{for}\ i = \bar{i}+1, \ldots, n.$$

$\mathcal{B}$ chooses random $(\nu_{c,1}, \nu_{c,2}, \nu_{c,3}) \in \mathbb{Z}_p^3$. Let $\boldsymbol{v}_c^p = \nu_{c,1}\boldsymbol{\chi}_1 + \nu_{c,2}\boldsymbol{\chi}_2$ and $\boldsymbol{v}_c^q = \nu_{c,3}\boldsymbol{\chi}_3$, in the following simulation, $\mathcal{B}$ will implicitly set

$$\boldsymbol{v}_c = a^{-1}\boldsymbol{v}_c^p + \boldsymbol{v}_c^q.$$

$\mathcal{B}$ creates a ciphertext $\langle R^*, (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows:

1. For each $i \in [n]$:
   - if $i < \bar{i}$: it chooses random $\hat{s}_i \in \mathbb{Z}_p$, and sets

$$\begin{aligned}
&\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i},\\
&\boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = Q_i^{(\theta'+\sum_{j'\in\bar{R}^*} \upsilon_j)} Z_i^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2},\\
&\boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\
&T_i = e(g,g)^{\hat{s}_i}.
\end{aligned}$$

   - if $i = \bar{i}$: it sets

$$\begin{aligned}
&\boldsymbol{R}_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}},\\
&\boldsymbol{Q}_i = g^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)} A^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)(\boldsymbol{d}_1+\boldsymbol{d}_2)},\\
&\boldsymbol{Q}'_i = Q_i^{(\theta'+\sum_{j'\in\bar{R}^*} \upsilon_j)} Z_i^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2},\\
&\boldsymbol{Q}''_i = g^{t_{\bar{i}}(\boldsymbol{d}_1+\boldsymbol{d}_2)},\\
&T_i = M\frac{e_3(\boldsymbol{Q}_i, g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*})}{e_3(g^{\boldsymbol{d}_1+\boldsymbol{d}_2}, (h\prod_{j'\in\bar{R}^*} h_{j'})^{\alpha_1\boldsymbol{d}_1^*+\alpha_2\boldsymbol{d}_2^*})^{\tau' s'_{\bar{i}} \boldsymbol{v}_{\bar{i}} \boldsymbol{v}_c^p} e_3(A^{\boldsymbol{d}_1+\boldsymbol{d}_2}, (h\prod_{j'\in\bar{R}^*} h_{j'})^{\alpha_1\boldsymbol{d}_1^*+\alpha_2\boldsymbol{d}_2^*})^{\tau' s'_{\bar{i}} \boldsymbol{v}_{\bar{i}} \boldsymbol{v}_c^q} F_1^{\pi_1} F_2^{\pi_2}}.
\end{aligned}$$

   - if $i > \bar{i}$: it sets

$$\begin{aligned}
&\boldsymbol{R}_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r_i s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r_i s_i \boldsymbol{v}_i},\\
&\boldsymbol{Q}_i = B^{\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \quad \boldsymbol{Q}'_i = Q_i^{(\theta'+\sum_{j'\in\bar{R}^*} \upsilon_j)} Z_i^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2},\\
&\boldsymbol{Q}''_i = g^{t_i(\boldsymbol{d}_1+\boldsymbol{d}_2)},\\
&T_i = M\frac{e_3(\boldsymbol{Q}_i, g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*})}{e_3(B^{\boldsymbol{d}_1+\boldsymbol{d}_2}, (h\prod_{j'\in\bar{R}^*} h_{j'})^{\alpha_1\boldsymbol{d}_1^*+\alpha_2\boldsymbol{d}_2^*})^{\tau' s_i \boldsymbol{v}_{\bar{i}} \boldsymbol{v}_c^p} F_1^{\pi_1} F_2^{\pi_2}}.
\end{aligned}$$

2. For each $j \in [n]$:
   - if $j < \bar{j}$: it chooses random $\mu'_j \in \mathbb{Z}_p$ and implicitly sets the value of $\mu_j$ such that $(\frac{\mu'_j}{ab} - 1)\nu_{c,3} \equiv \mu_j \bmod p$, then sets

$$\boldsymbol{C}_j = (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^p} (g^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\mu'_j\boldsymbol{v}_c^q} (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{y_j\boldsymbol{w}_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}.$$

- if $j = \bar{j}$:

$$\boldsymbol{C}_j = (T^{c'_{\bar{j},1}\boldsymbol{b}_1^* + c'_{\bar{j},2}\boldsymbol{b}_2^*})^{\tau' \boldsymbol{v}_c^q} (B^{c'_{j,1}\boldsymbol{b}_1^* + c'_{j,2}\boldsymbol{b}_2^*})^{y'_{\bar{j}} \boldsymbol{w}'_{\bar{j}}}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_{\bar{j}})^{\boldsymbol{w}'_{\bar{j}}} (C^{c'_{\bar{j},1}\boldsymbol{b}_1^* + c'_{\bar{j},2}\boldsymbol{b}_2^*})^{-\tau' \boldsymbol{v}_c^p}.$$

- if $j > \bar{j}$:

$$\boldsymbol{C}_j = (B^{c_{j,1}\boldsymbol{b}_1^* + c_{j,2}\boldsymbol{b}_2^*})^{\tau' \boldsymbol{v}_c^p} (B^{c_{j,1}\boldsymbol{b}_1^* + c_{j,2}\boldsymbol{b}_2^*})^{y_j \boldsymbol{w}'_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}'_j} (A^{c_{j,1}\boldsymbol{b}_1^* + c_{j,2}\boldsymbol{b}_2^*})^{-\tau' \boldsymbol{v}_c^q}.$$

3. $\boldsymbol{P}_0 = g^{\theta(\pi_1 \boldsymbol{b}_{0,1} + \pi_2 \boldsymbol{b}_{0,2})}, \quad \{\boldsymbol{P}_k = (g^{\theta})^{(A_k \cdot \boldsymbol{u}_1 + \xi_{k,1})\boldsymbol{b}_{\rho(k),1} - \xi_{k,1}\boldsymbol{b}_{\rho(k),2} + (A_k \cdot \boldsymbol{u}_2 + \xi_{k,2})\boldsymbol{b}_{\rho(k),3} - \xi_{k,2}\boldsymbol{b}_{\rho(k),4}}\}_{k \in [l]}$.

Note that $\mathcal{B}$ implicitly chooses $\kappa, \tau,\ s_{\bar{i}},\ t_i(i \in [n] \setminus \{\bar{i}\}),\ \pi_1, \pi_2\ \in \mathbb{Z}_p$ and $\boldsymbol{w}_j \in \mathbb{Z}_p^3(\bar{j} \le j \le n)$ such that

$$\begin{aligned} b \equiv \kappa \bmod p, \quad ab\tau' &\equiv \tau \bmod p, \\ s'_{\bar{i}}/b &\equiv s_{\bar{i}} \bmod p, \\ \boldsymbol{w}'_{\bar{j}} - c\tau' \boldsymbol{v}_c^p / y'_{\bar{j}} &\equiv \boldsymbol{w}_{\bar{j}} \bmod p, \\ \boldsymbol{w}'_j - a\tau' \boldsymbol{v}_c^q / y_j &\equiv \boldsymbol{w}_j \bmod p\ \forall j \in \{\bar{j}+1, \ldots, n\}. \end{aligned}$$

If $T = g^{abc}$, then the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$. If $T$ is randomly chosen, say $T = g^r$ for some random $r \in \mathbb{Z}_p$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j}+1)$ with implicitly setting $\mu_{\bar{j}}$ such that $(\frac{r}{abc} - 1)\nu_{c,3} \equiv \mu_{\bar{j}} \bmod p$.

**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this $b'$ to the challenger as its answer to the D3DH game.

When $\mathcal{B}$ does not abort, $\mathcal{B}$'s advantage in the index-hiding game for our AugR-CP-ABE scheme is $\epsilon \cdot$ Pr $[\mathcal{A}$ is not in **Case II.3** $\wedge (c = 0)] = \epsilon \cdot$ **Pr** $[\overline{\mathcal{A}.\textbf{II}.3} \wedge (c = 0)]$**.**

**Case B**: $\mathcal{B}$ receives the D3DH challenge from the challenger as $((p, \mathbb{G}, \mathbb{G}_T, e), g, A = g^a, B = g^b, C = g^c, T)$, and it is expected to guess if $T$ is $g^{abc}$ or if it is random.

**Setup.** Firstly, $\mathcal{B}$ randomly chooses an attribute $\bar{x} \in [\mathcal{U}]$ to guess that $\bar{x}$ will be in the challenge attribute set $S^*$ (regardless of whether $\mathcal{A}$ behaves in **Case I** or **Caes II**) and will not be in $S_{(\bar{i},\bar{j})}$ if $\mathcal{A}$ behaves in **Case II.2** or **Case II.3**.

$\mathcal{B}$ chooses random dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$, $(\mathbb{D}_0, \mathbb{D}_0^*)$ of dimension 3 and $(\mathbb{D}_x, \mathbb{D}_x^*)$ of dimension 6, all with the same value of $\psi$. It then implicitly sets $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ and $\{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ as follows:

$$\begin{aligned} &\boldsymbol{b}_1 = \boldsymbol{d}_1, \quad \boldsymbol{b}_2 = \boldsymbol{d}_2, \quad \boldsymbol{b}_3 = \boldsymbol{d}_3, \quad \boldsymbol{b}_1^* = \boldsymbol{d}_1^*, \quad \boldsymbol{b}_2^* = \boldsymbol{d}_2^*, \quad \boldsymbol{b}_3^* = \boldsymbol{d}_3^*, \\ &\boldsymbol{b}_{0,1} = (c)^{-1}\boldsymbol{d}_{0,1}, \boldsymbol{b}_{0,2} = (c)^{-1}\boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3} = \boldsymbol{d}_{0,3}, \boldsymbol{b}_{0,1}^* = (c)\boldsymbol{d}_{0,1}^*, \boldsymbol{b}_{0,2}^* = (c)\boldsymbol{d}_{0,2}^*, \boldsymbol{b}_{0,3}^* = \boldsymbol{d}_{0,3}^*. \end{aligned}$$

$$\begin{aligned} &\boldsymbol{b}_{x,1} = \boldsymbol{d}_{x,1}, \quad \ldots, \boldsymbol{b}_{x,6} = \boldsymbol{d}_{x,6}, \quad \boldsymbol{b}_{x,1}^* = \boldsymbol{d}_{x,1}^*, \quad \ldots, \boldsymbol{b}_{x,6}^* = \boldsymbol{d}_{x,6}^*\ \forall x \in [\mathcal{U}] \setminus \{\bar{x}\}; \\ &\boldsymbol{b}_{\bar{x},1} = (c)^{-1}\boldsymbol{d}_{\bar{x},1}, \ldots, \boldsymbol{b}_{\bar{x},6} = c^{-1}\boldsymbol{d}_{\bar{x},6}, \boldsymbol{b}_{\bar{x},1}^* = (c)\boldsymbol{d}_{\bar{x},1}^*, \ldots, \boldsymbol{b}_{\bar{x},6}^* = (c)\boldsymbol{d}_{\bar{x},6}^*. \end{aligned}$$

We note $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ and $\{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ are properly distributed.

$\mathcal{B}$ chooses random exponents

$$\begin{aligned} &\theta', \alpha_1, \alpha_2 \in \mathbb{Z}_p,\ \{\alpha_{i,1}, \alpha_{i,2} \in \mathbb{Z}_p\}_{i \in [n]}, \quad \{r_i,\ z'_i \in \mathbb{Z}_p\}_{i \in [n] \setminus \{\bar{i}\}}, \quad \{c_{j,1}, c_{j,2},\ y_j \in \mathbb{Z}_p\}_{j \in [n] \setminus \{\bar{j}\}}, \\ &r'_{\bar{i}},\ z_{\bar{i}},\ c'_{\bar{j},1},\ c'_{\bar{j},1},\ y'_{\bar{j}},\ \{\upsilon'_j\}_{j \in [n]} \in \mathbb{Z}_p. \end{aligned}$$

$\mathcal{B}$ gives $\mathcal{A}$ the following public parameter $\mathsf{PP}$:

$$\Big( g, h = C^{\theta'},\ g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}, \{h_j = C^{\upsilon'_j}\}_{j\in[n]}, h^{\boldsymbol{b}_1} = C^{\theta'\boldsymbol{d}_1}, h^{\boldsymbol{b}_2} = C^{\theta'\boldsymbol{d}_2},$$
$$\{h_j^{\boldsymbol{b}_1} = (C^{\boldsymbol{b}_1})^{\upsilon'_j},\ \ h_j^{\boldsymbol{b}_2} = (C^{\boldsymbol{b}_2})^{\upsilon'_j}\}_{j\in[n]},\ h^{\boldsymbol{b}_{0,1}} = g^{\theta'\boldsymbol{d}_{0,1}}, h^{\boldsymbol{b}_{0,2}} = g^{\theta'\boldsymbol{d}_{0,2}},$$
$$\{h^{\boldsymbol{b}_{x,1}} = C^{\theta'\boldsymbol{d}_{x,1}}, \ldots, h^{\boldsymbol{b}_{x,4}} = C^{\theta'\boldsymbol{d}_{x,4}}\}_{x\in[\mathcal{U}]\setminus\{\bar{x}\}},\ \ \{h^{\boldsymbol{b}_{x,1}} = g^{\theta'\boldsymbol{d}_{x,1}}, \ldots, h^{\boldsymbol{b}_{x,4}} = g^{\theta'\boldsymbol{d}_{x,4}}\}_{x=\bar{x}},$$
$$F_1 = e(g,h)^{\psi\alpha_1},\ F_2 = e(g,h)^{\psi\alpha_2},\ \ \{F_{1,j} = e(g,h_j)^{\psi\alpha_1}, F_{2,j} = e(g,h_j)^{\psi\alpha_2}\}_{j\in[n]},$$
$$\{E_{i,1} = e(g,g)^{\psi\alpha_{i,1}},\ E_{i,2} = e(g,g)^{\psi\alpha_{i,2}}\}_{i\in[n]},$$
$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{d}_1+\boldsymbol{d}_2)},\ \boldsymbol{Z}_i = C^{z'_i(\boldsymbol{d}_1+\boldsymbol{d}_2)},\}_{i\in[n]\setminus\{\bar{i}\}},\ \ \boldsymbol{G}_{\bar{i}} = B^{r'_{\bar{i}}(\boldsymbol{d}_1+\boldsymbol{d}_2)},\ \boldsymbol{Z}_{\bar{i}} = g^{z_{\bar{i}}(\boldsymbol{d}_1+\boldsymbol{d}_2)},$$
$$\{\boldsymbol{H}_j = g^{c_{j,1}\boldsymbol{d}_1^*+c_{j,2}\boldsymbol{d}_2^*},\ \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]\setminus\{\bar{j}\}},\ \ \boldsymbol{H}_{\bar{j}} = C^{c'_{\bar{j},1}\boldsymbol{d}_1^*+c'_{\bar{j},2}\boldsymbol{d}_2^*},\ \boldsymbol{Y}_{\bar{j}} = (g^{c'_{\bar{j},1}\boldsymbol{d}_1^*+c'_{\bar{j},2}\boldsymbol{d}_2^*})^{y'_{\bar{j}}}\ \Big).$$

Note that $\mathcal{B}$ implicitly chooses $r_{\bar{i}},\ c_{\bar{j},1},\ c_{\bar{j},2},\ y_{\bar{j}} \in \mathbb{Z}_p$ and $\{z_i \in \mathbb{Z}_p\}_{i\in[n]\setminus\{\bar{i}\}}$ such that

$$br'_{\bar{i}} \equiv r_{\bar{i}} \bmod p,\ \ cc'_{\bar{j},1} \equiv c_{\bar{j},1} \bmod p,\ cc'_{\bar{j},2} \equiv c_{\bar{j},2} \bmod p,\ y'_{\bar{j}}/c = y_{\bar{j}} \bmod p,$$
$$cz'_i \equiv z_i \bmod p\ \forall i \in [n]\setminus\{\bar{i}\}.$$

**Key Query.** To respond to a query for $((i,j), S_{(i,j)})$,

- if $(i,j) \neq (\bar{i},\bar{j})$: $\mathcal{B}$ randomly chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, then creates a private key $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j},\ \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ where

$$\boldsymbol{K}_{i,j} = \begin{cases} g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} g^{r_i c_{j,1}\boldsymbol{d}_1^*+r_i c_{j,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}, & : i \neq \bar{i}, j \neq \bar{j} \\ g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} B^{r'_i c_{j,1}\boldsymbol{d}_1^*+r'_i c_{j,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}, & : i = \bar{i}, j \neq \bar{j} \\ g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*} C^{r_i c'_{j,1}\boldsymbol{d}_1^*+r_i c'_{j,2}\boldsymbol{d}_2^*} (hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}, & : i \neq \bar{i}, j = \bar{j} \end{cases}$$
$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*},$$
$$\boldsymbol{K}''_{i,j} = \begin{cases} (C^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*})^{z'_i}, & : i \neq \bar{i}, j \neq \bar{j} \\ (\boldsymbol{K}'_{i,j})^{z_{\bar{i}}}, & : i = \bar{i}, j \neq \bar{j} \\ (C^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*})^{z'_i}, & : i \neq \bar{i}, j = \bar{j} \end{cases}$$
$$\{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}\}_{j'\in[n]\setminus\{j\}}, \qquad : i \neq \bar{i}, j \neq \bar{j}$$
$$\boldsymbol{K}_{i,j,0} = C^{\delta_{i,j,1}\boldsymbol{d}_{0,1}^*+\delta_{i,j,2}\boldsymbol{d}_{0,2}^*},$$
$$\boldsymbol{K}_{i,j,x} = \begin{cases} g^{\sigma_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}. & : x \neq \bar{x} \\ C^{\sigma_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}. & : x = \bar{x} \end{cases}$$

- if $(i,j) = (\bar{i},\bar{j})$: it means that $\mathcal{A}$ behaves in **Case II.2** or **Case II.3**. if $\bar{x} \in S_{(i,j)}$ then $\mathcal{B}$ aborts and outputs a random $b' \in \{0,1\}$ to the challenger. Otherwise $\mathcal{B}$ chooses random $\sigma'_{i,j,1}, \sigma'_{i,j,2} \in \mathbb{Z}_p$ and sets the value of $\sigma_{i,j,1}, \sigma_{i,j,2}$ by implicitly setting $\sigma'_{i,j,1} - br'_i c'_{j,1}/(\theta'+\upsilon'_j) \equiv \sigma_{i,j,1} \bmod p$, $\sigma'_{i,j,2} - br'_i c'_{j,2}/(\theta'+\upsilon'_j) \equiv \sigma_{i,j,2} \bmod p$. In addition $\mathcal{B}$ randomly chooses $\delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$. $\mathcal{B}$ creates a private key $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j},\ \{\bar{\boldsymbol{K}}_{i,j,j'}\}_{j'\in[n]\setminus\{j\}},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ where

$$\boldsymbol{K}_{i,j} = g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*}(hh_j)^{(\sigma'_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma'_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*},$$
$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma'_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma'_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}(B^{c'_{\bar{j},1}\boldsymbol{d}_1^*+c'_{\bar{j},2}\boldsymbol{d}_2^*})^{-r'_{\bar{i}}/(\theta'+\upsilon'_j)},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_{\bar{i}}},$$
$$\{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma'_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\sigma'_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*}\}_{j'\in[n]\setminus\{j\}},$$
$$\boldsymbol{K}_{i,j,0} = C^{\delta_{i,j,1}\boldsymbol{d}_{0,1}^*+\delta_{i,j,2}\boldsymbol{d}_{0,2}^*},$$
$$\boldsymbol{K}_{i,j,x} = g^{\sigma'_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma'_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}(B^{-r'_{\bar{i}}/(\theta'+\upsilon'_j)})^{c'_{j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+c'_{j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}\ \forall x \in S_{(i,j)}.$$

**Challenge.** $\mathcal{A}$ submits a revocation list $R^*$, a message $M$ and an attribute set $S^*$. If $\bar{x} \notin S^*$ then $\mathcal{B}$ aborts and outputs a random $b' \in \{0,1\}$ to the challenger. Otherwise, $\mathcal{B}$ constructs the LSSS matrix $(A,\rho)$ for $\mathbb{A}_{S^*}$. Let $l \times m$ be the size of $A$.

Note that $S^* \setminus \{\bar{x}\}$ does not satisfy $(A,\rho)$, $\mathcal{B}$ first computes a vector $\boldsymbol{w} \in \mathbb{Z}_p^m$ that has first entry equal to 1 and is orthogonal to all of the rows $A_k$ of $A$ such that $\rho(k) \in S^* \setminus \{\bar{x}\}$ (such a vector must exist since $S^* \setminus \{\bar{x}\}$ fails to satisfy $(A,\rho)$, and it is efficiently computable).

$\mathcal{B}$ chooses random

$$\begin{aligned} \tau', \quad s_1,\ldots,s_{\bar{i}-1}, s'_{\bar{i}}, s_{\bar{i}+1},\ldots,s_n, \quad t'_1,\ldots,t'_{\bar{i}-1}, t_{\bar{i}}, t'_{\bar{i}+1},\ldots,t'_n \ &\in \mathbb{Z}_p, \\ \boldsymbol{w}_1,\ldots,\boldsymbol{w}_{\bar{j}-1}, \boldsymbol{w}'_{\bar{j}},\ldots,\boldsymbol{w}'_n \ &\in \mathbb{Z}_p^3, \\ \xi_{1,1},\xi_{1,2},\ldots,\xi_{l,1},\xi_{l,2} \ \in \mathbb{Z}_p, \quad \pi'_1,\pi'_2 \ \in \mathbb{Z}_p, \quad \boldsymbol{u}'_1,\boldsymbol{u}'_2 \ &\in \mathbb{Z}_p^m, \end{aligned}$$

where the first entries of $\boldsymbol{u}'_1, \boldsymbol{u}'_2$ are equal to zero.

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$, then it chooses random

$$\begin{aligned} \boldsymbol{v}_i \in \mathbb{Z}_p^3 \ for \ i = 1,\ldots,\bar{i}, \\ \boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} \ for \ i = \bar{i}+1,\ldots,n. \end{aligned}$$

$\mathcal{B}$ chooses random $(\nu_{c,1}, \nu_{c,2}, \nu_{c,3}) \in \mathbb{Z}_p^3$. Let $\boldsymbol{v}_c^p = \nu_{c,1}\boldsymbol{\chi}_1 + \nu_{c,2}\boldsymbol{\chi}_2$ and $\boldsymbol{v}_c^q = \nu_{c,3}\boldsymbol{\chi}_3$, in the following simulation, $\mathcal{B}$ will implicitly set

$$\boldsymbol{v}_c = a^{-1}\boldsymbol{v}_c^p + \boldsymbol{v}_c^q.$$

$\mathcal{B}$ creates a ciphertext $\langle R^*, (A,\rho), \ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, \ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n, \ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows:

1. For each $i \in [n]$:
   - if $i < \bar{i}$: it chooses random $\hat{s}_i \in \mathbb{Z}_p$, and sets

$$\begin{aligned} &\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \\ &\boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = (h \prod_{j' \in \bar{R}^*} h_{j'})^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} C^{z'_i t'_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} h^{\pi'_1\boldsymbol{b}_1+\pi'_2\boldsymbol{b}_2}, \\ &\boldsymbol{Q}''_i = (g^{t'_i} A^{(\theta'+\sum_{j'\in\bar{R}^*} \upsilon_{j'})\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)/z'_i})^{(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \\ &T_i = e(g,g)^{\hat{s}_i}. \end{aligned}$$

   - if $i = \bar{i}$: it sets

$$\begin{aligned} &\boldsymbol{R}_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}}, \\ &\boldsymbol{Q}_i = g^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)} A^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \\ &\boldsymbol{Q}'_i = (h \prod_{j'\in\bar{R}^*} h_{j'})^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)} \boldsymbol{Z}_i^{t_{\bar{i}}} h^{\pi'_1\boldsymbol{d}_1+\pi'_2\boldsymbol{d}_2}, \\ &\boldsymbol{Q}''_i = g^{t_{\bar{i}}(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \\ &T_i = M \frac{e_3(\boldsymbol{Q}_i, g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*})}{(F'_1F'_2)^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)} F_1^{\pi'_1} F_2^{\pi'_2}}, \end{aligned}$$

   where ${F_1}' = F_1 \prod_{j'\in\bar{\bar{R}}_i^*} F_{1,j'}$ and ${F_2}' = F_2 \prod_{j'\in\bar{\bar{R}}_i^*} F_{2,j'}$ respectively.

- if $i > \bar{i}$: it sets

$$\begin{aligned}
&\boldsymbol{R}_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r_i s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r_i s_i \boldsymbol{v}_i},\\
&\boldsymbol{Q}_i = B^{\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \quad \boldsymbol{Q}'_i = C^{z'_i t'_i(\boldsymbol{d}_1+\boldsymbol{d}_2)} h^{\pi'_1\boldsymbol{d}_1+\pi'_2\boldsymbol{d}_2},\\
&\boldsymbol{Q}''_i = (g^{t'_i} B^{-(\theta'+\sum_{j'\in\bar{R}^*}\upsilon_{j'})\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)/z'_i} A^{(\theta'+\sum_{j'\in\bar{R}^*}\upsilon_{j'})\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)/z'_i})^{(\boldsymbol{d}_1+\boldsymbol{d}_2)},\\
&T_i = M\frac{e_3(\boldsymbol{Q}_i, g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*})}{e_3\big(\boldsymbol{Q}_i, (h\prod_{j'\in\bar{R}^*} h_{j'})^{\alpha_1\boldsymbol{d}_1^*+\alpha_2\boldsymbol{d}_2^*}\big) F_1^{\pi'_1} F_2^{\pi'_2} e_3\big(A^{\boldsymbol{d}_1+\boldsymbol{d}_2}, (h\prod_{j'\in\bar{R}^*} h_{j'})^{\alpha_1\boldsymbol{d}_1^*+\alpha_2\boldsymbol{d}_2^*}\big)^{-\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)}}.
\end{aligned}$$

2. For each $j \in [n]$:
   - if $j < \bar{j}$: it chooses random $\mu'_j \in \mathbb{Z}_p$ and implicitly sets the value of $\mu_j$ such that $(\frac{\mu'_j}{ab}-1)\nu_{c,3} \equiv \mu_j \bmod p$, then sets

$$\boldsymbol{C}_j = (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^p}(g^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\mu'_j\boldsymbol{v}_c^q}(B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{y_j\boldsymbol{w}_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}.$$

   - if $j = \bar{j}$:

$$\boldsymbol{C}_j = (T^{c'_{\bar{j},1}\boldsymbol{b}_1^*+c'_{\bar{j},2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^q}(B^{c'_{j,1}\boldsymbol{b}_1^*+c'_{\bar{j},2}\boldsymbol{b}_2^*})^{y'_{\bar{j}}\boldsymbol{w}'_{\bar{j}}}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_{\bar{j}})^{\boldsymbol{w}'_{\bar{j}}}(C^{c'_{\bar{j},1}\boldsymbol{b}_1^*+c'_{\bar{j},2}\boldsymbol{b}_2^*})^{-\tau'\boldsymbol{v}_c^p}.$$

   - if $j > \bar{j}$:

$$\boldsymbol{C}_j = (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^p}(B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{y_j\boldsymbol{w}'_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}'_j}(A^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{-\tau'\boldsymbol{v}_c^q}.$$

3.

$$\begin{aligned}
\boldsymbol{P}_0 =& g^{\theta'(\pi'_1\boldsymbol{d}_{0,1}+\pi'_2\boldsymbol{d}_{0,2})} A^{-((\theta'+\sum_{j'\in\bar{R}^*}\upsilon_{j'})/\theta')\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)(\boldsymbol{d}_{0,1}+\boldsymbol{d}_{0,2})},\\
\boldsymbol{P}_k =& (g^{\theta'})^{(A_k\cdot(\pi'_1\boldsymbol{w}+\boldsymbol{u}'_1)+\xi_{k,1})\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{1}}-\xi_{k,1}\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{2}}+(A_k\cdot(\pi'_2\boldsymbol{w}+\boldsymbol{u}'_2)+\xi_{k,2})\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{3}}-\xi_{k,2}\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{4}}}\\
&\qquad A^{-((\theta'+\sum_{j'\in\bar{R}^*}\upsilon_{j'})/\theta')\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)(A_k\cdot\boldsymbol{w})(\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{1}}+\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{3}})} \ \forall k\in[l] \ s.t.\ \rho(k)=\bar{x},\\
\boldsymbol{P}_k =& (C^{\theta'})^{(A_k\cdot\boldsymbol{u}'_1+\xi_{k,1})\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{1}}-\xi_{k,1}\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{2}}+(A_k\cdot\boldsymbol{u}'_2+\xi_{k,2})\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{3}}-\xi_{k,2}\boldsymbol{d}_{\boldsymbol{\rho(k)},\boldsymbol{4}}} \ \forall k\in[l] \ s.t.\ \rho(k)\neq\bar{x}.
\end{aligned}$$

Note that $\mathcal{B}$ implicitly chooses $\kappa, \tau,\ s_{\bar{i}},\ t_i(i \in [n]\setminus\{\bar{i}\}),\ \pi_1, \pi_2\ \in \mathbb{Z}_p$ and $\boldsymbol{w}_j \in \mathbb{Z}_p^3(\bar{j} \le j \le n)$ such that

$$\begin{aligned}
b \equiv \kappa \bmod p, \quad ab\tau' &\equiv \tau \bmod p,\\
s'_{\bar{i}}/b &\equiv s_{\bar{i}} \bmod p,\\
t'_i + a(\theta' + \sum_{j'\in\bar{R}^*}\upsilon_{j'})\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)/z'_i &\equiv t_i \bmod p \ \forall i \in \{1,\ldots,\bar{i}-1\},\\
t'_i - b(\theta' + \sum_{j'\in\bar{R}^*}\upsilon_{j'})\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)/z'_i + a(\theta' + \sum_{j'\in\bar{R}^*}\upsilon_{j'})\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)/z'_i &\equiv t_i \bmod p \ \forall i \in \{\bar{i}+1,\ldots,n\},\\
\boldsymbol{w}'_{\bar{j}} - c\tau'\boldsymbol{v}_c^p/y'_{\bar{j}} &\equiv \boldsymbol{w}_{\bar{j}} \bmod p,\\
\boldsymbol{w}'_j - a\tau'\boldsymbol{v}_c^q/y_j &\equiv \boldsymbol{w}_j \bmod p \ \forall j \in \{\bar{j}+1,\ldots,n\},\\
\pi'_1 - a\tau'(\theta' + \sum_{j'\in\bar{R}^*}\upsilon_{j'})s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)/\theta' &\equiv \pi_1 \bmod p,\\
\pi'_2 - a\tau'(\theta' + \sum_{j'\in\bar{R}^*}\upsilon_{j'})s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)/\theta' &\equiv \pi_2 \bmod p,
\end{aligned}$$

and implicitly sets

$$\begin{aligned}\boldsymbol{u}_1 =&(\pi'_1 - a\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q))\boldsymbol{w} + \boldsymbol{u}'_1,\\ \boldsymbol{u}_2 =&(\pi'_2 - a\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q))\boldsymbol{w} + \boldsymbol{u}'_2.\end{aligned}$$

If $T = g^{abc}$, then the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$. If $T$ is randomly chosen, say $T = g^r$ for some random $r \in \mathbb{Z}_p$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j}+1)$ with implicitly setting $\mu_{\bar{j}}$ such that $(\frac{r}{abc} - 1)\nu_{c,3} \equiv \mu_{\bar{j}} \bmod p$.

**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this $b'$ to the challenger as its answer to the D3DH game.

Note that when $\mathcal{B}$ does not abort, the distributions of the public parameter, private keys and challenge ciphertext are same as the real scheme. As $S^* \neq \emptyset$ and when $\mathcal{A}$ doesn't behave in **Case II.1** the attribute set $S_{(\bar{i},\bar{j})}$ must satisfy $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$, the event that $\mathcal{B}$ does not abort will happen at least $1/|\mathcal{U}|$. Thus, $\mathcal{B}$'s advantage in the D3DH game will be at least $\epsilon\cdot$ Pr [$\mathcal{A}$ is not in **Case II.1** $\wedge(S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset)$] $= \epsilon\cdot$ Pr $[\overline{\mathcal{A}.\mathbf{II}.3} \wedge (S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset)]$. As of the fully secure CP-ABE schemes in [10,19,11,12,13], the size of attribute universe (i.e. $|\mathcal{U}|$) in our scheme is also polynomial in the security parameter $\lambda$. Thus,

$$\begin{aligned}&\epsilon \cdot \Pr[\overline{\mathcal{A}.\mathbf{II.3}} \wedge (c=0)] + \epsilon \cdot \Pr[\overline{\mathcal{A}.\mathbf{II.3}} \wedge (S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset)]\\ &\quad = \epsilon \cdot \Pr[\overline{\mathcal{A}.\mathbf{II.3}}] \cdot \Pr[c=0] + \epsilon \cdot \Pr[\overline{\mathcal{A}.\mathbf{II.3}}] \cdot \Pr[S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset]\\ &\quad = \epsilon \cdot (1 - \Pr[\mathcal{A}.\mathbf{II.3}]) \cdot \Pr[c=0] + \epsilon \cdot (1 - \Pr[\mathcal{A}.\mathbf{II.3}]) \cdot \Pr[S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset]\\ &\quad = \epsilon \cdot (1 - \Pr[\mathcal{A}.\mathbf{II.3}]) \cdot \frac{1}{2} + \epsilon \cdot (1 - \Pr[\mathcal{A}.\mathbf{II.3}]) \cdot \frac{1}{|\mathcal{U}|}\\ &\quad \geq \frac{1}{2} \cdot \epsilon,\end{aligned}$$

Since Pr [$\mathcal{A}$.**II.3**]+ Pr [$\mathcal{A}$.**II.1**] $\leq$ Pr [$\mathcal{A}$.**I**] + Pr [$\mathcal{A}$.**II.1**] + Pr [$\mathcal{A}$.**II.2**] + Pr [$\mathcal{A}$.**II.3**] and $|\mathcal{U}| \geq 2$.

### B.3 Proof of Lemma 2

**Lemma 10.** *If the D3DH assumption holds, then no PPT adversary can distinguish between games $H_1$ and $H_2$ with non-negligible probability.*

*Proof.* This lemma can be proved by applying the result of Lemma 1.

**Lemma 11.** *If the D3DH assumption holds, then no PPT adversary can distinguish between games $H_2$ and $H_3$ with non-negligible probability.*

*Proof.* Consider an adversary $\mathcal{A}$ that can distinguish between $H_2$ and $H_3$ with a probability greater than $\epsilon$. We build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the D3DH problem. $\mathcal{B}$ receives the D3DH challenge as $((p, \mathbb{G}, \mathbb{G}_T, e), g, A = g^a, B = g^b, C = g^c, T)$, and it is expected to guess if $T$ is $g^{abc}$ or if it is random. $\mathcal{B}$ interacts with $\mathcal{A}$ in the $\mathsf{Game}_{\mathsf{IH}}$ as follows:

**Setup.** $\mathcal{B}$ randomly chooses two pairs of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ of dimension 3 and $\mathcal{U}$ pairs of dual orthonormal bases $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ of dimension 6, subject to the constraint that all of these share the same value of $\psi$.

$\mathcal{B}$ also randomly chooses

$$\theta,\ \alpha_1, \alpha_2 \in \mathbb{Z}_p,\ \{r_i,\ \alpha_{i,1}, \alpha_{i,2} \in \mathbb{Z}_p\}_{i\in[n]\setminus\{\bar{i}\}},\ \alpha_{\bar{i},1}, \alpha_{\bar{i},2} \in \mathbb{Z}_p,\ \{z_i \in \mathbb{Z}_p\}_{i\in[n]},\ \{c'_{j,1}, c'_{j,2},\ y_j, \upsilon_j \in \mathbb{Z}_p\}_{j\in[n]}.$$

$\mathcal{B}$ sets the public parameters to

$$\Big( g, h = g^{\theta},\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, \{h_j\}_{j\in[n]}, h^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\theta}, h^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\theta},\ \{h_j^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\upsilon_j}, h_j^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\upsilon_j}\}_{j\in[n]},$$

$$h^{\boldsymbol{b}_{0,1}} = (g^{\boldsymbol{b}_{0,1}})^{\theta}, h^{\boldsymbol{b}_{0,2}} = (g^{\boldsymbol{b}_{0,2}})^{\theta},\quad \{h^{\boldsymbol{b}_{x,1}}, \dots, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]},$$

$$F_1 = e(g,h)^{\psi\alpha_1}, F_2 = e(g,h)^{\psi\alpha_2},\ \{F_{1,j} = e_3(g^{\boldsymbol{b}_1}, g^{\eta\boldsymbol{b}_1^*})^{\upsilon_j\alpha_1}, F_{2,j} = e_3(g^{\boldsymbol{b}_2}, g^{\eta\boldsymbol{b}_2^*})^{\upsilon_j\alpha_2}\}_{j\in[n]},$$

$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad E_{i,1} = e(g,g)^{\psi\alpha_{i,1}}, E_{i,2} = e(g,g)^{\psi\alpha_{i,2}}\}_{i\in[n]\setminus\{\bar{i}\}},$$

$$\boldsymbol{G}_{\bar{i}} = B^{(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad E_{\bar{i},1} = e(A,B)^{\psi}e(g,g)^{\psi\alpha'_{\bar{i},1}}, E_{\bar{i},2} = e(A,B)^{\psi}e(g,g)^{\psi\alpha'_{\bar{i},2}}$$

$$\{\boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}\}_{i\in[n]},\quad \{\boldsymbol{H}_j = g^{c'_{j,1}\boldsymbol{b}_1^*+c'_{j,2}\boldsymbol{b}_2^*}A^{-(\boldsymbol{b}_1^*+\boldsymbol{b}_2^*)},\quad \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]}\Big).$$

Note that $\mathcal{B}$ implicitly sets

$$r_{\bar{i}} = b,\quad \alpha_{\bar{i},1} = ab + \alpha'_{i,1},\ \alpha_{\bar{i},2} = ab + \alpha'_{i,2},\ \{c_{j,1} = c'_{j,1} - a,\ c_{j,2} = c'_{j,2} - a\}_{j\in[n]}.$$

**Key Query.** To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ randomly chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, then creates a private key as

$$\bar{\boldsymbol{K}}_{i,j} = \begin{cases} g^{(\alpha_{i,1}+r_ic'_{j,1})\boldsymbol{b}_1^*+(\alpha_{i,2}+r_ic'_{j,2})\boldsymbol{b}_2^*}A^{-r_i(\boldsymbol{b}_1^*+\boldsymbol{b}_2^*)}(hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}, & : i \neq \bar{i} \\ g^{\alpha'_{\bar{i},1}\boldsymbol{b}_1^*+\alpha'_{\bar{i},2}\boldsymbol{b}_2^*}B^{(c'_{j,1}\boldsymbol{b}_1^*+c'_{j,2}\boldsymbol{b}_2^*)}(hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}, & : i = \bar{i} \end{cases}$$

$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},$$

$$\{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}\}_{j'\in[n]\setminus\{j\}},$$

$$\boldsymbol{K}_{i,j,0} = g^{\delta_{i,j,1}\boldsymbol{b}_{0,1}^*+\delta_{i,j,2}\boldsymbol{b}_{0,2}^*},$$

$$\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{b}_{x,1}^*+\boldsymbol{b}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{b}_{x,3}^*+\boldsymbol{b}_{x,4}^*)}\ \forall x \in S_{(i,j)}.$$

**Challenge.** $\mathcal{A}$ submits a message $M$, a revocation list $R$ and an attribute set $S^*$. $\mathcal{B}$ constructs the LSSS matrix $(A,\rho)$ for $\mathbb{A}_{S^*}$. Let $l \times m$ be the size of $A$.

$\mathcal{B}$ chooses random

$$\begin{aligned} \kappa, \tau,\quad s_1, \dots, s_n,\quad t_1, \dots, t_n &\in \mathbb{Z}_p, \\ \boldsymbol{w}_1, \dots, \boldsymbol{w}_n &\in \mathbb{Z}_p^3, \\ \xi_{1,1}, \xi_{1,2}, \dots, \xi_{l,1}, \xi_{l,2} \in \mathbb{Z}_p,\quad \boldsymbol{u}_1, \boldsymbol{u}_2 &\in \mathbb{Z}_p^m, \end{aligned}$$

where the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are equal to $\pi_1$ and $\pi_2$ respectively.

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_yr_z, -r_xr_z, r_xr_y)$, then it chooses random

$$\boldsymbol{v}_i \in \mathbb{Z}_p\ for\ i = 1, \dots, \bar{i},$$
$$\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ for\ i = \bar{i}+1, \dots, n.$$

$\mathcal{B}$ chooses random $(\nu_{c,1}, \nu_{c,2}, \nu_{c,3}) \in \mathbb{Z}_p^3$. Let $\boldsymbol{v}_c^p = \nu_{c,1}\boldsymbol{\chi}_1 + \nu_{c,2}\boldsymbol{\chi}_2$ and $\boldsymbol{v}_c^q = \nu_{c,3}\boldsymbol{\chi}_3$, in the following simulation, $\mathcal{B}$ will implicitly set

$$\boldsymbol{v}_c = \boldsymbol{v}_c^p + (c)\boldsymbol{v}_c^q.$$

$\mathcal{B}$ creates a ciphertext $\langle R, (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l\rangle$ as follows:

1. For each $i \in [n]$:

- if $i < \bar{i}$: it chooses random $\hat{s}_i \in \mathbb{Z}_p$, and sets

$$\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa},$$
$$\boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = h^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},$$
$$T_i = e(g,g)^{\hat{s}_i}.$$

- if $i = \bar{i}$: it sets

$$\boldsymbol{R}_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{s_{\bar{i}}\boldsymbol{v}_{\bar{i}}}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\kappa s_{\bar{i}}\boldsymbol{v}_{\bar{i}}},$$
$$\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{b}_1+\boldsymbol{b}_2)}C^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{(\theta+\sum_{j'\in\bar{R}^*} \upsilon_{j'})}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},$$
$$T_i = M\frac{e(A,B)^{2\psi\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}e(g,T)^{2\psi\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)}e(g,g)^{\psi(\alpha'_{i,1}+\alpha'_{i,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}e(g,C)^{\psi(\alpha'_{i,1}+\alpha'_{i,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)}}{(F'_1F'_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}\big(e(C,h)\prod_{j'\in\bar{R}_i}e(C,h_{j'})\big)^{\psi(\alpha_1+\alpha_2)\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)}F_1^{\pi_1}F_2^{\pi_2}},$$

where ${F_1}' = F_1 \prod_{j'\in\bar{R}_i} F_{1,j'}$ and ${F_2}' = F_2 \prod_{j'\in\bar{R}_i} F_{2,j'}$ respectively.

- if $i > \bar{i}$: it sets

$$\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{r_i s_i\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{\kappa r_i s_i\boldsymbol{v}_i},$$
$$\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{(\theta+\sum_{j'\in\bar{R}^*} \upsilon_{j'})}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},$$
$$T_i = M\frac{e(g,g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}}{(F'_1F'_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}F_1^{\pi_1}F_2^{\pi_2}},$$

where ${F_1}' = F_1 \prod_{j'\in\bar{R}_i} F_{1,j'}$ and ${F_2}' = F_2 \prod_{j'\in\bar{R}_i} F_{2,j'}$ respectively.

2. For each $j \in [n]$: Since $j < n+1$, $\mathcal{B}$ chooses random $\mu'_j \in \mathbb{Z}_p$ and implicitly sets the value of $\mu_j$ such that $\mu_j = \mu'_j - c\nu_{c,3}$, then sets

$$\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau(\boldsymbol{v}_c^p+\mu'_j\boldsymbol{\chi}_3)}(\boldsymbol{Y}_j)^{\kappa\boldsymbol{w}_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}.$$

3. $\boldsymbol{P}_0 = h^{\pi_1\boldsymbol{b}_{0,1}+\pi_2\boldsymbol{b}_{0,2}}$, $\{\boldsymbol{P}_k = h^{(A_k\cdot\boldsymbol{u}_1+\xi_{k,1})\boldsymbol{b}_{\rho(k),1}-\xi_{k,1}\boldsymbol{b}_{\rho(k),2}+(A_k\cdot\boldsymbol{u}_2+\xi_{k,2})\boldsymbol{b}_{\rho(k),3}-\xi_{k,2}\boldsymbol{b}_{\rho(k),4}}\}_{k\in[l]}$.

If $T$ corresponds to $g^{abc}$, then the encryption corresponds to game $H_2$; and if $T$ is randomly chosen, then the encryption corresponds to game $H_3$.

**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this $b'$ to the challenger.

The advantage of $\mathcal{B}$ is exactly equal to the advantage of the adversary $\mathcal{A}$.

**Lemma 12.** *If the D3DH assumption holds, then no PPT adversary can distinguish between games $H_3$ and $H_4$ with non-negligible probability.*

*Proof.* $H_3$ to $H_4$ can be expressed as a series of games $H_{3,n+1}, H_{3,n}, \ldots, H_{3,1}$. In the game $H_{3,\hat{j}}$ all column ciphertexts $(\boldsymbol{C}_j, \boldsymbol{C}'_j)$ are well-formed for all $j$ such that $\hat{j} \le j \le n$. It can be seen that $H_{3,1}$ is the same as $H_4$, and $H_{3,n+1}$ is the same as $H_3$. We prove the indistinguishability of games $H_{3,\hat{j}}$ and $H_{3,\hat{j}+1}$ for all $\hat{j}$ where $1 \le \hat{j} \le n$. The proof for this is similar to that of Lemma 1.

Consider an adversary $\mathcal{A}$ that solves the index hiding game with a probability greater than $\epsilon$. The adversary is considered successful if it can distinguish between games $H_{3,\hat{j}}$ and $H_{3,\hat{j}+1}$. We build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the D3DH problem. $\mathcal{B}$ receives the D3DH challenge as $((p, \mathbb{G}, \mathbb{G}_T, e), g, A = g^a, B = g^b, C = g^c, T)$, and it is expected to guess if $T$ is $g^{abc}$ or if it is random. $\mathcal{B}$ interacts with $\mathcal{A}$ in the $\mathsf{Game}_{\mathsf{IH}}$ as follows:

**Setup.** $\mathcal{B}$ randomly chooses two pairs of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ of dimension 3 and $\mathcal{U}$ pairs of dual orthonormal bases $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ of dimension 6, subject to the constraint that all of these share the same value of $\psi$.

$\mathcal{B}$ also randomly chooses

$$\theta,\ \alpha_1, \alpha_2 \in \mathbb{Z}_p,\quad \{r_i,\ z_i,\ \alpha_{i,1}, \alpha_{i,2} \in \mathbb{Z}_p\}_{i\in[n]},$$
$$\{c_{j,1}, c_{j,2},\ y_j, \in \mathbb{Z}_p\}_{j\in[n]\setminus\{\hat{j}\}},\quad c'_{\hat{j},1}, c'_{\hat{j},2}, y'_{\hat{j}} \in \mathbb{Z}_p,\quad \{\upsilon_j \in \mathbb{Z}_p\}_{j\in[n]}.$$

$\mathcal{B}$ sets the public parameter to

$$\Big( g, h = g^{\theta}, g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, \{h_j\}_{j\in[n]},\ h^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\theta}, h^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\theta},$$
$$\{h_j^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\upsilon_j}, h_j^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\upsilon_j}\}_{j\in[n]},\quad h^{\boldsymbol{b}_{0,1}}, h^{\boldsymbol{b}_{0,2}},\ \{h^{\boldsymbol{b}_{x,1}}, \ldots, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]},$$
$$F_1 = e(g,h)^{\psi\alpha_1}, F_2 = e(g,h)^{\psi\alpha_2},\ \{F_{1,j} = e(g,h_j)^{\psi\alpha_1}, F_{2,j} = e(g,h_j)^{\psi\alpha_2}\}_{j\in[n]},$$
$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad E_{i,1} = e(g,g)^{\psi\alpha_{i,1}}, E_{i,2} = e(g,g)^{\psi\alpha_{i,2}}\}_{i\in[n]},$$
$$\{\boldsymbol{H}_j = g^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*},\quad \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]\setminus\{\hat{j}\}},\quad \boldsymbol{H}_{\hat{j}} = C^{c'_{\hat{j},1}\boldsymbol{b}_1^*+c'_{\hat{j},2}\boldsymbol{b}_2^*},\quad \boldsymbol{Y}_{\hat{j}} = g^{y'_{\hat{j}}(c'_{\hat{j},1}\boldsymbol{b}_1^*+c'_{\hat{j},2}\boldsymbol{b}_2^*)} \Big).$$

Note that $\mathcal{B}$ implicitly sets

$$c_{\hat{j},1} = cc'_{\hat{j},1},\ c_{\hat{j},2} = cc'_{\hat{j},2},\quad y_j = y'_j/c.$$

**Key Query.** To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ randomly chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, then creates a private key as

$$\bar{\boldsymbol{K}}_{i,j} = \begin{cases} g^{(\alpha_{i,1}+r_ic_{j,1})\boldsymbol{b}_1^*+(\alpha_{i,2}+r_ic_{j,2})\boldsymbol{b}_2^*}(hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}, & : j \neq \hat{j} \\ g^{\alpha_{i,1}\boldsymbol{b}_1^*+\alpha_{i,2}\boldsymbol{b}_2^*}C^{r_i(c'_{j,1}\boldsymbol{b}_1^*+c'_{j,2}\boldsymbol{b}_2^*)}(hh_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}, & : j = \hat{j} \end{cases}$$
$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},$$
$$\{\bar{\boldsymbol{K}}_{i,j,j'} = h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}\}_{j'\in[n]\setminus\{j\}},$$
$$\boldsymbol{K}_{i,j,0} = g^{\delta_{i,j,1}\boldsymbol{b}_{0,1}^*+\delta_{i,j,2}\boldsymbol{b}_{0,2}^*},$$
$$\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{b}_{x,1}^*+\boldsymbol{b}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{b}_{x,3}^*+\boldsymbol{b}_{x,4}^*)}\ \forall x \in S_{(i,j)}.$$

**Challenge.** $\mathcal{A}$ submits a revocation list $R$, a message $M$ and an attribute set $S^*$. $\mathcal{B}$ constructs the LSSS matrix $(A, \rho)$ for $\mathbb{A}_{S^*}$. Let $l \times m$ be the size of $A$.

$\mathcal{B}$ chooses random

$$\tau',\quad s_1, \ldots, s_n,\quad t_1, \ldots, t_n\ \in \mathbb{Z}_p,$$
$$\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\hat{j}-1}, \boldsymbol{w}'_{\hat{j}}, \ldots, \boldsymbol{w}'_n\ \in \mathbb{Z}_p^3,$$
$$\xi_{1,1}, \xi_{1,2}, \ldots, \xi_{l,1}, \xi_{l,2}\ \in \mathbb{Z}_p,\quad \boldsymbol{u}_1, \boldsymbol{u}_2\ \in \mathbb{Z}_p^m,$$

where the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are equal to $\pi_1$ and $\pi_2$ respectively.

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_yr_z, -r_xr_z, r_xr_y)$, then it chooses random

$$\boldsymbol{v}_i \in \mathbb{Z}_p^3\ \textit{for}\ i = 1, \ldots, \bar{i},$$
$$\boldsymbol{v}_i \in \textit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ \textit{for}\ i = \bar{i}+1, \ldots, n.$$

$\mathcal{B}$ chooses random $(\nu_{c,1}, \nu_{c,2}, \nu_{c,3}) \in \mathbb{Z}_p^3$. Let $\boldsymbol{v}_c^p = \nu_{c,1}\boldsymbol{\chi}_1 + \nu_{c,2}\boldsymbol{\chi}_2$ and $\boldsymbol{v}_c^q = \nu_{c,3}\boldsymbol{\chi}_3$, in the following simulation, $\mathcal{B}$ will implicitly set

$$\boldsymbol{v}_c = a^{-1}\boldsymbol{v}_c^p + \boldsymbol{v}_c^q.$$

$\mathcal{B}$ creates a ciphertext $\langle R, (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows:

1. For each $i \in [n]$:
   - if $i \leq \bar{i}$: it chooses random $\hat{s}_i \in \mathbb{Z}_p$, and sets

$$\begin{aligned}&\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i},\\ &\boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = h^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ &T_i = e(g,g)^{\hat{s}_i}.\end{aligned}$$

   - if $i > \bar{i}$: it sets

$$\begin{aligned}&\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{r_i s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{r_i s_i \boldsymbol{v}_i},\\ &\boldsymbol{Q}_i = B^{\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{(\theta+\sum_{j'\in\bar{R}^*} \upsilon_{j'})}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ &T_i = M\frac{e(B,g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}}{\big(e(B,h)\prod_{j'\in\bar{R}^*} e(g,h_{j'})\big)^{\psi(\alpha_1+\alpha_2)\tau' s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}F_1^{\pi_1}F_2^{\pi_2}}.\end{aligned}$$

2. For each $j \in [n]$:
   - if $j < \hat{j}$: it chooses random $\mu'_j \in \mathbb{Z}_p$ and implicitly sets the value of $\mu_j$ such that $(\frac{\mu'_j}{ab}-1)\nu_{c,3} \equiv \mu_j \bmod p$, then sets

$$\boldsymbol{C}_j = (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^p}\ (g^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\mu'_j\tau'\boldsymbol{v}_c^q}\ (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{y_j\boldsymbol{w}_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}.$$

   - if $j = \hat{j}$:

$$\boldsymbol{C}_j = (T^{c'_{\hat{j},1}\boldsymbol{b}_1^*+c'_{\hat{j},2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^q}\ (B^{c'_{j,1}\boldsymbol{b}_1^*+c'_{j,2}\boldsymbol{b}_2^*})^{y'_{\hat{j}}\boldsymbol{w}'_{\hat{\boldsymbol{j}}}}, \quad \hat{\boldsymbol{C}}_{\boldsymbol{j}} = (\boldsymbol{Y}_{\hat{j}})^{\boldsymbol{w}'_{\hat{\boldsymbol{j}}}}\ (C^{c'_{\hat{j},1}\boldsymbol{b}_1^*+c'_{\hat{j},2}\boldsymbol{b}_2^*})^{-\tau'\boldsymbol{v}_c^p}.$$

   - if $j > \hat{j}$:

$$\boldsymbol{C}_j = (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{\tau'\boldsymbol{v}_c^p}\ (B^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{y_j\boldsymbol{w}'_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}'_j}\ (A^{c_{j,1}\boldsymbol{b}_1^*+c_{j,2}\boldsymbol{b}_2^*})^{-\tau'\boldsymbol{v}_c^q}.$$

3. $\boldsymbol{P}_0 = h^{\pi_1\boldsymbol{b}_{0,1}+\pi_2\boldsymbol{b}_{0,2}}$, $\{\boldsymbol{P}_k = h^{(A_k\cdot\boldsymbol{u}_1+\xi_{k,1})\boldsymbol{b}_{\rho(k),1}-\xi_{k,1}\boldsymbol{b}_{\rho(k),2}+(A_k\cdot\boldsymbol{u}_2+\xi_{k,2})\boldsymbol{b}_{\rho(k),3}-\xi_{k,2}\boldsymbol{b}_{\rho(k),4}}\}_{k\in[l]}$.

Note that $\mathcal{B}$ implicitly chooses $\kappa, \tau \in \mathbb{Z}_p$ and $\boldsymbol{w}_j \in \mathbb{Z}_p^3(\hat{j} \leq j \leq n)$ such that

$$\begin{aligned}b \equiv \kappa \bmod p, \quad ab\tau' &\equiv \tau \bmod p,\\ \boldsymbol{w}'_{\hat{j}} - c\tau'\boldsymbol{v}_c^p/y'_{\hat{j}} &\equiv \boldsymbol{w}_{\hat{j}} \bmod p,\\ \boldsymbol{w}'_j - a\tau'\boldsymbol{v}_c^q/y_j &\equiv \boldsymbol{w}_j \bmod p\ \forall j \in \{\hat{j}+1,\ldots,n\}.\end{aligned}$$

If $T = g^{abc}$, then the encryption corresponds to the game $H_{3,\hat{j}}$; and if $T$ is randomly chosen, say $T = g^r$ for some random $r \in \mathbb{Z}_p$, then the encryption corresponds the game $H_{3,\hat{j}+1}$ with implicitly setting $\mu_{\hat{j}}$ such that $(\frac{r}{abc} - 1)\nu_{c,3} \equiv \mu_{\hat{j}} \bmod p$.

**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this $b'$ to the challenger.

The advantage of $\mathcal{B}$ is exactly equal to the advantage of the adversary $\mathcal{A}$.

**Lemma 13.** *If the DLIN assumption holds, then no PPT adversary can distinguish between games $H_4$ and $H_5$ with non-negligible probability.*

*Proof.* Consider an adversary $\mathcal{A}$ that can distinguish between $H_4$ and $H_5$ with a probability greater than $\epsilon$. We build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the DLIN problem. $\mathcal{B}$ receives the DLIN challenge as $(\mathbb{G}, g, g^a, g^b, g^c, g^{ax}, g^{by}, T)$, and it is expected to guess if $T$ is $g^{c(x+y)}$ or if it is random. Then $\mathcal{B}$ interacts with $\mathcal{A}$ in the $\mathsf{Game}_{\mathsf{IH}}$ as follows:

**Setup.** $\mathcal{B}$ randomly chooses two pairs of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ of dimension 3 and $\mathcal{U}$ pairs of dual orthonormal bases $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*)$ of dimension 6, subject to the constraint that all of these share the same value of $\psi$.
$\mathcal{B}$ also randomly chooses

$$\theta,\ \alpha_1, \alpha_2 \in \mathbb{Z}_p,\quad \{r_i,\ z_i,\ \alpha_{i,1}, \alpha_{i,2} \in \mathbb{Z}_p\}_{i\in[n]},\quad \{c_{j,1}, c_{j,2},\ y_j, \upsilon_j \in \mathbb{Z}_p\}_{j\in[n]}.$$

$\mathcal{B}$ sets the public parameter to

$$\Big( g, h = g^{\theta},\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2},\ \{h_j\}_{j\in[n]}, h^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\theta}, h^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\theta},\ \{h_j^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\upsilon_j}, h_j^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\upsilon_j}\}_{j\in[n]},$$
$$h^{\boldsymbol{b}_{0,1}} = (g^{\boldsymbol{b}_{0,1}})^{\theta}, h^{\boldsymbol{b}_{0,2}} = (g^{\boldsymbol{b}_{0,2}})^{\theta},\ \{h^{\boldsymbol{b}_{x,1}}, \ldots, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]},$$
$$F_1 = e(g,h)^{\psi\alpha_1}, F_2 = e(g,h)^{\psi\alpha_2},\ \{F_{1,j} = e(g,h_j)^{\psi\alpha_1}, F_{2,j} = e(g,h_j)^{\psi\alpha_2}\}_{j\in[n]},$$
$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad E_{i,1} = e(g,g)^{\psi\alpha_{i,1}}, E_{i,2} = e(g,g)^{\psi\alpha_{i,2}}\}_{i\in[n]},$$
$$\{\boldsymbol{H}_j = g^{c_{j,1}\boldsymbol{b}_1^* + c_{j,2}\boldsymbol{b}_2^*},\quad \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]} \Big).$$

**Key Query.** To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ randomly chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, then creates a private key as

$$\boldsymbol{K}_{i,j} = g^{(\alpha_{i,1}+r_i c_{j,1})\boldsymbol{b}_1^* + (\alpha_{i,2}+r_i c_{j,2})\boldsymbol{b}_2^*} (h h_j)^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},$$
$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},$$
$$\{\bar{\boldsymbol{K}} = h_{j'}^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}\}_{j'\in[n]\setminus\{j\}},$$
$$\boldsymbol{K}_{i,j,0} = g^{\delta_{i,j,1}\boldsymbol{b}_{0,1}^* + \delta_{i,j,2}\boldsymbol{b}_{0,2}^*},$$
$$\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{b}_{x,1}^* + \boldsymbol{b}_{x,2}^*) + \sigma_{i,j,2}(\boldsymbol{b}_{x,3}^* + \boldsymbol{b}_{x,4}^*)}\ \forall x \in S_{(i,j)}.$$

**Challenge.** $\mathcal{A}$ submits a revocation list $R$, a message $M$ and an attribute set $S^*$. $\mathcal{B}$ constructs the LSSS matrix $(A, \rho)$ for $\mathbb{A}_{S^*}$. Let $l \times m$ be the size of $A$.
$\mathcal{B}$ chooses random

$$\begin{aligned} \kappa, \tau, \quad s_1, \ldots, s_n, \quad t_1, \ldots, t_n &\in \mathbb{Z}_p, \\ \boldsymbol{v}_c, \quad \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n &\in \mathbb{Z}_p^3, \\ \xi_{1,1}, \xi_{1,2}, \ldots, \xi_{l,1}, \xi_{l,2} \in \mathbb{Z}_p, \quad \boldsymbol{u}_1, \boldsymbol{u}_2 &\in \mathbb{Z}_p^m, \end{aligned}$$

where the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are equal to $\pi_1$ and $\pi_2$ respectively.
$\mathcal{B}$ implicitly sets $\boldsymbol{\chi}_1 = (a, 0, c), \boldsymbol{\chi}_2 = (0, b, c), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-bc, -ac, ab)$. Note that a valid DLIN tuple will lie in the subspace formed by vectors $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$. In the following, a DLIN problem tuple will be used for setting row ciphertext for row $\bar{i}+1$. A valid tuple leads to encryption as in game $H_4$, and a random tuple will cause the encryption to be as in game $H_5$.
$\mathcal{B}$ creates a ciphertext $\langle R, (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l\rangle$ as follows:

1. For each $i \in [n]$:
   - if $i \leq \bar{i}$: it chooses random $\boldsymbol{v}_i \in \mathbb{Z}_p^3$ and $\hat{s}_i \in \mathbb{Z}_p$. Then it sets

$$\begin{aligned} &\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa}, \\ &\boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = h^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} h^{\pi_1 \boldsymbol{b}_1 + \pi_2 \boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \\ &T_i = e(g,g)^{\hat{s}_i}. \end{aligned}$$

   - if $i = \bar{i}+1$: $\mathcal{B}$ implicitly chooses $\boldsymbol{v}_i \in \mathbb{Z}_p^3$ such the $g^{\boldsymbol{v}_i} = (g^{ax}, g^{by}, T)$. Since $\mathcal{B}$ knows the values of $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{v}_c$, it can compute the value of $(g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}$ and $g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$. Then it sets

$$\begin{aligned} &\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\kappa s_i \boldsymbol{v}_i}, \\ &\boldsymbol{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{(\theta + \sum_{j' \in \bar{R}^*} \upsilon_{j'})} \boldsymbol{Z}_i^{t_i} h^{\pi_1 \boldsymbol{d}_1 + \pi_2 \boldsymbol{d}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \\ &T_i = M \frac{e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau s_i}}{\big(e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, h) \prod\limits_{j' \in \bar{R}^*} e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, h_{j'})\big)^{\psi(\alpha_1+\alpha_2)\tau s_i} F_1^{\pi_1} F_2^{\pi_2}}. \end{aligned}$$

   - if $i > \bar{i}+1$: it chooses random $\boldsymbol{v}_i \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, i.e., chooses random $\nu_{i,1}, \nu_{i,2} \in \mathbb{Z}_p$ and sets $\boldsymbol{v}_i = \nu_{i,1}\boldsymbol{\chi}_1 + \nu_{i,2}\boldsymbol{\chi}_2$. $\mathcal{B}$ cannot compute the value of $\boldsymbol{v}_i$, but it can compute the value of $g^{\boldsymbol{v}_i}$, i.e., $g^{\boldsymbol{v}_i} = \big((g^a)^{\nu_{i,1}}, (g^b)^{\nu_{i,2}}, (g^c)^{\nu_{i,1}+\nu_{i,2}}\big)$. Also, since $\mathcal{B}$ knows the values of $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{v}_c$, it can compute the value of $(g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i}$ and $g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$. Then it sets

$$\begin{aligned} &\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{r_i s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\kappa r_i s_i \boldsymbol{v}_i}, \\ &\boldsymbol{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{(\theta + \sum_{j' \in \bar{R}^*} \upsilon_{j'})} \boldsymbol{Z}_i^{t_i} h^{\pi_1 \boldsymbol{d}_1 + \pi_2 \boldsymbol{d}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \\ &T_i = M \frac{e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau s_i}}{\big(e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, h) \prod\limits_{j' \in \bar{R}^*} e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, h_{j'})\big)^{\psi(\alpha_1+\alpha_2)\tau s_i} F_1^{\pi_1} F_2^{\pi_2}}. \end{aligned}$$

2. For each $j \in [n]$: since $j \geq 1$, $\mathcal{B}$ sets

$$\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau \boldsymbol{v}_c} (\boldsymbol{Y}_j)^{\kappa \boldsymbol{w}_j}, \quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}.$$

3. $\boldsymbol{P}_0 = h^{\pi_1 \boldsymbol{b}_{0,1} + \pi_2 \boldsymbol{b}_{0,2}}$, $\{\boldsymbol{P}_k = h^{(A_k \cdot \boldsymbol{u}_1 + \xi_{k,1})\boldsymbol{b}_{\rho(k),1} - \xi_{k,1}\boldsymbol{b}_{\rho(k),2} + (A_k \cdot \boldsymbol{u}_2 + \xi_{k,2})\boldsymbol{b}_{\rho(k),3} - \xi_{k,2}\boldsymbol{b}_{\rho(k),4}}\}_{k \in [l]}$.

If $T$ corresponds to $g^{c(x+y)}$, then the encryption corresponds to game $H_4$; and if $T$ is randomly chosen, then it corresponds to game $H_5$.

**Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this $b'$ to the challenger.

The advantage of $\mathcal{B}$ is exactly equal to the advantage of the adversary $\mathcal{A}$.

## C Access Structure and Linear Secret-Sharing Schemes

**Definition 6. (Access Structure)** *[23] Let $\mathcal{P}$ be a set of attributes. A collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is monotone if $\forall B, C$ : $B \in \mathbb{A}$ and $B \subseteq C$ imply $C \in \mathbb{A}$. An access structure (resp., monotone access structure) is a collection (resp., monotone collection) $\mathbb{A}$ of non-empty subsets of $\mathcal{P}$, i.e., $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets. Also, for an attribute set $S \subseteq \mathcal{P}$, if $S \in \mathbb{A}$ then we say $S$ satisfies the access structure $\mathbb{A}$, otherwise we say $S$ does not satisfy $\mathbb{A}$.*

As shown in [2], any monotonic access structure can be realized by a linear secret sharing scheme.

**Definition 7. (Linear Secret-Sharing Schemes (LSSS))** *[23] A secret sharing scheme $\Pi$ over a set of attributes $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if*

1. *The shares for each attribute form a vector over $\mathbb{Z}_p$.*
2. *There exists a matrix $A$ called the share-generating matrix for $\Pi$. The matrix $A$ has $l$ rows and $n$ columns. For $i = 1, \ldots, l$, the $i^{th}$ row $A_i$ of $A$ is labeled by an attribute $\rho(i)$ ($\rho$ is a function from $\{1, \ldots, l\}$ to $\mathcal{P}$). When we consider the column vector $\boldsymbol{v} = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $A\boldsymbol{v}$ is the vector of $l$ shares of the secret $s$ according to $\Pi$. The share $\lambda_i = (A\boldsymbol{v})_i$, i.e., the inner product $A_i \cdot \boldsymbol{v}$, belongs to attribute $\rho(i)$.*

Also shown in [2], every LSSS as defined above enjoys the linear reconstruction property, which is defined as follows: Suppose that $\Pi$ is an LSSS for access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be an authorized set, and $I \subset \{1, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. There exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i A_i = (1, 0, \ldots, 0)$, so that if $\{\lambda_i\}$ are valid shares of a secret $s$ according to $\Pi$, $\sum_{i \in I} \omega_i \lambda_i = s$. Furthermore, these constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix $A$. For any unauthorized set, no such constants exist.