Traceable CP-ABE on Prime Order Groups: Fully Secure and Fully Collusion-resistant Blackbox Traceable

Zhen Liu and Duncan S. Wong

Security and Data Sciences, ASTRI, Hong Kong SAR, China
zhenliu7-c@my.cityu.edu.hk, duncanwong@astri.org

Abstract. In Ciphertext-Policy Attribute-Based Encryption (CP-ABE), access policies associated with the ciphertexts are generally role-based and the attributes satisfying the policies are generally *shared* by multiple users. If a malicious user, with his attributes shared with multiple other users, created a decryption blackbox for sale, this malicious user could be difficult to identify from the blackbox. Hence in practice, a useful CP-ABE scheme should have some tracing mechanism to identify this 'traitor' from the blackbox. In this paper, we propose the first CP-ABE scheme which simultaneously achieves (1) fully collusion-resistant blackbox traceability in the standard model, (2) full security in the standard model, and (3) on prime order groups. When compared with the latest fully collusion-resistant blackbox traceable CP-ABE schemes, this new scheme achieves the same efficiency level, enjoying the sub-linear overhead of $O(\sqrt{N})$, where N is the number of users in the system. This new scheme is highly expressive and can take any monotonic access structures as ciphertext policies.

Keywords: Traceable, Ciphertext-policy Attribute Based Encryption, Prime Order Groups

1 Introduction

In a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [10,2] system, each user possesses a set of attributes and a private key which is generated according to his attributes, and the encrypting party does not need to know or specify the exact identities of the targeted receivers, instead, the encrypting party can define an *access policy* over role-based/descriptive *attributes* to encrypt a message, so that only the users whose attributes satisfy the access policy can decrypt the ciphertext. For example, a school secretary, say Alice, may encrypt some messages using "(Mathematics AND (PhD Student OR Alumni))", which is an *access policy* defined over descriptive *attributes*, say "Mathematics", "PhD Student", and "Alumni", so that only PhD students and alumni in the Department of Mathematics have access to the messages. Due to the high flexibility and expressivity of the access policy, CP-ABE has promising applications related to access control, such as secure cloud storage access and sharing, and has attracted great attention in the research community. Among the CP-ABE schemes recently proposed, [2,5,9,24,14,22,11,15,23], progress has been made on 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 [14,22,15], the traceability of traitors which intentionally expose their decryption keys has been becoming an important concern related to the applicability of CP-ABE. Specifically, due to the nature of CP-ABE, access policies associated with the ciphertexts do not have to contain the exact identities of the eligible receivers. Instead, access policies are role-based and 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 there is little risk of getting caught. While all the aforementioned CP-ABE schemes lack the traitor tracing functionality, recently a handful of traceable CP-ABE schemes have been proposed in [18,17,6].

In the aforementioned non-traceable CP-ABE schemes, an easy and attractive way for a malicious user to make money is to sell a well-formed decryption key where the corresponding attribute set does not contain his identity-related attributes. For example, a malicious user with attributes {Bob, PhD, Mathematics} may build and sell a new decryption key with attributes {PhD, Mathematics}, and does not worry getting caught, since many other users share the attributes {PhD, Mathematics}. Liu et al. [18] proposed a whitebox traceable CP-ABE scheme that can deter users from such malicious behaviours, i.e., given a well-formed decryption key as input, a tracing algorithm can find out the malicious user who created the key from his/her original key. To avoid the whitebox traceability, instead of selling a well-formed decryption key, a more sophisticated malicious user may build and sell a decryption device/blackbox while keeping the embedded decryption key and algorithm hidden. Liu et al. [17] proposed a blackbox traceable CP-ABE scheme that can deter users from these more practical attacks, i.e., given a decryption blackbox/device, while the decryption key and even the decryption algorithm could be hidden, the tracing algorithm, which treats the decryption blackbox as an oracle, can still find out the malicious user whose key must have been used in constructing the decryption blackbox. Liu et al. proved that the CP-ABE scheme in [17] is fully secure in the standard model and 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 is not limited and can be arbitrary. In addition, the scheme in [17] is highly expressive (i.e. supporting any monotonic access structures), and as a fully collusion-resistant blackbox traceable CP-ABE scheme, it achieves the most efficient level to date, i.e. the overhead for the fully collusion-resistant blackbox traceability is in $O(\sqrt{N})$, where N is the number of users in the system. However, the scheme in [17] is based on composite order groups with order being the product of three large primes, and this severely limits its applicability. Liu and Wong [19] proposed a fully collusion-resistant blackbox traceable CP-ABE scheme on prime order groups, but achieves only selective security, where the adversary is required to declare his attacking target before seeing the system public key. Another recent blackbox traceable CP-ABE scheme is due to Deng et al. [6], which is only *t-collusion-resistant* traceable, where the number of colluding users is limited, i.e., less than a parameter t. In addition, the scheme in [6] is only selectively secure and the security is proven in the random oracle model.

1.1 Our Results

In this paper, we propose a new CP-ABE scheme that is fully secure in the standard model, fully collusion-resistant blackbox traceable in the standard model, and highly expressive (i.e. supporting any monotonic access structures). On the efficiency, as a fully collusion-resistant blackbox traceable CP-ABE scheme, this new scheme also achieves the most efficient level to date, i.e. the overhead for the fully collusion-resistant blackbox traceability is in $O(\sqrt{N})$. When compared with the CP-ABE scheme in [17], the advantage of this new scheme is that this scheme is constructed on prime order groups. Note that this implies this new scheme has better security and performance than the scheme in [17], although both of them are fully secure in the standard model and have overhead in $O(\sqrt{N})$. More specifically, as it has been shown (e.g. in [8,13]), 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. To the best of our knowledge, this is the first CP-ABE scheme that is fully collusion-resistant blackbox traceable, fully secure, and constructed on prime order groups. Table 1 compares this new scheme with that in [17] in terms of performance, as both of the schemes are fully secure in the standard

	Ciphertext Size	Private Key Size	Public Key Size	Pairing Computation in Decryption	On Prime Order Groups	Order of the Groups
[17]	$2l + 17\sqrt{N}$	$\lvert S\rvert + 4$	$\lvert\mathcal{U}\rvert + 3 + 4\sqrt{N}$	$2\lvert I\rvert + 10$	×	$p_1p_2p_3$
this work	$6l + 3 + 46\sqrt{N}$	$6\lvert S\rvert + 12$	$24\lvert\mathcal{U}\rvert + 20 + 14\sqrt{N}$	$6\lvert I\rvert + 30$	$\surd$	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. Comparison with the fully collusion-resistant blackbox CP-ABE in [17]

model, fully collusion-resistant blackbox traceable in the standard model, and highly expressive (i.e. supporting any monotonic access structures).

Related Work. In [17] 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. It will be tempting to obtain a prime order construction by applying the existing general tools of converting constructions from composite order groups to prime order groups, e.g. [7,13], to the composite order group construction of [17]. However, as the traceability is a new feature of CP-ABE and these tools focus on the conventional security (i.e. hiding the messages), it is not clear whether these tools are applicable to the traceable CP-ABE of [17].

Outline. In this paper, we also follow the framework in [17]. In particular, in Section 2 we review the definitions and security models of AugCP-ABE, then in Section 3 we propose our AugCP-ABE construction on prime order groups and prove that our AugCP-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 CP-ABE scheme on prime order groups.

2 Augmented CP-ABE Definitions

In this section, we review the definitions of Augmented CP-ABE, which is proposed by Liu et al. [17] as a primitive that help constructing fully collusion-resistant blackbox traceable CP-ABE.

2.1 Definitions and Security Models

Given a positive integer n, let $[n]$ be the set $\{1, 2, \ldots, n\}$. An Augmented CP-ABE (AugCP-ABE) system consists of the following four algorithms:

- $\mathsf{Setup}_\mathsf{A}(\lambda, \mathcal{U}, N) \to (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter λ, the attribute universe $\mathcal{U}$, and the number of users N in the system, then runs in polynomial time in λ, and outputs the public parameter PP and a master secret key MSK.
- $\mathsf{KeyGen}_\mathsf{A}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. The algorithm takes as input PP, MSK, and an attribute set S, and outputs a private key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in [N]$.

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

$\mathsf{Decrypt}_\mathsf{A}(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) \to M$ or $\perp$. The algorithm takes as input PP, a ciphertext CT, and a private key $\mathsf{SK}_{k,S}$. If S satisfies the ciphertext access policy, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.

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

Security. The security of AugCP-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. It is worth noticing that, as pointed in [17], in the three games: (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.

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 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 an access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M_b, \mathbb{A}^*, \bar{k})$ 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 $CT \leftarrow \mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M_b, \mathbb{A}^*, 1)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that $\mathbb{A}^*$ cannot be satisfied by any of the queried attribute sets $S_{k_1}, \ldots, S_{k_Q}$. 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 $CT \leftarrow \mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M_b, \mathbb{A}^*, N+1)$ 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 1. *A N-user Augmented 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 λ.*

$\mathsf{Game}^\mathsf{A}_\mathsf{IH}$. 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 $(\mathbb{A}_{S^*}, \bar{k})$ and $(\mathbb{A}_{S^*}, \bar{k}+1)$ without a private decryption key $\mathsf{SK}_{\bar{k}, S_{\bar{k}}}$ such that $S_{\bar{k}} \supseteq S^*$. 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 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 non-empty attribute set S^*. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}_{S^*}, \bar{k}+b)$ 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 $\{(k_t, S_{k_t})\}_{1\le t\le Q}$ can satisfy $(k_t = \bar{k}) \wedge (S_{k_t}$ satisfies $\mathbb{A}_{S^*})$, i.e. $(k_t = \bar{k}) \wedge (S_{k_t} \supseteq S^*)$. The advantage of $\mathcal{A}$ is defined as $\mathsf{IH^A Adv}_{\mathcal{A}}[\bar{k}] = |\Pr[b'=b] - \frac{1}{2}|$.

Definition 2. *A N-user Augmented CP-ABE system is index-hiding if for all PPT adversaries $\mathcal{A}$ the advantages $\mathsf{IH^A Adv}_{\mathcal{A}}[\bar{k}]$ for $\bar{k} = 1, \ldots, N$ are negligible in λ.*

2.2 The Reduction of Traceable CP-ABE to Augmented CP-ABE

Let $\Sigma_\mathsf{A} = (\mathsf{Setup_A}, \mathsf{KeyGen_A}, \mathsf{Encrypt_A}, \mathsf{Decrypt_A})$ be an AugCP-ABE, define $\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}) = \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}, 1)$, and let $\Sigma = (\mathsf{Setup_A}, \mathsf{KeyGen_A}, \mathsf{Encrypt}, \mathsf{Decrypt_A})$. It is apparent that Σ is a 'functional' CP-ABE that has the same functionality as the conventional CP-ABE, except that the number of users in the system is predefined and each user is assigned and identified by a unique index[1]. As shown in [17], with the following Trace algorithm [17], Σ achieves fully collusion-resistant blackbox traceability against key-like decryption blackbox[2].

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq \{1, \ldots, N\}$: 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 $k = 1$ 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 $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}_{S_{\mathcal{D}}}, k)$, where $\mathbb{A}_{S_{\mathcal{D}}}$ is the strictest access policy of $S_{\mathcal{D}}$.
      iii. Query oracle $\mathcal{D}$ on input CT, and compare the output of $\mathcal{D}$ with M.
   (b) Let $\hat{p}_k$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
2. Let $\mathbb{K}_T$ be the set of all $k \in \{1, \ldots, N\}$ for which $\hat{p}_k - \hat{p}_{k+1} \ge \epsilon/(4N)$. Then output $\mathbb{K}_T$ as the index set of the private keys of malicious users.

Theorem 1. *[17, Theorem 1] If Σ_A is message-hiding and index-hiding, then Σ is secure, and using the* Trace *algorithm, Σ is traceable against key-like decryption blackbox.*

Please refer to [17, Section 2 and 3] for more details, including the formal definitions of the security and traceability, the Trace algorithm, and the proof of the theorem.

3 An Augmented CP-ABE Construction on Prime Order Groups

Now we construct an AugCP-ABE scheme on prime order groups, and prove that this AugCP-ABE scheme is message-hiding and index-hiding in the standard model. Combined with the results in Section 2.2, we obtain a 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.

[1] Note that as pointed in [17], predefining the number of users is indeed a weakness as well as a necessary cost for achieving blackbox traceability, but in practice this should not incur much concern, and all the existing blackbox traceable systems (e.g. [12,3,4,8]) have the same setting.

[2] Roughly speaking, a key-like decryption blackbox $\mathcal{D}$ is described by a non-empty attribute set $S_{\mathcal{D}}$ and a non-negligible probability value ϵ, and for any access policy $\mathbb{A}$, if it is satisfied by $S_{\mathcal{D}}$, this blackbox $\mathcal{D}$ can decrypt the corresponding ciphertext associated with $\mathbb{A}$ with probability at least ϵ. Please refer to [17] for more formal details.

3.1 Preliminaries

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

Bilinear Groups. Let $\mathcal{G}$ be a group generator, which takes a security parameter λ 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 AugCP-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 AugCP-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 [16]. 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 [20,21,22] and developed by Lewko [13] and Lewko and Waters [16]. 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

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

where ψ is a non-zero element of $\mathbb{Z}_p$. (This is a slight abuse of the terminology "orthonormal", since ψ 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 $i \ne j$, 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 AugCP-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 [16], 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 AugCP-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 ρ. An attribute set S satisfies the LSSS access matrix (A, ρ) 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 D.

Notations. Suppose the number of users N in the system equals n^2 for some n [3]. We arrange the users in a $n \times n$ matrix and uniquely assign a tuple (i, j) where $1 \leq i, j \leq n$, 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 $(i, j) \geq (\bar{i}, \bar{j})$ means that $((i > \bar{i}) \vee (i = \bar{i} \wedge j \geq \bar{j}))$. The use of pairwise notation (i, j) is purely a notational convenience, as $k = (i-1) * n + j$ defines a bijection between $\{(i, j) | 1 \leq i, j \leq n\}$ and $\{1, \ldots, N\}$. We conflate the notation and consider the attribute universe to be $[\mathcal{U}] = \{1, 2 \ldots, \mathcal{U}\}$, so $\mathcal{U}$ servers 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 $\mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ be the subspace spanned by $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, i.e. $\mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} = \{\nu_1 \boldsymbol{\chi}_1 + \nu_2 \boldsymbol{\chi}_2 | \nu_1, \nu_2 \in \mathbb{Z}_p\}$. We can see that $\boldsymbol{\chi}_3$ is orthogonal to the subspace $\mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and that $\mathbb{Z}_p^3 = \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\} = \{\nu_1 \boldsymbol{\chi}_1 + \nu_2 \boldsymbol{\chi}_2 + \nu_3 \boldsymbol{\chi}_3 | \nu_1, \nu_2, \nu_3 \in \mathbb{Z}_p\}$. For any $\boldsymbol{v} \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, we have $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) = 0$, and for random $\boldsymbol{v} \in \mathbb{Z}_p^3$, $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) \neq 0$ happens with overwhelming probability.

3.2 AugCP-ABE Construction

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

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

The public parameter PP and the master secret key MSK are set to

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

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

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

$\mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S) \rightarrow \mathsf{SK}_{(i,j),S}$. The algorithm first sets $\mathit{ctr} = \mathit{ctr} + 1$ and computes the corresponding index in the form of (i, j) where $1 \leq i, j \leq n$ and $(i-1) * n + j = \mathit{ctr}$. Then 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

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

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

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A} = (A, \rho), (\bar{i}, \bar{j})) \to CT$. A is an $l \times m$ LSSS matrix and ρ maps each row A_k of A to an attribute $\rho(k) \in [\mathcal{U}]$. 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

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

Let π_1 and π_2 be the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ respectively. The algorithm creates a ciphertext $\langle (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows:

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

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

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

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

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

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT, \mathsf{SK}_{(i,j),S}) \to M$ or $\perp$. The algorithm parses CT and $\mathsf{SK}_{(i,j),S}$ to $\langle (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ and $\langle\ (i,j), S\ \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S}\ \rangle$ respectively. If S does not satisfy (A, ρ), 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. Computes $D_I = \frac{e_3(\boldsymbol{K}_{i,j}, \boldsymbol{Q}_i) \cdot e_3(\boldsymbol{K}''_{i,j}, \boldsymbol{Q}''_i) \cdot e_9(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{K}'_{i,j}, \boldsymbol{Q}'_i) \cdot e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)}$.
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 $\boldsymbol{v}_i \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$), 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 $\mathbb{Z}_p^3$). The correctness can be found in Appendix B.

Remarks: We borrow the ideas of [16, Section 5] to achieve the full security for prime order group constructions, and borrow the ideas of [17] to achieve fully collusion-resistant blackbox traceability. But the above construction and the later security proof are not trivial combinations of the two schemes. In particular, the public parameter components $g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1}, h^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_{0,1}}, h^{\boldsymbol{b}_{0,2}}$, $\{h^{\boldsymbol{b}_{x,1}}, h^{\boldsymbol{b}_{x,2}}, h^{\boldsymbol{b}_{x,3}}, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]}, F_1$, F_2, the key components $\boldsymbol{K}'_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S}$, and ciphertext components $\boldsymbol{P}_0, \{\boldsymbol{P}_k\}_{k\in[l]}$ are designed using the ideas of [16, Section 5]. To achieve fully collusion-resistant blackbox traceability, $\{E_{i,1}, E_{i,2}, \boldsymbol{G}_i, \boldsymbol{Z}_i\}_{i\in[n]}, \{\boldsymbol{H}_j\}_{j\in[n]}$ are put in the public parameter, and $\boldsymbol{K}_{i,j}, \boldsymbol{K}''_{i,j}$ are introduced into the private key. Note that $\boldsymbol{G}_i$ and $\boldsymbol{H}_j$ will be used to generate ciphertext components $\boldsymbol{R}_i$ and $\boldsymbol{C}_j$ respectively, and $e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)$ will be computed during decryption, so that $\boldsymbol{G}_i$ and $\boldsymbol{H}_j$ must use the basis vectors of a pair of dual orthonormal bases, i.e. $\boldsymbol{G}_i$ uses $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ and $\boldsymbol{H}_j$ uses $(\boldsymbol{b}_1^*, \boldsymbol{b}_2^*)$. This prevents us from trivially using the proof of [16, Section 5], because in the construction of [16, Section 5], only $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,2}, \{\boldsymbol{b}_{x,1}, \boldsymbol{b}_{x,2}, \boldsymbol{b}_{x,3}, \boldsymbol{b}_{x,4}\}_{x\in[\mathcal{U}]}$ appear in the exponents of the public parameter components. As an informal evidence, while the AugCP-ABE scheme of [17] reduces its message-hiding property (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) to the security of the CP-ABE scheme of [15], it is impossible to make a similar reduction here, since the public parameter of the above AugCP-ABE construction contains $(\boldsymbol{b}_1^*, \boldsymbol{b}_2^*)$ while the public parameter of [16, Section 5] does not contain them. To address this problem, we introduce a new and crucial public parameter component $\boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}$ which does not have counterpart in the AugCP-ABE scheme of [17] or the CP-ABE scheme in [16, Section 5], and we reduce the message-hiding property of our construction directly to the underlying assumptions.

3.3 Security of The AugCP-ABE Construction

The following Theorem 2 and Theorem 3 show that our AugCP-ABE construction is message-hiding, and Theorem 4 shows that our AugCP-ABE construction is index-hiding.

Theorem 2. *Suppose the DLIN assumption, the D3DH assumption, and the source group q-parallel BDHE assumption hold. Then no PPT adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ *with non-negligible advantage.*

Proof. Our message-hiding proof route here is quite similar to the security proof route of the conventional CP-ABE scheme by Lewko and Waters [16, Section 5]. But as discussed previously, this is not a trivial work.

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 $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}$, $\boldsymbol{K}_{i,j,0}$, $\{\boldsymbol{K}_{i,j,x}\}_{x\in S}$ with index (i,j). One then chooses random value γ. The semi-functional key is

$$\boldsymbol{K}_{i,j} h^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}'_{i,j} g^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}''_{i,j} g^{z_i \gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}_{i,j,0},\ \{\boldsymbol{K}_{i,j,x}\}_{x\in S}.$$

Semi-functional Ciphertexts. To produce a semi-functional ciphertext for an LSSS matrix (A,ρ) of size $l \times m$, one first calls the normal encryption algorithm to produce a normal ciphertext consisting of $\langle (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n, (\boldsymbol{P}_k)_{k=0}^l \rangle$. One then chooses random values $\pi_3, \xi_{k,3} (1 \le k \le l) \in \mathbb{Z}_p$ and a random vector $\boldsymbol{u}_3 \in \mathbb{Z}_p^m$ with first entry equal to π_3. The semi-functional ciphertext is:

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

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

Game_{real}: The real message-hiding game (i.e. $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) as defined in the Section 2.1.

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 Game_t as follows: In Game_t, the ciphertext given to the attacker is semi-functional, as are the first t keys. The remaining keys are normal.

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 Game_{real} to Game_0, then to Game_1, next to Game_2, and so on. We ultimately arrive at Game_Q, where the ciphertext and all of the keys given to the attacker are semi-functional. We then transition to Game_{final}, which is defined to be like 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 Game_{real} to Game_0 and from Game_Q to Game_{final} are relatively easy, and can be accomplished directly via computational assumptions. The transitions from Game_{t-1} to 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 $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S}$ 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: $\boldsymbol{K}_{i,j} h^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}_3^*}$, $\boldsymbol{K}'_{i,j} g^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}_3^*}$, $\boldsymbol{K}''_{i,j} g^{z_i(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}_3^*}$, $\boldsymbol{K}_{i,j,0} g^{\delta_{i,j,3}\boldsymbol{b}_{0,3}^*}$, $\{\boldsymbol{K}_{i,j,x} g^{\sigma_{i,j,3}(\boldsymbol{b}_{x,5}^*+\boldsymbol{b}_{x,6}^*)}\}_{x \in S}$. 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 $\boldsymbol{K}_{i,j}$ and $\boldsymbol{K}''_{i,j}$ 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 $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}$, $\{\boldsymbol{K}_{i,j,x}\}_{x \in S}$ 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:

$$\boldsymbol{K}_{i,j} h^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}'_{i,j} g^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}''_{i,j} g^{z_i \gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}_{i,j,0} g^{\delta_{i,j,3}\boldsymbol{b}_{0,3}^*},\ \{\boldsymbol{K}_{i,j,x} g^{\sigma_{i,j,3}(\boldsymbol{b}_{x,5}^*+\boldsymbol{b}_{x,6}^*)}\}_{x \in S}.$$

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

Game_t^N: This is like 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.

Game_t^T: This is like 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 Game_{t-1} to Game_t in our hybrid argument, we will transition first from Game_{t-1} to Game_t^N, then to Game_t^T, and finally to Game_t. The transition from Game_t^N to 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 C.1.

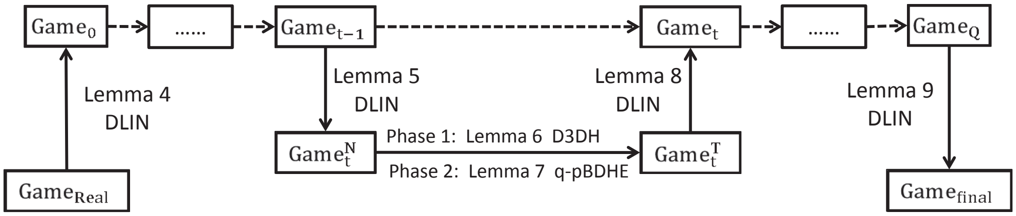

Fig. 1. Lemmas 4, 5, 8, and 9 rely on the subspace assumption (w.r.t. Definition 3), which is implied by DLIN assumption, Lemma 6 relies on the D3DH assumption, and Lemma 7 relies on the source group q-parallel BDHE assumption.

Theorem 3. *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 4. *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 4 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 $S_{(\bar{i},\bar{j})}$ (i.e., $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$).

The simulation for **Case I** is very similar to that of [8] because the simulator does not need to generate private key indexed $(\bar{i}, \bar{j})$ and there is not any restriction on the attribute set S^*. The **Case II** captures the security that even when a user has a key indexed $(\bar{i}, \bar{j})$ he cannot distinguish between an encryption to $(\mathbb{A}_{S^*}, (\bar{i}, \bar{j}))$ and one to $(\mathbb{A}_{S^*}, (\bar{i}, \bar{j}+1))$ if the corresponding attribute set $S_{(\bar{i},\bar{j})}$ is not a superset of S^*. With the crucial components $\boldsymbol{Z}_i^{t_i}$ (in $\boldsymbol{Q}'_i$) and $\boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}$ in the ciphertext, and $\boldsymbol{Y}_j$ in the public parameter, our particular construction guarantees that $\mathcal{B}$ can successfully finish the simulation with probability $|S^* \setminus S_{(\bar{i},\bar{j})}|/|\mathcal{U}|$, which is at least $1/|\mathcal{U}|$ since $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$. As of the fully secure CP-ABE schemes in [14,22,15,16,17], we assume that the size of attribute universe (i.e. $|\mathcal{U}|$) is polynomial in the security parameter λ, 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 C.2.

Lemma 2. *Suppose the D3DH assumption and the DLIN assumption hold. Then for any $1 \leq \bar{i} \leq n$ 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 C.3.

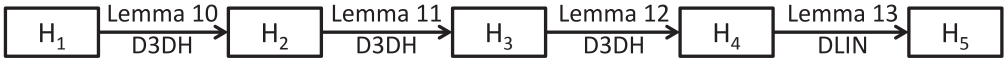

Fig. 2. Lemmas 10, 11, and 12 rely on the D3DH assumption, and Lemma 13 relies on the DLIN assumption.

4 Conclusion

In this paper, we proposed a new Augmented CP-ABE construction on prime order groups, and proved its message-hiding and index-hiding properties in the standard model. This implies the first CP-ABE that simultaneously achieves (1) fully collusion-resistant blackbox traceability in the standard model, (2) full security in the standard model, and (3) on prime order groups. The scheme is highly expressive in supporting any monotonic access structures, and as a fully collusion-resistant blackbox traceable CP-ABE scheme, it achieves the most efficient level to date, with the overhead in $O(\sqrt{N})$ only.

References

1. A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
2. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
3. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT*, pages 573–592, 2006.
4. D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *ACM Conference on Computer and Communications Security*, pages 211–220, 2006.
5. L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. In *ACM Conference on Computer and Communications Security*, pages 456–465, 2007.
6. 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.

7. D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, pages 44–61, 2010.
8. 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.
9. V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.
10. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.
11. J. Herranz, F. Laguillaumie, and C. Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In *Public Key Cryptography*, pages 19–34, 2010.
12. J. Katz and D. Schröder. Tracing insider attacks in the context of predicate encryption schemes. In *ACITA*, 2011. `https://www.usukita.org/node/1779`.
13. A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. Z. Liu and D. S. Wong. Practical attribute based encryption: Traitor tracing, revocation, and large universe. *IACR Cryptology ePrint Archive*, 2014:616, 2014.
20. T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.
21. T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.
22. T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.
23. 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.
24. 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 λ 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 λ for any PPT algorithm $\mathcal{A}$.

The Source Group q-Parallel BDHE Assumption [16] Given a group generator $\mathcal{G}$ and a positive integer q, define the following distribution:

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

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 λ 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. *[13] 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 [13], and is generalized by Lewko and Waters [16]. 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 [16, 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 [16] 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 3. *(The Subspace Assumption [16]) 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\} : \\
&\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 λ.

B Correctness of Our AugCP-ABE Construction

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

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

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 $\boldsymbol{v}_i \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$), then we have

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

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

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

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'.

C Proofs

C.1 Proof of Theorem 2

Lemma 4. *Under the subspace assumption, no PPT attacker can achieve a non-negligible difference in advantage between* Game_{real} *and* Game_0.

Proof. Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between Game_{real} and 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 $(\mathbb{D}, \mathbb{D}^*), (\mathbb{D}_0, \mathbb{D}_0^*) \in \mathit{Dual}(Z_p^3, \psi)$ and $(\mathbb{D}_1, \mathbb{D}_1^*), \ldots, (\mathbb{D}_{\mathcal{U}}, \mathbb{D}_{\mathcal{U}}^*) \in \mathit{Dual}(Z_p^6, \psi)$. $\mathcal{B}$ is given (we will ignore the $\boldsymbol{U}$ terms and μ_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 τ_3 contributions are present or not.

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

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

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

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 ψ value).

$\mathcal{B}$ chooses random exponents

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

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

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

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 $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}_{i,j}', \boldsymbol{K}_{i,j}'', \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\rangle$ as:

$$\begin{aligned}\boldsymbol{K}_{i,j} =& g^{(\alpha_{i,1}+r_ic_{j,1})\boldsymbol{b}_1^*+(\alpha_{i,2}+r_ic_{j,2})\boldsymbol{b}_2^*} h^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^*+(\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*}\\ =& (g^{\boldsymbol{d}_1})^{\alpha_{i,1}'+r_ic_{j,1}'+\theta(\sigma_{i,j,1}'+\delta_{i,j,1}')}(g^{\boldsymbol{d}_2})^{\alpha_{i,2}'+r_ic_{j,2}'+\theta(\sigma_{i,j,2}'+\delta_{i,j,2}')},\\ \boldsymbol{K}_{i,j}' =& (g^{\boldsymbol{d}_1})^{\alpha_1'+\sigma_{i,j,1}'+\delta_{i,j,1}'}(g^{\boldsymbol{d}_2})^{\alpha_2'+\sigma_{i,j,2}'+\delta_{i,j,2}'}, \ \boldsymbol{K}_{i,j}'' = (\boldsymbol{K}_{i,j}')^{z_i},\\ \boldsymbol{K}_{i,j,0} =& (g^{\boldsymbol{d}_{0,1}})^{\delta_{i,j,1}'}(g^{\boldsymbol{d}_{0,2}})^{\delta_{i,j,2}'},\\ \boldsymbol{K}_{i,j,x} =& (g^{\boldsymbol{d}_{x,1}})^{\sigma_{i,j,1}'}(g^{\boldsymbol{d}_{x,2}})^{\sigma_{i,j,1}'}(g^{\boldsymbol{d}_{x,3}})^{\sigma_{i,j,2}'}(g^{\boldsymbol{d}_{x,4}})^{\sigma_{i,j,2}'} \ \forall x \in S_{(i,j)}.\end{aligned}$$

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 $\mathcal{B}$ an LSSS matrix (A, ρ) 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 $\langle (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l\rangle$ as follows (note that $\bar{i} = 1, \bar{j} = 1$):

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

$$\begin{aligned} &\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i\boldsymbol{v}_i},\quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa},\\ &\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad \boldsymbol{Q}'_i = h^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}\boldsymbol{Z}_i^{t_i}\boldsymbol{T}_1^{\theta},\quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ &T_i = M_b\frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{(F_1F_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}e_3(g^{\boldsymbol{d}_1},\boldsymbol{T}_1)^{\theta\alpha'_1}e_3(g^{\boldsymbol{d}_2},\boldsymbol{T}_1)^{\theta\alpha'_2}}. \end{aligned}$$

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 τ_3 terms, then $\boldsymbol{Q}'_i$ and $\boldsymbol{P}_0$ are in their normal forms, and the exponent vector of $\boldsymbol{P}_k$ is

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

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

If the exponents of the T terms *do* include the τ_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 τ_3 terms are absent, $\mathcal{B}$ properly simulates Game_{real}, and when the τ_3 terms are present, $\mathcal{B}$ properly simulates 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* Game_{t-1} *and* 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 Game_{t-1} and 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 μ_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 τ_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

$$\theta,\ \alpha'_1, \alpha'_2 \in \mathbb{Z}_p,\ \ \{r_i,\ z_i,\ \alpha'_{i,1}, \alpha'_{i,2}\ \in \mathbb{Z}_p\}_{i\in[n]},\ \ \{c'_{j,1}, c'_{j,2},\ y_j\ \in \mathbb{Z}_p\}_{j\in[n]}.$$

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

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

Note that $\mathcal{B}$ implicitly sets

$$\alpha_1 = \eta\alpha'_1,\ \alpha_2 = \beta\alpha'_2,\ \ \{\alpha_{i,1} = \eta\alpha'_{i,1},\ \alpha_{i,2} = \beta\alpha'_{i,2}\}_{i\in[n]},\ \ \{c_{j,1} = \eta c'_{j,1},\ c_{j,2} = \beta c'_{j,2}\}_{j\in[n]}.$$

Phase 1. To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ 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 $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S_{(i,j)}}\ \rangle$ as:

$$\begin{aligned}\boldsymbol{K}_{i,j} =&(g^{\eta \boldsymbol{b}_1^*})^{\alpha'_{i,1}+r_i c'_{j,1}+\theta(\sigma'_{i,j,1}+\delta'_{i,j,1})}(g^{\beta \boldsymbol{b}_2^*})^{\alpha'_{i,2}+r_i c'_{j,2}+\theta(\sigma'_{i,j,2}+\delta'_{i,j,2})} g^{\theta\gamma \boldsymbol{b}_3^*},\\ \boldsymbol{K}'_{i,j} =&(g^{\eta \boldsymbol{b}_1^*})^{\alpha'_1+\sigma'_{i,j,1}+\delta'_{i,j,1}}(g^{\beta \boldsymbol{b}_2^*})^{\alpha'_2+\sigma'_{i,j,2}+\delta'_{i,j,2}} g^{\gamma \boldsymbol{b}_3^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},\\ \boldsymbol{K}_{i,j,0} =&(g^{\eta \boldsymbol{b}_{0,1}^*})^{\delta'_{i,j,1}}(g^{\beta \boldsymbol{b}_{0,2}^*})^{\delta'_{i,j,2}},\\ \boldsymbol{K}_{i,j,x} =&(g^{\eta \boldsymbol{b}_{x,1}^*})^{\sigma'_{i,j,1}}(g^{\eta \boldsymbol{b}_{x,2}^*})^{\sigma'_{i,j,1}}(g^{\beta \boldsymbol{b}_{x,3}^*})^{\sigma'_{i,j,2}}(g^{\beta \boldsymbol{b}_{x,4}^*})^{\sigma'_{i,j,2}}\ \forall x \in S_{(i,j)}.\end{aligned}$$

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

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

- 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 $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S_{(i,j)}})$ as:

$$\begin{aligned}\boldsymbol{K}_{i,j} =&(g^{\eta \boldsymbol{b}_1^*})^{\alpha'_{i,1}+r_i c'_{j,1}+\theta\delta'_{i,j,1}}(g^{\beta \boldsymbol{b}_2^*})^{\alpha'_{i,2}+r_i c'_{j,2}+\theta\delta'_{i,j,2}} \boldsymbol{T}_1^{\theta} \boldsymbol{T}_1^{\theta\delta'_{i,j,3}},\\ \boldsymbol{K}'_{i,j} =&(g^{\eta \boldsymbol{b}_1^*})^{\alpha'_1+\delta'_{i,j,1}}(g^{\beta \boldsymbol{b}_2^*})^{\alpha'_2+\delta'_{i,j,2}} \boldsymbol{T}_1(\boldsymbol{T}_1)^{\delta'_{i,j,3}},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},\\ \boldsymbol{K}_{i,j,0} =&(g^{\eta \boldsymbol{b}_{0,1}^*})^{\delta'_{i,j,1}}(g^{\beta \boldsymbol{b}_{0,2}^*})^{\delta'_{i,j,2}} \boldsymbol{T}_{0,1}^{\delta'_{i,j,3}},\\ \boldsymbol{K}_{i,j,x} =&\boldsymbol{T}_{x,1}\boldsymbol{T}_{x,2}\ \forall x \in S_{(i,j)}.\end{aligned}$$

  Note that $\mathcal{B}$ implicitly sets

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

  If the exponents of the T terms *do not* include the τ_3 terms, then this is a properly distributed normal key. If they *do* include the τ_3 terms, then this is a properly distributed nominal semi-functional key with $\sigma_{i,j,3} = \tau_3$ and $\delta_{i,j,3} = \delta'_{ij,3}\tau_3$. (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 $\{t+1, \ldots, Q\}$ 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 $\mathsf{SK}_{(i,j),S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S_{(i,j)}}\ \rangle$ as:

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

  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 $\mathcal{B}$ an LSSS matrix (A, ρ) 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

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

where the first entries of $\boldsymbol{u}'_1$ and $\boldsymbol{u}'_2$ are equal to 0. It also chooses a random vector $\boldsymbol{u} \in \mathbb{Z}_p^m$ 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

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

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

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

$$\begin{aligned} &\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i\boldsymbol{v}_i},\quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa},\\ &\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad \boldsymbol{Q}'_i = h^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}\boldsymbol{Z}_i^{t_i}U_1^{\theta},\quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ &T_i = M_b\frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{(F_1F_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}e_3(U_1, g^{\eta\boldsymbol{b}_1^*})^{\theta\alpha'_1}e_3(U_1, g^{\eta\boldsymbol{b}_2^*})^{\theta\alpha'_2}}. \end{aligned}$$

2. For each $j \in [n]$: it sets $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau\boldsymbol{v}_c}(\boldsymbol{Y}_j)^{\kappa\boldsymbol{w}_j},\quad \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}.$
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 τ_3 terms are absent, $\mathcal{B}$ properly simulates Game_{t-1}, and when the τ_3 terms are present, $\mathcal{B}$ properly simulates 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* Game_t^N *and* 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 Game_t^N and 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 Game_t^N or 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 ψ. It then implicitly sets $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{B}_0, \mathbb{B}_0^*)$ as follows:

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

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:

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

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 $\theta, \alpha_1, \alpha_2, r_i, \alpha_{i,1}, \alpha_{i,2}, z_i (i \in [n]), c_{j,1}, c_{j,2}, y_j (j \in [n]) \in \mathbb{Z}_p$ randomly. We observe that $\mathcal{B}$ can now produce the public parameter (with $h = g^\theta$), and also knows 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 $\boldsymbol{K}''_{i,j}$ by $g^{\theta\gamma'\boldsymbol{d}^*_3}, g^{\gamma'\boldsymbol{d}^*_3}$ and $g^{z_i\gamma'\boldsymbol{d}^*_3}$ 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 [16, 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

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

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 to $\mathcal{B}$ an LSSS matrix (A, ρ) 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, ρ), $\mathcal{B}$ 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, ρ), 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 π_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 π_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:

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

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 θ, $\mathcal{B}$ can produce the semi-functional components using the value of θ 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 Game_t^N, and if T is a random group element, then $\mathcal{B}$ has properly simulated 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* Game_t^N *and* Game_t^T *for a* $t > Q_1$ *using an access matrix* (A, ρ) *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 Game_t^N and 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 Game_t^N or 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 ψ. 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 $\theta, \alpha_1, \alpha_2, r_i, z_i, \alpha_{i,1}, \alpha_{i,2} (i \in [n]), c_{j,1}, c_{j,2}, y_j (j \in [n]) \in \mathbb{Z}_p$ randomly. We observe that $\mathcal{B}$ can now produce the public parameter (with $h = g^\theta$), and also knows 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 $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}'_{i,j}$ and $\boldsymbol{K}''_{i,j}$ by $g^{\theta\gamma'\boldsymbol{d}_3^*}, g^{\gamma'\boldsymbol{d}_3^*}$ and $g^{z_i\gamma'\boldsymbol{d}_3^*}$ 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 access matrix (A, ρ) 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 η_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 $u'_2, \ldots, u'_m \in \mathbb{Z}_p$ and random values $\xi'_{k,3} \in \mathbb{Z}_p$ for each $k \in [l]$. It implicitly sets $\boldsymbol{u}_3 = (cdf, dfc^2 + u'_2, \ldots, dfc^m + u'_m)$. [4] This is distributed as a random vector with first entry equal to π_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:

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

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

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

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 θ.
$\mathcal{B}$ 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 $\mathcal{A}$.

[4] 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.

Phase 2. To create the $Q_1^{th}, \ldots, (t-1)^{th}$ 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 $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}'_{i,j}$ and $\boldsymbol{K}''_{i,j}$ by $g^{\theta\gamma'\boldsymbol{d}_3^*}, g^{\gamma'\boldsymbol{d}_3^*}$ and $g^{z_i\gamma'\boldsymbol{d}_\mathbf{3}^*}$ 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, ρ), $\mathcal{B}$ can (efficiently) compute a vector $\boldsymbol{w} = (w_1, \ldots, w_m) \in \mathbb{Z}_p^m$ 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 $\sigma_{i,j,3} = w_1c^q + \cdots + w_mc^{q-m+1}$, 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 $\delta_{i,j,3} = -w_2c^{q-1} - \cdots - w_mc^{q-m+1} + fc^{-1}\delta'_{i,j,3}$. 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

$$(\sigma_{i,j,3} + \delta_{i,j,3})\boldsymbol{b}_3^* = (w_1dc^{q+1} + df\delta'_{i,j,3})\boldsymbol{d}_3^*.$$

$\mathcal{B}$ 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 $\boldsymbol{K}_{i,j}$ and $\boldsymbol{K}'_{i,j}$ as $(T^{w_1\boldsymbol{d}_3^*}(g^{df})^{\delta'_{i,j,3}\boldsymbol{d}_3^*})^{\theta}$ and $(T^{w_1\boldsymbol{d}_3^*}(g^{df})^{\delta'_{i,j,3}\boldsymbol{d}_3^*})^{z_{i_t}}$ 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

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

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 Game_t^N, and if T is distributed randomly, then $\mathcal{B}$ has properly simulated 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* Game_t^T *and* 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 $g^{\theta\gamma \boldsymbol{b}_3^*}$, $g^{\gamma \boldsymbol{b}_3^*}$ and $g^{z_i \gamma \boldsymbol{b}_3^*}$ to $\boldsymbol{K}_{i,j}$, $\boldsymbol{K}'_{i,j}$ and $\boldsymbol{K}''_{i,j}$ respectively for the t^{th} key (where it chooses $\gamma \in \mathbb{Z}_p$ randomly). This ensures that when the τ_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* Game_Q *and* Game_{final}.

Proof. Given a PPT attacker $\mathcal{A}$ achieving a non-negligible difference in advantage between Game_Q and 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 μ_3 and $\boldsymbol{T}_{0,1}, \{\boldsymbol{T}_{x,1}, \boldsymbol{T}_{x,2}\}_{x \in [\mathcal{U}]}$ because they do not be needed):

$$\mathbb{G}, p, g,\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2},\ g^{\boldsymbol{b}_{0,1}}, g^{\boldsymbol{b}_{0,2}},\ \{g^{\boldsymbol{b}_{x,1}}, g^{\boldsymbol{b}_{x,2}}, g^{\boldsymbol{b}_{x,3}}, g^{\boldsymbol{b}_{x,4}}\}_{x \in [\mathcal{U}]},$$

$$g^{\eta \boldsymbol{b}_1^*}, g^{\beta \boldsymbol{b}_2^*}, g^{\boldsymbol{b}_3^*},\ g^{\eta \boldsymbol{b}_{0,1}^*}, g^{\beta \boldsymbol{b}_{0,2}^*}, g^{\boldsymbol{b}_{0,3}^*},\ \{g^{\eta \boldsymbol{b}_{x,1}^*}, g^{\eta \boldsymbol{b}_{x,2}^*}, g^{\beta \boldsymbol{b}_{x,3}^*}, g^{\beta \boldsymbol{b}_{x,4}^*}, g^{\boldsymbol{b}_{x,5}^*}, g^{\boldsymbol{b}_{x,6}^*}\}_{x \in [\mathcal{U}]},$$

$$\boldsymbol{U}_1 = g^{\mu_1 \boldsymbol{b}_1 + \mu_2 \boldsymbol{b}_2 + \mu_3 \boldsymbol{b}_3},\ \boldsymbol{U}_{0,1} = g^{\mu_1 \boldsymbol{b}_{0,1} + \mu_2 \boldsymbol{b}_{0,2} + \mu_3 \boldsymbol{b}_{0,3}},$$

$$\{\boldsymbol{U}_{x,1} = g^{\mu_1 \boldsymbol{b}_{x,1} + \mu_2 \boldsymbol{b}_{x,3} + \mu_3 \boldsymbol{b}_{x,5}},\ \boldsymbol{U}_{x,2} = g^{\mu_1 \boldsymbol{b}_{x,2} + \mu_2 \boldsymbol{b}_{x,4} + \mu_3 \boldsymbol{b}_{x,6}}\}_{x \in [\mathcal{U}]},$$

$$\boldsymbol{T}_1.$$

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 τ_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

$$\theta, \alpha'_1, \alpha'_2\ \in \mathbb{Z}_p,\ \ \{r_i,\ z_i,\ \alpha'_{i,1}, \alpha'_{i,2}\ \in \mathbb{Z}_p\}_{i \in [n]},\ \ \{c'_{j,1}, c'_{j,2}, y_j\ \in \mathbb{Z}_p\}_{j \in [n]}.$$

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

$$\Big(g, h = g^{\theta},\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1} = (g^{\boldsymbol{b}_1})^{\theta}, h^{\boldsymbol{b}_2} = (g^{\boldsymbol{b}_2})^{\theta}, h^{\boldsymbol{b}_{0,1}} = (g^{\boldsymbol{b}_{0,1}})^{\theta}, h^{\boldsymbol{b}_{0,2}} = (g^{\boldsymbol{b}_{0,2}})^{\theta},$$

$$\{h^{\boldsymbol{b}_{x,1}} = (g^{\boldsymbol{b}_{x,1}})^{\theta}, \ldots, h^{\boldsymbol{b}_{x,4}} = (g^{\boldsymbol{b}_{x,4}})^{\theta}\}_{x \in [\mathcal{U}]},\ \ F_1 = e_3(g^{\boldsymbol{b}_1}, T_1)^{\theta},\ F_2 = e_3(g^{\boldsymbol{b}_2}, T_1)^{\theta},$$

$$\{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1 + \boldsymbol{b}_2)},\ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1 + \boldsymbol{b}_2)},$$

$$E_{i,1} = e_3(g^{\boldsymbol{b}_1}, T_1^{\theta}) e_3(g^{\boldsymbol{b}_1}, g^{\eta \boldsymbol{b}_1^*})^{\alpha'_{i,1}}, E_{i,2} = e_3(g^{\boldsymbol{b}_2}, T_1^{\theta}) e_3(g^{\boldsymbol{b}_2}, g^{\beta \boldsymbol{b}_2^*})^{\alpha'_{i,2}}\}_{i \in [n]},$$

$$\{\boldsymbol{H}_j = (g^{\eta \boldsymbol{b}_1^*})^{c'_{j,1}} (g^{\beta \boldsymbol{b}_2^*})^{c'_{j,2}},\ \boldsymbol{Y}_j = (\boldsymbol{H}_j)^{y_j}\}_{j \in [n]} \Big).$$

Note that $\mathcal{B}$ implicitly sets

$$\alpha_1 = \eta \tau_1, \alpha_2 = \beta \tau_2,$$

$$\{\alpha_{i,1} = \eta(\theta \tau_1 + \alpha'_{i,1}),\ \alpha_{i,2} = \beta(\theta \tau_2 + \alpha'_{i,2})\}_{i \in [n]},\ \ \{c_{j,1} = \eta c'_{j,1},\ c_{j,2} = \beta c'_{j,2}\}_{j \in [n]}.$$

Phase 1. To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ 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 $\mathsf{SK}_{(i,j), S_{(i,j)}} = \langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x \in S_{(i,j)}}\ \rangle$ as:

$$\boldsymbol{K}_{i,j} = \boldsymbol{T}_1^{\theta} (g^{\eta \boldsymbol{b}_1^*})^{\alpha'_{i,1} + r_i c'_{j,1} + \theta(\sigma'_{i,j,1} + \delta'_{i,j,1})} (g^{\beta \boldsymbol{b}_2^*})^{\alpha'_{i,2} + r_i c'_{j,2} + \theta(\sigma'_{i,j,2} + \delta'_{i,j,2})} g^{\theta \gamma' \boldsymbol{b}_3^*},$$

$$\boldsymbol{K}'_{i,j} = \boldsymbol{T}_1 (g^{\eta \boldsymbol{b}_1^*})^{\sigma'_{i,j,1} + \delta'_{i,j,1}} (g^{\beta \boldsymbol{b}_2^*})^{\sigma'_{i,j,2} + \delta'_{i,j,2}} g^{\gamma' \boldsymbol{b}_3^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},$$

$$\boldsymbol{K}_{i,j,0} = (g^{\eta \boldsymbol{b}_{0,1}^*})^{\delta'_{i,j,1}} (g^{\beta \boldsymbol{b}_{0,2}^*})^{\delta'_{i,j,2}},$$

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

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

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

We note that the multiple of $\boldsymbol{b}_3^*$ appearing in the exponent of $\boldsymbol{K}'_{i,j}(\boldsymbol{K}_{i,j}, \boldsymbol{K}''_{i,j}, resp.)$ is either equal to γ' (γ', $z_i\gamma'$, resp.) or $\gamma' + \tau_3$ ($\gamma' + \tau_3$, $z_i(\gamma' + \tau_3)$, resp.), depending on the nature of $\boldsymbol{T}_1$. Either way, this is a properly distributed semi-functional key (whose distribution is independent of τ_3 even if it is present).

Challenge. $\mathcal{A}$ submits to $\mathcal{B}$ an LSSS matrix (A, ρ) 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

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

where the first entries of $\boldsymbol{u}'_1$ and $\boldsymbol{u}'_2$ are equal to 0. It also chooses a random vector $\boldsymbol{u} \in \mathbb{Z}_p^m$ 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

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

$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_yr_z, -r_xr_z, r_xr_y)$, 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 $\langle(A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l\rangle$ as follows (note that $\bar{i} = 1, \bar{j} = 1$):

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

$$\begin{aligned} &\boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i\boldsymbol{v}_i},\quad \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa}, \\ &\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)},\quad \boldsymbol{Q}'_i = h^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}\boldsymbol{Z}_i^{t_i}\boldsymbol{U}_1^{\theta},\quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \\ &T_i = M_b\frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{(F_1F_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}e_3(\boldsymbol{U}_1, \boldsymbol{T}_1)^{\theta}}. \end{aligned}$$

2. For each $j \in [n]$: 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. 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 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 τ_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 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.

C.2 Proof of Lemma 1

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ that breaks the Index Hiding Game with advantage ϵ. We build a simulator $\mathcal{B}$ to solve a D3DH problem instance as follows.

$\mathcal{B}$ 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**.

$\mathcal{B}$ 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 ψ. It then implicitly sets $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{B}_0, \mathbb{B}_0^*)$ and $\{(\mathbb{B}_x, \mathbb{B}_x^*)\}$ as follows:

$$\begin{array}{llllll}\boldsymbol{b}_1 = \boldsymbol{d}_1, & \boldsymbol{b}_2 = \boldsymbol{d}_2, & \boldsymbol{b}_3 = \boldsymbol{d}_3, & \boldsymbol{b}_1^* = \boldsymbol{d}_1^*, & \boldsymbol{b}_2^* = \boldsymbol{d}_2^*, & \boldsymbol{b}_3^* = \boldsymbol{d}_3^*, \\ \multicolumn{6}{l}{\boldsymbol{b}_{0,1} = (c)^{-1}\boldsymbol{d}_{0,1}, \boldsymbol{b}_{0,2} = (c)^{-1}\boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3} = \boldsymbol{d}_{0,3}, \boldsymbol{b}_{0,1}^* = (c)\boldsymbol{d}_{0,1}^*, \boldsymbol{b}_{0,2}^* = (c)\boldsymbol{d}_{0,2}^*, \boldsymbol{b}_{0,3}^* = \boldsymbol{d}_{0,3}^*.}\end{array}$$

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

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

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

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

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

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

$$\begin{array}{l}br_{\bar{i}}' \equiv r_{\bar{i}} \bmod p, \quad cc_{\bar{j},1}' \equiv c_{\bar{j},1} \bmod p, \ cc_{\bar{j},2}' \equiv c_{\bar{j},2} \bmod p, \ y_{\bar{j}}'/c = y_{\bar{j}} \bmod p, \\ cz_i' \equiv z_i \bmod p \ \forall i \in [n] \setminus \{\bar{i}\}.\end{array}$$

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 $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ where

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

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

$$\boldsymbol{K}''_{i,j} = \begin{cases} (C^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*})^{z'_i}, & : i\neq\bar{i}, j\neq\bar{j} \\ (\boldsymbol{K}'_{i,j})^{z_{\bar{i}}}, \quad : i=\bar{i}, j\neq\bar{j} \\ (C^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{d}_1^*+(\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{d}_2^*})^{z'_i}, & : i\neq\bar{i}, j=\bar{j} \end{cases}$$

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

$$\boldsymbol{K}_{i,j,x} = \begin{cases} g^{\sigma_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}. & : x\neq\bar{x} \\ C^{\sigma_{i,j,1}(\boldsymbol{d}_{x,1}^*+\boldsymbol{d}_{x,2}^*)+\sigma_{i,j,2}(\boldsymbol{d}_{x,3}^*+\boldsymbol{d}_{x,4}^*)}. & : x=\bar{x} \end{cases}$$

- if $(i,j) = (\bar{i},\bar{j})$: it means that $\mathcal{A}$ behaves in **Case II**. 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 $\sigma'_{i,j,1} - br'_i c'_{j,1}/\theta' \equiv \sigma_{i,j,1} \bmod p$, $\sigma'_{i,j,2} - br'_i c'_{j,2}/\theta' \equiv \sigma_{i,j,2} \bmod p$. In addition $\mathcal{B}$ randomly chooses $\delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$. $\mathcal{B}$ creates a private key $\langle\ (i,j), S_{(i,j)}, \boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j},\ \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S_{(i,j)}}\ \rangle$ where

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

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

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

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

Challenge. $\mathcal{A}$ submits a 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,ρ) for $\mathbb{A}_{S^*}$. Let $l\times m$ be the size of A.

Note that $S^*\setminus\{\bar{x}\}$ does not satisfy (A,ρ), $\mathcal{B}$ 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,ρ), 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 $\boldsymbol{\chi}_1=(r_x,0,r_z), \boldsymbol{\chi}_2=(0,r_y,r_z), \boldsymbol{\chi}_3=\boldsymbol{\chi}_1\times\boldsymbol{\chi}_2=(-r_yr_z,-r_xr_z,r_xr_y)$, 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 $\langle (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l\rangle$ as follows:

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

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

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

$$\begin{aligned}
&\boldsymbol{R}_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}},\\
&\boldsymbol{Q}_i = g^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)} A^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \quad \boldsymbol{Q}'_i = h^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{d}_1+\boldsymbol{d}_2)} \boldsymbol{Z}_i^{t_{\bar{i}}} h^{\pi'_1\boldsymbol{d}_1+\pi'_2\boldsymbol{d}_2},\\
&\boldsymbol{Q}''_i = g^{t_{\bar{i}}(\boldsymbol{d}_1+\boldsymbol{d}_2)}, \quad T_i = M\frac{e_3(\boldsymbol{Q}_i, g^{\alpha_{i,1}\boldsymbol{d}_1^*+\alpha_{i,2}\boldsymbol{d}_2^*})}{(F_1F_2)^{\tau' s'_{\bar{i}}(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)} F_1^{\pi'_1} F_2^{\pi'_2}}.
\end{aligned}$$

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

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

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 μ_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 $j = \bar{j}$:

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

   - 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.

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

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

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

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 $S^* \ne \emptyset$ and when $\mathcal{A}$ behaves in **Case II** the attribute set $S_{(\bar{i},\bar{j})}$ must satisfy $S^* \setminus S_{(\bar{i},\bar{j})} \ne \emptyset$, 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 $\epsilon/|\mathcal{U}|$. As of the fully secure CP-ABE schemes in [14,22,15,16,17], the size of attribute universe (i.e. $|\mathcal{U}|$) in our scheme is also polynomial in the security parameter λ. Thus a degradation of $1/|\mathcal{U}|$ in the security reduction is acceptable.

C.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 ϵ. 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 $\mathsf{Game_{IH}}$ 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 ψ.

$\mathcal{B}$ also randomly chooses

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

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

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

Note that $\mathcal{B}$ implicitly sets

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

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

$$\boldsymbol{K}_{i,j} = \begin{cases} g^{(\alpha_{i,1} + r_i c'_{j,1})\boldsymbol{b}_1^* + (\alpha_{i,2} + r_i c'_{j,2})\boldsymbol{b}_2^*} A^{-r_i(\boldsymbol{b}_1^* + \boldsymbol{b}_2^*)} h^{(\sigma_{i,j,1} + \delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2} + \delta_{i,j,2})\boldsymbol{b}_2^*}, & : i \neq \bar{i} \\ g^{\alpha'_{\bar{i},1}\boldsymbol{b}_1^* + \alpha'_{\bar{i},2}\boldsymbol{b}_2^*} B^{(c'_{j,1}\boldsymbol{b}_1^* + c'_{j,2}\boldsymbol{b}_2^*)} h^{(\sigma_{i,j,1} + \delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2} + \delta_{i,j,2})\boldsymbol{b}_2^*}, & : i = \bar{i} \end{cases}$$
$$\boldsymbol{K}'_{i,j} = g^{(\alpha_1 + \sigma_{i,j,1} + \delta_{i,j,1})\boldsymbol{b}_1^* + (\alpha_2 + \sigma_{i,j,2} + \delta_{i,j,2})\boldsymbol{b}_2^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},$$
$$\boldsymbol{K}_{i,j,0} = g^{\delta_{i,j,1}\boldsymbol{b}_{0,1}^* + \delta_{i,j,2}\boldsymbol{b}_{0,2}^*},$$
$$\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{b}_{x,1}^* + \boldsymbol{b}_{x,2}^*) + \sigma_{i,j,2}(\boldsymbol{b}_{x,3}^* + \boldsymbol{b}_{x,4}^*)}\ \forall x \in S_{(i,j)}.$$

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

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

where the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are equal to π_1 and π_2 respectively.
$\mathcal{B}$ chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$, then it chooses random

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

$\mathcal{B}$ 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

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

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

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

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

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

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

$$\begin{aligned}&\boldsymbol{R}_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{s_{\bar{i}}\boldsymbol{v}_{\bar{i}}}, \quad \boldsymbol{R}'_i = (B^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\kappa s_{\bar{i}}\boldsymbol{v}_{\bar{i}}},\\ &\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{b}_1+\boldsymbol{b}_2)}C^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{\theta}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ &T_i = M\frac{e(A,B)^{2\psi\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}e(g,T)^{2\psi\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)}e(g,g)^{\psi(\alpha'_{i,1}+\alpha'_{i,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}e(g,C)^{\psi(\alpha'_{i,1}+\alpha'_{i,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)}}{(F_1F_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}e(C,h)^{\psi(\alpha_1+\alpha_2)\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^q)}F_1^{\pi_1}F_2^{\pi_2}}.\end{aligned}$$

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

$$\begin{aligned}&\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{r_i s_i\boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (g^{\boldsymbol{d}_1+\boldsymbol{d}_2})^{\kappa r_i s_i\boldsymbol{v}_i},\\ &\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{\theta}\boldsymbol{Z}_i^{t_i}h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ &T_i = M\frac{e(g,g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}}{(F_1F_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c^p)}F_1^{\pi_1}F_2^{\pi_2}}.\end{aligned}$$

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 μ_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 ϵ. 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 ψ.

$\mathcal{B}$ also randomly chooses

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

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

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

Note that $\mathcal{B}$ implicitly sets

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

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

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

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

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

where the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are equal to π_1 and π_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

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

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

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

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

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

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

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 $\mu_{\bar{j}}$ 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 ϵ. 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 ψ.
$\mathcal{B}$ also randomly chooses

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

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

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

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

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

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

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

where the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are equal to π_1 and π_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 $\langle (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$ as follows:

1. For each $\underline{i} \in [n]$:
   - 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

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

   - 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

$$\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{s_i\boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\kappa s_i\boldsymbol{v}_i},$$
$$\boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \ \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{\theta} \boldsymbol{Z}_i^{t_i} h^{\pi_1\boldsymbol{d}_1+\pi_2\boldsymbol{d}_2},\ \ \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{d}_1+\boldsymbol{d}_2)},$$
$$T_i = M\frac{e(g^{(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}, g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau s_i}}{e(g^{(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}, h)^{\psi(\alpha_1+\alpha_2)\tau s_i} F_1^{\pi_1} F_2^{\pi_2}}.$$

- 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., $g^{\boldsymbol{v}_i} = ((g^a)^{\nu_{i,1}}, (g^b)^{\nu_{i,2}}, (g^c)^{\nu_{i,1}+\nu_{i,2}})$. 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

$$\boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{r_i s_i \boldsymbol{v}_i}, \quad \boldsymbol{R}'_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\kappa r_i s_i \boldsymbol{v}_i},$$
$$\boldsymbol{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad \boldsymbol{Q}'_i = \boldsymbol{Q}_i^{\theta} \boldsymbol{Z}_i^{t_i} h^{\pi_1 \boldsymbol{d}_1 + \pi_2 \boldsymbol{d}_2}, \quad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{d}_1+\boldsymbol{d}_2)},$$
$$T_i = M \frac{e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, g)^{\psi(\alpha_{i,1}+\alpha_{i,2})\tau s_i}}{e(g^{(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, h)^{\psi(\alpha_1+\alpha_2)\tau s_i} F_1^{\pi_1} F_2^{\pi_2}}.$$

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

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 $\mathcal{A}$.

D Access Structure and Linear Secret-Sharing Schemes

Definition 4. (Access Structure) *[24] 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 [1], any monotonic access structure can be realized by a linear secret sharing scheme.

Definition 5. (Linear Secret-Sharing Schemes (LSSS)) *[24] A secret sharing scheme Π 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 Π. 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 $\rho(i)$ (ρ 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 Π. 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 [1], every LSSS as defined above enjoys the linear reconstruction property, which is defined as follows: Suppose that Π 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 Π, $\sum_{i \in I} \omega_i \lambda_i = s$. 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.