Multi-Client Functional Encryption with Fine-Grained Access Control

Ky Nguyen, Duong Hieu Phan, David Pointcheval

Multi-Client Functional Encryption with Fine-Grained Access Control

Ky Nguyen[1], Duong Hieu Phan[2], and David Pointcheval[1]

[1] DIENS, École normale supérieure, CNRS, Inria, PSL University, Paris, France
[2] LTCI, Telecom Paris, Institut Polytechnique de Paris, France

Abstract. Multi-Client Functional Encryption (MCFE) and Multi-Input Functional Encryption (MIFE) are very interesting extensions of Functional Encryption for practical purpose. They allow to compute joint function over data from multiple parties. Both primitives are aimed at applications in multi-user settings where decryption can be correctly output for users with appropriate functional decryption keys only. While the definitions for a single user or multiple users were quite general and can be realized for general classes of functions as expressive as Turing machines or all circuits, efficient schemes have been proposed so far for concrete classes of functions: either only for access control, *i.e.* the identity function under some conditions, or linear/quadratic functions under no condition.

In this paper, we target classes of functions that explicitly combine some evaluation functions independent of the decrypting user under the condition of some access control. More precisely, we introduce a framework for MCFE with fine-grained access control and propose constructions for both single-client and multi-client settings, for inner-product evaluation and access control via Linear Secret Sharing Schemes (LSSS), with selective and adaptive security. The only known work that combines functional encryption in multi-user setting with access control was proposed by Abdalla *et al.* (Asiacrypt '20), which relies on a generic transformation from the single-client schemes to obtain MIFE schemes that suffer a quadratic factor of n (where n denotes the number of clients) in the ciphertext size. We follow a different path, via MCFE: we present a *duplicate-and-compress* technique to transform the single-client scheme and obtain a MCFE with fine-grained access control scheme with only a linear factor of n in the ciphertext size. Our final scheme thus outperforms the Abdalla *et al.*'s scheme by a factor n, as one can obtain MIFE from MCFE by making all the labels in MCFE a fixed public constant. The concrete constructions are secure under the SXDH assumption, in the random oracle model for the MCFE scheme, but in the standard model for the MIFE improvement.

1 Introduction

Encryption enables people to securely communicate and share sensitive data in an *all-or-nothing* fashion: once the recipients have the secret key then they will recover the original data, otherwise the recipients have no information about the plaintext data. Functional Encryption (FE) [SW05, BSW11], introduced by Boneh, Sahai and Waters, overcomes this all-or-nothing limitation of PKE by allowing recipients to recover encrypted data in a more fine-grained manner: instead of revealing the whole original encrypted data, recipients can get the result of evaluation of some function on the data, according to the function associated to the decryption key, called *functional decryption key*. By allowing computation of partial data, one can aim at getting both: the utility of analysis on large data while preserving personal information private.

FE received large interest from the cryptographic community, first as a generalization of Identity-Based Encryption (IBE) [Sha84, Coc01, BF01, BGH07] and Attribute-Based Encryption (ABE) [SW05, GPSW06, OSW07, ALdP11, OT12b], which are unfortunately only access control, with all-or-nothing decryption as a result. Abdalla *et al.* [ABDP15] proposed the first construction for evaluating a concrete function: the inner product between a vector in the ciphertext and a vector in the functional decryption key, hence coined IPFE. The interest in FE then increased, especially in the multi-user setting in which the inputs come from different users, possibly in competition, and the output characterizes a joint function on the inputs [CDG+18a, LT19]. Applications are then numerous, and the encryptors can even be the final recipients of aggregated results. Then, this might look similar to multi-party computation (MPC), where several players

privately provide their inputs to allow computations on them. But the main difference is that functional encryption is expected as a non-interactive process, and thus quite more interesting in practice. While FE with a single encryptor might be of theoretical interest, in real-life, the number of really useful functions may be limited. When this number of functions is small, any PKE can be converted into FE by additionally encrypting the evaluations by the various functions under specific keys. This approach is impossible for multiple users, even when a unique fixed function is considered.

In the multi-user case, Goldwasser *et al.* [GGG+14, GKL+13] introduced the notion of Multi-Input Functional Encryption (MIFE) and Multi-Client Functional Encryption (MCFE) where the single input x to the encryption procedure is broken down into an input vector $(x_1, \ldots, x_n)$ with independent components. An index i for each client and, in the case of MCFE, a (typically time-based) tag tag are used for every encryption: $(c_1 = \mathsf{Enc}(1, x_1, \mathsf{tag}), \ldots, c_n = \mathsf{Enc}(n, x_n, \mathsf{tag}))$. Anyone owning a functional decryption key dk_f, for an n-ary function f and multiple ciphertexts (for the same tag tag, in the case of MCFE) can compute $f(x_1, \ldots, x_n)$ but nothing else about the individual x_i's. Implicitly, clients have to be able to coordinate together on the tags, and different usability in practice. In particular, in MCFE, the combination of ciphertexts generated for different tags does not give a valid global ciphertext and the adversary learns nothing from it. This leads to more versatility since encrypting x_i under tag has a different meaning from encrypting x_i under $\mathsf{tag}' \neq \mathsf{tag}$. On the other hand, MIFE does not use tags and once a ciphertext of x_i is computed, it can be reused for different combinations. However, in both situations, encryption must require a private key, otherwise anybody could complete the vector initiated by a user in many ways, and then obtain many various evaluations from a unique functional decryption key. But then, since encryption needs a private key per user, for each component c_i, some of these keys might get corrupted. Therefore, there are two main distinguishing aspects regarding MCFE that have to be dealt with: the role of tags in construction and the danger of corruption for security.

Another classical issue with encryption is the decryption key, even if legitimately obtained: once delivered, it can be used forever. One may expect revocation, or access control with more fine-grained authentication. This has been extensively studied with broadcast encryption, revocation systems and more generally, with attribute-based encryption (ABE) [Wee21]. Finally, as already explained, FE is a generalization of IBE and ABE, and after having been illustrated with IBE and ABE, linear evaluations [ALS16, ABDP16, BBL17, CLT18] and quadratic evaluations [BCFG17, Gay20, AS17, Lin17] have been proposed. However, there are still very few works that combine function evaluation and access control with concrete schemes. This could provide FE, with concrete function evaluation for some target users, or revocation (of users or functions). Abdalla *et al.* [ACGU20] have been the first to address this problem, for enhancing FE and MIFE with access control. In addition, they informally argue that from an ABE for MIFE one can lift it for free to get MCFE, thus solving both problems at the same time. Precisely, they mentioned "*by resorting for instance, to the notion of multi-client IPFE, where ciphertexts are associated with time-stamps, and only ciphertext with matching time-stamps can be combined (e.g. [CDG+18a]) we believe that our proposed primitive provides a more general and versatile solution to the problem*". Their idea can be interpreted as: tags can be used as specific attributes, and tags can be embedded in policies to automatically obtain multi-client settings. This argument seems formally valid when considering the general form of MIFE and MCFE. However, when considering concrete classes of functions, which is our main focus in this paper, it is unlikely to be efficiently feasible and we will explain the reason in the technical overview in Section 3. We underline that the principal difference between MCFE and MIFE is the presence of tags for producing the ciphertext components, which can be jointly decrypted only if all tags are equal. Thus, we can retrieve an MIFE from MCFE by fixing and publishing one tag, which retains the *same* ciphertext's size from the MCFE scheme to the new MIFE one. Moreover, since the combination of ciphertext components in MCFE is restrained by the tags, its security model is far less restrictive than the

security model of MIFE that has to deal with arbitrary combination of ciphertext components. For these reasons, our main objective becomes constructing an MCFE having smaller ciphertext size while permitting access control over decryption keys.

We take a completely different approach than in [ACGU20] to answer this question. Borrowing the terminology from ABE, our work will focus on *key-policy* (KP) constructions, where the policy is defined at the moment of key extraction and a ciphertext associated with certain attributes can be decrypted only if those attributes satisfy the policy. The dual notion of *ciphertext-policy* (CP) constructions is already studied in [ACGU20]. We concentrate solely on particular functionality classes whose description contains two separate parts: a description of functions exclusively for evaluation and a binary relation exclusively for modeling access control. Although this conceptual point of view does *not* take us out of the FE realm and thus can be captured by the general FE notion, it suits perfectly our purpose to compute inner-products along with fine-grained access control provided by Linear Secret Sharing Schemes (LSSS) in this paper. Then, we start from single-client IPFE schemes with LSSS access control and leverage them to get an MCFE scheme, where only tags are needed for hashing during encryption, and the hash function is modeled as a random oracle. Removing labels by fixing a public tag for all ciphertexts leads to an MIFE scheme in the standard model that is more efficient than the one from [ACGU20].

1.1 Related Work

Recently, [LLW21] improves upon the single-client construction based on Learning with Errors (LWE) from [ACGU20], for IPFE with access control expressed by bounded depth boolean circuits, achieving better security along with smaller ciphertexts. In another work, [PD21] also studies LWE-based single client constructions for IPFE with access control expressed by general boolean functions but under selective challenge attributes. The single-client LWE-based construction in [PD21] is later lifted to an MIFE using the generic transformation from [ACGU20].

Also in the single-client setting, another line of works attempts to construct FE for a general uniform functionality class such as Turing machines (TMFE), which naturally captures inner-product evaluation under LSSS access control. The work of Agrawal *et al.* [AMVY21] provided a non-adaptively simulation-based secure construction for TMFE in the *dynamic bounded collusion* model under sub-exponential LWE. The construction is later improved in [AKM+22] to achieve adaptive security under polynomial LWE, DDH or bilinear decisional Diffie-Hellman in specific groups, or quadratic residuosity. Towards this goal, both works of [AMVY21, AKM+22] additionally gave constructions of FE for circuits of *unbounded* size and depth, which can also encompass inner-product computation under LSSS access control, based on various standard assumptions such as computational Diffie-Hellman, factoring, or polynomial LWE. All single-client constructions from [AMVY21, AKM+22] use a wide range of cryptographic primitives in a generic manner, which deviates from our goal to give explicit constructions in the multi-user setting.

1.2 Our Contributions

Single-client setting. We propose new single-client schemes whose selectively-secure version is almost as efficient as the selectively-secure version in [ACGU20] and the adaptively-secure version is nearly three times as efficient as the adaptively-secure version in [ACGU20]. More importantly, our schemes can be extended to multi-client settings. Our constructions exploit the *Dual Pairing Vector Spaces* proposed by Okamoto-Takashima [OT10, OT12b].

Multi-client setting. Our main contribution is an extension from single-client to multi-client without linearly increasing the complexity in the number n of clients. The generic transformation proposed by Abdalla *et al.* [ACGU20, Theorem 6.3] results in a degradation of factor n in both construction and security reduction. As previously stated, Abdalla *et al.*'s generic transformation

Scheme	$\mathcal{P}$	$\mathcal{F}$	$\lvert\mathsf{ct}\rvert$	Security
[ACGU20, Sect. 3.1]	MSP; CP	$\mathcal{F}^{\mathsf{IP},\mathsf{poly}}_{n,q,\mathsf{MSP}}$	$n+2d+2$	sel-sim
[ACGU20, Sect. 3.2]	roMSP; CP	$\mathcal{F}^{\mathsf{IP},\mathsf{poly}}_{n,q,\mathsf{roMSP}}$	$3nd+3d+2$	ad-ind
Sect. 4, Fig. 1	LSSS; KP	$\mathcal{F}^{\mathsf{IP},\mathsf{poly}}_{n,q,\mathsf{LSSS}}$	$n+8d+4$	sel-ind
	LSSS; KP		$nd+2n+7d+3$	ad-ind
[ACGU20, Sect. 6.2] applied to [ACGU20, Sect. 3.1]	MSP; CP	$\mathcal{F}^{\mathsf{IP},\mathsf{poly}}_{n,q,\mathsf{MSP}}$	$n^2+2nd+2n$	mi-ad-ind
Sect. 5.2	LSSS; KP	$\mathcal{F}^{\mathsf{IP},\mathsf{poly}}_{n,q,\mathsf{LSSS}}$	$8nd+5n$	mc-ad-ind

Table 1: We compare our constructions with existing works, in terms of the number of group elements in the ciphertext (column $\lvert\mathsf{ct}\rvert$), the largest predicate class that can be handled (column $\mathcal{P}$), the function class (column $\mathcal{F}$), security (column **Security**). We denote by d the number of attributes needed by the policy in a ciphertext. All our schemes are defined for the functionality class $\mathcal{F}^{\mathsf{IP},\mathsf{poly}}_{n,q,\mathsf{LSSS}} = \mathcal{F}^{\mathsf{IP}} \times \mathsf{LSSS}$ constituted by $\mathcal{F}^{\mathsf{IP}} = \{F_{\mathbf{y}} : \mathbb{Z}_q^n \to \mathbb{Z}_q; \mathbf{x} \mapsto \langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{R}(\mathbb{Z}_q)\}$ and LSSS of Linear Secret Sharing Schemes over attributes in $\mathbb{Z}_q$, where $n, q \in \mathbb{N}$, q is prime and $|\mathcal{R}(\mathbb{Z}_q)| = \mathsf{poly}(\log q)$. The schemes from [ACGU20] are constructed for $\mathcal{F}^{\mathsf{IP}} \times \mathsf{MSP}$ and $\mathcal{F}^{\mathsf{IP}} \times \mathsf{roMSP}$, where $\mathsf{MSP}, \mathsf{roMSP}$ are classes of *monotone span programs, read-once monotone span programs* over attributes in $\mathbb{Z}_q$. The shorthands $(\mathsf{mc}, \mathsf{mi}, \mathsf{sel}, \mathsf{ad}, \mathsf{ind}, \mathsf{sim})$ denote multi-client setting, multi-input setting, selective security, adaptive security, indistinguishability-based, simulation-based.

can only help to achieve a multi-input scheme and is unlikely to be generalized to a multi-client scheme without further seriously degrading efficiency. On the other hand, because MIFE can be defined as MCFE with a fixed public constant tag, our construction yields a much more efficient MIFE with access control than the Abdalla *et al.*'s scheme (in fact, n times more efficient). More concretely, the total communication among n clients in our MCFE construction is a linear function in n and does not suffer a quadratic blow-up of n^2 group elements as in [ACGU20].

Comparisons. Our concrete constructions focus on the functionality class whose member's description contains inner-product evaluation functions and binary relations to describe access control. In the pairing-based setting, we give comparisons with existing works in Table 1. Recall that in MCFE, n can be a large number of clients, while d is the number of attributes, generally small, used in a policy. Concretely, we can consider identity-based functional encryption, as outlined in [ACGU20], where $d = 1$, whatever the size of n: our ciphertext's size is linear instead of quadratic in n as in [ACGU20].

Organization. We first give the necessary preliminaries in Section 2, then we present the high-level ideas and intuitions of our results in Section 3, before going into purely technical details in Section 4 for the single-client schemes and in Section 5 for the multi-client schemes.

2 Preliminaries

We write $[n]$ to denote the set $\{1, 2, \ldots, n\}$ for an integer n. For any $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers with addition and multiplication modulo q. For a prime q and an integer N, we denote by $GL_N(\mathbb{Z}_q)$ the general linear group of of degree N over $\mathbb{Z}_q$. We write vectors as row-vectors, unless stated otherwise. For a vector $\mathbf{x}$ of dimension n, the notation $\mathbf{x}[i]$ indicates the i-th coordinate of $\mathbf{x}$, for $i \in [n]$. We will follow the implicit notation in [EHK+13] and use $[\![a]\!]$ to denote g^a in a cyclic group $\mathbb{G}$ of prime order q generated by g, given $a \in \mathbb{Z}_q$. This implicit notation extends to matrices and vectors having entries in $\mathbb{Z}_q$. We use the shorthand ppt for "probabilistic polynomial time". In the security proofs, whenever we use an ordered sequence of

games $(\mathsf{G}_0, \mathsf{G}_1, \ldots, \mathsf{G}_i, \ldots, \mathsf{G}_L)$ indexed by $i \in \{0, 1, \ldots, L\}$, we refer to the predecessor of G_j by G_{j-1}, for $j \in [L]$.

2.1 Hardness Assumptions

We state the assumptions needed for our constructions.

Definition 1. *In a cyclic group $\mathbb{G}$ of prime order q, the* **Decisional Diffie-Hellman** (DDH) *problem is to distinguish the distributions*

$$D_0 = \{([\![1]\!], [\![a]\!], [\![b]\!], [\![ab]\!])\} \qquad D_1 = \{([\![1]\!], [\![a]\!], [\![b]\!], [\![c]\!])\}.$$

for $a, b, c \xleftarrow{\$} \mathbb{Z}_q$. The DDH *assumption in $\mathbb{G}$ assumes that no* ppt *adversary can solve the* DDH *problem with non-negligible probability.*

Definition 2. *In the bilinear setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$, the* **Symmetric eXternal Diffie-Hellman** (SXDH) *assumption makes the* DDH *assumption in both $\mathbb{G}_1$ and $\mathbb{G}_2$.*

2.2 Dual Pairing Vector Spaces

Our constructions rely on the *Dual Pairing Vector Spaces* (DPVS) framework in prime-order bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$ and $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}$ are all written additively. The DPVS technique dates back to the seminal work by Okamoto-Takashima [OT10, OT12a, OT12b] aiming at adaptive security for ABE as well as IBE, together with the *dual system methodology* introduced by Waters [Wat09]. In [LW10], the setting for dual systems is composite-order bilinear groups. Continuing on this line of works, Chen *et al.* [CLL+13] used prime-order bilinear groups under the SXDH assumption. Let us fix $N \in \mathbb{N}$ and consider $\mathbb{G}_1^N$ having N copies of $\mathbb{G}_1$. Any $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1 \in \mathbb{G}_1^N$ is identified as the vector $(x_1, \ldots, x_N) \in \mathbb{Z}_q^N$. There is no ambiguity because $\mathbb{G}_1$ is a cyclic group of order q prime. The $\mathbf{0}$-vector is $\mathbf{0} = [\![(0, \ldots, 0)]\!]_1$. The addition of two vectors in $\mathbb{G}_1^N$ is defined by coordinate-wise addition. The scalar multiplication of a vector is defined by $t \cdot \mathbf{x} := [\![t \cdot (x_1, \ldots, x_N)]\!]_1$, where $t \in \mathbb{Z}_q$ and $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1$. The additive inverse of $\mathbf{x} \in \mathbb{G}_1^N$ is defined to be $-\mathbf{x} := [\![(-x_1, \ldots, -x_N)]\!]_1$. Viewing $\mathbb{Z}_q^N$ as a vector space of dimension N over $\mathbb{Z}_q$ with the notions of bases, we can obtain naturally a similar notion of bases for $\mathbb{G}_1^N$. More specifically, any invertible matrix $B \in GL_N(\mathbb{Z}_q)$ identifies a basis $\mathbf{B}$ of $\mathbb{G}_1^N$, whose i-th row $\mathbf{b}_i$ is $[\![B^{(i)}]\!]_1$, where $B^{(i)}$ is the i-th row of B. The canonical basis $\mathbf{A}$ of $\mathbb{G}_1^N$ consists of $\mathbf{a}_1 := [\![(1, 0 \ldots, 0)]\!]_1, \mathbf{a}_2 := [\![(0, 1, 0 \ldots, 0)]\!]_1, \ldots, \mathbf{a}_N := [\![(0, \ldots, 0, 1)]\!]_1$. It is straightforward that we can write $\mathbf{B} = B \cdot \mathbf{A}$ for any basis $\mathbf{B}$ of $\mathbb{G}_1^N$ corresponding to an invertible matrix $B \in GL_N(\mathbb{Z}_q)$. We write $\mathbf{x} = (x_1, \ldots, x_N)_\mathbf{B}$ to indicate the representation of $\mathbf{x}$ in the basis $\mathbf{B}$, i.e. $\mathbf{x} = \sum_{i=1}^N x_i \cdot \mathbf{b}_i$. By convention the writing $\mathbf{x} = (x_1, \ldots, x_N)$ concerns the canonical basis $\mathbf{A}$.

Treating $\mathbb{G}_2^N$ similarly, we can furthermore define a product of two vectors $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1 \in \mathbb{G}_1^N, \mathbf{y} = [\![(y_1, \ldots, y_N)]\!]_2 \in \mathbb{G}_2^N$ by $\mathbf{x} \times \mathbf{y} := \prod_{i=1}^N \mathbf{e}(\mathbf{x}[i], \mathbf{y}[i]) = [\![\langle (x_1, \ldots, x_N), (y_1, \ldots, y_N) \rangle]\!]_\mathsf{t}$. Given a basis $\mathbf{B} = (\mathbf{b}_i)_{i \in [N]}$ of $\mathbb{G}_1^N$, we define $\mathbf{B}^*$ to be a basis of $\mathbb{G}_2^N$ by first defining $B' := (B^{-1})^\top$ and the i-th row $\mathbf{b}_i^*$ of $\mathbf{B}^*$ is $[\![B'^{(i)}]\!]_2$. It holds that $B \cdot (B')^\top = I_N$ the identity matrix and $\mathbf{b}_i \times \mathbf{b}_j^* = [\![\delta_{i,j}]\!]_\mathsf{t}$ for every $i, j \in [N]$, where $\delta_{i,j} = 1$ if and only if $i = j$. We call the pair $(\mathbf{B}, \mathbf{B}^*)$ a *pair of dual orthogonal bases* of $(\mathbb{G}_1^N, \mathbb{G}_2^N)$. If $\mathbf{B}$ is constructed by a random invertible matrix $B \xleftarrow{\$} GL_N(\mathbb{Z}_q)$, we call the resulting $(\mathbf{B}, \mathbf{B}^*)$ a pair of random dual bases. A DPVS is a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q, N)$ with dual orthogonal bases. In this work, we also use extensively *basis changes* over dual orthogonal bases of a DPVS to argue the steps of switching key as well as ciphertext vectors to semi-functional mode in our proofs. The details of such basis changes are recalled in the appendix A.2.

2.3 Access Structure and Linear Secret Sharing Schemes

We recall below the vocabularies of access structures and linear secret sharing schemes that will be used in this work. Let $\mathsf{Att} = \{\mathsf{att}_1, \mathsf{att}_2, \ldots, \mathsf{att}_m\}$ be a finite universe of attributes. An *access structure* over Att is a family $\mathbb{A} \subseteq 2^{\mathsf{Att}} \setminus \{\varnothing\}$. A set in $\mathbb{A}$ is said to be *authorized*; otherwise it is *unauthorized*. An access structure $\mathbb{A}$ is *monotone* if $\mathsf{S}_1 \subseteq \mathsf{S}_2 \subseteq \mathsf{Att}$ and $\mathsf{S}_1 \in \mathbb{A}$ imply $\mathsf{S}_2 \in \mathbb{A}$. Given a set of attributes $\mathsf{S} \subseteq \mathsf{Att}$, we write $\mathbb{A}(\mathsf{S}) = 1$ if and only if there exists $A \subseteq \mathsf{S}$ such that A is authorized. A secret sharing scheme for an access structure $\mathbb{A}$ over the attributes $\mathsf{Att} = \{\mathsf{att}_1, \mathsf{att}_2, \ldots, \mathsf{att}_m\}$ allows sharing a secret s among the m attributes att_j for $1 \leq j \leq m$, such that: (1) Any authorized set in $\mathbb{A}$ can be used to reconstruct s from the shares of its elements; (2) Given any unauthorized set and its shares, the secret s is statistically identical to a uniform random value. We will use *linear secret sharing schemes* (LSSS), which is recalled below:

Definition 3 (LSSS [Bei96]). *Let K be a field, $d, f \in \mathbb{N}$, and Att be a finite universe of attributes. A* Linear Secret Sharing Scheme LSSS *over K for an access structure $\mathbb{A}$ over Att is specified by a share-generating matrix $\mathbf{A} \in K^{d \times f}$ such that for any $I \subset [d]$, there exists a vector $\mathbf{c} \in K^d$ with support I and $\mathbf{c} \cdot \mathbf{A} = (1, 0, \ldots, 0)$ if and only if $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$.*

In order to share s using an LSSS over K, one first picks uniformly random values $v_2, v_3, \ldots, v_f \xleftarrow{\$} K$ and the share for an attribute att_i is the i-th coordinate $\mathbf{s}[i]$ of the share vector $\mathbf{s} := (s, v_2, v_3, \ldots, v_f) \cdot \mathbf{A}^\top$. Then, only an authorized set $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$ for some $I \subseteq [d]$ can recover $\mathbf{c}$ to reconstruct s from the shares by: $\mathbf{c} \cdot \mathbf{s}^\top = \mathbf{c} \cdot (\mathbf{A} \cdot (s, v_2, v_3, \ldots, v_f)^\top) = s$. Some canonical examples of LSSS include Shamir's secret sharing scheme for any f-out-of-d threshold gate [Sha79] or Benaloh and Leichter's scheme for any monotone formula [BL90]. An access structure $\mathbb{A}$ is said to be LSSS-*realizable* if there exists a linear secret sharing scheme implementing $\mathbb{A}$.

Let $y \in \mathbb{Z}_q$ where q is prime and for the sake of simplicity, let $\mathsf{Att} \subset \mathbb{Z}_q$ be a set of attributes. Let $\mathbb{A}$ be a monotone access structure over Att realizable by an LSSS over $\mathbb{Z}_q$. A *random labeling* procedure $\Lambda_y(\mathbb{A})$ is a secret sharing of y using LSSS:

$$\Lambda_y(\mathbb{A}) := (y, v_2, v_3, \ldots, v_f) \cdot \mathbf{A}^\top \in \mathbb{Z}_q^d \tag{1}$$

where $\mathbf{A} \in \mathbb{Z}_q^{d \times f}$ is the share-generating matrix and $v_2, v_3, \ldots, v_f \xleftarrow{\$} \mathbb{Z}_q$.

2.4 The Masking Lemma

We state a technical lemma that is employed throughout our proofs. A detailed proof can be found in the appendix B.2. The general purposes of the variables τ, x, y, z_j in the lemma are discussed in the technical overview in Section 3.2.

Lemma 4 (Adapted from [OT10, OT12a, OT12b, DGP21]). *Let $\mathbb{A}$ be an LSSS-realizable over a set of attributes $\mathsf{Att} \subseteq \mathbb{Z}_q$. We denote by $\mathsf{List\text{-}Att}(\mathbb{A})$ the list of attributes appearing in $\mathbb{A}$ and by P the cardinality of $\mathsf{List\text{-}Att}(\mathbb{A})$. Let $\mathsf{S} \subseteq \mathsf{Att}$ be a set of attributes. Let $(\mathbf{H}, \mathbf{H}^*)$ and $(\mathbf{F}, \mathbf{F}^*)$ be two random dual bases of $(\mathbb{G}_1^2, \mathbb{G}_2^2)$ and $(\mathbb{G}_1^8, \mathbb{G}_2^8)$, respectively. The vectors $(\mathbf{h}_1, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ are public, while all other vectors are secret. Suppose we have two random labelings $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ for $a_0, a'_0 \xleftarrow{\$} \mathbb{Z}_q$. Then, under the SXDH assumption in $(\mathbb{G}_1, \mathbb{G}_2)$, the following two distributions are computationally indistinguishable:*

$$D_1 := \left\{ \begin{array}{ll} x, y & \\ \forall\, j \in \mathsf{S}: & \mathbf{c}_j = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\ \forall\, j \in \mathsf{List\text{-}Att}(\mathbb{A}): & \mathbf{k}_j^* = (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\ & \mathbf{c}_{\mathsf{root}} = (\psi,\ 0)_{\mathbf{H}} \\ & \mathbf{k}^*_{\mathsf{root}} = (a_0 \cdot z,\ 0)_{\mathbf{H}^*} \end{array} \right\}$$

and

$$D_2 := \left\{ \begin{array}{lll} x, y & & \\ \forall\, j \in \mathsf{S}: & \mathbf{c}_j & = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ \tau z_j \cdot x,\ 0,\ 0)_{\mathbf{F}} \\ \forall\, j \in \mathsf{List\text{-}Att}(\mathbb{A}): & \mathbf{k}^*_j & = (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ a'_j \cdot y / z_j,\ 0,\ 0)_{\mathbf{F}^*} \\ & \mathbf{c}_{\mathsf{root}} & = (\psi,\ \tau \cdot x)_{\mathbf{H}} \\ & \mathbf{k}^*_{\mathsf{root}} & = (a_0 \cdot z,\ a'_0 \cdot y)_{\mathbf{H}^*} \end{array} \right\}$$

for any $x, y \in \mathbb{Z}_q$ *and* $z_j, \sigma_j, \pi_j, \psi, \tau, z, r'_0 \xleftarrow{\$} \mathbb{Z}_q$.

2.5 Functional Encryption with Fine-Grained Access Control

We first present the syntax of functional encryption with a fine-grained access control following the works in [ACGU20, LLW21, PD21]. The functionality class is $\mathcal{F} \times \mathsf{AC\text{-}K}$. The evaluation functions is taken from $\mathcal{F} := \{F_\lambda : \mathcal{D}_\lambda \to \mathcal{R}_\lambda\}_\lambda$ is a family of functions indexed by security parameters $\lambda \in \mathbb{N}$. When $F_\lambda, \mathcal{D}_\lambda$, and $\mathcal{R}_\lambda$ are clear from context, we drop the subscript λ and use the shorthands $F, \mathcal{D}$, and $\mathcal{R}$ respectively. The access control is captured by a relation $\mathsf{Rel} : \mathsf{AC\text{-}K} \times \mathsf{AC\text{-}Ct} \to \{0, 1\}$, for some sets $\mathsf{AC\text{-}Ct}$ and $\mathsf{AC\text{-}K}$. A plaintext consists of $(\mathsf{ac\text{-}ct}, x) \in \mathsf{AC\text{-}Ct} \times \mathcal{D}_\lambda$, whose corresponding ciphertext can be decrypted to $F_\lambda(x)$ using the functional key $\mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}}$ for $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$ if and only if $\mathsf{Rel}(\mathsf{ac\text{-}k}, \mathsf{ac\text{-}ct}) = 1$. The syntax of such functional encryption schemes is given below:

Definition 5 (Functional encryption with fine-grained access control). *A* functional encryption scheme with fine-grained access control *for* $\mathcal{F} \times \mathsf{AC\text{-}K}$ *consists of four algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$*:*

$\mathsf{Setup}(1^\lambda)$**:** *Given as input a security parameter* λ*, output a pair* $(\mathsf{pk}, \mathsf{msk})$.
$\mathsf{Extract}(\mathsf{msk}, F_\lambda, \mathsf{ac\text{-}k})$**:** *Given* $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$*, a function description* $F_\lambda \in \mathcal{F}$*, and the master secret key* msk*, output a secret key* $\mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}}$.
$\mathsf{Enc}(\mathsf{pk}, x, \mathsf{ac\text{-}ct})$**:** *Given as inputs* $\mathsf{ac\text{-}ct} \in \mathsf{AC\text{-}Ct}$*, the public key* pk*, and a message* $x \in \mathcal{D}_\lambda$*, output a ciphertext* ct.
$\mathsf{Dec}(\mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}}, \mathsf{ct})$**:** *Given the functional secret key* $\mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}}$*, and a ciphertext* ct*, output an element in* $\mathcal{R}_\lambda$ *or an invalid symbol* $\perp$.

Correctness. For sufficiently large $\lambda \in \mathbb{N}$, for all $(F_\lambda, \mathsf{ac\text{-}k}) \in \mathcal{F} \times \mathsf{AC\text{-}K}$ and $(\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda), \mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}} \leftarrow \mathsf{Extract}(\mathsf{msk}, F_\lambda, \mathsf{ac\text{-}k})$ for all $\mathsf{ac\text{-}ct}$ satisfying $\mathsf{Rel}(\mathsf{ac\text{-}k}, \mathsf{ac\text{-}ct}) = 1$, it holds with overwhelming probability that

$$\mathsf{Dec}(\mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}}, \mathsf{Enc}(\mathsf{pk}, x, \mathsf{ac\text{-}ct})) = F_\lambda(x) \text{ whenever } F_\lambda(x) \neq \perp^{3} \ ,$$

where the probability is taken over the random coins of the algorithms.

Security. We recall in the appendix A.3 the notion of *indistinguishability-based security against chosen-plaintext attacks (IND-CPA)* in the same manner as in [ABDP15], taking into account the attribute-based control using policies, as well as a *simulation-based* notion in a selective setting as in [ACGU20].

Remark 6. In Sections 4 and 5, our concrete constructions instantiate $\mathsf{AC\text{-}K}$ as a class of policies and $\mathsf{AC\text{-}Ct}$ as a superset over an attribute space, while the relation is the natural evaluation $\mathsf{Rel}(\mathbb{A} \in \mathsf{AC\text{-}K}, \mathsf{S} \in \mathsf{AC\text{-}Ct}) := \mathbb{A}(\mathsf{S})$. Following the terminology of ABE schemes, our constructions are *key-policy* (KP). By treating $\mathsf{AC\text{-}K}$ as a superset over an attribute space and $\mathsf{AC\text{-}Ct}$ as a class of policies, we will obtain *ciphertext-policy* (CP) schemes. The KP and CP notions are symmetric in terms of how we determine the support $\mathsf{AC\text{-}K} \times \mathsf{AC\text{-}Ct}$ of Rel.

[3] See [BO13, ABN10] for discussions about this relaxation. The general reason is that some functionality might contain $\perp$ in its range and if $F_\lambda(x) = \perp$ we do not impose $\mathsf{Dec}(\mathsf{sk}_{F_\lambda, \mathsf{ac\text{-}k}}, \mathsf{Enc}(\mathsf{pk}, x, \mathsf{ac\text{-}ct})) = F_\lambda(x)$, neither do we disallow it.

3 Technical Overview

3.1 Formalizing Access Control in Functional Encryption

First of all, we discuss how we formalize access control in the notion of functional encryption, which will affect our formal definitions in both single-client setting (Definition 5) and multi-client setting (Definition 8). On the one hand, accompanying an encryption scheme with access control over decryption keys is already expressed by ABE, which in itself is a special case of FE. Thus, FE schemes with fine-grained access control can be described by the general FE notion for any class of functions that can handle the desired access control along with the required computation.

On the other hand, when working with concrete functionality, we usually find ourselves in the context where the evaluation *cannot* express the access control and they cannot be described abstractly using a single functionality. Therefore, in this paper we consider FE with access control as FE schemes for *particular* functionality class whose description can be separated into two parts $\mathcal{F} \times \mathsf{AC\text{-}K}$: (1) a first part $F \in \mathcal{F}$ for evaluation, (2) and a second part for access control captured by a binary relation $\mathsf{Rel} : \mathsf{AC\text{-}K} \times \mathsf{AC\text{-}Ct} \to \{0,1\}$, for some sets $\mathsf{AC\text{-}K}, \mathsf{AC\text{-}Ct}$. The key extraction is done with respect to $(\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}, F)$, meanwhile the encryption procedure will receive $(\mathsf{ac\text{-}ct} \in \mathsf{AC\text{-}Ct}, x)$. A key $\mathsf{sk}_{\mathsf{ac\text{-}k},F}$ can decrypt a ciphertext $\mathsf{ct}_{\mathsf{ac\text{-}ct}}(x)$ to $F(x)$ if and only if $\mathsf{Rel}(\mathsf{ac\text{-}k}, \mathsf{ac\text{-}ct}) = 1$. We stress that this way of formulation does not take us out of the FE regime, as it is still captured by the general FE notion.

We show how the above fomalization is used in a concrete case. In the following discussion we will distinguish the $\boxed{\text{input}}$ during encryption from the $\boxed{\text{parameters}}$ during key extraction. In this paper we focus on $F \in \mathcal{F}^{\mathsf{IP}} = \{F_{\mathbf{y}} : \mathbb{Z}_q^n \to \mathbb{Z}_q\}$ for computing inner products over $\mathbb{Z}_q^n$ for some prime q and $n \in \mathbb{N}$, where $F_{\mathbf{y}}(\mathbf{x}) := \langle \mathbf{x}, \mathbf{y} \rangle$. The simplest non-trivial example for access control is identity-based control, i.e. $\mathsf{AC\text{-}K} = \mathsf{AC\text{-}Ct} = \mathsf{ID}$ for some identity space ID and $\mathsf{Rel}_{\mathsf{ibe}}(\mathsf{id\text{-}k}, \mathsf{id\text{-}ct}) = \left(\mathsf{id\text{-}k} \stackrel{?}{=} \mathsf{id\text{-}ct}\right)$. The functional keys are extracted using $\boxed{(\mathsf{id\text{-}k}, \mathbf{y})}$ and the ciphertexts are encrypted using $\boxed{(\mathsf{id\text{-}ct}, \mathbf{x})}$. First of all, it is *not* immediate how $\mathcal{F}^{\mathsf{IP}}$ can be used to implement the check $\tau z \cdot (\boxed{\mathsf{id\text{-}k}} - \boxed{\mathsf{id\text{-}ct}})$ for the identity-based control, where τ and z are random values generated for encryption and key extraction, respectively, together acting as a mask of the decryption value. Notably, the value z *cannot* be specified as part of the inner-product evaluation function, because the inner-product evaluation itself must be independent of users at the time of generating functional keys, *nor* as part of the ciphertext. It thus seems indispensable to treat the functionality as $\mathcal{F}^{\mathsf{IP}} \times \mathsf{ID}$: the functional key is generated w.r.t $F_{\boxed{\mathbf{y}}} \in \mathcal{F}^{\mathsf{IP}}$ and $\boxed{\mathsf{id\text{-}k}} \in \mathsf{ID}$, while the ciphertext is encrypted w.r.t $\boxed{(\mathsf{id\text{-}ct} \in \mathsf{ID}, \mathbf{x} \in \mathbb{Z}_q^n)}$. During decryption for obtaining $\langle \boxed{\mathbf{x}}, \boxed{\mathbf{y}} \rangle + \tau z \cdot (\boxed{\mathsf{id\text{-}k}} - \boxed{\mathsf{id\text{-}ct}})$, the ID-part of the functional key will implement the control $\tau z \cdot (\boxed{\mathsf{id\text{-}k}} - \boxed{\mathsf{id\text{-}ct}})$ whilst the $\mathcal{F}^{\mathsf{IP}}$-part will compute $\langle \boxed{\mathbf{x}}, \boxed{\mathbf{y}} \rangle$.

Treatment of Tags in MCFE with Access Control. As mentioned in the introduction, our current objective is constructing MCFE schemes with access control having smaller ciphertexts. We use the functionality $\mathcal{F}^{\mathsf{IP}} \times \mathsf{ID}$ as a running example. The input $\boxed{\mathbf{x}}$ for inner-product calculation is broken down into n components for the entries x_i of $\boxed{\mathbf{x}}$. The encryption procedure takes $\boxed{(x_i, \mathsf{id\text{-}ct}_i, \mathsf{tag}_i)}$ and outputs a ciphertext component ct_i, for some identity $\boxed{\mathsf{id\text{-}ct}_i}$ and a tag $\boxed{\mathsf{tag}_i}$. The decryption procedure receives a functional key, which is derived from $F_{\boxed{\mathbf{y}}} \in \mathcal{F}^{\mathsf{IP}}$ and $\boxed{\mathsf{id\text{-}k}} \in \mathsf{ID}$, and the n ciphertext components $(\mathsf{ct}_i)_{i=1}^n$. The decrypted result is $\langle \boxed{\mathbf{x}}, \boxed{\mathbf{y}} \rangle$ if $\boxed{\mathsf{id\text{-}ct}_i} = \boxed{\mathsf{id\text{-}k}}$ for all i and $\boxed{\mathsf{tag}_i} = \boxed{\mathsf{tag}_j}$ for all i, j. In the setting that the identities and tags can be public, if the identity control does not pass or if the tags are not the same, a totally random value is returned by the decryption procedure. We now face the same problem of checking equality among $\boxed{\mathsf{tag}_i}$ in the same manner that has to be done for identities from ID.

First of all, it is unlikely that we want to embed the checks $\boxed{\mathsf{tag}_i} \stackrel{?}{=} \boxed{\mathsf{tag}_j}$ in the $\mathcal{F}^{\mathsf{IP}}$-part. More specifically, we would have to make the decryption compute $(\sum_{i=1}^n \boxed{x_i}, \boxed{y_i}) + \tau z \cdot (\boxed{\mathsf{id\text{-}k}} - \boxed{\mathsf{id\text{-}ct}_i}) + \sum_{i=1}^{n-1} z_i(\boxed{\mathsf{tag}_i} - \boxed{\mathsf{tag}_{i+1}})$ from n ciphertext components ct_i of $\boxed{(x_i, \mathsf{id\text{-}ct}_i)}$, for some

random values $z, z_i \xleftarrow{\$} \mathbb{Z}_q$ and $\boxed{\mathbf{y} = (y_1, ..., y_n)}$. It is worth noting that the check $z_i(\boxed{\mathsf{tag}_i} - \boxed{\mathsf{tag}_{i+1}})$ needs two values defined at encryption time and not key extraction time. Therefore, in order for the functional key to "perform" the n required checks, all n tags $\boxed{(\mathsf{tag}_1, \ldots, \mathsf{tag}_n)}$ must be encrypted in an IBE-style in ct_i. Roughly speaking, this makes each ct_i of size linear in n, due to the number of group elements required for encrypting the n tags, in addition to a constant number of group elements for encrypting $\boxed{(x_i, \mathsf{id\text{-}ct}_i)}$. Thus the total communication increases to quadratic in n over all n components ct_i, which is exactly what we are trying to avoid.

Furthermore, it might be tempting to embed the equality checks in the access control but because $\boxed{\mathsf{tag}_i, \mathsf{tag}_j}$ are defined only at encryption time, they are unknown to the key extraction for the ID-part. More generally, in a setting that permits a *different* [4] attribute set $\boxed{\mathsf{S}_i}$ in each individual ciphertext, one can try to regard $\boxed{\mathsf{tag}_i}$ as an attribute in $\boxed{\mathsf{S}_i}$. The correctness insists on the condition $\ulcorner\mathbb{A}\urcorner(\boxed{\mathsf{S}_i}) = 1$ for all i and the equality checks $\boxed{\mathsf{tag}_i} \stackrel{?}{=} \boxed{\mathsf{tag}_j}$ must somehow be done by $\ulcorner\mathbb{A}\urcorner(\boxed{\mathsf{S}_i})$, which is not possible due to the fact that $\boxed{\mathsf{tag}_j}$ is independent of both $\ulcorner\mathbb{A}\urcorner$ and $\boxed{\mathsf{S}_i}$. Consequently, we have to cope with the tags independently from the functionality's description. As a final remark, this also demonstrates the gap between MIFE and MCFE for the concrete functionality to compute inner products under access control by access structures, even though the general notion of MIFE can describe MCFE, provided that the evaluation functions of the underlying functionality class can test equality between $\boxed{\mathsf{tag}_i}$.

3.2 Adaptively Secure Single-Client Construction

Our construction for functional encryption schemes with fine-grained access control is using *Dual Pairing Vector Spaces* (DPVSes). We highlight our main ideas to achieve adaptive security. We refer to Section 2.2 for background on DPVSes. Our schemes are key-policy, such that the access structure $\mathbb{A}$ is expressed in the key using vectors $\{(\mathbf{k}^*_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}_{\mathsf{root}}\}$ over $\mathbb{G}_2$ and a set S of attributes are embedded in the ciphertext using vectors $\{(\mathbf{c}_j)_{j \in \mathsf{S}}, \mathbf{c}_{\mathsf{root}}\}$ over $\mathbb{G}_1$, where $\mathsf{List\text{-}Att}(\mathbb{A})$ is the list of attributes appearing in the access structure $\mathbb{A}$. We use a linear secret sharing scheme based on $\mathbb{A}$ to create the shares $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ of $a_0 \xleftarrow{\$} \mathbb{Z}_q$. The shares will then be embedded in the functional secret key components $(\mathbf{k}^*_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$. When all the components corresponding to an authorized set in $\mathbb{A}$ are present, the shares can be combined to reconstruct the secret value a_0, which is now embedded in a key component $\mathbf{k}^*_{\mathsf{root}}$. In all vectors $(\mathbf{c}_j)_j$ and $\mathbf{c}_{\mathsf{root}}$, we put a random value ψ. Intuitively, $[\![\psi a_0]\!]_{\mathsf{t}}$ is masking the IPFE-related ciphertext of Agrawal *et al.*'s type [ALS16]. The vectors $((\mathbf{k}^*_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}^*_{\mathsf{root}})$ and $((\mathbf{c}_j)_{j \in \mathsf{S}}, \mathbf{c}_{\mathsf{root}})$ lie in the dual orthogonal bases. Performing the products $\mathbf{c}_j \times \mathbf{k}^*_j$ and combining over $j \in \mathsf{S}$, where S is an authorized set, will permit recovering $[\![\psi a_0]\!]_{\mathsf{t}}$ that can be used to cancel out $[\![\psi a_0]\!]_{\mathsf{t}}$ in $\mathbf{c}_{\mathsf{root}} \times \mathbf{k}_{\mathsf{root}}$:

$$\begin{array}{cccccccccc} \mathbf{c}_j & (& \cdots & \psi & 0 & \cdots &)_{\mathbf{F}} ; & \mathbf{c}_{\mathsf{root}} & (& \cdots & \psi & 0 & \cdots &)_{\mathbf{H}} \\ \mathbf{k}^*_j & (& \cdots & a_j & 0 & \cdots &)_{\mathbf{F}^*} ; & \mathbf{k}^*_{\mathsf{root}} & (& \cdots & a_0 & 0 & \cdots &)_{\mathbf{H}^*} \end{array}$$

We use the techniques for adaptively-secure ABE introduced in the original work of Okamoto and Takashima [OT10, OT12a, OT12b] in the ensuing steps. In vein of the *dual-system methodology*, there are two modes of operation for keys and ciphertexts: a normal mode and a *semi-functional* mode. A normal key can decrypt any ciphertext, a semi-functional key can decrypt only normal ciphertexts, and decrypting semi-functional ciphertexts using semi-functional keys gives totally random values. The dual-system method proves security by a sequence of indistinguishable changes to make the challenge ciphertext semi-functional, then to make the keys semi-functional and in the end the challenge message will be perfectly hidden from the adversary.

[4] If all clients must use the *same* set of attributes $\boxed{\mathsf{S}}$, we can treat $\boxed{\mathsf{tag}_i}$ as a virtual attribute in S, while enforcing the same $\boxed{\mathsf{S}}$ for all i. This implies that all $\boxed{\mathsf{tag}_i}$ must be the same. However, this approach requires a consensus among all n clients on S, which general might be more complicated than agreeing on tag.

Interestingly, there is a twist stemming from the security model when integrating this technique into our security proofs for FE with access control: an adversary can additionally query for keys that work with the challenge ciphertext, i.e. the key's policy is satisfied. So as to achieve adaptive security, we have to be much more careful about which key to turn semi-functional, because the keys whose policies are satisfied should be capable of decrypting the (semi-functional) challenge ciphertext.

Our goal is to mask the value a_0 in $\mathbf{k}^*_{\mathsf{root}}$ by introducing a random mask $a'_0 y$ in the coordinate of *hidden* basis vectors, i.e. those that are not used at all in real life and are defined only for the proof, while the facing coordinate in $\mathbf{c}_{\mathsf{root}}$ is also changed to τx so as to mask ψ:

$$\begin{array}{cccccccc;cccccccc} \mathbf{c}_j & (& \cdots & \psi & \tau x z_j & \cdots &)_{\mathbf{F}} & ; & \mathbf{c}_{\mathsf{root}} & (& \cdots & \psi & \tau x & \cdots &)_{\mathbf{H}} \\ \mathbf{k}^*_j & (& \cdots & a_j & a'_j y / z_j & \cdots &)_{\mathbf{F}^*} & ; & \mathbf{k}^*_{\mathsf{root}} & (& \cdots & a_0 & a'_0 y & \cdots &)_{\mathbf{H}^*} \end{array} .$$

The values x, y are known constants, $\tau, a'_0, (z_j)_j \xleftarrow{\$} \mathbb{Z}_q$, and $(a'_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ is another ensemble of secret shares for a'_0. Consequently, this will introduce a value $[\![\tau a'_0 xy]\!]_{\mathsf{t}}$ masking $[\![\psi a_0]\!]_{\mathsf{t}}$ when performing the product $\mathbf{c}_{\mathsf{root}} \times \mathbf{k}_{\mathsf{root}}$. We note that the value a'_0 is related to $(a'_j/z_j)_j$ by $a'_0 = \sum_{j \in \mathsf{S}'} z_j \cdot (a'_j/z_j)$ for any S' such that $\mathbb{A}(\mathsf{S}') = 1$. In the end, if $\mathbb{A}(\mathsf{S}) = 1$, from $\mathbf{c}_j$ and $\mathbf{k}_j$ it is possible to reconstruct $[\![\tau a'_0 xy]\!]_{\mathsf{t}}$ and recover $[\![\psi a_0]\!]_{\mathsf{t}}$. Otherwise, the entropy of a'_0 is preserved thanks to the randomness provided by $z_j \xleftarrow{\$} \mathbb{Z}_q$ for randomizing $(a'_j)_j$ to $(a'_j/z_j)_j$ in the components $(\mathbf{c}_j)_j$ of the *unique* challenge ciphertext[5], as well as the fact that $\mathbb{A}(\mathsf{S}) = 0$ means there will be some a'_j/z_j missing in the components $(\mathbf{k}^*_j)_j$ and the value z_j is information-theoretically hidden. Hence, if $\mathbb{A}(\mathsf{S}) = 0$ we will be able to change a'_0 to an independent and uniformly random value $r_0 \xleftarrow{\$} \mathbb{Z}_q^*$. It is obligatory that we apply this argument *key by key*, while considering the key's capability to decrypt the challenge ciphertext, because two different keys might mutually leak information about the same z_j and our statistically argument no longer holds. After a sequence of hybrids on the functional key queries, we can mask all the keys as desired so that the key and the challenge ciphertext will become readily semi-functional for later steps in the proof.

However, only for functional keys whose policy is not satisfied can we perform such a change from a'_0 to r_0, and we can decide the satisfiability only when the adversary adaptively queries for functional keys. Our idea is to introduce r_0 in *all* key components and at the same time use a mechanism to "cancel out" the masks $((a'_j/z_j)_j, r_0)$ in $((\mathbf{k}^*_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}^*_{\mathsf{root}})$ if $\mathbb{A}(\mathsf{S}) = 1$. It is indispensable to have this mechanism because otherwise, as soon as we change a'_0 to r_0, even the reconstruction $\sum_{j \in \mathsf{S}'} z_j \cdot (a'_j/z_j) = a'_0$ is not able to remove r_0 for a correct decryption. In our particular setting for computing inner-products, we observe that if $\mathbb{A}(\mathsf{S}) = 1$, then $\langle \Delta\mathbf{x}, \mathbf{y} \rangle = 0$ for the sake of avoiding trivial attacks, where $\Delta\mathbf{x} := \mathbf{x}^*_1 - \mathbf{x}^*_0$ is the difference of the two left-or-right challenge messages and $\mathbf{y}$ is specified the functional key. In the selective setting where $\Delta\mathbf{x}$ is known in advance, the key and ciphertext components can simply be masked using the constants $(x, y) := (1, \langle \Delta\mathbf{x}, \mathbf{y} \rangle)$. However, for the goal of adaptive security where $\Delta\mathbf{x}$ is unknown at the time of key extraction, we have to make a trade-off and use DPVSes of dimensions linear in the dimension n of vectors for inner-products and mask the key and ciphertext components as follows:

$$\begin{array}{cccccccc} \mathbf{c}_j & (& \cdots & \psi & \tau z_j \Delta\mathbf{x}[1] & \cdots & \tau z_j \Delta\mathbf{x}[n] & \cdots &)_{\mathbf{F}} \\ \mathbf{k}^*_j & (& \cdots & a_j & a'_j \mathbf{y}[1]/z_j & \cdots & a'_j \mathbf{y}[n]/z_j & \cdots &)_{\mathbf{F}^*} \\ \hline \mathbf{c}_{\mathsf{root}} & (& \cdots & \psi & \tau \Delta\mathbf{x}[1] & \cdots & \tau \Delta\mathbf{x}[n] & \cdots &)_{\mathbf{H}} \\ \mathbf{k}^*_{\mathsf{root}} & (& \cdots & a_0 & r_0 \mathbf{y}[1] & \cdots & r_0 \mathbf{y}[n] & \cdots &)_{\mathbf{H}^*} \end{array}$$

where each i-th pair of constants (x, y) is set to $(\Delta\mathbf{x}[i], \mathbf{y}[i])$ for all $i \in [n]$. Our arguments resort to a slight variant of the technique in [OT10, OT12a, OT12b], stated as a technical lemma (see Lemma 4) in Section 2.4. The lemma will use some auxiliary hidden vectors (which we do not show here) during the masking process and so as to economize the dimensions of our DPVSes, we apply the lemma n times in a sequence of hybrids to introduce $(\tau \Delta\mathbf{x}[i], r_0 \mathbf{y}[i])_i$ while reusing

[5] Since our single-client scheme is public-key, we can obtain multi-challenge security using a standard hybrid argument.

and cleaning those auxiliary hidden vectors after each application. After successfully laying $(r_0\mathbf{y}[i])_i$ in place, the rest of the proof will use r_0 as a source of randomness to completely hide the challenge message. Our single-client constructions are presented in Section 4.

3.3 The "Duplicate-and-Compress" Technique

We give a glimpse of our main technical method to obtain a multi-client construction from our single-client construction, while maintaining the total ciphertext's size of order linear in n. The intriguing point we observe is as long as each client uses an independent DPVS, the technique we use to take care of the ciphertext/key vectors in the single-client case can be carried out in a *parallel* manner, to some extent. Therefore, in the security proof, we can distribute and accumulate in parallel the necessary information in small-dimension vectors rather than centralizing such information in few vectors of big dimension. Our treatment for the multi-client setting is twofold and we give below the main technical ideas.

The more restrictive MCFE. Firstly, Section 5.2 presents a construction that enforces the same $\mathsf{S}_1 = \cdots = \mathsf{S}_n = \mathsf{S}$ for all clients, by hashing it using a full-domain hash function modeled as a random oracle (RO), along with the tag at the time of encryption. Indeed, we will use an argument resembling what we do in the single-client construction and perform a masking procedure key by key, where the functional key query for $(\mathbb{A}, \mathbf{y}^{(\ell)})$ is indexed by ℓ. For each $i \in [n]$,we mask $(\mathbf{k}^*_{i,j})_j = (..., a^{(\ell)}_{i,j}, a'^{(\ell)}_{i,j} y/z_j, ...)_j$,$\mathbf{k}^*_{i,\mathsf{root}} = (..., a^{(\ell)}_{i,0}, a'^{(\ell)}_{i,0} y, ...)$ and $(\mathbf{c}_{i,j})_j = (..., \psi_i, \tau x z_j, ...)_j, \mathbf{c}_{i,\mathsf{root}} = (..., \psi_i, \tau x, ...)$, where $(a^{(\ell)}_{i,j})_j, (a'^{(\ell)}_{i,j})_j$ are secret shares of $a^{(\ell)}_{i,0}, a'^{(\ell)}_{i,0}$ respectively. In this more restrictive case of Section 5.2 where all n clients use the same S, it entails all clients $i \in [n]$ using the same $a^{(\ell)}_0, a'^{(\ell)}_0$ with their secret shares $(a^{(\ell)}_j)_j, (a'^{(\ell)}_j)_j$ in $(\mathbf{k}^*_{i,j})_j = (..., a^{(\ell)}_j, a'^{(\ell)}_j y/z_j, ...)_j$ and $\mathbf{k}^*_{i,\mathsf{root}} = (..., a^{(\ell)}_0, a'^{(\ell)}_0 y, ...)$. Afterwards, we want to replace $a'^{(\ell)}_0$ by an independent and uniformly random value $r^{(\ell)}_0 \xleftarrow{\$} \mathbb{Z}^*_q$ if $\mathbb{A}(\mathsf{S}_i) = 0$ and clearing the masks otherwise. As our first observation, the reasoning is still based crucially on the fact that in S there will lack some j whose corresponding z_j permits recovering $a'^{(\ell)}_0 = \sum_j z_j (a'^{(\ell)}_j / z_j)$ if $\mathbb{A}(\mathsf{S}) = 0$. It gets clear that as long as $\mathbb{A}(\mathsf{S}) = 0$, for all i independently, the same argument will hold because all i use the same set S of attributes. This observation leads to a *compression* of all $(\mathbf{c}_{i,j})_j, (\mathbf{k}^*_{i,j})_j$ into one pair of dual bases $(\mathbf{F}, \mathbf{F}^*)$ instead of n separate pairs for each $i \in [n]$. As a second observation,when $\mathbb{A}(\mathsf{S}) = 1$, all ciphertext components must be combined together for a correct decryption. As a result, to program the canceling mechanism, instead of naively embedding n pairs of constants $(\Delta\mathbf{x}[k], \mathbf{y}^{(\ell)}[k])_{k=1}^n$ in $(\mathbf{c}_{i,\mathsf{root}}, (\mathbf{c}_{i,j})_j, \mathbf{k}^*_{i,\mathsf{root}}, (\mathbf{k}^*_{i,j})_j)$ for *each* i, we only need to embed $(\Delta\mathbf{x}[i], \mathbf{y}^{(\ell)}[i])$ in $(\mathbf{c}_{i,\mathsf{root}}, (\mathbf{c}_{i,j})_j, \mathbf{k}^*_{i,\mathsf{root}}, (\mathbf{k}^*_{i,j})_j)$. The grouping by i of the products $\mathbf{c}_{i,\mathsf{root}} \times \mathbf{k}^*_{i,\mathsf{root}}$ as well as $\sum_j \mathbf{c}_{i,j} \times \mathbf{k}^*_{i,j}$ will retrieve $[\![\tau r^{(\ell)}_0 \langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle]\!]_\mathsf{t}$ and we proceed the remaining as in the single-client proof. We point out that in the multi-client setting, it might be the case that some i are corrupted and the retrieval of $[\![\tau r^{(\ell)}_0 \langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle]\!]_\mathsf{t}$ is more complicated when regrouping over i. However, by carefully defining (see Definition 9) and considering only *admissible* adversaries, i.e. they cannot win by trivial attacks[6], it remains the case. This individual insertion of $(\Delta\mathbf{x}[i], \mathbf{y}^{(\ell)}[i])$ for each i leads to a *duplication* of one pair of dual bases $(\mathbf{H}_i, \mathbf{H}^*_i)$ for each $(\mathbf{c}_{i,\mathsf{root}}, \mathbf{k}^*_{i,\mathsf{root}})$, while all $(\mathbf{c}_{i,j})_j, (\mathbf{k}^*_{i,j})_j$ are readily put in the same basis following our first observation:

(Compressing) for all $i \in [n]$	$\mathbf{c}_{i,j}$	(	$\cdots$	ψ	$\tau\Delta\mathbf{x}[i]z_j$	$\cdots$	$)_\mathbf{F}$
	$\mathbf{k}^*_{i,j}$	(	$\cdots$	$a^{(\ell)}_j$	$a'^{(\ell)}_j \mathbf{y}^{(\ell)}[i]/z_j$	$\cdots$	$)_{\mathbf{F}^*}$
(Duplicating) for each $i \in [n]$	$\mathbf{c}_{i,\mathsf{root}}$	(	$\cdots$	ψ	$\tau\Delta\mathbf{x}[i]$	$\cdots$	$)_{\mathbf{H}_i}$
	$\mathbf{k}^*_{i,\mathsf{root}}$	(	$\cdots$	$a^{(\ell)}_0$	$a'^{(\ell)}_0 \mathbf{y}^{(\ell)}[i]$	$\cdots$	$)_{\mathbf{H}^*_i}$

We emphasize that this parallel process is feasible thanks to a conveniently smooth control, as low as the level of the vectors' coordinates in DPVSes. This potential of parallelization helps

[6] For instance, the adversary might corrupt i^*, query a left-or-right challenge $(\mathbf{x}_0, \mathbf{x}_1)$ where $\Delta\mathbf{x}[i^*] := \mathbf{x}_0[i^*] - \mathbf{x}_1[i^*] \neq 0$ and $\Delta\mathbf{x}[i] = 0$ for $i \neq i^*$, then decrypt the challenge ciphertext with a satisfied key for $\mathbf{y}^{(\ell)}$ whose i^*-th entry is non-zero.

us spread the necessary information for answering adaptive key queries, which accounts for the linearly large dimension, into n collections $\{(\mathbf{k}^*_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}^*_{i,\mathsf{root}}\}_{i\in[n]}$. On the one hand, we change the vectors $(\mathbf{k}^*_{i,j}, \mathbf{c}_{i,j})_{i,j}$ in parallel for all i, while these vectors are written in bases $(\mathbf{F}, \mathbf{F}^*)$. On the other hand, we change the vectors $(\mathbf{k}^*_{i,\mathsf{root}}, \mathbf{c}_{i,\mathsf{root}})_i$ independently for each client i, using the fact that each pair $(\mathbf{k}^*_{i,\mathsf{root}}, \mathbf{c}_{i,\mathsf{root}})$ belong to a separate pair of dual bases $(\mathbf{H}_i, \mathbf{H}^*_i)$. In the end, instead of using n bases of dimension n, we can use n bases of *constant* dimension for $(\mathbf{k}^*_{i,\mathsf{root}})_i$ along with one *constant*-dimension basis for all $\{(\mathbf{k}^*_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}\}_i$, saving a factor n in the ciphertext's size.

The more flexible MCFE. Section 5.4 discusses an extension of the above MCFE construction where we do not impose the same set of attributes among n clients. Each client i can now encrypt using a different S_i and the decryption can decrypt the inner-product if and only if $\mathbb{A}(\mathsf{S}_i) = 1$ for all i. Unsurprisingly, our argument as it is from the previous construction, for masking and for replacing $a'^{(\ell)}_{i,0}$ by an independent and uniformly random value, does not hold anymore because there might be two keys corresponding to $\mathbb{A}^{(\ell)}$ and $\mathbb{A}^{(\ell')}$ such that $\mathbb{A}^{(\ell)}(\mathsf{S}_i) \neq \mathbb{A}^{(\ell')}(\mathsf{S}_i)$ and the adversary might try to use key components of the ℓ'-th query to recover $a'^{(\ell)}_{i,0}$ in the ℓ-th query. We thus make use of another layer of random secret shares $(d_{\ell,i})_{i=1}^n$ over n components of each ℓ-th functional key, facing θ_i in the ciphertext components such that $\sum_{i=1}^n \theta_i d_{\ell,i} = 0$. The values $(\theta_i)_i$ are generated as part of the master secret key but $(d_{\ell,i})_{i=1}^n$ are chosen independently for each key. A fully working key can be obtain only if all the n components corresponding to $(d_{\ell,i})_{i=1}^n$ are combined. That will prevent the adversary from trying to mix components between two different keys, i.e. if $\mathbb{A}^{(\ell)}(\mathsf{S}_i) = 0$ we can be sure that $a'^{(\ell)}_{i,0}$ retains its entropy and stays hidden. After a similar masking step using the secret shares $(a'^{(\ell)}_{i,j})_j$ of $a'^{(\ell)}_{i,0}$ independently generated for each i, the randomness provided by $(d_{\ell,i})_{i=1}^n$ allows us to tweak $a'^{(\ell)}_{i,0}$ with a uniformly random value $r^{(\ell)}_0$:

$$
\begin{array}{l||cccc|c|c}
\text{(Compressing) for all } i\in[n] & \mathbf{c}_{i,j} & (\ \cdots & \psi & \tau\Delta\mathbf{x}[i]z_j & \cdots & \cdots)_{\mathbf{F}} \\
 & \mathbf{k}^*_{i,j} & (\ \cdots & a^{(\ell)}_j & a'^{(\ell)}_{i,j}\mathbf{y}^{(\ell)}[i]/z_j & \cdots & \cdots)_{\mathbf{F}^*} \\
\hline
\text{(Duplicating) for each } i\in[n] & \mathbf{c}_{i,\mathsf{root}} & (\ \cdots & \psi & \tau\Delta\mathbf{x}[i] & \theta_i & \cdots)_{\mathbf{H}_i} \\
 & \mathbf{k}^*_{i,\mathsf{root}} & (\ \cdots & a^{(\ell)}_0 & (a'^{(\ell)}_{i,0}+r^{(\ell)}_0)\mathbf{y}^{(\ell)}[i] & d_{\ell,i} & \cdots)_{\mathbf{H}^*_i}
\end{array}
$$

It is of the utmost importance that we rely on $(d_{\ell,i})_{i=1}^n$, which is particular for each ℓ-th key, to carry out this change from $a'^{(\ell)}_{i,0}$ to $a'^{(\ell)}_{i,0} + r^{(\ell)}_0$. Or else, the adversary can mix and match the ℓ-th and ℓ'-th keys to remove $a'^{(\ell)}_{i,0}$ and distinguish the adding of $r^{(\ell)}_0$, regardless whether S_i is authorized or not. The argument is now computational, in contrast to the information-theoretical indistinguishability when changing from $a'^{(\ell)}_0$ to $r^{(\ell)}_0$ in the more restrictive MCFE. We now perform an unmasking by going backwards to remove the sharing $(a'^{(\ell)}_{i,j})_j$ and $a'^{(\ell)}_{i,0}$ in the key. This transition is completely symmetric. If $\mathbb{A}(\mathsf{S}_i) = 1$ for all i, then the admissibility requires $\langle\Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle = 0$ and the noise $\tau r^{(\ell)}_0$ can be removed. Otherwise, in case $\langle\Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle \neq 0$, the mask $\tau r^{(\ell)}_0$ persists but the admissbility implies there exists i such that $\mathbb{A}(\mathsf{S}_i) = 0$ and the functional key cannot decrypt the challenge ciphertext. We emphasize that the incapability of the key when $\mathbb{A}(\mathsf{S}_i) = 0$ is ensured by $(d_{\ell,i})_{i=1}^n$. After introducing $r^{(\ell)}_0$, the remaining steps resemble the proof of the less flexible construction in Section 5.2. A desirable byproduct of this more flexible construction is that the hash function, which is modeled as a random oracle (RO), is now applied only on the tag. Therefore, we can obtain an MIFE in the standard model that is comparable to the work in [ACGU20] by fixing the hash value of a tag for all ciphertexts and publishing it as a parameter of the scheme.

4 Single-Client Functional Encryption For Inner-product with Fine-Grained Access Control via LSSS

We present constructions of FE for the inner-product functionality with attribute-based control expressed using linear secret sharing schemes, starting with the simpler single-client setting.

We are in the bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$ and $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ are written additively. The function class of interests is $\mathcal{F}^{\mathsf{IP}} \times \mathsf{LSSS}$ where $\mathcal{F}^{\mathsf{IP}}$ contains $F_{\mathbf{y}} : (\mathbb{Z}_q^*)^n \to \mathbb{Z}_q$ defined as $F_{\mathbf{y}}(\mathbf{x}) \coloneqq \langle \mathbf{x}, \mathbf{y} \rangle$. The access control is given by $\mathsf{Rel} : \mathsf{LSSS} \times 2^{\mathsf{Att}} \to \{0,1\}$, where $\mathsf{Rel}(\mathbb{A}, \mathsf{S}) = \mathbb{A}(\mathsf{S})$, the class LSSS contains Linear Secret Sharing Schemes over Att, and 2^{Att} denotes the superset of an attribute space $\mathsf{Att} \subseteq \mathbb{Z}_q$. Our constructions are key-policy, where $\mathbb{A}$ is embedded in the key and S is specified in the ciphertext. In order to facilitate the understanding and the motivation of our later multi-client constructions in Section 5, we present both selectively-secure and adaptively-secure single-client constructions in Figure 1. We leverage the selectively-secure scheme to obtain the adaptively-secure one by replacing certain elements in the former by the corresponding boxed components for the latter.

The main difference between the adaptive version and the selectively-secure version is the increase in the dimension of dual bases, from constant dimensions to dimensions linear in n. The details can be found in Figure 1. The computation for encrypting and decrypting stays essentially the same. We refer to the technical overview in Section 3 for the main ideas why using bigger DPVSes allows us to achieve the stronger adaptive notion. The *correctness* can be verified in a straightforward manner. Theorem 7 proves the adaptive IND-security for the construction corresponding to boxed components in Figure 1, where the adversary can query a unique challenge ciphertext and multiple functional keys. Using a standard hybrid argument and recalling that our scheme is public-key provide us with adaptive security against multiple challenge ciphertexts. The easier selective security can be proved using similar techniques. Figure 9 in Appendix B.3 describes the main ideas including the sequence of games employed in the proof. Full details can be also found therein.

Theorem 7. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an IPFE scheme with fine-grained access control via LSSS presented in Figure 1 in a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$, for the functionality class $\mathcal{F}^{\mathsf{IP}} \times \mathsf{LSSS}$. Then, $\mathcal{E}$ is secure against chosen-plaintext attacks, adaptively in the attributes and the challenge messages, if the SXDH assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$. More precisely, for $\lambda \in \mathbb{N}$ and for any ppt adversary $\mathcal{A}$, let n be the dimension of vectors for inner-product computation, K denote the total number of functional key queries, and P denote the total number of attributes used by the adversary. We have the following bound:*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}(1^\lambda) \leq (2nK \cdot (P(6P+3)+2)+5) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$$

where $\mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$ denotes the maximum advantage over ppt adversaries against the SXDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ set up with parameter λ.

5 Multi-Client Functional Encryption for Inner-Product with Fine-Grained Access Control via LSSS

First of all, we define and give the model of security for *multi-client functional encryption with fine-grained access control* in Section 5.1. We then present our main contribution by extending our FE scheme in Section 4 from the single-client setting to the multi-client setting in Section 5.2, for the functionality class to evaluate inner-products under access control by linear secret-sharing schemes. Theorem 14 proves its adaptive security. Finally, in Section 5.4 we discuss further our construction and revisit the MIFE regime for comparison with [ACGU20].

5.1 Definitions

We extend the notion of functional encryption with fine-grained access control to the multi-client setting. The access control is defined via a relation $\mathsf{Rel} : \mathsf{AC\text{-}K} \times \mathsf{AC\text{-}Ct}_1 \times \cdots \times \mathsf{AC\text{-}Ct}_n \to \{0,1\}$, for some sets $\mathsf{AC\text{-}Ct}_1, \ldots, \mathsf{AC\text{-}Ct}_n$ and $\mathsf{AC\text{-}K}$. A plaintext for client i consists of $(\mathsf{ac\text{-}ct}_i, x_i) \in \mathsf{AC\text{-}Ct}_i \times \mathcal{D}_\lambda$, whose corresponding ciphertext can be decrypted to $F_\lambda(x)$ using the functional key $\mathsf{sk}_{F_\lambda,\mathsf{ac\text{-}k}}$ for $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$ if and only if $\mathsf{Rel}(\mathsf{ac\text{-}k}, (\mathsf{ac\text{-}ct}_i)_i) = 1$.

$\underline{\mathsf{Setup}(1^\lambda)}$**:** Choose two pairs of dual orthogonal bases $(\mathbf{F}, \mathbf{F}^*)$ and $(\mathbf{H}, \mathbf{H}^*)$ where $(\mathbf{H}, \mathbf{H}^*)$ is a pair of bases of the dual pairing vector spaces $(\mathbb{G}_1^4, \mathbb{G}_2^4)$ $\boxed{(\mathbb{G}_1^{n+3}, \mathbb{G}_2^{n+3})}$, and $(\mathbf{F}, \mathbf{F}^*)$ are dual bases of $(\mathbb{G}_1^8, \mathbb{G}_2^8)$ $\boxed{(\mathbb{G}_1^{n+7}, \mathbb{G}_2^{n+7})}$. We write

$$\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4) \qquad \mathbf{H}^* = (\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*, \mathbf{h}_4^*)$$
$$\boxed{\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4, \ldots, \mathbf{h}_{n+3})} \qquad \boxed{\mathbf{H}^* = (\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*, \mathbf{h}_4^*, \ldots, \mathbf{h}_{n+3}^*)}$$
$$\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_6, \mathbf{f}_7, \mathbf{f}_8) \qquad \mathbf{F}^* = (\mathbf{f}_1^*, \mathbf{f}_2^*, \mathbf{f}_3^*, \mathbf{f}_4^*, \mathbf{f}_5^*, \mathbf{f}_6^*, \mathbf{f}_7^*, \mathbf{f}_8^*)$$
$$\boxed{\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \ldots, \mathbf{f}_{n+5}, \mathbf{f}_{n+6}, \mathbf{f}_{n+7})} \qquad \boxed{\mathbf{F}^* = (\mathbf{f}_1^*, \mathbf{f}_2^*, \mathbf{f}_3^*, \mathbf{f}_4^*, \ldots, \mathbf{f}_{n+5}^*, \mathbf{f}_{n+6}^*, \mathbf{f}_{n+7}^*)}$$

and sample $\mu, z \xleftarrow{\$} \mathbb{Z}_q^*, S, U \xleftarrow{\$} (\mathbb{Z}_q^*)^n$ and write $S = (s_1, \ldots, s_n)$, $U = (u_1, \ldots, u_n)$. Output the public key and the master secret key as

$$\begin{cases} \mathsf{pk} := \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu \cdot u_i]\!]_1)_{i\in[n]}\right) \\ \mathsf{msk} := (z,\ S,\ U,\ (\mathbf{f}_i^*)_{i\in[3]},\ (\mathbf{h}_i^*)_{i\in[3]}) \ . \end{cases}$$

$\underline{\mathsf{Extract}(\mathsf{msk}, \mathbb{A}, \mathbf{y} \in \mathbb{Z}_q^n)}$**:** Let $\mathbb{A}$ be an LSSS-realizable monotone access structure over a set of attributes $\mathsf{Att} \subseteq \mathbb{Z}_q$. First, sample $a_0 \xleftarrow{\$} \mathbb{Z}_q$ and run the labeling algorithm $\Lambda_{a_0}(\mathbb{A})$ (see (1)) to obtain the labels $(a_j)_j$ where j runs over the attributes in Att. In the end, it holds that $a_0 = \sum_{j\in A} c_j \cdot a_j$ where j runs over an authorized set $A \in \mathbb{A}$ and $\mathbf{c}_A = (c_j)_{j\in A}$ is the reconstruction vector from LSSS w.r.t A. We denote by $\mathsf{List\text{-}Att}(\mathbb{A})$ the list of attributes appearing in $\mathbb{A}$, with possible repetitions. Parse $\mathsf{msk} = (z,\ S,\ U,\ (\mathbf{f}_i^*)_{i\in[3]},\ (\mathbf{h}_i^*)_{i\in[3]})$. Compute:

$$\mathbf{k}_j^* := (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$
$$\boxed{\mathbf{k}_j^* := (\pi_j \cdot (j, 1),\ a_j \cdot z, \overbrace{0, \ldots, 0}^{n \text{ times}},\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})}$$
$$\mathbf{m}_i^* := [\![\mathbf{y}[i]]\!]_2 \text{ for } i \in [n]$$
$$\mathbf{k}_{\mathsf{ipfe}}^* := (\langle S, \mathbf{y}\rangle, \langle U, \mathbf{y}\rangle,\ a_0 \cdot z,\ 0)_{\mathbf{H}^*} \quad \boxed{\mathbf{k}_{\mathsf{ipfe}}^* := (\langle S, \mathbf{y}\rangle, \langle U, \mathbf{y}\rangle,\ a_0 \cdot z, \overbrace{0, \ldots, 0}^{n \text{ times}})_{\mathbf{H}^*}}$$

where $\pi_j \xleftarrow{\$} \mathbb{Z}_q$. Output $\mathsf{sk}_{\mathbb{A},\mathbf{y}} := \left((\mathbf{k}_j^*)_j, (\mathbf{m}_i^*)_{i\in[n]}, \mathbf{k}_{\mathsf{ipfe}}^*\right)$.

$\underline{\mathsf{Enc}(\mathsf{pk}, \mathbf{x}, \mathsf{S})}$**:** Parse the public key $\mathsf{pk} = \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu \cdot u_i]\!]_1)_{i\in[n]}\right)$ and $\mathsf{S} \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ as the set of attributes, then sample $\omega, \psi \xleftarrow{\$} \mathbb{Z}_q$. Compute

$$\mathbf{c}_j = \sigma_j \cdot \mathbf{f}_1 - j \cdot \sigma_j \cdot \mathbf{f}_2 + \psi \cdot \mathbf{f}_3 = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \text{ for each } j \in \mathsf{S}$$
$$\boxed{\mathbf{c}_j = (\sigma_j \cdot (1, -j),\ \psi, \overbrace{0, \ldots, 0}^{n \text{ times}}, 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \text{ for each } j \in \mathsf{S}}$$

where $\sigma_j \xleftarrow{\$} \mathbb{Z}_q$. Finally, compute

$$\mathbf{t}_i = \omega \cdot [\![s_i + \mu \cdot u_i]\!]_1 + [\![\mathbf{x}[i]]\!]_1 = [\![\omega \cdot (s_i + \mu u_i) + \mathbf{x}[i]]\!]_1 \text{ for } i \in [n]$$
$$\mathbf{c}_{\mathsf{ipfe}} = \omega \cdot (\mathbf{h}_1 + \mu\mathbf{h}_2) + \psi \cdot \mathbf{h}_3 = (\omega,\ \mu\omega,\ \psi,\ 0)_{\mathbf{H}} \quad \boxed{\mathbf{c}_{\mathsf{ipfe}} = (\omega,\ \mu\omega,\ \psi,\ \overbrace{0, \ldots, 0}^{n \text{ times}})_{\mathbf{H}}}$$

where $\sigma_i \xleftarrow{\$} \mathbb{Z}_q$ for every $i \in [n]$ and output $\mathsf{ct} := \left((\mathbf{c}_j)_{j\in\mathsf{S}}, (\mathbf{t}_i)_{i\in[n]}, \mathbf{c}_{\mathsf{ipfe}}\right)$.

$\underline{\mathsf{Dec}(\mathsf{sk}_{\mathbb{A},\mathbf{y}}, \mathsf{ct})}$**:** Parse $\mathsf{ct} = \left((\mathbf{c}_j)_{j\in\mathsf{S}},\ (\mathbf{t}_i)_{i\in[n]},\ \mathbf{c}_{\mathsf{ipfe}}\right)$ and $\mathsf{sk}_{\mathbb{A},\mathbf{y}} := \left((\mathbf{k}_j^*)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})},\ (\mathbf{m}_i^*)_{i\in[n]},\ \mathbf{k}_{\mathsf{ipfe}}^*\right)$. If there exists $A \subseteq \mathsf{S}$ and $A \in \mathbb{A}$, then compute the reconstruction vector $\mathbf{c} = (c_j)_j$ of the LSSS for A and

$$[\![\mathsf{out}]\!]_{\mathsf{t}} = \sum_{j\in A} \mathbf{c}_j \times (c_j \cdot \mathbf{k}_j^*) + \sum_{i=1}^{n} (\mathbf{e}(\mathbf{t}_i, \mathbf{m}_i^*)) - \left(\mathbf{c}_{\mathsf{ipfe}} \times \mathbf{k}_{\mathsf{ipfe}}^*\right)$$

Finally, compute the discrete logarithm and output $\mathsf{out} \in \mathbb{Z}_q$. Else, output $\perp$.

Fig. 1: The selectively-secure and $\boxed{\text{adaptively-secure}}$ single-client constructions for IPFE with fine-grained access control via LSSS. The high-level ideas can be found in the technical overview of Section 3 and more details are presented in the appendix B.3.

Definition 8 (Multi-client functional encryption with fine-grained access control). *A* multi-client functional encryption (MCFE) scheme with fine-grained access control *for the functionality class* $\mathcal{F} \times \mathsf{AC\text{-}K}$ *consists of four algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$*:*

$\mathsf{Setup}(1^\lambda)$**:** *Given as input a security parameter* λ*, output a master secret key* msk *and* $n = n(\lambda)$ *encryption keys* $(\mathsf{ek}_i)_{i\in[n]}$ *where* $n : \mathbb{N} \to \mathbb{N}$ *is a function.*

$\mathsf{Extract}(\mathsf{msk}, F_\lambda, \mathsf{ac\text{-}k})$**:** *Given* $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$*, a function description* $F_\lambda \in \mathcal{F}$*, and the master secret key* msk*, output a decryption key* $\mathsf{dk}_{F_\lambda,\mathsf{ac\text{-}k}}$*.*

$\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$**:** *Given as inputs* $\mathsf{ac\text{-}ct}_i \in \mathsf{AC\text{-}Ct}_i$*, an encryption key* ek_i*, a message* $x_i \in \mathcal{D}_\lambda$*, and a tag* tag*, output a ciphertext* $\mathsf{ct}_{\mathsf{tag},i}$*.*

$\mathsf{Dec}(\mathsf{dk}_{F_\lambda,\mathsf{ac\text{-}k}}, \mathbf{c})$**:** *Given the decryption key* $\mathsf{dk}_{F_\lambda,\mathsf{ac\text{-}k}}$ *and a vector of ciphertexts* $\mathbf{c} := (\mathsf{ct}_{\mathsf{tag},i})_i$ *of length* n*, output an element in* $\mathcal{R}_\lambda$ *or an invalid symbol* $\perp$*.*

Correctness. For sufficiently large $\lambda \in \mathbb{N}$, for all $(\mathsf{msk}, (\mathsf{ek}_i)_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(F_\lambda, \mathsf{ac\text{-}k}) \in \mathcal{F} \times \mathsf{AC\text{-}K}$ and $\mathsf{dk}_{F_\lambda,\mathsf{ac\text{-}k}} \leftarrow \mathsf{Extract}(\mathsf{msk}, F_\lambda, \mathsf{ac\text{-}k})$, for all tag and $(\mathsf{ac\text{-}ct}_i)_i$ satisfying $\mathsf{Rel}(\mathsf{ac\text{-}k}, (\mathsf{ac\text{-}ct}_i)_i) = 1$, for all $(x_i)_{i\in[n]} \in \mathcal{D}_\lambda^n$, if $F_\lambda(x_1, \ldots, x_n) \neq \perp$, the following holds with overwhelming probability:

$$\mathsf{Dec}\left(\mathsf{dk}_{F_\lambda,\mathsf{ac\text{-}k}}, (\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{ac\text{-}ct}_i))_{i\in[n]}\right) = F_\lambda(x_1, \ldots, x_n)$$

where $F_\lambda : \mathcal{D}_\lambda^n \to \mathcal{R}_\lambda$ and the probability is taken over the coins of algorithm.

Security. We define an indistinguishability-based security notion taking into account the attribute-based access control as well as the possibility of collusion among multiple clients. Below we define the *admissibility* of an adversary $\mathcal{A}$ in the security game against $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$. Intuitively, we consider only admissible adversaries who do not win our security game in a trivial manner as well as other meaningful restrictions in the multi-client setting. The admissibility additionally takes into account the satisfiability of the relation for access control, which also complicates the way we model the security notion. In the plain setting, interested readers can refer to [CDG+18a] or [LT19] for more details.

Definition 9 (Admissible adversaries). *Let* $\mathcal{A}$ *be a* ppt *adversary and let* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *be an MCFE scheme with fine-grained access control for the functionality class* $\mathcal{F} \times \mathsf{AC\text{-}K}$*. In the security game given in Figure 2 for* $\mathcal{A}$ *considering* $\mathcal{E}$*, let the sets* $(\mathcal{C}, \mathcal{Q}, \mathcal{H})$ *be the sets of corrupted clients, functional key queries, and honest clients, in that order. We say that* $\mathcal{A}$ is NOT admissible w.r.t $(\mathcal{C}, \mathcal{Q}, \mathcal{H})$ *if any of the following conditions holds:*

1. *There exist two different partial ciphertexts for* $x_i^{(b)} \neq x_i^{(b)\prime}$*, for some* $b \in \{0, 1\}$*, under one challenge tag* tag *that is queried to* **LoR***.*
2. *There exist a tag* tag *and* $i, j \in \mathcal{H}$ *such that* $i \neq j$*, there exists a query* $(i, x_i^{(0)}, x_i^{(1)}, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$ *to* **LoR** *but there exist no query* $(j, x_j^{(0)}, x_j^{(1)}, \mathsf{tag}, \mathsf{ac\text{-}ct}_j)$ *to* **LoR***.*
3. *There exists* $(\mathsf{tag}, \mathsf{ac\text{-}ct}_i)$ *for* $i \in [n]$*, a function* $F \in \mathcal{F}$*, and* $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$ *such that*
   - *We have* $\mathsf{Rel}(\mathsf{ac\text{-}k}, (\mathsf{ac\text{-}ct}_i)_i) = 1$ *and* $(F, \mathsf{ac\text{-}k}) \in \mathcal{Q}$*.*
   - *For all* $i \in \mathcal{H}$*, there exists a query* $(i, x_i^{(0)}, x_i^{(1)}, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$ *to* **LoR** *for* $(x_i^{(0)}, x_i^{(1)})$*.*
   - *For all* $i \in \mathcal{C}$*, it holds that* $x_i^{(0)} = x_i^{(1)}$*.*

Otherwise, we say that $\mathcal{A}$ is admissible w.r.t $(\mathcal{C}, \mathcal{Q}, \mathcal{H})$*.*

Remark 10. As in the plain MCFE with no attribute-based access control in [CDG+18a, LT19], we will consider security with no repetitions, i.e. the adversary cannot query **Enc** nor **LoR** for multiple ciphertexts under the same $(i, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$. Moreover, the adversary is not allowed to query the encryption oracle **Enc** for ciphertexts under the challenge tag^* that was previously queried to **LoR**. The intuition of this restriction is to prevent trivial attacks where, by querying for ciphertexts under tag^*, the adversary can combine them with the challenge ciphertext under the same tag^* to learn much more information about the challenge bit b and win the game. In addition, for every honest clients i, there must be a ciphertext query to **LoR** under the challenge

$(\mathsf{tag}, \mathsf{ac\text{-}ct}_i)$. That is, we do not take into account the scenario where only partial (in terms of honest clients) challenge ciphertext is queried to **LoR**. We can relax this condition and allow partial challenge ciphertexts by adding a layer of *All-or-Nothing Encapsulation* (AoNE). The AoNE encapsulates the partial components from clients and guarantees that all encapsulated components can be decapsulated if and only if all components are gathered, otherwise the original information remain hidden. The work by Chotard *et al.* [CDSG+20] presents constructions for AoNE in the prime-order (asymmetric) bilinear groups compatible with our current setting. In the MIFE realm, the work of [ACGU20] considers the similar restriction and expects all honest slots $i \in [n]$ are queried to **LoR**.

Remark 11. Our syntax and model of MCFE with fine-grained access control require that in order to combine the ciphertext components, they must be encrypted under the same tag and the same set of attributes. One can aim for a more flexible notion in which each client i can encrypt their ciphertext component under a different $(\mathsf{tag}, \mathsf{ac\text{-}ct}_i)$. However, this creates a much more intricate situation and we have to take into account non-trivial attacks where two different functional keys, whose policies are satisfied by different subsets of clients, may be combined to evaluate the underlying plaintext components of the union of the foregoing subsets. By hashing the tags and attributes during encryption, our concrete constructions enforce the same set of attributes embedded in the ciphertext components. In Section 5.4, we discuss how to relax the constraint and achieve the flexible notion where each client i can use a different $(\mathsf{tag}, \mathsf{ac\text{-}ct}_i)$ and hash only tag. As a result, this more flexible MCFE scheme in the RO model can be morphed into an MIFE scheme in the *standard* model by fixing a public tag and publishing its hash.

We are now ready to give the definition for the indistinguishability-based security.

Definition 12 (IND-security for MCFE with fine-grained access control). *An MCFE scheme with fine-grained access control* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the functionality class* $\mathcal{F} \times \mathsf{AC\text{-}K}$ *is* IND-secure *if for all* ppt *adversaries* $\mathcal{A}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*, the following probability is negligible*

$$\mathsf{Adv}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) := \left| \Pr[\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \right| .$$

The game $\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ *is depicted in Figure 2. The probability is taken over the random coins of* $\mathcal{A}$ *and the algorithms.*

In a more relaxed notion, the scheme $\mathcal{E}$ *is* selectively IND-secure *if the following probability is negligible*

$$\mathsf{Adv}^{\mathsf{mc\text{-}sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) := \left| \Pr[\mathsf{Expr}^{\mathsf{mc\text{-}sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \right| .$$

We also define a notion of security where only one challenge tag tag^* *is allowed. That is, the scheme* $\mathcal{E}$ *is* one-time IND-secure *if the following probability is negligible*

$$\mathsf{Adv}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) := \left| \Pr[\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \right| .$$

Lemma 13 allows us to concentrate on the notion of one-time IND-security for our construction. The proof is a standard hybrid argument and we give it in the appendix B.1 for completeness.

Lemma 13. *Let* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the function class* $\mathcal{F} \times \mathsf{AC\text{-}K}$ *be an MCFE scheme with fine-grained access control. If* $\mathcal{E}$ *is one-time IND-secure, then* $\mathcal{E}$ *is IND-secure.*

Initialise(1^λ) | **Initialise**$(1^\lambda, (x_i^{(0)}, x_i^{(1)})_{i\in[n]})$

$b \xleftarrow{\$} \{0,1\}$
$(\mathsf{msk}, (\mathsf{ek}_i)_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathcal{Q} := \varnothing,\ \mathcal{C} := \varnothing,\ \mathcal{H} := [n]$
Return pk

LoR$(i, x_i^{(0)}, x_i^{(1)}, \mathsf{tag}^*, \mathsf{ac\text{-}ct}_i^*)$ | **LoR**$(i, \mathsf{tag}^*, \mathsf{ac\text{-}ct}_i^*)$

If $(i, \mathsf{tag}^*, \mathsf{ac\text{-}ct}_i^*)$ appears previously:
or another $(i, \mathsf{tag}', \mathsf{ac\text{-}ct}_i')$ was queried:
  Ignore
Else:$\mathsf{Enc}(\mathsf{ek}_i, x_i^{(b)}, \mathsf{tag}^*, \mathsf{ac\text{-}ct}_i^*) \to \mathsf{ct}_{\mathsf{tag}^*,i}^{(b)}$
Return $\mathsf{ct}_{\mathsf{tag}^*,i}^{(b)}$

Enc$(i, x_i, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$

If $(i, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$ appears previously
or $\mathsf{tag} = \mathsf{tag}^*$:
  Ignore
Else: return $\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{ac\text{-}ct}_i)$

Corrupt(i)

$\mathcal{C} := \mathcal{C} \cup \{i\}$
$\mathcal{H} := \mathcal{H} \setminus \{i\}$
Return ek_i

Finalise(b')

If $\mathcal{A}$ is NOT admissible w.r.t $(\mathcal{C}, \mathcal{Q}, \mathcal{H})$:
  return 0
Else return $\left(b' \stackrel{?}{=} b\right)$

Extract$(F, \mathsf{ac\text{-}k})$

$\mathcal{Q} := \mathcal{Q} \cup \{(F, \mathsf{ac\text{-}k})\}$
$\mathsf{dk}_{F,\mathsf{ac\text{-}k}} \leftarrow \mathsf{Extract}(\mathsf{msk}, F, \mathsf{ac\text{-}k})$
Return $\mathsf{dk}_{F,\mathsf{ac\text{-}k}}$

Fig. 2: The security games $\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$, $\mathsf{Expr}^{\mathsf{mc\text{-}sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ and $\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ for Definition 12

5.2 Construction

This section presents a multi-client FE scheme with fine-grained access control, as defined in Section 5.1. We are in the bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$ and $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}$ are written additively. In our concrete construction, the functionality class of interests is $\mathcal{F}^{\mathsf{IP}} \times \mathsf{LSSS}$ and $\mathcal{F}^{\mathsf{IP}}$ contains $F_\mathbf{y} : (\mathbb{Z}_q^*)^n \to \mathbb{Z}_q$ that is defined as $F_\mathbf{y}(\mathbf{x}) := \langle \mathbf{x}, \mathbf{y} \rangle$. The access control is given by $\mathsf{Rel} : \mathsf{LSSS} \times \left(\prod_{i=1}^n 2^{\mathsf{Att}}\right) \to \{0,1\}$, where $\mathsf{Rel}(\mathbb{A}, (\mathsf{S}_i)_i) = \prod_i \mathbb{A}(\mathsf{S}_i)$, the class LSSS contains Linear Secret Sharing Schemes over Att, and 2^{Att} denotes the superset of an attribute space $\mathsf{Att} \subseteq \mathbb{Z}_q$. Our constructions are key-policy, where $\mathbb{A}$ is embedded in the key and S is specified in the ciphertext. The tag space Tag contains the tags that accompany plaintext components at the time of encryption.

We also need a full domain hash function $\mathsf{H} : \mathsf{Tag} \times 2^{\mathsf{Att}} \to \mathbb{G}_1^2$, where Tag denotes the set of tags and 2^{Att} contains the subsets of attributes of Att. The details of our construction is given in Figure 3. We remark that currently all clients $i \in [n]$ must use the same S for encrypting their inputs x_i, because S is hashed together with tag by H. Section 5.4 presents another construction that relaxes the matching condition on S and H then receives only tag as inputs. We note that the *duplicate-and-compress* technique is used by putting the vectors $\{(\mathbf{c}_{i,j}, \mathbf{k}_{i,j})_j\}$ in the same pair of dual bases $(\mathbf{F}, \mathbf{F}^*)$ for all client $i \in [n]$, meanwhile each pair of vectors $(\mathbf{c}_{i,\mathsf{ipfe}}, \mathbf{k}_{i,\mathsf{ipfe}})$ is put in bases $(\mathbf{H}_i, \mathbf{H}_i^*)$ for each client $i \in [n]$. In the proof of Theorem 14 we detail how the basis changes in Lemma 4 can be done in parallel for $(\mathbf{H}_i, \mathbf{H}_i^*), (\mathbf{F}, \mathbf{F}^*)$ for all $i \in [n]$. The *correctness* of the scheme is verified by:

$$
\begin{aligned}
[\![\mathsf{out}]\!]_\mathsf{t} &= \sum_{i=1}^n \left(\left(\sum_{j\in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}) \right) - (\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}}) + \mathbf{e}(\mathbf{t}_i, \mathbf{m}_i) \right) \\
&= \sum_{i=1}^n \Big([\![\psi_i a_0 z]\!]_\mathsf{t} - [\![\omega p_i \cdot \langle S, \mathbf{y} \rangle + \omega' p_i \cdot \langle U, \mathbf{y} \rangle + \psi_i a_0 z]\!]_\mathsf{t} \\
&\quad + [\![(\omega s_i + \omega' u_i + x_i) y_i]\!]_\mathsf{t} \Big) = [\![\langle \mathbf{x}, \mathbf{y} \rangle]\!]_\mathsf{t}
\end{aligned}
$$

Setup(1^λ): Choose $n+1$ pairs of dual orthogonal bases $(\mathbf{H}_i, \mathbf{H}_i^*)$ for $i \in [n]$ and $(\mathbf{F}, \mathbf{F}^*)$where $(\mathbf{H}_i, \mathbf{H}_i^*)$ is a pair of dual bases for $(\mathbb{G}_1^4, \mathbb{G}_2^4)$ and $(\mathbf{F}, \mathbf{F}^*)$ is a pair of dual bases for $(\mathbb{G}_1^8, \mathbb{G}_2^8)$. We denote the basis changing matrices for $(\mathbf{F}, \mathbf{F}^*), (\mathbf{H}, \mathbf{H}_i^*)$ as $(F, F' \coloneqq (F^{-1})^\top), (H_i, H_i' \coloneqq (H_i^{-1})^\top)$ respectively (see the appendix A.2 for basis changes in DPVS):

$$(\mathbf{H}_i = H_i \cdot \mathbf{T};\ \mathbf{H}_i^* = H_i' \cdot \mathbf{T}^*)_{i\in[n]} \quad (\mathbf{F} = F \cdot \mathbf{W};\ \mathbf{F}^* = F' \cdot \mathbf{W}^*)$$

where $H_i, H_i' \in \mathbb{Z}_q^{4\times 4}, F, F' \in \mathbb{Z}_q^{8\times 8}$ and $(\mathbf{T} = [\![I_4]\!]_1, \mathbf{T}^* = [\![I_4]\!]_2)$, $(\mathbf{W} = [\![I_8]\!]_1, \mathbf{W}^* = [\![I_8]\!]_2)$ are canonical bases of $(\mathbb{G}_1^4, \mathbb{G}_2^4)$, $(\mathbb{G}_1^8, \mathbb{G}_2^8)$ respectively, for identity matrices I_4 and I_8. We recall that in the multi-client setting the scheme must be a private key encryption scheme. For each $i \in [n]$, we write

$$\mathbf{H}_i = (\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \mathbf{h}_{i,3}, \mathbf{h}_{i,4}) \qquad \mathbf{H}_i^* = (\mathbf{h}_{i,1}^*, \mathbf{h}_{i,2}^*, \mathbf{h}_{i,3}^*, \mathbf{h}_{i,4}^*)$$
$$\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_6, \mathbf{f}_7, \mathbf{f}_8) \qquad \mathbf{F}^* = (\mathbf{f}_1^*, \mathbf{f}_2^*, \mathbf{f}_3^*, \mathbf{f}_4^*, \mathbf{f}_5^*, \mathbf{f}_6^*, \mathbf{f}_7^*, \mathbf{f}_8^*)$$

and sample $\mu \xleftarrow{\$} \mathbb{Z}_q^*, S, U, \xleftarrow{\$} (\mathbb{Z}_q^*)^n$ and write $S = (s_1, \ldots, s_n)$, $U = (u_1, \ldots, u_n)$. Perform an n-out-of-n secret sharing on 1, that is, choose $p_i \in \mathbb{Z}_q$ such that $1 = p_1 + \cdots + p_n$. Output the master secret key and the encryption keys as

$$\begin{cases} \mathsf{msk} \coloneqq (S,\ U,\ \mathbf{f}_1^*,\ \mathbf{f}_2^*,\ \mathbf{f}_3^*,\ (\mathbf{h}_{i,1}^*, \mathbf{h}_{i,2}^*, \mathbf{h}_{i,3}^*)_{i\in[n]}) \\ \mathsf{ek}_i \coloneqq (s_i,\ u_i,\ p_i \cdot H_i^{(1)},\ p_i \cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3) \text{ for } i \in [n] \end{cases}$$

where $H_i^{(k)}$ denotes the k-th row of H_i.

Extract($\mathsf{msk}, \mathbb{A}, \mathbf{y} \in \mathbb{Z}_q^n$): Let $\mathbb{A}$ be an LSSS-realizable monotone access structure over a set of attributes $\mathsf{Att} \subseteq \mathbb{Z}_q$. First, sample $a_0 \xleftarrow{\$} \mathbb{Z}_q$ and run the labeling algorithm $\Lambda_{a_0}(\mathbb{A})$ (see Definition 1) to obtain the labels $(a_j)_j$ where j runs over the attributes in Att. In the end, it holds that $a_0 = \sum_{j\in A} c_j \cdot a_j$ where j runs over an authorized set $A \in \mathbb{A}$ and $\mathbf{c} = (c_j)_j$ is the reconstruction vector from LSSS w.r.t A. We denote by $\mathsf{List\text{-}Att}(\mathbb{A})$ the list of attributes appearing in $\mathbb{A}$, with possible repetitions. Parse $\mathsf{msk} = (S,\ U,\ \mathbf{f}_1^*,\ \mathbf{f}_2^*,\ \mathbf{f}_3^*,\ (\mathbf{h}_{i,1}^*, \mathbf{h}_{i,2}^*, \mathbf{h}_{i,3}^*)_{i\in[n]})$ and write $\mathbf{y} = (y_1, \ldots, y_n)$. For each $i \in [n]$, compute $\mathbf{m}_i \coloneqq [\![y_i]\!]_2$ and

$$\mathbf{k}_{i,j} = (\pi_{i,j} \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$
$$\mathbf{k}_{i,\mathsf{ipfe}} \coloneqq (\langle S, \mathbf{y}\rangle, \langle U, \mathbf{y}\rangle,\ a_0 \cdot z,\ 0)_{\mathbf{H}_i^*}$$

where $z, \pi_{i,j} \xleftarrow{\$} \mathbb{Z}_q$. Output $\mathsf{dk}_{\mathbb{A},\mathbf{y}} \coloneqq \left((\mathbf{k}_{i,j})_{i,j}, (\mathbf{m}_i, \mathbf{k}_{i,\mathsf{ipfe}})_{i\in[n]}\right)$.

Enc($\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{S}$): Parse $\mathsf{ek}_i \coloneqq (s_i,\ u_i,\ p_i \cdot H_i^{(1)},\ p_i \cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3)$ and $\mathsf{S} \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ as the set of attributes, compute $\mathsf{H}(\mathsf{tag}, \mathsf{S}) \to ([\![\omega]\!]_1, [\![\omega']\!]_1) \in \mathbb{G}_1^2$ and sample $\psi_i \xleftarrow{\$} \mathbb{Z}_q$. Use $p_i H_i^{(1)}$ and $p_i H_i^{(2)}$ to compute

$$p_i H_i^{(1)} \cdot [\![\omega]\!]_1 + p_i H_i^{(2)} \cdot [\![\omega']\!]_1 = p_i \cdot \left(\omega H_i^{(1)} \cdot g_1 + \omega' H_i^{(2)} \cdot g_1\right) = p_i \cdot (\omega \mathbf{h}_{i,1} + \omega' \mathbf{h}_{i,2}) \ .$$

For each $j \in \mathsf{S}$, compute

$$\mathbf{c}_{i,j} = \sigma_{i,j} \cdot \mathbf{f}_1 - j \cdot \sigma_{i,j} \cdot \mathbf{f}_2 + \psi_i \cdot \mathbf{f}_3 = (\sigma_{i,j} \cdot (1, -j),\ \psi_i,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}}$$

where $\sigma_{i,j} \xleftarrow{\$} \mathbb{Z}_q$. Finally, compute

$$\mathbf{t}_i = s_i \cdot [\![\omega]\!]_1 + u_i \cdot [\![\omega']\!]_1 + [\![x_i]\!]_1 = [\![\omega \cdot s_i + \omega' \cdot u_i + x_i]\!]_1$$
$$\mathbf{c}_{i,\mathsf{ipfe}} = p_i \cdot (\omega \cdot \mathbf{h}_{i,1} + \omega' \cdot \mathbf{h}_{i,2}) + \psi_i \cdot \mathbf{h}_{i,3} = (\omega p_i,\ \omega' p_i,\ \psi_i,\ 0)_{\mathbf{H}_i}$$

and output $\mathsf{ct}_{\mathsf{tag},i} \coloneqq \left((\mathbf{c}_{i,j})_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}}\right)$.

Dec($\mathsf{dk}_{\mathbb{A},\mathbf{y}}, \mathbf{c} \coloneqq (\mathsf{ct}_{\mathsf{tag},i})$): Parse $\mathsf{ct}_{\mathsf{tag},i} = ((\mathbf{c}_{i,j})_{j\in\mathsf{S}}, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}})$ and $\mathsf{dk}_{\mathbb{A},\mathbf{y}} \coloneqq ((\mathbf{k}_{i,j})_{i\in[n], j\in\mathsf{List\text{-}Att}(\mathbb{A})}, (\mathbf{m}_i, \mathbf{k}_{i,\mathsf{ipfe}})_{i\in[n]})$. If there exists $A \subseteq \mathsf{S}$ and $A \in \mathbb{A}$, then compute the reconstruction vector $\mathbf{c} = (c_j)_j$ of the LSSS for A and

$$[\![\mathsf{out}]\!]_\mathsf{t} = \sum_{i=1}^{n} \left(\left(\sum_{j\in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j})\right) - (\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}}) + \mathbf{e}(\mathbf{t}_i, \mathbf{m}_i)\right)$$

Finally, compute the discrete logarithm and output the small value out.

Fig. 3: The construction for multi-client IPFE with fine-grained access control via LSSS from Section 5.2.

5.3 Adaptive Security

We now present the main ideas of the adaptive proof for the multi-client construction described in Section 5.2, the detailed proof is presented in the appendix B.4. A high-level intuition can be revisited in Section 3.

Theorem 14. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be a multi-client IPFE scheme with fine-grained access control via LSSS for the functionality class $\mathcal{F}^{\mathsf{IP}} \times \mathsf{LSSS}$, constructed in Section 5.2 in a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$. Then, $\mathcal{E}$ is one-time IND-secure if the SXDH assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$. More specifically, for $\lambda \in \mathbb{Z}$ and for any adversary $\mathcal{A}$, let K denote the total number of functional key queries, P denote the total number of attributes used by $\mathcal{A}$, and Q denote the maximum number of random oracle (RO) queries. We have the following bound:*

$$\mathsf{Adv}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}(1^\lambda) \leq (2KP \cdot (6P+3) + 2K + 2Q + 5) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$$

where $\mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$ denotes the maximum advantage over ppt adversaries against the SXDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ set up with parameter λ.

By combining with Lemma 13, we have the following Corollary:

Corollary 15. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be a multi-client IPFE scheme with fine-grained access control via LSSS, for the functionality class $\mathcal{F}^{\mathsf{IP}} \times \mathsf{LSSS}$, constructed in Section 5.2 in a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$. Then, $\mathcal{E}$ is IND-secure if the SXDH assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Proof (of Theorem 14 - Main ideas). Recall that in the security proof for single-client adaptive security (Theorem 7) we switch the ℓ-th functional key to semi-functional by augmenting the dimension of the dual bases so that the challenge ciphertext is masked by $\tau\Delta\mathbf{x}[i]$, facing the mask $r_0^{(\ell)}\mathbf{y}^{(\ell)}[i]$ in the corresponding coordinate of the ℓ-th key and $\tau, r_0^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$ where $\Delta\mathbf{x} := \mathbf{x}_1^* - \mathbf{x}_0^*$. Afterwards, when doing the product of vectors in the dual bases, there will exist the quantity $\sum_{i=1}^n \tau r_0^{(\ell)} \Delta\mathbf{x}[i]\mathbf{y}^{(\ell)}[i] = \tau r_0^{(\ell)} \langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle$, which is non-zero when $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \neq 0$. The dual bases now must have dimension at least n in order to accommodate all the n terms $\Delta\mathbf{x}[i]\mathbf{y}[i]$. However, in the multi-client setting, we are already using n different dual basis pairs $(\mathbf{H}_i, \mathbf{H}_i^*)$ for n clients and the correctness of the construction in Section 5.2 makes sure that only when gathering all n ciphertext parts can we decrypt to obtain the inner product. Therefore, it suffices to introduce only $\tau_i\Delta\mathbf{x}[i]$ in the component $\mathbf{c}_{i,\mathsf{ipfe}}$ returned from $\mathbf{LoR}$ of client i and only $r_{i,0}^{(\ell)}\mathbf{y}^{(\ell)}[i]$ in the corresponding key component $\mathbf{k}^*_{i,\mathsf{ipfe}}$, while duplicating the pair of bases $(\mathbf{H}_i, \mathbf{H}_i^*)$ for each $i \in [n]$. Indeed, this is also the best we can do because a client i is not supposed to know other inputs $\mathbf{x}_b^*[j]$ of other clients j, where $b \xleftarrow{\$} \{0,1\}$ is the challenge bit. At the same time, we compress the components of the access control part $(\mathbf{c}_{i,j})_j, (\mathbf{k}^*_{i,j})_j$ into the same pair of bases $(\mathbf{F}, \mathbf{F}^*)$ for all clients i. We refer to the introduction for more intuition on this duplicate-and-compress process.

There are some further technical tweaks to be done when applying Lemma 4. First of all, we need the factors $\tau_i, r_{i,0}^{(\ell)}$ to be the same, for the grouping later when doing products of vectors in DPVS. This can be done by using the same $\tau_i = \tau$ for all i and during the basis change to mask the ciphertext component there will be a factor $\Delta\mathbf{x}[i]$. Our argument to introduce $r_{i,0}^{(\ell)}$ in fact does not depend on i and therefore we can use the same $r_{i,0}^{(\ell)} = r_0^{(\ell)}$ for all i as well. One might wonder if the dependence of the masks still relies on $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle$ because the adversary is not supposed to query $\mathbf{LoR}$ for corrupted clients and we can only introduce the masks in the vector components of honest i. As a result, the product of vectors in the dual bases in the end will have $\sum_{i \in \mathcal{H}} \tau r_0^{(\ell)} \Delta\mathbf{x}[i]\mathbf{y}^{(\ell)}[i]$. However, the security model imposes that for all corrupted i, the challenge message satisfies $\mathbf{x}_1^*[i] = \mathbf{x}_0^*[i]$ and consequently, $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$ if and only if $\sum_{i \in \mathcal{H}} \Delta\mathbf{x}[i]\mathbf{y}^{(\ell)}[i] = 0$. This implies that the mask $\tau r_0^{(\ell)} \sum_{i \in \mathcal{H}} \Delta\mathbf{x}[i]\mathbf{y}^{(\ell)}[i]$ persists only when $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \neq 0$, which is our goal. The masking of ciphertext and key components

results from the application of Lemma 4 as we are in the adaptive setting and not knowing what policy the ciphertext's attributes will satisfy. The lemma will mask all vectors $\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$ with $a_0'^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, using which we perform a random labeling, and under the constraint that all clients i use the same S, the mask $a_0'^{(\ell)}$ will either appear for all i or neither. This enables us to replace it with $r_0^{(\ell)}$, similarly to the *all-at-once-changing* step in the adaptive single-client proof. We recall that currently the constraint on using the same S for all i is guaranteed by hashing $(\mathsf{tag}, \mathsf{S})$ together. The more complicated and flexible case with possibly different S_i for each i is discussed in Section 5.4. The application of Lemma 4 needs some auxiliary vectors in the dual bases $(\mathbf{F}, \mathbf{F}^*)$, which are not needed in the real usage of the scheme. Following the terminology of Okamoto-Takashima [OT12b], those auxiliary vectors form a *hidden* part of the bases.

The final steps are to change (s_i, u_i) in the challenge ciphertext to (s_i', u_i') so that the ciphertext from $\mathbf{LoR}$ is encrypting $\mathbf{x}_0^*$ instead of $\mathbf{x}_b^*$ by solving a linear system for $(\Delta S, \Delta U)$ depending on $\mathbf{x}_b^* - \mathbf{x}_0^*$. We stress that the simulation of corrupted keys can still be done using (s_i, u_i) regardless of the order of $\mathbf{LoR}$ query, under the admissibility from condition 3 in Definition 9 that requires $\Delta\mathbf{x}[i] = \mathbf{x}_1^*[i] - \mathbf{x}_0^*[i] = 0$ if i is corrupted.

In the case of $\langle \Delta\mathbf{x}, \mathbf{y} \rangle \neq 0$, which then implies $\mathbb{A}(\mathsf{S}) \neq 0$, the functional key queries that are simulated using $(\langle S, \mathbf{y} \rangle, \langle U, \mathbf{y} \rangle)$ are computaionally indistinguishable from the ones in correct forms using $(\langle S', \mathbf{y} \rangle, \langle U', \mathbf{y} \rangle)$, under the SXDH assumption. However, the situation is more complicated than the single-client construction because the oracle $\mathbf{Enc}$ is using (s_i, u_i) as well. In order to be able to perform the correction step on the functional key, we have to program the full-domain hash function, which is modeled as an RO, such that for all queries $(\mathsf{tag}', \mathsf{S}')$ different from the challenge $(\mathsf{tag}, \mathsf{S})$, the value $\mathsf{H}(\mathsf{tag}', \mathsf{S}')$ belongs to $\mathrm{span}([\![(1,\ \mu)]\!]_1) \subseteq \mathbb{G}_1^2$, for $\mu \xleftarrow{\$} \mathbb{Z}_q$. For the challenge $(\mathsf{tag}, \mathsf{S})$, the value $\mathsf{H}(\mathsf{tag}, \mathsf{S})$ remains a pair of random group elements. The main reason behind this is that our correction step requires $\mathsf{H}(\mathsf{tag}', \mathsf{S}')$ belongs to $\mathrm{span}([\![(1,\ \mu)]\!]_1)$ so that it will not affect the normal ciphertext returned from $\mathbf{Enc}$. This implies a linear relation between $\Delta S := S' - S$ and $\Delta U := U' - U$. However, if we put $\mathsf{H}(\mathsf{tag}, \mathsf{S})$ on the line $\mathrm{span}([\![(1,\ \mu)]\!]_1)$ as well, then the intention to switch from $\mathbf{x}_0^*$ to $\mathbf{x}_b^*$ in the ciphertext from $\mathbf{LoR}$ will create another linear relation, which reduces significantly the degree of freedom to choose $(\Delta S, \Delta U)$ in order to make the simulation successful. In the end, the challenge ciphertext no longer depends on b and the advantage becomes 0, concluding the proof. □

5.4 Revisiting MIFE in the Standard Model

We recall that currently our MCFE scheme from Section 5.2 enforces the same $(\mathsf{tag}, \mathsf{S})$ when encrypting for all client $i \in [n]$, by hashing them using the full-domain hash function that is modeled as an RO in the security proof. In practice, this could render a significant cost for synchronisation among clients so as to agree on tag *and* the attributes S at the time of encryption. In addition, by fixing one public tag, one can only obtain an MIFE scheme whose security can be proven in the ROM because we still need the random oracle to process S.

If we allow different $(\mathsf{tag}, \mathsf{S}_i)$ for each client i and during encryption the input for hashing depends only on tag, i.e. $[\![(\omega_{\mathsf{tag}}, \omega'_{\mathsf{tag}})]\!]_1 \leftarrow \mathsf{H}(\mathsf{tag})$, there is a mix-and-match attack among functional keys that has to be considered. More precisely, suppose for two clients $\{1, 2\}$ encrypting $\mathbf{x} = (x_1, x_2)$ under different sets $(\mathsf{S}_1, \mathsf{S}_2)$ of attributes, the ℓ-th and ℓ'-th key queries have access structures $\mathbb{A}$ and $\mathbb{A}'$ where $\mathbb{A}(\mathsf{S}_1) = \mathbb{A}'(\mathsf{S}_2) = 1$ and $\mathbb{A}'(\mathsf{S}_1) = \mathbb{A}(\mathsf{S}_2) = 0$, for the same inner-product with $\mathbf{y} = \mathbf{y}' = (y_1, y_2)$. Neither of these keys should decrypt $x_1y_1 + x_2y_2$ for the sake of security. However, the construction from Figure 3 permits an adversary to use the vectors $\{(\mathbf{c}_{1,j})_j, (\mathbf{k}_{1,j})_j, \mathbf{c}_{1,\mathsf{ipfe}}, \mathbf{k}_{1,\mathsf{ipfe}}\}$ to recover $p_1\omega_{\mathsf{tag}}\langle S, \mathbf{y} \rangle + p_1\omega'_{\mathsf{tag}}\langle U, \mathbf{y} \rangle$. Similar computation allows the same adversary to obtain $p_2\omega_{\mathsf{tag}}\langle S, \mathbf{y} \rangle + p_2\omega'_{\mathsf{tag}}\langle U, \mathbf{y} \rangle$ using $\{(\mathbf{c}_{2,j})_j, (\mathbf{k}_{2,j})_j, \mathbf{c}_{2,\mathsf{ipfe}}, \mathbf{k}_{2,\mathsf{ipfe}}\}$. Finally, observing that $p_1 + p_2 = 1$, exploiting the linear combination $y_1 \cdot [\![\omega_{\mathsf{tag}} s_1 + \omega'_{\mathsf{tag}} u_1 + x_1]\!]_1 + y_2 \cdot [\![\omega_{\mathsf{tag}} s_2 + \omega'_{\mathsf{tag}} u_2 + x_2]\!]_1$ permits finding $\langle \mathbf{x}, \mathbf{y} \rangle$. This demonstrates the main reason why we put S as part of the input to the hash function H in our current scheme. The core of the above

problem is the fact that the construction from Section 5.2 does not prohibit combining different "root" vectors $\mathbf{k}_{1,\mathsf{ipfe}}$ and $\mathbf{k}_{2,\mathsf{ipfe}}$ w.r.t different access structure $\mathbb{A}$ and $\mathbb{A}'$.

In this section we present a solution, with minimal modifications to the scheme, to overcome the need for hashing S. Suppose now we are in the more flexible setting where $[\![(\omega_{\mathsf{tag}}, \omega'_{\mathsf{tag}})]\!]_1 \leftarrow \mathsf{H}(\mathsf{tag})$ during encryption. During setup phase, the pair $(\mathbf{H}_i, \mathbf{H}_i^*)$ is a pair of dual bases for $(\mathbb{G}_1^5, \mathbb{G}_2^5)$, with one more dimension compared to our less flexible construction. The master secret key msk stays the same, while the encryption key ek_i now contains furthermore $\theta_i \mathbf{h}_{i,5}$ for some $\theta_i \xleftarrow{\$} \mathbb{Z}_q$. Given an LSSS-realizable monotone access structure $\mathbb{A}$, the key extraction $\mathsf{Extract}(\mathsf{msk}, \mathbb{A}, \mathbf{y} \in \mathbb{Z}_q^n)$ returns $\mathsf{dk}_{\mathbb{A},\mathbf{y}} := ((\mathbf{k}_{i,j})_{i,j}, (\mathbf{m}_i, \mathbf{k}_{i,\mathsf{ipfe}})_{i\in[n]})$. The encryption $\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{S}_i)$ returns $\mathsf{ct}_{\mathsf{tag},i} := ((\mathbf{c}_{i,j})_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}})$ for each $i \in [n]$. There is a new element $d_{\mathbb{A},i}$ appearing in the extra coordinate in $\mathbf{k}_{i,\mathsf{ipfe}}$ for every $i \in [n]$, where $(d_{\mathbb{A},i})_i$ satisfies $\sum_{i=1}^n \theta_i d_{\mathbb{A},i} = 0$, independently chosen for each functional keys. The vectors are essentially the same as in Figure 3, except $(\mathbf{c}_{i,\mathsf{ipfe}}, \mathbf{k}_{i,\mathsf{ipfe}})$ for each i as follows:

$$
\begin{aligned}
\mathsf{ek}_i &:= (s_i,\ u_i,\ p_i \cdot H_i^{(1)},\ p_i \cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ \theta_i \mathbf{h}_{i,5},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3) \\
\mathsf{msk} &:= (S,\ U,\ (\theta_i)_i,\ \mathbf{f}_1^*,\ \mathbf{f}_2^*,\ \mathbf{f}_3^*,\ (\mathbf{h}_{i,1}^*, \mathbf{h}_{i,2}^*, \mathbf{h}_{i,3}^*)_{i\in[n]}) \\
\mathbf{c}_{i,\mathsf{ipfe}} &:= (\omega_{\mathsf{tag}} p_i,\ \omega'_{\mathsf{tag}} p_i,\ \psi_i,\ 0,\ \theta_i)_{\mathbf{H}_i} \\
\mathbf{k}_{i,\mathsf{ipfe}} &:= (\langle S, \mathbf{y}\rangle, \langle U, \mathbf{y}\rangle,\ a_{i,0} \cdot z,\ 0,\ d_{\mathbb{A},i})_{\mathbf{H}_i^*}
\end{aligned}
$$

The decryption calculation stays invariant because $\sum_{i=1}^n \theta_i d_{\mathbb{A},i} = 0$. In retrospection, the mix-and-match attack we gave at the beginning of this section no longer works, because $\mathbb{A} \neq \mathbb{A}'$ and $\theta_1 d_{\mathbb{A},1} + \theta_2 d_{\mathbb{A}',2} = 0$ only with negligible probability over the choices of $\theta_1, \theta_2, d_{\mathbb{A},1}, d_{\mathbb{A}',2} \xleftarrow{\$} \mathbb{Z}_q$, for two independent random families $(d_{\mathbb{A},i})_{i\in[2]}$ and $(d_{\mathbb{A}',i})_{i\in[2]}$. More formally, the security proof for this modified scheme, where we exploit the one extra 5-th coordinate in $(\mathbf{H}_i, \mathbf{H}_i^*)$, can be obtained with recourse to the proof of theorem 14 in section 5.2 under few changes. We sketch the proof and highlight the main differences compared to the less flexible scheme in the appendix B.5.

Remark 16. Adding this new layer of masking increases the ciphertext's size by only a factor linear in n. Moreover, given this new construction where the set of attributes does not involve in the computation of the full-domain hashing anymore, we can obtain an MIFE in the standard model by fixing one tag for every ciphertext. The random oracle can be removed by publishing a random fixed value corresponding to $\mathsf{H}(\mathsf{tag})$ for encryption. In the end, we obtain an attribute-based MIFE for inner-products with adaptive security in the standard model, where the adversary can make the challenge query to **LoR** at most once for each slot $i \in [n]$. To achieve security w.r.t multiple queries for same slot, we can apply the technique in [CDG+18b] to enhance our construction with repetitions. Finally, we can apply a layer of All-or-Nothing Encapsulation to the ciphertext components of construction in Section 5.4, so as to remove the tradeoff with respect to partial challenge ciphertexts in case of $(\mathsf{tag}, \mathsf{S}_i)$ for different S_i.

Acknowledgments

We thank Romain Gay for insightful discussions regarding their constructions in [ACGU20]. This work was supported in part by the European Union Horizon 2020 ERC Programme (Grant Agreement no. 966570 – CryptAnalytics), the Beyond5G project and the French ANR Project ANR-19-CE39-0011 PRESTO.

References

ABDP15. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 733–751. Springer, Heidelberg, March / April 2015.

ABDP16. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, Report 2016/011, 2016. https://eprint.iacr.org/2016/011.

ABN10. Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 480–497. Springer, Heidelberg, February 2010.

ACGU20. Michel Abdalla, Dario Catalano, Romain Gay, and Bogdan Ursu. Inner-product functional encryption with fine-grained access control. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 467–497. Springer, Heidelberg, December 2020.

AKM+22. Shweta Agrawal, Fuyuki Kitagawa, Anuja Modi, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Bounded functional encryption for turing machines: Adaptive security from general assumptions. Cryptology ePrint Archive, Report 2022/316, 2022. https://ia.cr/2022/316.

ALdP11. Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 90–108. Springer, Heidelberg, March 2011.

ALS16. Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 333–362. Springer, Heidelberg, August 2016.

AMVY21. Shweta Agrawal, Monosij Maitra, Narasimha Sai Vempati, and Shota Yamada. Functional encryption for turing machines with dynamic bounded collusion from LWE. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 239–269, Virtual Event, August 2021. Springer, Heidelberg.

AS17. Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 152–181. Springer, Heidelberg, April / May 2017.

BBL17. Fabrice Benhamouda, Florian Bourse, and Helger Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 36–66. Springer, Heidelberg, March 2017.

BCFG17. Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 67–98. Springer, Heidelberg, August 2017.

Bei96. Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Technion - Israel Institute of Technology, Haifa, Israel, June 1996. https://www.cs.bgu.ac.il/~beimel/Papers/thesis.pdf.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

BGH07. Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th FOCS*, pages 647–657. IEEE Computer Society Press, October 2007.

BL90. Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 27–35. Springer, Heidelberg, August 1990.

BO13. Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *CANS 13*, volume 8257 of *LNCS*, pages 218–234. Springer, Heidelberg, November 2013.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

CDG+18a. Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized multi-client functional encryption for inner product. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 703–732. Springer, Heidelberg, December 2018.

CDG+18b. Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Multi-client functional encryption with repetition for inner product. Cryptology ePrint Archive, Report 2018/1021, 2018. https://eprint.iacr.org/2018/1021.

CDSG+20. Jérémy Chotard, Edouard Dufour-Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Dynamic decentralized functional encryption. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 747–775. Springer, Heidelberg, August 2020.

CLL+13. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. In Michel Abdalla and Tanja Lange, editors, *PAIRING 2012*, volume 7708 of *LNCS*, pages 122–140. Springer, Heidelberg, May 2013.

CLT18. Guilhem Castagnos, Fabien Laguillaumie, and Ida Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 733–764. Springer, Heidelberg, December 2018.

Coc01. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, Heidelberg, December 2001.

DGP21. Cécile Delerablée, Lénaïck Gouriou, and David Pointcheval. Key-policy abe with delegation of rights. Cryptology ePrint Archive, Report 2021/867, 2021. `https://ia.cr/2021/867`.

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

Gay20. Romain Gay. A new paradigm for public-key functional encryption for degree-2 polynomials. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 95–120. Springer, Heidelberg, May 2020.

GGG+14. Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 578–602. Springer, Heidelberg, May 2014.

GKL+13. S. Dov Gordon, Jonathan Katz, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. Cryptology ePrint Archive, Report 2013/774, 2013. `https://eprint.iacr.org/2013/774`.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

Lin17. Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 599–629. Springer, Heidelberg, August 2017.

LLW21. Qiqi Lai, Feng-Hao Liu, and Zhedong Wang. New lattice two-stage sampling technique and its applications to functional encryption - stronger security and smaller ciphertexts. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 498–527. Springer, Heidelberg, October 2021.

LT19. Benoît Libert and Radu Titiu. Multi-client functional encryption for linear functions in the standard model from LWE. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 520–551. Springer, Heidelberg, December 2019.

LW10. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Heidelberg, February 2010.

OSW07. Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 195–203. ACM Press, October 2007.

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Heidelberg, August 2010.

OT12a. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 591–608. Springer, Heidelberg, April 2012.

OT12b. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, December 2012.

PD21. Tapas Pal and Ratna Dutta. Attribute-based access control for inner product functional encryption from LWE. In *LATIN 2021*, 2021.

Sha79. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, August 2009.

Wee21. Hoeteck Wee. Broadcast encryption with size $N^{1/3}$ and more from k-lin. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 155–178, Virtual Event, August 2021. Springer, Heidelberg.

A Appendix

A.1 Decisional Separation Diffie-Hellman (DSDH) Assumption

Definition 17. *In a cyclic group $\mathbb{G}$ of prime order q, the* **Decisional Separation Diffie-Hellman** (DSDH) *problem is to distinguish the distributions*

$$D_0 = \{(x, y, [\![1]\!], [\![a]\!], [\![b]\!], [\![ab + x]\!])\} \qquad D_1 = \{x, y, ([\![1]\!], [\![a]\!], [\![b]\!], [\![ab + y]\!])\}$$

for any $x, y \in \mathbb{Z}_q$, and $a, b \xleftarrow{\$} \mathbb{Z}_q$. The DSDH *assumption in $\mathbb{G}$ assumes that no* ppt *adversary can solve the* DSDH *problem with non-negligible probability.*

A.2 Dual Pairing Vector Spaces

Basis changes. In this work, we use extensively *basis changes* over dual orthogonal bases of a DPVS. We again use $\mathbb{G}_1^N$ as a running example. Let $(\mathbf{A}, \mathbf{A}^*)$ be the dual canonical bases of $(\mathbb{G}_1^N, \mathbb{G}_2^N)$. Let $(\mathbf{U} = (U_i)_i, \mathbf{U}^* = (U_i^*)_i)$ be a pair of dual bases of $(\mathbb{G}_1^N, \mathbb{G}_2^N)$, corresponding to an invertible matrix $U \in \mathbb{Z}_q^{N \times N}$. Given an invertible matrix $B \in \mathbb{Z}_q^{N \times N}$, the basis change from $\mathbf{U}$ w.r.t B is defined to be $\mathbf{B} := B \cdot \mathbf{U}$, which means:

$$\begin{aligned}(x_1, \ldots, x_N)_{\mathbf{B}} &= \sum_{i=1}^{N} x_i \mathbf{b}_i = (x_1, \ldots, x_N) \cdot \mathbf{B} = (x_1, \ldots, x_N) \cdot B \cdot \mathbf{U} \\ &= (y_1, \ldots, y_N)_{\mathbf{U}} \text{ where } (y_1, \ldots, y_N) := (x_1, \ldots, x_N) \cdot B \ .\end{aligned}$$

Under a basis change $\mathbf{B} = B \cdot \mathbf{U}$, we have

$$(x_1, \ldots, x_N)_{\mathbf{B}} = ((x_1, \ldots, x_N) \cdot B)_{\mathbf{U}} \,;\ (y_1, \ldots, y_N)_{\mathbf{U}} = \left((y_1, \ldots, y_N) \cdot B^{-1}\right)_{\mathbf{B}} \ .$$

The computation is extended to the dual basis change $\mathbf{B}^* = B' \cdot \mathbf{U}^*$, where $B' = \left(B^{-1}\right)^\top$:

$$(x_1, \ldots, x_N)_{\mathbf{B}^*} = \left((x_1, \ldots, x_N) \cdot B'\right)_{\mathbf{U}^*} \,;\ (y_1, \ldots, y_N)_{\mathbf{U}^*} = \left((y_1, \ldots, y_N) \cdot B^\top\right)_{\mathbf{B}^*} \ .$$

In can be checked that $(\mathbf{B}, \mathbf{B}^*)$ remains a pair of dual orthogonal bases. When we consider a basis change $\mathbf{B} = B \cdot \mathbf{U}$, if $B = (b_{i,j})_{i,j}$ affects only a subset $J \subseteq [N]$ of indices in the representation w.r.t basis $\mathbf{U}$, we will write B as the square block containing $(b_{i,j})_{i,j}$ for $i, j \in J$ and implicitly the entries of B outside this block is taken from I_N.

A.3 IND-security of Functional Encryption with Fine-grained Access Control

Definition 18 (IND-CPA security). *A functional encryption scheme with fine-grained access control $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ for the functionality class $\mathcal{F} \times \mathsf{AC\text{-}K}$ is* secure against chosen-plaintext attacks (IND-secure) *if for all* ppt *adversaries $\mathcal{A}$, and for all sufficiently large $\lambda \in \mathbb{N}$, the following probability is negligible*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) := \left| \Pr[\mathsf{Expr}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \right| \ .$$

In a more relaxed notion, the scheme $\mathcal{E}$ is selectively secure against chosen-plaintext attacks (selectively IND-secure) *if the following probability is negligible*

$$\mathsf{Adv}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) := \left| \Pr[\mathsf{Expr}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \right| \ .$$

For $b \in \{0, 1\}$, the games $\mathsf{Expr}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ and $\mathsf{Expr}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ are depicted in Figure 4. The probability is taken over the random coins of $\mathcal{A}$ and the algorithms.

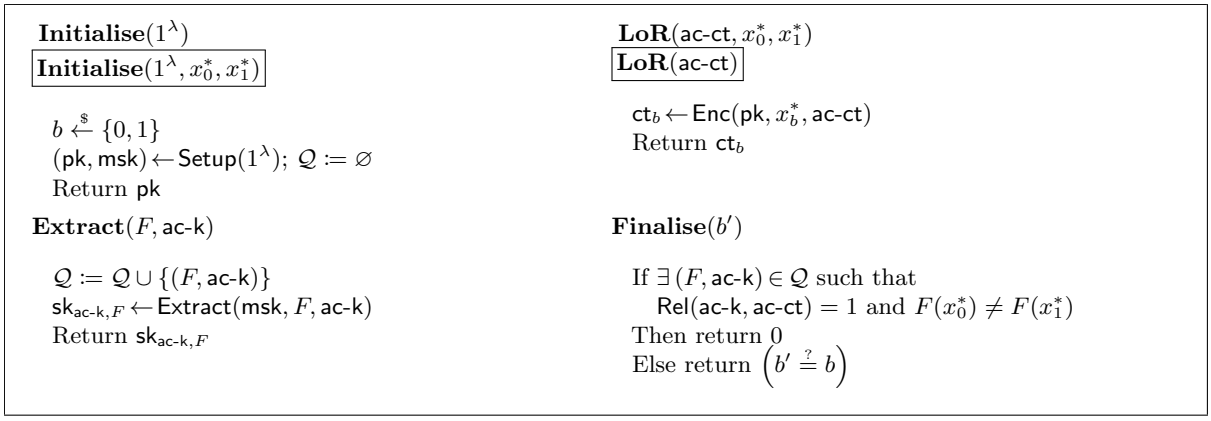

Fig. 4: The security games $\mathsf{Expr}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ and $\boxed{\mathsf{Expr}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)}$ for Definition 18

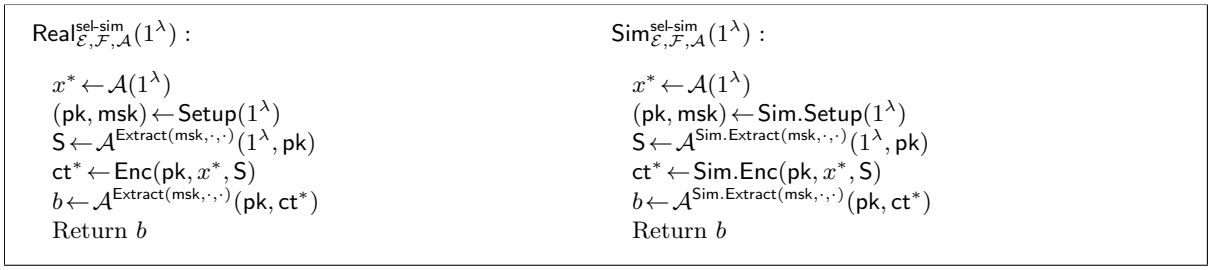

Fig. 5: The security games $\mathsf{Real}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ and $\mathsf{Sim}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda)$ for Definition 19

A.4 Selective Simulation-based Security for IPFE with Fine-Grained Access Control

Regarding this notion, an IPFE scheme with fine-grained access control is *selectively simulation-based* secure if there exists a ppt simulator that can setup the public information, derive functional keys, and encrypt a selective challenge message in a way that is indistinguishable from an execution of the real scheme.

Definition 19 (SEL-SIM security). *An IPFE scheme with fine-grained access control* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the function class* $\mathcal{F} \times \mathsf{AC\text{-}K}$ *is* selectively simulation-based secure *if for all* ppt *adversaries* $\mathcal{A}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*, there exists a* ppt *simulator* $\mathsf{Sim} = (\mathsf{Sim.Setup}, \mathsf{Sim.Extract}, \mathsf{Sim.Enc})$ *such that the following probability is negligible:*

$$\mathsf{Adv}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) :=:= \left|\Pr[\mathsf{Real}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1] - \Pr[\mathsf{Sim}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathsf{AC\text{-}K},\mathcal{A}}(1^\lambda) = 1]\right| \ .$$

The experiments $\mathsf{Real}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ *and* $\mathsf{Sim}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ *are described in Figure 5. The probability is taken over the random coins of* $\mathcal{A}$ *and the algorithms.*

B Supporting Materials - Deferred Proofs

B.1 Proof of Lemma 13

Lemma 13. *Let* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the function class* $\mathcal{F} \times \mathsf{AC\text{-}K}$ *be an MCFE scheme with fine-grained access control. If* $\mathcal{E}$ *is one-time IND-secure, then* $\mathcal{E}$ *is IND-secure.*

Proof. Suppose $\mathcal{E}$ is one-time IND-secure but not IND-secure. This means there exists a ppt adversary $\mathcal{A}$ such that

$$\mathsf{Adv}^{\text{mc-ind-cpa}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{A}}(1^\lambda) := \left|\Pr[\mathsf{Expr}^{\text{mc-ind-cpa}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2}\right|$$

is non-negligible. We construct a ppt adversary $\mathcal{B}$ using a black-blox access to $\mathcal{A}$ to break the one-time IND-security of $\mathcal{E}$. The adversary $\mathcal{B}$ works as follows:

1. $\mathcal{B}$ first obtains pk from its one-time challenger and sends pk to $\mathcal{A}$.
2. $\mathcal{B}$ chooses uniformly at random $k \in [Q]$.
3. For all challenge ciphertext queries $\mathbf{LoR}(i, x^{(0)}_{k,i}, x^{(1)}_{k,i}, \mathsf{tag}^*_k, \mathsf{ac\text{-}ct}^*_k)$: $\mathcal{B}$ make the queries to **LoR** of its one-time challenger and transfer the responses to $\mathcal{A}$.
4. For all challenge ciphertext queries $\mathbf{LoR}(i, x^{(0)}_{\ell,i}, x^{(1)}_{\ell,i}, \mathsf{tag}^*_\ell \neq \mathsf{tag}^*_k, \mathsf{ac\text{-}ct}^*_\ell)$ where $\ell < k$: $\mathcal{B}$ makes the queries for $x^{(0)}_{\ell,i}$ to **Enc** of its one-time challenger and transfer the responses to $\mathcal{A}$.
5. For all challenge ciphertext queries $\mathbf{LoR}(i, x^{(0)}_{\ell,i}, x^{(1)}_{\ell,i}, \mathsf{tag}^*_\ell \neq \mathsf{tag}^*_k, \mathsf{ac\text{-}ct}^*_\ell)$ where $\ell > k$: $\mathcal{B}$ makes the queries for $x^{(1)}_{\ell,i}$ to **Enc** of its one-time challenger and transfer the responses to $\mathcal{A}$.
6. For all **Enc** and **Extract** queries by $\mathcal{A}$, $\mathcal{B}$ relay them to its one-time challenger and transfers the responses to $\mathcal{A}$.
7. Finally, $\mathcal{A}$ outputs a bit b'. The adversary $\mathcal{B}$ outputs the same bit b'.

Let Q denote the number of challenge tags that are queried by $\mathcal{A}$ to $\mathcal{B}$. We use a sequence of hybrids $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_Q$ between $\mathcal{A}$ and its IND-security challenger such that in $\mathcal{H}_k$ all **LoR** queries for tag^*_ℓ, where $\ell \leq k$, are answered by encrypting $x^{(0)}_{\ell,i}$ and by encrypting $x^{(1)}_{\ell,i}$ if $\ell > k$. We denote by $\mathcal{H}_k = 1$ the event where the challenger outputs 1. We have $2 \cdot \mathsf{Adv}^{\text{mc-ind-cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda) = |\Pr[\mathcal{H}_Q = 1] - \Pr[\mathcal{H}_0 = 1]|$.

On the other hand, the advantage of $\mathcal{B}$ against the one-time IND-security experiment of $\mathcal{E}$ is

$$\begin{aligned}
2 \cdot \mathsf{Adv}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda) &= |\Pr[\mathsf{Expr}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda) = 1 \mid b = 0] \\
&\quad - \Pr[\mathsf{Expr}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda) = 1 \mid b = 1]| \\
&= \frac{1}{Q} \cdot \left| \sum_{k=1}^{Q} \Big(\Pr[\mathsf{Expr}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda) = 1 \mid b = 0, \mathcal{B} \text{ picks } k] \right. \\
&\quad \left. - \Pr[\mathsf{Expr}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda) = 1 \mid b = 1, \mathcal{B} \text{ picks } k] \Big) \right| \\
&\overset{(*)}{\geq} \frac{1}{Q} \cdot |\sum_{k=1}^{Q} (\Pr[\mathcal{H}_k] - \Pr[\mathcal{H}_{k-1}])| \\
&= \frac{1}{Q} \cdot |\Pr[\mathcal{H}_Q = 1] - \Pr[\mathcal{H}_0 = 1]| \\
&= \frac{1}{Q} \cdot 2 \cdot \mathsf{Adv}^{\text{mc-ind-cpa}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{A}}(1^\lambda)
\end{aligned}$$

where $(*)$ comes from the observation that conditioned on $b = 0$ (resp. $b = 1$) and $\mathcal{B}$ picks k, $\mathsf{Expr}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda) = 1$ is identical to $\mathcal{H}_k$ (resp. $\mathcal{H}_{k-1}$), where $\mathcal{B}$ is simulating the challenger for $\mathcal{A}$. Finally, we have $\mathsf{Adv}^{\text{mc-ind-cpa}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{A}}(1^\lambda) \leq Q \cdot \mathsf{Adv}^{\text{mc-ind-cpa-1-chal}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{B}}(1^\lambda)$ and because $\mathsf{Adv}^{\text{mc-ind-cpa}}_{\mathcal{E},\mathcal{F},\text{AC-K},\mathcal{A}}(1^\lambda)$ is non-negligible, $\mathcal{B}$ is breaking the one-time IND-security of $\mathcal{E}$ with non-negligible advantage. □

B.2 Proof of Lemma 4

Remark 20. The proof of Lemma 4 is a direct subsequence of the games we use to prove Lemma 21, i.e. from G_0 to G_4 in Figure 6, with an additional cleaning at coordinate 3 (based on

the subspace indistinguishability) of the vectors $\mathbf{c}_j$ at the end. It is important to note that the foregoing subsequence of games does not make use of the hypothesis $\mathbb{A}(\mathsf{S}) = 0$, which is used only for going from G_4 to G_5 and from G_5 to G_6 in Figure 6.

Lemma 4. *Let $\mathbb{A}$ be an* LSSS*-realizable over a set of attributes* Att $\subseteq \mathbb{Z}_q$*. We denote by* List-Att$(\mathbb{A})$ *the list of attributes appearing in $\mathbb{A}$ and by P the cardinality of* List-Att$(\mathbb{A})$*. Let* $\mathsf{S} \subseteq$ Att *be a set of attributes. Let $(\mathbf{H}, \mathbf{H}^*)$ and $(\mathbf{F}, \mathbf{F}^*)$ be two random dual bases of $(\mathbb{G}_1^2, \mathbb{G}_2^2)$ and $(\mathbb{G}_1^8, \mathbb{G}_2^8)$, respectively. The vectors $(\mathbf{h}_1, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ are public, while all other vectors are secret. Suppose we have two random labelings $(a_j)_{j \in \text{List-Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ for $a_0, a'_0 \xleftarrow{\$} \mathbb{Z}_q$. Then, under the* SXDH *assumption in $(\mathbb{G}_1, \mathbb{G}_2)$, the following two distributions are computationally indistinguishable:*

$$D_1 := \left\{ \begin{array}{lll} x, y & & \\ \forall\, j \in \mathsf{S}: & \mathbf{c}_j & = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\ \forall\, j \in \text{List-Att}(\mathbb{A}): & \mathbf{k}^*_j & = (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\ & \mathbf{c}_{\mathsf{root}} & = (\psi,\ 0)_{\mathbf{H}} \\ & \mathbf{k}^*_{\mathsf{root}} & = (a_0 \cdot z,\ 0)_{\mathbf{H}^*} \end{array} \right\}$$

and

$$D_2 := \left\{ \begin{array}{lll} x, y & & \\ \forall\, j \in \mathsf{S}: & \mathbf{c}_j & = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ \tau z_j \cdot x,\ 0,\ 0)_{\mathbf{F}} \\ \forall\, j \in \text{List-Att}(\mathbb{A}): & \mathbf{k}^*_j & = (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ a'_j \cdot y / z_j,\ 0,\ 0)_{\mathbf{F}^*} \\ & \mathbf{c}_{\mathsf{root}} & = (\psi,\ \tau \cdot x)_{\mathbf{H}} \\ & \mathbf{k}^*_{\mathsf{root}} & = (a_0 \cdot z,\ a'_0 \cdot y)_{\mathbf{H}^*} \end{array} \right\}$$

where for any $x, y \in \mathbb{Z}_q$ and $z_j, \sigma_j, \pi_j, \psi, \tau, z, r'_0 \xleftarrow{\$} \mathbb{Z}_q$.

Lemma 21. *Let $\mathbb{A}$ be an* LSSS*-realizable over a set of attributes* Att $\subseteq \mathbb{Z}_q$*. We denote by* List-Att$(\mathbb{A})$ *the list of attributes appearing in $\mathbb{A}$ and by P the cardinality of* List-Att$(\mathbb{A})$*. Let* $\mathsf{S} \subseteq$ Att *such that $\mathbb{A}(\mathsf{S}) = 0$, i.e. S does not contain any authorized set. Let $(\mathbf{H}, \mathbf{H}^*)$ and $(\mathbf{F}, \mathbf{F}^*)$ be two random dual bases of $(\mathbb{G}_1^2, \mathbb{G}_2^2)$ and $(\mathbb{G}_1^8, \mathbb{G}_2^8)$, respectively. The vectors $(\mathbf{h}_1, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ are public, while all other vectors are secret. Suppose we have a random labeling $(a_j)_{j \in \text{List-Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ for some $a_0 \xleftarrow{\$} \mathbb{Z}_q$. Then, under the* SXDH *assumption in $(\mathbb{G}_1, \mathbb{G}_2)$, the following two distributions are computationally indistinguishable:*

$$D_1 := \left\{ \begin{array}{lll} x, y & & \\ \forall\, j \in \mathsf{S}: & \mathbf{c}_j & = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\ \forall\, j \in \text{List-Att}(\mathbb{A}): & \mathbf{k}^*_j & = (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\ & \mathbf{c}_{\mathsf{root}} & = (\psi,\ 0)_{\mathbf{H}} \\ & \mathbf{k}^*_{\mathsf{root}} & = (a_0 \cdot z,\ 0)_{\mathbf{H}^*} \end{array} \right\}$$

and

$$D_2 := \left\{ \begin{array}{lll} x, y & & \\ \forall\, j \in \mathsf{S}: & \mathbf{c}_j & = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\ \forall\, j \in \text{List-Att}(\mathbb{A}): & \mathbf{k}^*_j & = (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\ & \mathbf{c}_{\mathsf{root}} & = (\psi,\ \tau \cdot x)_{\mathbf{H}} \\ & \mathbf{k}^*_{\mathsf{root}} & = (a_0 \cdot z,\ r'_0 \cdot y)_{\mathbf{H}^*} \end{array} \right\}$$

where $\sigma_j, \pi_j, \psi, \tau, z, r'_0 \xleftarrow{\$} \mathbb{Z}_q$ and x, y are constants.

Game G_0 :

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & 0 &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & 0 &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_1 : \tau \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau \cdot x & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & \tau \cdot x &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & 0 &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_2 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & 0 & \tau z_j \cdot x & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & \tau \cdot x &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & 0 &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_3 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q, a'_0 \xleftarrow{\$} \mathbb{Z}_q, (a'_j)_{j \in \mathcal{J}} \leftarrow \Lambda_{a'_0}(\mathbb{A})$

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau \cdot x & 0 & \tau z_j \cdot x & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & a'_j \cdot y & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & \tau \cdot x &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & a'_0 \cdot y &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_4 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q, a'_0 \xleftarrow{\$} \mathbb{Z}_q, (a'_j)_{j \in \mathcal{J}} \leftarrow \Lambda_{a'_0}(\mathbb{A})$

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau \cdot x & 0 & \tau z_j \cdot x & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & 0 & 0 & a'_j \cdot y / z_j & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & \tau \cdot x &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & a'_0 \cdot y &)_{\mathbf{H}^*} \end{array}$$

Two additional games for Lemma 21, if we know in advance $\mathbb{A}(\mathsf{S}) = 0$:

Game $\mathsf{G}_5 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q, a'_0 \xleftarrow{\$} \mathbb{Z}_q, (a'_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_0}(\mathbb{A})$

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau \cdot x & 0 & \tau z_j \cdot x & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & 0 & 0 & a'_j \cdot y / z_j & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & \tau \cdot x &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & r'_0 \cdot y &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_6 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q, a'_0 \xleftarrow{\$} \mathbb{Z}_q, (a'_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_0}(\mathbb{A})$

$$\begin{array}{llllllllll} \mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}_j^* & (& \pi_j \cdot (j,1) & a_j \cdot z & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{root}} & (& \psi & \tau \cdot x &)_{\mathbf{H}} \\ \mathbf{k}_{\mathsf{root}}^* & (& a_0 z & r'_0 \cdot y &)_{\mathbf{H}^*} \end{array}$$

Fig. 6: Games $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3, \mathsf{G}_4$ for the proof of Lemma 4. The index j runs over the list $\mathsf{List\text{-}Att}(\mathbb{A})$ for the $\mathbf{k}$-vectors and runs over the attributes in S for the $\mathbf{c}$-vectors. Games $\mathsf{G}_5, \mathsf{G}_6$ demonstrate a few extra steps to be done if more conveniently we know in advance $\mathbb{A}(\mathsf{S}) = 0$, and thus we regain the totally random masking from the works of Okamoto-Takashima (see Lemma 21 in Appendix B.2).

Proof (Of Lemma 21). The proof is done through a sequence of games, starting from G_0 where the adversary receives D_1 and ending in G_6 where the adversary receives D_2. The games are depicted in Figure 6.

The changes that make the transitions between games are highlighted in gray. The advantage of an adversary $\mathcal{A}$ in a game G_i is denoted by

$$\mathsf{Adv}(\mathsf{G}_i) \coloneqq \Pr[\mathsf{G}_i = 1] \ .$$

Game G_0: The vectors $\mathbf{c}_j, \mathbf{c}_{\mathsf{root}}$ and $\mathbf{k}^*_j, \mathbf{k}^*_{\mathsf{root}}$ are taken from D_1:

$$\begin{aligned}
\forall\, j \in \mathsf{S}: \ \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\forall\, j \in \mathsf{List\text{-}Att}(\mathbb{A}): \ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{root}} &= (\psi,\ 0)_{\mathbf{H}} \\
\mathbf{k}^*_{\mathsf{root}} &= (a_0 \cdot z,\ 0)_{\mathbf{H}^*}
\end{aligned}$$

Game G_1: We introduce a mask $\tau \xleftarrow{\$} \mathbb{Z}_q$ in the vectors $\mathbf{c}_j$ and $\mathbf{c}_{\mathsf{root}}$

$$\begin{aligned}
\forall\, j \in \mathsf{S}: \ \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau \cdot x,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\forall\, j \in \mathsf{List\text{-}Att}(\mathbb{A}): \ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau \cdot x)_{\mathbf{H}} \\
\mathbf{k}^*_{\mathsf{root}} &= (a_0 \cdot z,\ 0)_{\mathbf{H}^*}
\end{aligned}$$

Initially, let $(\mathbf{T}, \mathbf{T}^*), (\mathbf{W}, \mathbf{W}^*)$ be pairs of random dual bases. In the reduction from a DDH instance $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ where $c = ab + \tau$ with $\tau = 0$ or $\tau \xleftarrow{\$} \mathbb{Z}_q$, the bases will be changed as follows:

$$\begin{aligned}
F &\coloneqq \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}_{3,4} & F' &\coloneqq \left(F^{-1}\right)^{\top} = \begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}_{3,4} \\
\mathbf{F} &= F \cdot \mathbf{W}; & \mathbf{F}^* &= F' \cdot \mathbf{W}^* \\
H &\coloneqq \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}_{1,2} & H' &\coloneqq \left(H^{-1}\right)^{\top} = \begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}_{1,2} \\
\mathbf{H} &= H \cdot \mathbf{T}; & \mathbf{H}^* &= H' \cdot \mathbf{T}^*
\end{aligned}$$

Note that we can compute all the basis vectors except $\mathbf{h}^*_2$ and $\mathbf{f}^*_4$ but currently they are not needed because their coordinates are 0 in all the keys. The simulator can virtually set

$$\begin{aligned}
\mathbf{c}_{\mathsf{root}} &= (b \cdot x,\ c \cdot x)_{\mathbf{T}} \\
&= (b \cdot x,\ \tau \cdot x)_{\mathbf{H}} \\
\mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ b \cdot x,\ c \cdot x,\ 0,\ 0)_{\mathbf{W}} \text{ for } j \in \mathsf{S} \\
&= (\sigma_j \cdot (1, -j),\ b \cdot x,\ \tau \cdot x,\ 0,\ 0)_{\mathbf{F}} \text{ for } j \in \mathsf{S}
\end{aligned}$$

and $\psi \coloneqq b \cdot x$. If $\tau = 0$ then above vectors are computed as in G_0, otherwise we are in G_1. Therefore the difference in advantage is $|\mathsf{Adv}(\mathsf{G}_1) - \mathsf{Adv}(\mathsf{G}_0)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$, where $\mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$ denotes the advantage against the DDH problem in $\mathbb{G}_1$ set up with parameter λ.

Game G_2: In this game we introduce further a mask τz_j where $z_j \xleftarrow{\$} \mathbb{Z}_q$ into each vector $\mathbf{c}_j$:

$$\begin{aligned}
\forall\, j \in \mathsf{S}: \ \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau \cdot x,\ 0,\ \tau z_j \cdot x,\ 0,\ 0)_{\mathbf{F}} \\
\forall\, j \in \mathsf{List\text{-}Att}(\mathbb{A}): \ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau \cdot x)_{\mathbf{H}} \\
\mathbf{k}^*_{\mathsf{root}} &= (a_0 \cdot z,\ 0)_{\mathbf{H}^*}
\end{aligned}$$

Initially, let $(\mathbf{T}, \mathbf{T}^*), (\mathbf{V}, \mathbf{V}^*), (\mathbf{W}, \mathbf{W}^*)$ be pairs of random dual bases. Given a DDH instance $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ where $c = ab + \zeta$ with $\zeta = 0$ or $\zeta \xleftarrow{\$} \mathbb{Z}_q$, the bases will be changed as follows:

$$\begin{aligned}
&\mathbf{H} = \mathbf{T}; && \mathbf{H}^* = \mathbf{T}^* \\
&F \coloneqq \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & -a \\ 0 & 0 & 1 \end{bmatrix}_{1,2,6} && F' \coloneqq \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & a & 1 \end{bmatrix}_{1,2,6} \\
&\mathbf{F} = F \cdot \mathbf{W}; && \mathbf{F}^* = F' \cdot \mathbf{W}^*
\end{aligned}$$

Under this basis change, we can compute all basis vectors except $\mathbf{f}_6^*$, which is not a problem because the coordinate of $\mathbf{f}_6^*$ in the keys are 0 (and thus their representations do not alter under this basis change).
For $j \in \mathsf{S}$, the simulator can sample $\alpha_j, \beta_j \xleftarrow{\$} \mathbb{Z}_q$, compute (in the exponent) $b_j = \alpha_j \cdot b + \beta_j$ and $c_j = \alpha_j \cdot c + \beta_j \cdot a$, then virtually set

$$\begin{aligned}
\mathbf{c}_j &= (b_j \cdot x \cdot (1, -j),\ \psi,\ \tau,\ 0,\ c_j \cdot (1+j) \cdot x,\ 0,\ 0)_{\mathbf{W}} \\
&= (b_j x \cdot (1, -j),\ \psi,\ \tau,\ 0,\ (c_j \cdot (1+j) - a \cdot b_j - a \cdot b_j \cdot j) \cdot x,\ 0,\ 0)_{\mathbf{F}} \\
&= (b_j x \cdot (1, -j),\ \psi,\ \tau,\ 0,\ (c_j - a \cdot b_j) \cdot (1+j) \cdot x,\ 0,\ 0)_{\mathbf{F}} \\
&= (b_j x \cdot (1, -j),\ \psi,\ \tau,\ 0,\ (\alpha_j \cdot c - \alpha_j \cdot ab) \cdot (1+j) \cdot x,\ 0,\ 0)_{\mathbf{F}} \\
&= (b_j x \cdot (1, -j),\ \psi,\ \tau,\ 0,\ \tau z_j \cdot x,\ 0,\ 0)_{\mathbf{F}}
\end{aligned}$$

where $z_j = \alpha_j(1+j)\zeta/\tau$. If $\zeta = 0$ then $\mathbf{c}_j$ is computed as in G_1, else we are in the current game. We remark that we use the random self-reducibility of DDH in this transition to avoid a linear blow-up. Consequently, the difference in advantages of an adversary against G_0 and G_1 is bounded by

$$|\mathsf{Adv}(\mathsf{G}_2) - \mathsf{Adv}(\mathsf{G}_1)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) \ .$$

Game G_3: In this game, we start to change the vectors $\mathbf{k}_j^*$ and $\mathbf{k}_{\mathsf{root}}^*$. We sample $a_0' \xleftarrow{\$} \mathbb{Z}_q$ and perform a random labeling of a_0' to obtain $(a_j')_j \leftarrow \Lambda_{a_0'}(\mathbb{A})$. The vectors are masked as follows:

$$\begin{aligned}
\forall\, j \in \mathsf{S}:\ \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau \cdot x,\ 0,\ \tau z_j \cdot x,\ 0,\ 0)_{\mathbf{F}} \\
\forall\, j \in \mathsf{List\text{-}Att}(\mathbb{A}):\ \mathbf{k}_j^* &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ \boxed{a_j' \cdot y},\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau \cdot x)_{\mathbf{H}} \\
\mathbf{k}_{\mathsf{root}}^* &= (a_0 \cdot z,\ \boxed{a_0' \cdot y})_{\mathbf{H}^*}
\end{aligned}$$

Initially, let $(\mathbf{T}, \mathbf{T}^*), (\mathbf{W}, \mathbf{W}^*)$ be pairs of random dual bases. Given a DDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $c = ab + \rho$ with $\rho = 0$ or $\rho \xleftarrow{\$} \mathbb{Z}_q$, the bases will be changed as follows:

$$\begin{aligned}
&F \coloneqq \begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}_{3,4} && F' \coloneqq \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}_{3,4} \\
&\mathbf{F} = F \cdot \mathbf{W}; && \mathbf{F}^* = F' \cdot \mathbf{W}^* \\
&H \coloneqq \begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}_{1,2} && H' \coloneqq \left(H^{-1}\right)^\top = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}_{1,2} \\
&\mathbf{H} = H \cdot \mathbf{T}; && \mathbf{H}^* = H' \cdot \mathbf{T}^*
\end{aligned}$$

From the basis changes w.r.t $\mathbf{F}$ and $\mathbf{H}$, we can compute all vectors in those two bases except $\mathbf{h}_2$ and $\mathbf{f}_3$, but we can express those $\mathbf{c}$-vectors in $\mathbf{T}$ and $\mathbf{W}$. More precisely, the simulator can

virtually set:

$$\begin{aligned}\mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau\cdot x)_{\mathbf{T}}\\ &= (\psi + a\tau\cdot x,\ \tau\cdot x)_{\mathbf{H}}\\ \mathbf{c}_j &= (\sigma_j\cdot(1,-j),\ \psi,\ \tau\cdot x,\ 0,\ \tau z_j\cdot x,\ 0,\ 0)_{\mathbf{W}} \text{ for } j\in\mathsf{S}\\ &= (\sigma_j\cdot(1,-j),\ \psi + a\tau\cdot x,\ \tau\cdot x,\ 0,\ \tau z_j\cdot x,\ 0,\ 0)_{\mathbf{F}} \text{ for } j\in\mathsf{S}\ .\end{aligned}$$

Let $(d'_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ be a random labeling obtained from $\Lambda_1(\mathbb{A})$, i.e. we perform a secret sharing of 1 using the LSSS realizing $\mathbb{A}$. The simulator can virtually set:

$$\begin{aligned}\mathbf{k}^*_{\mathsf{root}} &= (a_0 z,\ 0)_{\mathbf{H}^*} + (b\cdot y,\ c\cdot y)_{\mathbf{T}^*}\\ &= (a_0 z + b\cdot y,\ \rho\cdot y)_{\mathbf{H}^*}\\ \mathbf{k}^*_j &= (\pi_j\cdot(j,1),\ a_j\cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*}\\ &\quad + (0,\ 0,\ bd'_j\cdot y,\ cd'_j\cdot y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{W}^*}\\ &= (\pi_j\cdot(j,1),\ a_j\cdot z + b\cdot y\cdot d'_j,\ \rho\cdot d'_j\cdot y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \forall\, j\in\mathsf{List\text{-}Att}(\mathbb{A})\ .\end{aligned}$$

When $\rho = 0$ we are in the previous game, where $\psi + a\tau\cdot y$ is used instead of ψ and the labeling is updated to:

$$\begin{aligned}&a_0 + b\cdot y/z\\ \text{For each } j\in\mathsf{List\text{-}Att}(\mathbb{A})\quad &a_j + b\cdot y\cdot d'_j/z\ .\end{aligned}$$

Otherwise, we are in the current game having additionally

$$a'_0 = \rho$$

that corresponds to the labels $a'_j = \rho\cdot d'_j$ for $j\in\mathsf{List\text{-}Att}(\mathbb{A})$. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}(\mathsf{G}_2)| \le \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

Game G_4: In this game, we swap $a'_j\cdot y$ from the 4-th coordinate to the 6-th coordinate, while multiplying it with $1/z_j$:

$$\begin{aligned}\forall\, j\in\mathsf{S}:\ \mathbf{c}_j &= (\sigma_j\cdot(1,-j),\ \psi,\ \tau\cdot x,\ 0,\ \tau z_j\cdot x,\ 0,\ 0)_{\mathbf{F}}\\ \forall\, j\in\mathsf{List\text{-}Att}(\mathbb{A}):\ \mathbf{k}^*_j &= (\pi_j\cdot(j,1),\ a_j\cdot z,\ 0,\ 0,\ a'_j\cdot y/z_j,\ 0,\ 0)_{\mathbf{F}^*}\\ \mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau\cdot x)_{\mathbf{H}}\\ \mathbf{k}^*_{\mathsf{root}} &= (a_0\cdot z,\ a'_0\cdot y)_{\mathbf{H}^*}\end{aligned}$$

This transition is discussed separately in Lemma 22. In the end, the difference in advantages is

$$\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3) \le P\cdot(6P+3)\cdot\mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)\ .$$

Game G_5: In this game, we replace a'_0 in the vector $\mathbf{k}_{\mathsf{root}}$ with a totally random value $r'_0 \xleftarrow{\$} \mathbb{Z}_q$:

$$\begin{aligned}\forall\, j\in\mathsf{S}:\ \mathbf{c}_j &= (\sigma_j\cdot(1,-j),\ \psi,\ \tau\cdot x,\ 0,\ \tau z_j\cdot x,\ 0,\ 0)_{\mathbf{F}}\\ \forall\, j\in\mathsf{List\text{-}Att}(\mathbb{A}):\ \mathbf{k}^*_j &= (\pi_j\cdot(j,1),\ a_j\cdot z,\ 0,\ 0,\ a'_j\cdot y/z_j,\ 0,\ 0)_{\mathbf{F}^*}\\ \mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau\cdot x)_{\mathbf{H}}\\ \mathbf{k}^*_{\mathsf{root}} &= (a_0\cdot z,\ r'_0\cdot y)_{\mathbf{H}^*}\end{aligned}$$

We first observe that the attributes in $(\mathbf{c}_j)_{j\in\mathsf{S}}$ do not satisfy the access structure $\mathbb{A}$ embedded in $(\mathbf{k}_j)_{j\in\mathcal{J}}$. Therefore, there are not enough a'_j/z_j from $\mathbf{k}_j$ to recover $\tau a'_0\cdot xy$, i.e. we cannot find an authorized set $A\subseteq\mathsf{S}$ having a reconstruction vector $\mathbf{c} = (c_j)$ such that

$$\sum_{j\in A}\tau x z_j\cdot\frac{c_j a'_j y}{z_j} = \tau a'_0\sum_{j\in A}c_j a'_j = \tau a'_0\cdot xy\ .$$

Moreover, because $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ is a secret sharing of a'_0 using the LSSS of $\mathbb{A}$, then all $(a'_j)_j$ are randomized into $(a'_j/z_j)_j$ and become independent uniformly random values, it holds that a'_0 will be perfectly indistinguishable from a random value $r'_0 \xleftarrow{\$} \mathbb{Z}_q$, which is not depending on $(a'_j)_j$ whatsoever, even under the view of an unbounded adversary. The advantage stays the same $\mathsf{Adv}(\mathsf{G}_5) = \mathsf{Adv}(\mathsf{G}_4)$.

Game G_6: In this game, we clean the masks in the vector components $\mathbf{c}_j, \mathbf{k}_j$:

$$\begin{aligned}
\forall\, j \in \mathsf{S}:\ \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \boxed{0},\ 0,\ \boxed{0},\ 0,\ 0)_{\mathbf{F}} \\
\forall\, j \in \mathcal{J}:\ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ \boxed{0},\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau \cdot x)_{\mathbf{H}} \\
\mathbf{k}^*_{\mathsf{root}} &= (a_0 \cdot z,\ r'_0 \cdot y)_{\mathbf{H}^*}
\end{aligned}$$

The transition is done by first applying the transition from G_3 to G_4 in reverse order (only the basis changes concerning $(\mathbf{F}, \mathbf{F}^*)$) to clear the masks in $\mathbf{k}_j$, then the transitions from G_0 to G_3 in reverse order to clear the masks in $\mathbf{c}_j$. Note that we are using the condition $\mathbb{A}(\mathsf{S}) = 0$ while cleaning the a'_j, without paying attention to a'_0 that is already replaced by r'_0, because there are not enough a'_j/z_j in the $\mathbf{k}$-vectors to recover a'_0 anyway.

The difference in advantages is

$$\begin{aligned}
|\mathsf{Adv}(\mathsf{G}_6) - \mathsf{Adv}(\mathsf{G}_0)| &\le \sum_{i=1}^{6} |\mathsf{Adv}(\mathsf{G}_i) - \mathsf{Adv}(\mathsf{G}_{i-1})| \\
&\le 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda) + 2P(6P+3) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \\
&\quad + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda) \\
&\le (2P \cdot (6P+3) + 6) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)
\end{aligned}$$

and the proof is concluded. □

Lemma 22. *Assuming the* SXDH *assumption for* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, the difference between advantages* $|\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3)|$ *in the proof of Lemma 21 is negligible.*

Proof. The idea is that we consider the swapping of $a'_j y$ to $a'_j y/z_j$ by each component in the list $\mathsf{List\text{-}Att}(\mathbb{A})$ of the attributes in $\mathbb{A}$ and analyse a sequence of games indexed by those attributes. More precisely, the game $\mathsf{G}_{3.m}$ is indexed by $m \in \{0, \ldots, P\}$, where P is the number of attributes in $\mathsf{List\text{-}Att}(\mathbb{A})$ and :

$$\begin{aligned}
\text{For } j \le m\ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ a'_j \cdot y/z_j,\ 0,\ 0)_{\mathbf{F}^*} \\
\text{For } j > m\ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ a'_j \cdot y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*}\ .
\end{aligned}$$

This leads to $\mathsf{G}_{3.0} = \mathsf{G}_3$ and $\mathsf{G}_{3.P} = \mathsf{G}_4$. The current form of other vectors is:

$$\begin{aligned}
\forall\, j \in \mathsf{S}:\ \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau \cdot x,\ 0,\ \tau z_j \cdot x,\ 0,\ 0)_{\mathbf{F}} \\
\forall\, j \ne m \in \mathsf{List\text{-}Att}(\mathbb{A}):\ \mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ a'_j \cdot y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{root}} &= (\psi,\ \tau \cdot x)_{\mathbf{H}} \\
\mathbf{k}^*_{\mathsf{root}} &= (a_0 \cdot z,\ a'_0 \cdot y)_{\mathbf{H}^*}
\end{aligned}$$

where $\tau, z_j \xleftarrow{\$} \mathbb{Z}_q$ are chosen uniformly at random. The labels $a_0, a'_0, (a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ and $(a'_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ satisfy $(a_j)_j \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$.

We first observe that the family of labelings, when viewed as a vector space over $\mathbb{Z}_q$, is closed under linear operations. In other words, a linear combination of vectors of labels gives a vector

of labels. Hence, following the idea from [DGP21], we can "factor out" the current labels in $\mathbf{k}$-vectors and manipulate the appropriate random linear factor for obtaining the desired new labels (multiplicatively). This requires some rewriting. For two labelings $\tilde{\mathbf{a}} := (\tilde{a}_0, (\tilde{a}_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}) \leftarrow \Lambda_{\tilde{a}_0}(\mathbb{A})$ and $(a''_0, (a''_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}) \leftarrow \Lambda_{a''_0}(\mathbb{A})$, together with uniformly random scalars $\rho, \delta \xleftarrow{\$} \mathbb{Z}_q^*$ we rewrite the vectors as follows

$$\begin{aligned}\mathbf{k}^*_{\mathsf{root}} &= (\tilde{a}_0 z,\ 0)_{\mathbf{H}^*} + a''_0 \cdot (\delta \cdot z,\ \rho y)_{\mathbf{H}^*}\\ \mathbf{k}^*_j &= (\Pi_j \cdot (j,1),\ \tilde{a}_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*}\\ &\quad + a''_j \cdot (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})\end{aligned}$$

and thus we have

$$\begin{aligned} a'_0 &= \rho \cdot a''_0; & a_0 &= \tilde{a}_0 + \delta \cdot a''_0\\ a'_j &= \rho \cdot a''_j; & a_j &= \tilde{a}_j + \delta \cdot a''_j\\ \pi_j &= \Pi_j + a''_j \cdot \tilde{\pi}_j\ . \end{aligned} \tag{2}$$

We can concentrate solely on the changes of the vectors $\mathbf{k}^*_j$. We can define

$$\mathbf{h}^*_j := (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$

and as a result we concentrate on the changes of the vectors $\mathbf{h}^*_j$. We note that changing multiplicatively the vectors $\mathbf{h}^*_j$ means changing *multiplicatively* the factor ρ. Thanks to the relations in (2), this means we are changing *multiplicatively* a'_0 and $(a'_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ as required for introducing $1/z_j$ in a'_j.

First, we fix an ordering of the attributes in the list $\mathsf{List\text{-}Att}(\mathbb{A})$, which is of size P. Given $m \in \{1, \ldots, P\}$, we write $j = m$ if $\mathbf{h}^*_j$ is the m-th vector component among $\mathbf{h}^*_j$ and the notation extends to $j < m$ and $j > m$. We now give a sequence of games for the transition from $\mathsf{G}_{3.m-1}$ to $\mathsf{G}_{3.m}$. This sequence of games can be found in Figure 7. We start from $\mathsf{G}_{3.m-1.0} = \mathsf{G}_{3.m-1}$:

Game $\mathsf{G}_{3.m-1.0}$: The vectors are specified as follows:

$$\begin{aligned}\mathbf{c}_j &= (\sigma_j \cdot (1,-j),\ \psi,\ \tau x,\ 0,\ \tau z_j x,\ 0,\ 0)_{\mathbf{F}}\\ \mathbf{h}^*_j &= \begin{cases} (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ 0,\ 0,\ \rho y/z_j,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j < m\\ (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j \geq m \end{cases}\end{aligned}$$

Game $\mathsf{G}_{3.m-1.1}$: In this game we do a formal basis change to duplicate the 5-th component into the 6-th one of the $\mathbf{c}$-vectors:

$$\mathbf{c}_j = (\sigma_j \cdot (1,-j),\ \psi,\ \tau x,\ \boxed{\tau x},\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}}$$

The basis change is done following these matrices:

$$\begin{aligned} F &:= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}_{4,5} & F' &:= \left(F^{-1}\right)^{\top} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}_{4,5}\\ \mathbf{F} &= F \cdot \mathbf{W}; & \mathbf{F}^* &= F' \cdot \mathbf{W}^* \end{aligned}$$

and the simulator can set

$$\begin{aligned}\mathbf{c}_j &= (\sigma_j \cdot (1,-j),\ \psi,\ \tau x,\ 0,\ \tau x z_j,\ 0,\ 0)_{\mathbf{W}}\\ &= (\sigma_j \cdot (1,-j),\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}}\ .\end{aligned}$$

We note that this affect all $\mathbf{c}$-vectors, for all $j \in \mathsf{S}$. This changes the vectors $\mathbf{f}_4$ and $\mathbf{f}^*_5$ but since they are all hidden from the adversary and the facing coordinates in $\mathbf{k}$-vectors are 0, the transition is perfectly indistinguishable and $\mathsf{Adv}(\mathsf{G}_{3.m-1.1}) = \mathsf{Adv}(\mathsf{G}_{3.m-1.0})$.

Game $\mathsf{G}_{3.m-1.0} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllllllll}
\mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & 0 & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j \geq m
\end{array}$$

Game $\mathsf{G}_{3.m-1.1} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllllllll}
\mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j \geq m
\end{array}$$

Game $\mathsf{G}_{3.m-1.2} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllllllll}
\mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j \geq m
\end{array}$$

Game $\mathsf{G}_{3.m-1.3} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllllllll}
\mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j / z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j \geq m
\end{array}$$

Game $\mathsf{G}_{3.m-1.4} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllllllll}
\mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & 0 & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \alpha y & \rho y / z_m & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j \geq m
\end{array}$$

Game $\mathsf{G}_{3.m-1.5} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{lllllllll}
\mathbf{c}_j & (& \sigma_j \cdot (1,-j) & \psi & \tau x & 0 & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y / z_m & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (& \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j \geq m
\end{array}$$

Fig. 7: Games for Lemma 22. The changes are made for the m-th key component $\mathbf{h}^*_m$ (with an ordering on $j \in \mathsf{List\text{-}Att}(\mathbb{A})$). See (2) for the rewriting of $\mathbf{k}^*_j$ into $\mathbf{h}^*_j$. The hybrids to go from $\mathsf{G}_{3.m-1.2}$ to $\mathsf{G}_{3.m-1.3}$ can be found in Figure 8.

Game $\mathsf{G}_{3.m-1.2}$: We do a swap between 4-th and 5-th components w.r.t the m-th attribute-wise key components:

$$\mathbf{h}_j^* = \begin{cases} (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ 0,\ 0,\ \rho y/z_j,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j < m \\ (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ \boxed{0},\ \boxed{\rho y},\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j = m \\ (\tilde{\pi}_j \cdot (j,1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j > m \end{cases}$$

Given a DSDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$, where $c = ab + \theta$ for $\theta = 0$ or $\theta = \rho$, the basis change is performed following the matrices:

$$F := \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ -a & 0 & 1 \end{bmatrix}_{2,4,5} \qquad F' := \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & -a & a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{2,4,5}$$

$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^*$$

The $\mathbf{c}$-vectors can be expressed in the bases $(\mathbf{W}, \mathbf{W}^*)$:

$$\begin{aligned} \mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{W}} \\ &= (\sigma_j, -j \cdot \sigma_j - ax\tau + ax\tau,\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \\ &= (\sigma_j, -j \cdot \sigma_j,\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \ . \end{aligned}$$

On the other hand, the simulator can set the $\mathbf{k}$-vectors as below: if $j = m$

$$\begin{aligned} \mathbf{h}_j^* &= (\tilde{\pi}'_j \cdot (j,1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\ &\quad + (by \cdot (j,1),\ 0,\ -cy,\ cy,\ 0)_{\mathbf{W}^*} \\ &= (\tilde{\pi}'_j \cdot (j,1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\ &\quad + (by \cdot (j,1),\ 0,\ -(c-ab)y,\ (c-ab)y,\ 0)_{\mathbf{F}^*} \\ &= ((\tilde{\pi}'_j + by) \cdot (j,1),\ \delta \cdot z,\ \rho y - \theta y,\ \theta y,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \ . \end{aligned}$$

The other vector components stay as in the previous game. When $\theta = 0$, we are in $\mathsf{G}_{3.m-1.1}$, otherwise we are in the current game and the difference between advantages is $|\mathsf{Adv}(\mathsf{G}_{3.m-1.2}) - \mathsf{Adv}(\mathsf{G}_{3.m-1.1})| \leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

Game $\mathsf{G}_{3.m-1.3}$: We now change the $\mathbf{c}$-vector component such that for every $j \neq m$, the 5-th coordinate, which is τx from the duplication in $\mathsf{G}_{3.m-1.1}$, will be changed to $\tau x z_j / z_m$:

$$\mathbf{c}_j = \begin{cases} (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \boxed{\tau x z_j / z_m},\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \text{ if } j \neq m \\ (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \text{ if } j = m \end{cases}$$

We apply Lemma 23 to consider the transition from $\mathsf{G}_{3.m-1.2}$ to $\mathsf{G}_{3.m-1.3}$. We do a sequence of hybrids indexed by $m' \in \mathsf{List\text{-}Att}(\mathbb{A}) \setminus \{m\}$. The coordinates affected are $(1, 2, 5, 7, 8)$ of $(\mathbf{F}, \mathbf{F}^*)$. We note that during each application of the lemma for an index m', only the vectors $\mathbf{c}_{m'}$ and $\mathbf{k}^*_m$ are taken into account and affected by the basis changes (w.r.t the gray boxes). The main reason that we have to do index by index, for $m' \in \mathsf{List\text{-}Att}(\mathbb{A}) \setminus \{m\}$, to change $\mathbf{c}_{m'}$ is the fact that we use formal basis changes to randomize the $(7, 8)$ coordinates, which in turn provide randomness to change the 5-th coordinate of $\mathbf{c}_{m'}$. Indeed, if we change more than 2 vectors $\mathbf{c}_{m'}$ at the same time, there will be more than 2 linear relations in a linear system binding the $(7, 8)$ coordinates. The solution of this system uses the fact that $m' - m \neq 0$ and $1/(m' - m)$ is well-defined. The more relations it has, the more restrictive it becomes and in the end our formal basis change cannot be well-defined, i.e. we cannot obtain an invertible matrix. Thus, we can only deal with 1 vector $\mathbf{c}_{m'}$, where $m' \in \mathsf{List\text{-}Att}(\mathbb{A}) \setminus \{m\}$. For other vectors, the concerning coordinates can be written directly in the target bases because they

Game $\mathsf{G}_{3.m-1.2.m'-1.0} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllll}
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j < m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j \geq m' \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y/z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j > m
\end{array}$$

Game $\mathsf{G}_{3.m-1.2.m'-1.1} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllll}
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j < m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j \geq m' \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y/z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & j\theta_j & \theta_j \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j > m
\end{array}$$

Game $\mathsf{G}_{3.m-1.2.m'-1.2} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllll}
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j < m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & \mu_j & -j\mu_j \ )_{\mathbf{F}} \text{ if } m \neq j = m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j > m' \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y/z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & j\theta_j & \theta_j \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j > m
\end{array}$$

Game $\mathsf{G}_{3.m-1.2.m'-1.3} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllll}
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j < m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & \mu_1 & \mu_2 \ )_{\mathbf{F}} \text{ if } m \neq j = m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j > m' \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y/z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & \theta_1 & \theta_2 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j > m
\end{array}$$

Game $\mathsf{G}_{3.m-1.2.m'-1.4} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllll}
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j < m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & \mu_1 & \mu_2 \ )_{\mathbf{F}} \text{ if } m \neq j = m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j > m' \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y/z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & \theta_1 & \theta_2 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j > m
\end{array}$$

Game $\mathsf{G}_{3.m-1.2.m'-1.5} = \mathsf{G}_{3.m-1.2.m'} : z_j \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{llllllll}
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x z_j/z_m & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j \leq m' \\
\mathbf{c}_j & (\ \sigma_j \cdot (1,-j) & \psi & \tau x & \tau x & \tau z_j x & 0 & 0 \ )_{\mathbf{F}} \text{ if } m \neq j > m' \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & 0 & \rho y/z_j & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j < m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & 0 & \rho y & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j = m \\
\mathbf{h}^*_j & (\ \tilde{\pi}_j \cdot (j,1) & \delta \cdot z & \rho y & 0 & 0 & 0 & 0 \ )_{\mathbf{F}^*} \text{ if } j > m
\end{array}$$

Fig. 8: The hybrids to go from $\mathsf{G}_{3.m-1.2}$ to $\mathsf{G}_{3.m-1.3}$, by applying Lemma 23. The changes are made for the m-th key component $\mathbf{h}^*_m$ (with an ordering on $j \in \mathsf{List\text{-}Att}(\mathbb{A})$). See (2) for the rewriting of $\mathbf{k}^*_j$ into $\mathbf{h}^*_j$.

are all 0. We proceed by a sequence of games depicted in Figure 8. The changes that make the transitions between games are highlighted in gray. The difference in advantages is

$$|\mathsf{Adv}(\mathsf{G}_{3.m-1.3}) - \mathsf{Adv}(\mathsf{G}_{3.m-1.2})| \leq P \cdot (4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)) \ .$$

Game $\mathsf{G}_{3.m-1.4}$: The goal of this game is to introduce ρ/z_m in the 6-th coordinate of the m-th $\mathbf{h}$-vector component, and at the same time to clean the τ in the 6-th coordinate of the $\mathbf{c}$-vector components. After $\mathsf{G}_{3.m-1.3}$, the vectors are of the form:

$$\mathbf{c}_j = \begin{cases} (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \tau z_j x/z_m,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \text{ if } j \neq m \\ (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \text{ if } j = m \end{cases}$$

$$\mathbf{h}^*_j = \begin{cases} (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ 0,\ 0,\ \rho y/z_j,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j < m \\ (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ 0,\ \rho y,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j = m \\ (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j > m \end{cases}$$

We now change the basis w.r.t $(\mathbf{F}, \mathbf{F}^*)$ using the following matrices:

$$F \coloneqq \begin{bmatrix} \alpha/\rho & 0 \\ 1/z_m & 1 \end{bmatrix}_{5,6} \qquad F' \coloneqq \left(F^{-1}\right)^\top = \begin{bmatrix} \rho/\alpha & -\rho/(z_m \alpha) \\ 0 & 1 \end{bmatrix}_{5,6}$$
$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^* \ .$$

Note that this basis change will affect only the $\mathbf{h}$-vector of attribute $m \in \mathsf{List\text{-}Att}(\mathbb{A})$, because by construction the other components have coordinate 0 for $\mathbf{f}^*_5$ and have the same writing before and after the basis change. Moreover, the basis change can be applied before the simulator sees the vectors along with $\mathbb{A}$ and S, by first sampling a value $z \xleftarrow{\$} \mathbb{Z}_q$ and use z in the basis change. Afterwards, when all attributes are declared, z would be the mask at the attribute m corresponding to the current hybrid.
We have

$$\begin{aligned} \mathbf{c}_j &= \begin{cases} (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \tau z_j x/z_m,\ \tau x z_j,\ 0,\ 0)_{\mathbf{W}} \text{ if } j \neq m \\ (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ \tau x,\ \tau x z_j,\ 0,\ 0)_{\mathbf{W}} \text{ if } j = m \end{cases} \\ &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ 0,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \text{ for all } j \\ \mathbf{h}^*_j &= (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ 0,\ \rho y,\ 0,\ 0,\ 0)_{\mathbf{W}^*} \text{ if } j = m \\ &= (\tilde{\pi}_m \cdot (m, 1),\ \delta \cdot z,\ 0,\ \alpha y,\ \rho y/z_m,\ 0,\ 0)_{\mathbf{F}^*} \end{aligned}$$

and because $\mathbf{f}_5, \mathbf{f}_6, \mathbf{f}^*_5, \mathbf{f}^*_6$ are hidden from the adversary, this change is a formal basis change. For other $j \neq m$, $\mathbf{h}^*_j$ does not use $\mathbf{f}^*_5$, which is affected, then we can write directly:

$$\mathbf{h}^*_j = (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ *,\ 0,\ *,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j \neq m \ .$$

The transition is perfectly indistinguishable. In the end, the difference in advantage is $\mathsf{Adv}(\mathsf{G}_{3.m-1.3}) = \mathsf{Adv}(\mathsf{G}_{3.m-1.4})$.

Game $\mathsf{G}_{3.m-1.5}$: The goal of this game is to put the m-th attribute-wise $\mathbf{h}$-vector component in to the form required by $\mathsf{G}_{3.m}$, i.e. remove the random value αy in the 5-th coordinate. After $\mathsf{G}_{3.m-1.4}$, the vectors are of the form:

$$\mathbf{c}_j = (\sigma_j \cdot (1, -j),\ \psi,\ \tau x,\ 0,\ \tau x z_j,\ 0,\ 0)_{\mathbf{F}} \text{ for all } j$$

$$\mathbf{h}^*_j = \begin{cases} (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ 0,\ 0,\ \rho y/z_j,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j < m \\ (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ 0,\ \alpha y,\ \rho y/z_j,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j = m \\ (\tilde{\pi}_j \cdot (j, 1),\ \delta \cdot z,\ \rho y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ if } j > m \end{cases}$$

where $\alpha \xleftarrow{\$} \mathbb{Z}_q$. Given an instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $c = ab + \alpha$ and either $\alpha = 0$ or $\alpha \xleftarrow{\$} \mathbb{Z}_q$, the simulator performs a basis change following the matrices:

$$F \coloneqq \begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}_{2,5} \qquad F' \coloneqq \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}_{2,5}$$
$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^* \ .$$

We cannot compute $\mathbf{f}_5$ but this is not problematic because all the 5-th coordinates of the $\mathbf{c}$-vector components are 0. In addition, the vectors $\mathbf{h}_j^*$ for $j \neq m$ can be written directly in $(\mathbf{F}, \mathbf{F}^*)$ thanks to the fact that their coordinates in $\mathbf{f}_5^*$ are 0. The simulator can then virtually set for $j = m$,

$$\begin{aligned} \mathbf{h}_j^* &= (by \cdot (j,1),\ \delta \cdot z,\ 0,\ cy,\ \rho y / z_m,\ 0,\ 0)_{\mathbf{W}^*} \\ &= (by \cdot (j,1),\ \delta \cdot z,\ 0,\ \alpha y,\ \rho y / z_m,\ 0,\ 0)_{\mathbf{F}^*} \end{aligned}$$

and when $\alpha \xleftarrow{\$} \mathbb{Z}_q$, we are in the previous game, otherwise we are in the current game that is identical to $\mathsf{Adv}(\mathsf{G}_{3.m})$. The difference in advantages is

$$\begin{aligned} |\mathsf{Adv}(\mathsf{G}_{3.m}) - \mathsf{Adv}(\mathsf{G}_{3.m-1.4})| &= |\mathsf{Adv}(\mathsf{G}_{3.m-1.5}) - \mathsf{Adv}(\mathsf{G}_{3.m-1.4})| \\ &\leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) \ . \end{aligned}$$

The difference in advantages from $\mathsf{G}_{3.m-1} = \mathsf{G}_{3.m-1.0}$ to $\mathsf{G}_{3.m} = \mathsf{G}_{3.m-1.5}$ is

$$\begin{aligned} |\mathsf{Adv}(\mathsf{G}_{3.m}) - \mathsf{Adv}(\mathsf{G}_{3.m-1})| &\leq \sum_{i=1}^{5} |\mathsf{Adv}(\mathsf{G}_{3.m-1.i}) - \mathsf{Adv}(\mathsf{G}_{3.m-1.i-1})| \\ &\leq P \cdot (4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)) \\ &\quad + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda) + \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) \\ &\leq (6P + 3) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1, \mathbb{G}_2}(1^\lambda) \ . \end{aligned}$$

After changing all P components $\mathbf{k}_j$, for $j \in \mathsf{List\text{-}Att}(\mathbb{A})$, we arrive at $\mathsf{G}_{3.P} = \mathsf{G}_4$ and the total difference in advantages is:

$$|\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3)| \leq P \cdot (6P + 3) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1, \mathbb{G}_2}(1^\lambda) \ .$$

The proof is concluded. □

Lemma 23. *Let $(\mathbf{F}, \mathbf{F}^*)$ be the dual bases of $\mathbb{G}_1^5$ and $\mathbb{G}_2^5$ respectively. Suppose that the vectors $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ are public, while all others are kept secret. Let $j \neq m$ and $\beta, \alpha, \gamma \in \mathbb{Z}_q$ are chosen constants. Then, under the* SXDH *assumption, the following two distributions are computationally indistinguishable:*

$$D_1 \coloneqq \left\{ \begin{array}{l} \mathbf{c} \ = (\sigma \cdot (1, -j),\ \gamma,\ 0,\ 0)_{\mathbf{F}} \\ \mathbf{k}^* = (\pi \cdot (m, 1),\ \beta,\ 0,\ 0)_{\mathbf{F}^*} \end{array} \right\}$$

and

$$D_2 \coloneqq \left\{ \begin{array}{l} \mathbf{c} \ = (\sigma \cdot (1, -j),\ \alpha,\ 0,\ 0)_{\mathbf{F}} \\ \mathbf{k}^* = (\pi \cdot (m, 1),\ \beta,\ 0,\ 0)_{\mathbf{F}^*} \end{array} \right\}$$

where $\sigma, \pi \xleftarrow{\$} \mathbb{Z}_q$ are unknown and random.

Proof. The advantage of an adversary $\mathcal{A}$ in a game G_i is denoted by

$$\mathsf{Adv}(\mathsf{G}_i) \coloneqq \Pr[\mathsf{G}_i = 1]$$

where the probability is taken over the random choices of $\mathcal{A}$ and coins of G_i.

Game G_0: In this game, the adversary receives from the distribution D_1:

$$\begin{aligned}\mathbf{c} &= (\sigma \cdot (1, -j),\ \gamma,\ 0,\ 0)_{\mathbf{F}} \\ \mathbf{k}^* &= (\pi \cdot (m, 1),\ \beta,\ 0,\ 0)_{\mathbf{F}^*}\ .\end{aligned}$$

Game G_1: In this game, we duplicate the first two coordinates of $\mathbf{k}^*$ into the 4-th and 5-th coordinates:

$$\begin{aligned}\mathbf{c} &= (\sigma \cdot (1, -j),\ \gamma,\ 0,\ 0)_{\mathbf{F}} \\ \mathbf{k}^* &= (\pi \cdot (m, 1),\ \beta,\ \rho m,\ \rho)_{\mathbf{F}^*}\ .\end{aligned}$$

Let $(\mathbf{W}, \mathbf{W}^*)$ be the canonical bases of $\mathbb{G}_1^5$ and $\mathbb{G}_2^5$. Given a DDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $\rho := c - ab$ is either 0 or uniformly random, we use the following basis changing matrices (F, F'):

$$F := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & 1 & 0 \\ 0 & -a & 0 & 1 \end{bmatrix}_{1,2,4,5} \qquad F' := \left(F^{-1}\right)^{\top} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{1,2,4,5}$$

$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^*$$

We cannot compute the basis vectors $\mathbf{f}_4$ and $\mathbf{f}_5$ but they are not used in $\mathbf{c}$. The vector $\mathbf{k}^*$ can be simulated as follows:

$$\begin{aligned}\mathbf{k}^* &= (b \cdot (m, 1),\ \beta,\ c \cdot m,\ c)_{\mathbf{W}^*} \\ &= (b \cdot (m, 1),\ \beta,\ c \cdot m - ab \cdot m,\ c - ab)_{\mathbf{F}^*} \\ &= (b \cdot (m, 1),\ \beta,\ \rho \cdot m,\ \rho)_{\mathbf{F}^*}\end{aligned}$$

If $\rho = 0$ we are in G_0, otherwise we are in G_1. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_1) - \mathsf{Adv}(\mathsf{G}_0)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

Game G_2:In this game, we duplicate the first two coordinates of $\mathbf{c}$ into the 4-th and 5-th coordinates:

$$\begin{aligned}\mathbf{c} &= (\sigma \cdot (1, -j),\ \gamma,\ \tau,\ -j\tau)_{\mathbf{F}} \\ \mathbf{k}^* &= (\pi \cdot (m, 1),\ \beta,\ \rho m,\ \rho)_{\mathbf{F}^*}\ .\end{aligned}$$

Let $(\mathbf{W}, \mathbf{W}^*)$ be the canonical bases of $\mathbb{G}_1^5$ and $\mathbb{G}_2^5$. Given a DDH instance $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ where $\tau := c - ab$ is either 0 or uniformly random, we use the following basis changing matrices (F, F'):

$$F := \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{1,2,4,5} \qquad F' := \left(F^{-1}\right)^{\top} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & 1 & 0 \\ 0 & -a & 0 & 1 \end{bmatrix}_{1,2,4,5}$$

$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^*$$

The vector $\mathbf{c}$ can be simulated as follows:

$$\begin{aligned}\mathbf{c} &= (b \cdot (1, -j),\ \gamma,\ c, -j \cdot c)_{\mathbf{W}} \\ &= (b \cdot (1, -j),\ \gamma,\ c - ab, -j \cdot c - j \cdot ab)_{\mathbf{F}} \\ &= (b \cdot (1, -j),\ \gamma,\ \tau, -j\tau)_{\mathbf{F}}\ .\end{aligned}$$

We cannot compute the basis $\mathbf{F}^*$ but the vector $\mathbf{k}^*$ can be written in $\mathbf{W}^*$ and then we observe how it is affected under this basis change:

$$\begin{aligned}\mathbf{k}^* &= (\pi \cdot (m,1),\ \beta,\ \rho \cdot m,\ \rho)_{\mathbf{W}^*} \\ &= ((\pi + a\rho) \cdot (m,1),\ \beta,\ \rho \cdot m,\ \rho)_{\mathbf{F}^*}\end{aligned}$$

and π is updated to $\pi + a\rho$.

If $\rho = 0$ we are in G_1, otherwise we are in G_2. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_2) - \mathsf{Adv}(\mathsf{G}_1)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$.

Game G_3: We randomise the last two coordinates in $\mathbf{c}$ and $\mathbf{k}^*$, which were changed from the previous games:

$$\begin{aligned}\mathbf{c} &= (\sigma \cdot (1,-j),\ \gamma,\ \mu_1,\ \mu_2)_{\mathbf{F}} \\ \mathbf{k}^* &= (\pi \cdot (m,1),\ \beta,\ \theta_1,\ \theta_2)_{\mathbf{F}^*}\end{aligned}$$

where $\theta_1, \theta_2 \xleftarrow{\$} \mathbb{Z}_q$ are chosen uniformly at random.

We consider the basis changing matrices (F, F'):

$$F \coloneqq \begin{bmatrix} z_1 & z_2 \\ z_3 & z_4 \end{bmatrix}_{4,5} \qquad F' \coloneqq \left(F^{-1}\right)^\top = \begin{bmatrix} z_4 & -z_3 \\ -z_2 & z_1 \end{bmatrix}_{4,5}$$
$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^*$$

where $z_1, z_2, z_3, z_4 \in \mathbb{Z}_q$ are chosen such that $z_1 z_4 - z_2 z_3 = 1$. The basis change affects the hidden vectors $(\mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_4^*, \mathbf{f}_5^*)$.

The two vectors $\mathbf{c}$ and $\mathbf{k}^*$ can be written directly in $\mathbf{W}$ and $\mathbf{W}^*$ respectively:

$$\begin{aligned}\mathbf{c} &= (\sigma \cdot (1,-j),\ \gamma,\ \tau, -j\tau)_{\mathbf{W}} \\ &= (\sigma \cdot (1,-j),\ \gamma,\ \tau z_4 + \tau j z_3, -\tau z_2 - \tau j z_1)_{\mathbf{F}} \\ \mathbf{k}^* &= (\pi \cdot (m,1),\ \beta,\ \rho m,\ \rho)_{\mathbf{W}^*} \\ &= (\pi \cdot (m,1),\ \beta,\ \rho m z_1 + z_2 \rho,\ \rho m z_3 + z_4 \rho)_{\mathbf{F}^*}\ .\end{aligned}$$

Let $\mu_1, \mu_2, \theta_1, \theta_2 \xleftarrow{\$} \mathbb{Z}_q$ and we consider the following system to solve for (z_1, z_2, z_3, z_4):

$$\begin{cases} \tau(z_4 + j z_3) = \mu_1 \\ -\tau(z_2 + j z_1) = \mu_2 \\ \rho(m z_1 + z_2) = \theta_1 \\ \rho(m z_3 + z_4) = \theta_2 \end{cases} \Leftrightarrow \begin{cases} z_4 + j z_3 = \mu_1/\tau \\ m z_3 + z_4 = \theta_2/\rho \\ z_2 + j z_1 = -\mu_2/\tau \\ m z_1 + z_2 = \theta_1/\rho \end{cases}$$
$$\Leftrightarrow \begin{cases} (j-m) z_3 = \mu_1/\tau - \theta_2/\rho \\ m z_3 + z_4 = \theta_2/\rho \\ (j-m) z_1 = -\mu_2/\tau - \theta_1/\rho \\ m z_1 + z_2 = \theta_1/\rho \end{cases}.$$

The system has a solution if and only if $j \neq m$, which is already our hypothesis. We note that since $\mu_1, \mu_2, \theta_1, \theta_2$ are uniformly random chosen values and fixed to determine (z_1, z_2, z_3, z_4), we can always perform normalization using $\mu_1, \mu_2, \theta_1, \theta_2$ to ensure $z_1 z_4 - z_2 z_3 = 1$ for the basis change. The basis change defined by (z_1, z_2, z_3, z_4) is totally formal and the difference in advantages is $\mathsf{Adv}(\mathsf{G}_3) = \mathsf{Adv}(\mathsf{G}_2)$.

Game G_4: In this game, we change the constant γ in $\mathbf{c}$ to another constant α:

$$\begin{aligned}\mathbf{c} &= (\sigma \cdot (1,-j),\ \alpha,\ \mu_1,\ \mu_2)_{\mathbf{F}} \\ \mathbf{k}^* &= (\pi \cdot (m,1),\ \beta,\ \theta_1,\ \theta_2)_{\mathbf{F}^*}\ .\end{aligned}$$

Let $(\mathbf{W}, \mathbf{W}^*)$ be the canonical bases of $\mathbb{G}_1^5$ and $\mathbb{G}_2^5$. Given a DSDH instance $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ where $\rho := c - ab$ is either γ or the constant α, we use the following basis changing matrices (F, F'):

$$F := \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix}_{3,4} \qquad F' := \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}_{3,4}$$
$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^* \ .$$

This basis change affects the vector $\mathbf{f}_4$ and $\mathbf{f}_3^*$, which are both kept secret from the adversary. The vector $\mathbf{c}$ can be simulated as follows:

$$\begin{aligned} \mathbf{c} &= (\sigma \cdot (1, -j),\ c,\ b,\ \mu_2)_{\mathbf{W}} \\ &= (\sigma \cdot (1, -j),\ \rho,\ b,\ \mu_2)_{\mathbf{F}} \ . \end{aligned}$$

Even though we cannot compute the basis vector $\mathbf{f}_3^*$, the vector $\mathbf{k}^*$ can be written directly in $\mathbf{W}^*$ to see how it will change:

$$\begin{aligned} \mathbf{k}^* &= (\pi \cdot (m, 1),\ \beta,\ \theta_1,\ \theta_2)_{\mathbf{W}^*} \\ &= (\pi \cdot (m, 1),\ \beta,\ \theta_1 + a\beta,\ \theta_2)_{\mathbf{F}^*} \end{aligned}$$

and θ_1 is updated to $\theta_1 + a\beta$. If $\rho = \gamma$ we are in the previous game, otherwise we are in the current game. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3)| \leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$.

Game G_5: In this game we clean the masks $\mu_1, \mu_2, \theta_1, \theta_2$ by doing the reverse transition from G_3 back to G_0. The total difference in advantages is $|\mathsf{Adv}(\mathsf{G}_5) - \mathsf{Adv}(\mathsf{G}_4)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

In G_5, the adversary receives from the distribution D_2 and we have

$$\begin{aligned} |\mathsf{Adv}(\mathsf{G}_5) - \mathsf{Adv}(\mathsf{G}_0)| &\leq \sum_{i=1}^{5} |\mathsf{Adv}(\mathsf{G}_i) - \mathsf{Adv}(\mathsf{G}_{i-1})| \\ &\leq 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda) \ . \end{aligned}$$

The proof of the lemma is concluded. □

B.3 Proof of Theorem 7

Theorem 7. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an IPFE scheme with fine-grained access control via LSSS presented in Figure 1 in a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$. Then, $\mathcal{E}$ is secure against chosen-plaintext attacks, adaptively in the attributes and the challenge messages, if the SXDH assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$. More precisely, let n be the dimension of vectors for inner-product computation, K denote the number of functional key queries, and P denote the maximum number of attributes in the access structure $\mathbb{A}$ queried for functional keys. We have the following bound:*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}(1^\lambda) \leq (2nK \cdot (P \cdot (6P + 3) + 2) + 5) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .$$

Proof (Main ideas). Using the dual-system methodology, we first change the challenge ciphertext into semi-functional and then we want to change the functional keys into semi-functional as well. This can be done only for the keys corresponding to $(\mathbb{A}, \mathbf{y})$ such that

$$\langle \mathbf{x}_0^*, \mathbf{y} \rangle \neq \langle \mathbf{x}_1^*, \mathbf{y} \rangle \ . \tag{3}$$

According to the model of security, condition (3) implies that the access structure $\mathbb{A}$ is not satisfied by the attributes S in the challenge ciphertext. Hence, changing the foregoing key into

semi-functional does not affect the fact that the ciphertext, which is already semi-functional, cannot be decrypted using this key. On the other hand, for the functional secret key associated to $(\mathbb{A}, \mathbf{y})$ where $\langle \Delta\mathbf{x}, \mathbf{y}' \rangle = 0$ and $\Delta\mathbf{x} := \mathbf{x}_1^* - \mathbf{x}_0^*$, it can remain normal. These keys include those whose policy is satisfied by the attributes in the challenge ciphertext and the decryption will return $\langle \mathbf{x}_0^*, \mathbf{y} \rangle = \langle \mathbf{x}_1^*, \mathbf{y} \rangle$ as expected. To prove the adaptive version, we need a strategy to change the challenge ciphertext and the keys into semi-functional such that the masks in the vectors exist only when condition (3) holds. Moreover, because the functional keys might be queried before the challenge messages are declared (we are in the adaptive setting), the keys should still allow correct decryption of normal ciphertexts, which the adversary can compute using pk as well as the later challenge ciphertext if the policy in the key is satisfied. Using the terminology from [OT12b], our main idea is using auxiliary *hidden* vectors $(\mathbf{f}_4, \ldots, \mathbf{f}_{n+7})$ over $\mathbf{F}$ and $(\mathbf{h}_4, \ldots, \mathbf{h}_{n+3})$ over $\mathbf{H}$, as well as their dual counterparts in $\mathbf{F}^*, \mathbf{H}^*$. These hidden subspace vectors will accommodate $\tau\Delta\mathbf{x}[i]$ in the $(i+3)$-th coordinate of the challenge ciphertext $\mathbf{c}_{\mathsf{ipfe}}$, and $r_0\mathbf{y}[i]$ in the $(i+3)$-th coordinate of functional key $\mathbf{k}_{\mathsf{ipfe}}$ corresponding to $\mathbf{y}$, for each $i \in [n]$ and the random masks $\tau, r_0 \xleftarrow{\$} \mathbb{Z}_q$. Then, when taking the products of vectors in DPVS, there will be a term $\tau r_0 \sum_{i\in[n]} \Delta\mathbf{x}[i]\mathbf{y}[i] = \tau r_0 \langle \Delta\mathbf{x}, \mathbf{y} \rangle$ and it will act as a mask only when $\langle \Delta\mathbf{x}, \mathbf{y} \rangle \neq 0$. The masking is done by each index $i \in [n]$, applying Lemma 4. For each $i \in [n]$, so as to introduce $r_0 \cdot \mathbf{y}[i]$ in $\mathbf{k}_{\mathsf{ipfe}}$ we will have to use 5 auxiliary hidden vectors in $\mathbf{c}_j$ for $(\tau\Delta\mathbf{x}[i],\ 0,\ \tau z_j \cdot \Delta\mathbf{x}[i], 0, 0)_{\mathbf{F}}$ for all $j \in \mathsf{S}$ and $z_j \xleftarrow{\$} \mathbb{Z}_q$. This explains why we need n more coordinates in $(\mathbf{F}, \mathbf{F}^*)$ to accommodate n values $(\tau z_j \cdot \Delta\mathbf{x}[i], a'_j\mathbf{y}[i]/z_j)$ in $(\mathbf{c}_j, \mathbf{k}_j^*) \in \mathbf{F} \times \mathbf{F}^*$, for each j, and 4 more auxiliary hidden vectors, besides the 3 vectors used in real life. The same goes for the need of $n+3$ basis vectors in $(\mathbf{H}, \mathbf{H}^*)$.

We remark that Lemma 4 only helps us mask the ℓ-th key components $\mathbf{k}^*_{\ell,\mathsf{ipfe}}$ by another random labeling based on $a'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$. However, after all the masks $(a'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ are in the vector $\mathbf{k}^*_{\ell,\mathsf{ipfe}}$, thanks to the fact that the product in DPVS will give us $\tau a'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y} \rangle$, we can change $(a'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ to $(r'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ *all at once*. If the access structure in the key is *not* satisfied by the challenge attributes, there does not exist any authorized set in S. In other words, there will exist $j \in \mathsf{S}$ such that $\tau z_j \cdot \Delta\mathbf{x}[i]$ appears in the *unique* challenge ciphertext but z_j is totally hidden in the current ℓ-th key. Thanks to the fact that $(a'_{\ell,j}/z_j)_j$ is perfectly randomized by $(z_j)_j$ from the labeling $(a'_{\ell,j})_j$ of $a'_{\ell,0}$, it implies that $a'_{\ell,0}$ is statistically hidden and $(a'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ can be replaced by $(r'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ for some uniformly independent random value $r'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$. Otherwise, if $\mathbb{A}(\mathsf{S}) = 1$, the security model enforces that $\langle \Delta\mathbf{x}, \mathbf{y} \rangle = 0$ and the result does not depend on $a'_{\ell,0}$ anymore. In either case, changing from $(a'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ to $(r'_{\ell,0} \cdot \mathbf{y}[i])_{i\in[n]}$ can be justified. We have to perform this masking by $r'_{\ell,0}\mathbf{y}[i]$ for only one key at a time; Or else two different ℓ-th and k-th keys containing the randomized labels $(a'_{\ell,j}/z_j, a'_{k,j}/z_j)_j$ might mutually leak information about z_j for some j embedded in $(\tau z_j)_j$ of the unique **LoR** ciphertext.

The last step is to virtually modify (S, U) in the master secret key msk so that the challenge ciphertext is now encrypting $\mathbf{x}_0^*[i]$ and is no longer depending on b. The new (S', U') will respect the relation dictated in pk, which is known by the adversary. For any functional key corresponding to $\mathbf{y}_\ell$ such that $\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle = 0$, simulating the key using (S, U) is identical to doing so using (S', U'). On the other hand, in the case where $\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle \neq 0$, simulating the functional key for $\mathbf{y}_\ell$ using (S, U) introduces errors when we update (S, U) to (S', U'). These errors can be corrected using the random mask from previous steps, under the SXDH assumption, to make the keys be in the correct form w.r.t (S', U'). Finally, because the challenge ciphertext no longer depends on b, the advantage becomes 0 and we can conclude. □

Proof (Of Theorem 7). We give the sequence of games in Figure 9. The changes that make the transitions between games are highlighted in gray. The advantage of an adversary $\mathcal{A}$ in a game G_i is denoted by

$$\mathsf{Adv}(\mathsf{G}_i) := |\Pr[\mathsf{G}_i = 1] - 1/2|$$

where the probability is taken over the random choices of $\mathcal{A}$ and coins of G_i.

Game G_0 : $a_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a_{\ell,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_{\ell,0}}(\mathbb{A})$, $\mathsf{pk} := \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\right)$, $\mathbf{F} \in \mathbb{G}_1^{(n+7)\times(n+7)}$, $\mathbf{H} \in \mathbb{G}_1^{(n+3)\times(n+3)}$

$$\begin{array}{lc|c|c|c|c|c|c|c|c|c|c}
\mathbf{c}_j & (& \sigma_j\cdot(1,-j) & \psi & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} & (& \pi_{\ell,j}\cdot(j,1) & a_{\ell,j}\cdot z & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{ll}
\mathbf{t}_i & [\![\omega\cdot(s_i+\mu u_i) + \mathbf{x}^*_b[i]]\!]_1 \\
\mathbf{m}^*_{\ell,i} & [\![\mathbf{y}_\ell[i]]\!]_2
\end{array}$$

$$\begin{array}{lc|c|c|c|c|c|c}
\mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & 0 & \cdots & 0 &)_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}z & 0 & \cdots & 0 &)_{\mathbf{H}^*}
\end{array}$$

Game G_1 : $r'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_b - \mathbf{x}^*_1$, $\mathsf{pk} := \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\right)$

$$\begin{array}{lc|c|c|c|c|c|c|c|c|c|c}
\mathbf{c}_j & (& \sigma_j\cdot(1,-j) & \psi & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} & (& \pi_{\ell,j}\cdot(j,1) & a_{\ell,j}\cdot z & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{lc|c|c|c|c|c|c}
\mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & r'_{\ell,0}\mathbf{y}_\ell[1] & \cdots & r'_{\ell,0}\mathbf{y}_\ell[n] &)_{\mathbf{H}^*}
\end{array}$$

Game G_2 : $\omega' \xleftarrow{\$} \mathbb{Z}_q$, $\mathsf{pk} := \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\right)$

$$\begin{array}{ll}
\mathbf{t}_i & \left[\!\!\left[\omega\cdot s_i + \omega'\cdot u_i + \mathbf{x}^*_b[i]\right]\!\!\right]_1 \\
\mathbf{m}^*_{\ell,i} & [\![\mathbf{y}_\ell[i]]\!]_2
\end{array}$$

$$\begin{array}{lc|c|c|c|c|c|c}
\mathbf{c}_{\mathsf{ipfe}} & (& \omega & \omega' & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & r'_{\ell,0}\mathbf{y}_\ell[1] & \cdots & r'_{\ell,0}\mathbf{y}_\ell[n] &)_{\mathbf{H}^*}
\end{array}$$

Game G_3 : $\omega', r''_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $\mathbf{s}' = \mathbf{s} + \Delta\mathbf{s}, \mathbf{u}' = \mathbf{u} + \Delta\mathbf{u}$, where $\Delta\mathbf{s}, \Delta\mathbf{u} \in \mathbb{Z}_q^n$ s.t. $\omega\cdot\Delta\mathbf{s} + \omega'\cdot\Delta\mathbf{u} = \mathbf{x}_b - \mathbf{x}_0$ and $\Delta\mathbf{s} + \mu\cdot\Delta\mathbf{u} = 0$, $\mathsf{pk} = \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\right)$

$$\begin{array}{ll}
\mathbf{t}_i & [\![\omega s'_i + \omega' u'_i + \mathbf{x}^*_0[i]]\!]_1 \\
\mathbf{m}^*_{\ell,i} & [\![\mathbf{y}_\ell[i]]\!]_2
\end{array}$$

$$\begin{array}{lc|c|c|c|c|c|c}
\mathbf{c}_{\mathsf{ipfe}} & (& \omega & \omega' & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s}',\mathbf{y}_\ell\rangle & \langle\mathbf{u}',\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & r''_{\ell,0}\mathbf{y}_\ell[1] & \cdots & r''_{\ell,0}\mathbf{y}_\ell[n] &)_{\mathbf{H}^*}
\end{array}$$

Fig. 9: Games for Theorem 7. The index i runs in $\{1,\ldots,n\}$. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries. The transition from G_0 to G_1 can be found in Lemma 24 in Appendix B.3, which will make use of the auxiliary vectors in $(\mathbf{F},\mathbf{F}^*)$ and $(\mathbf{H},\mathbf{H}^*)$.

Game G_0: This is the adaptive security game as given in Figure 4. We have $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}} = \mathsf{Adv}(\mathsf{G}_0)$.

Game G_1: In this game we introduce the masks in the key components $\mathbf{k}_{\ell,\mathsf{ipfe}}$:

$$\begin{aligned}
\mathbf{c}_j &= (\sigma_j\cdot(1,-j),\ \psi,\ 0,\cdots,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j}\cdot(j,1),\ a_{\ell,j}\cdot z,\ 0,\cdots,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ldots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S,\mathbf{y}_\ell\rangle,\ \langle U,\mathbf{y}_\ell\rangle,\ a_{\ell,0}\cdot z, r'_{\ell,0}\mathbf{y}_\ell[1],\ldots,r'_{\ell,0}\mathbf{y}_\ell[n])_{\mathbf{H}^*}
\end{aligned}$$

The transition from G_0 to G_1 is discussed separately in Lemma 24. The difference in advantages is

$$|\mathsf{Adv}(\mathsf{G}_1) - \mathsf{Adv}(\mathsf{G}_0)| \leq 2nK\cdot(P(6P+3)+2)\cdot\mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)\ .$$

Game G_2: In this game we replace the exponent $\mu\omega$ in the challenge ciphertext by a uniformly random ω'

$$\mathbf{t}_i = \left[\!\left[\omega \cdot s_i + \omega' \cdot u_i + \mathbf{x}_b^*[i]\right]\!\right]_1$$
$$\mathbf{c}_{\mathsf{ipfe}} = (\omega, \omega', \psi, \tau\Delta\mathbf{x}[1], \cdots, \tau\Delta\mathbf{x}[n])_{\mathbf{H}}$$

We note that if the attributes in the ciphertext satisfy some ℓ-th key's policy, it is still decryptable using this key. We double-check The change is indistinguishable under the adversary's view by a reduction to DDH in $\mathbb{G}_1$:

$$|\mathsf{Adv}(\mathsf{G}_2) - \mathsf{Adv}(\mathsf{G}_1)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) \ .$$

Game G_3: We are now ready to replace $\mathbf{x}_b^*[i]$ in the challenge ciphertext by $\mathbf{x}_0^*[i]$ making it not depend on b any more. The idea is similar to that of the proof for selective security. For all functional key queries, the simulator responds using the msk vectors (S, U), i.e. the component $\mathbf{k}_{\ell,\mathsf{ipfe}}$ is:

$$\mathbf{k}^*_{\ell,\mathsf{ipfe}} = (\langle S, \mathbf{y}_\ell\rangle, \ \langle U, \mathbf{y}_\ell\rangle, \ a_{\ell,0} \cdot z, \ r'_{\ell,0}\mathbf{y}_\ell[1], \cdots, \ r'_{\ell,0}\mathbf{y}_\ell[n])_{\mathbf{H}^*} \ .$$

Last but not least, we can require the adversary to query all functional keys conforming to the condition that $\mathbf{y}_\ell[1] \neq 0$. This does not reduce the power of the adversary because the entries of $\mathbf{y}_\ell$ that are 0 will not play any role in the final inner-product value.
When the adversary declares the challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$, the simulator updates the master secret vectors S, U to:

$$S' \coloneqq S + \Delta S$$
$$U' \coloneqq U + \Delta U$$

where $(\Delta S, \Delta U)$ satisfies:

$$\begin{aligned} &\Delta S + \mu\Delta U = 0 \\ &\omega \cdot \Delta S + \omega' \cdot \Delta U = \mathbf{x}_b - \mathbf{x}_0 \ . \end{aligned} \tag{4}$$

With overwhelming probability we can find a solution $(\Delta S, \Delta U)$ for the above system. It is straightforward to see that this change does not affect the public information pk that the adversary possesses, because $S + \mu U = S' + \mu U'$. The challenge ciphertext is now encrypting $\mathbf{x}_0^*$ under (S', U'), i.e.

$$\mathbf{t}_i = \left[\!\left[\omega s'_i + \omega' u'_i + \mathbf{x}_0^*[i]\right]\!\right]_1 = \left[\!\left[\omega s_i + \omega' u_i + \mathbf{x}_b^*[i]\right]\!\right]_1 \ .$$

Under this modification, the functional key component $\mathbf{k}^*_{\ell,\mathsf{ipfe}}$ becomes:

$$\mathbf{k}^*_{\ell,\mathsf{ipfe}} = (\langle S', \mathbf{y}_\ell\rangle - \langle \Delta S, \mathbf{y}_\ell\rangle, \ \langle U', \mathbf{y}_\ell\rangle - \langle \Delta U, \mathbf{y}_\ell\rangle, \ a_{\ell,0} \cdot z, r'_{\ell,0}\mathbf{y}_\ell[1], \ldots, r'_{\ell,0}\mathbf{y}_\ell[n])_{\mathbf{H}^*} \ .$$

Let us recall that $\Delta\mathbf{x} \coloneqq \mathbf{x}_1^* - \mathbf{x}_0^*$. We have to consider two cases:

- In the case $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle = 0$ or $b = 0$, there is no further changes to do because $S' = S$ and $U' = U$. The ℓ-th functional key still decrypts the challenge ciphertext to $\langle \mathbf{x}_b^*, \mathbf{y}_\ell\rangle = \langle \mathbf{x}_0^*, \mathbf{y}_\ell\rangle$ if the policy is satisfied by the ciphertext's attributes.
- In the case $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$ and $b = 1$, we need to remove the noises $\langle \Delta S, \mathbf{y}_\ell\rangle$ and $\langle \Delta U, \mathbf{y}_\ell\rangle$ so that the functional keys have the correct form w.r.t the new master secret vectors (S', U') and work as expected for normal ciphertexts that can be generated by the adversary using pk, including the group elements $[\![s'_i + \mu u'_i]\!]_1$. We note that the decryption of the challenge ciphertext is not taken into account anymore because the security model prohibits the access structure from being satisfied by the challenge attributes in the current case.

First we switch ω' back to $\mu\omega$. This change is indistinguishable under DDH:

$$\mathbf{t}_i = \left[\!\left[\omega \cdot s_i + \mu\omega \cdot u_i + \mathbf{x}_b^*[i]\right]\!\right]_1$$
$$\mathbf{c}_{\mathsf{ipfe}} = (\omega, \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ \cdots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}}$$

Then, given a DSDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $c - ab = \rho$ for either $\rho = \langle \Delta U, \mathbf{y}_\ell \rangle$ or $\rho = 0$, we perform a basis change on $(\mathbf{H}, \mathbf{H}^*)$ using:

$$H \coloneqq \begin{bmatrix} 1 & 0 & a\mu \\ 0 & 1 & -a \\ 0 & 0 & 1 \end{bmatrix}_{1,2,4} \qquad H' \coloneqq \left(H^{-1}\right)^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a\mu & a & 1 \end{bmatrix}_{1,2,4}$$
$$\mathbf{H} = H \cdot \mathbf{T}; \qquad \mathbf{H}^* = H' \cdot \mathbf{T}^* \ .$$

This changes $\mathbf{h}_1, \mathbf{h}_2$ and we do not have $[\![a]\!]_1$ to compute the full basis $\mathbf{H}$ but all the adversary sees from pk is $\mathbf{h}_1 + \mu\mathbf{h}_2$, which stays invariant. The vector $\mathbf{h}_4^*$ is also affected but it is already hidden from the adversary. The ciphertext component can be written directly in $\mathbf{T}$:

$$\begin{aligned}\mathbf{c}_{\mathsf{ipfe}} &= (\omega, \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ \cdots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{T}} \\ &= (\omega, \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1] - a\mu\omega + a\mu\omega,\ \tau\Delta\mathbf{x}[2],\ \cdots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}} \\ &= (\omega, \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ \tau\Delta\mathbf{x}[2],\ \cdots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}}\end{aligned}$$

and indeed $\mathbf{c}_{\mathsf{ipfe}}$ can still be simulated correctly. The key component $\mathbf{k}^*_{\ell,\mathsf{ipfe}}$ can be written:

$$\begin{aligned}\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S', \mathbf{y}_\ell\rangle - \langle \Delta S, \mathbf{y}_\ell\rangle,\ \langle U', \mathbf{y}_\ell\rangle - \langle \Delta U, \mathbf{y}_\ell\rangle,\ a_{\ell,0} \cdot z,\ r'_{\ell,0}\mathbf{y}_\ell[1], \ldots,\ r'_{\ell,0}\mathbf{y}_\ell[n])_{\mathbf{H}^*} \\ &+ \quad \left(-\mu c,\ c,\ 0,\ b,\ b \cdot \frac{\mathbf{y}_\ell[2]}{\mathbf{y}_\ell[1]} \ldots, b \cdot \frac{\mathbf{y}_\ell[n]}{\mathbf{y}_\ell[1]}\right)_{\mathbf{T}^*} \\ &= (\langle S', \mathbf{y}_\ell\rangle - \langle \Delta S, \mathbf{y}_\ell\rangle,\ \langle U', \mathbf{y}_\ell\rangle - \langle \Delta U, \mathbf{y}_\ell\rangle,\ a_{\ell,0} \cdot z,\ r'_{\ell,0}\mathbf{y}_\ell[1], \ldots,\ r'_{\ell,0}\mathbf{y}_\ell[n])_{\mathbf{H}^*} \\ &+ \quad \left(-\mu\rho,\ \rho,\ 0,\ b,\ b \cdot \frac{\mathbf{y}_\ell[2]}{\mathbf{y}_\ell[1]} \ldots, b \cdot \frac{\mathbf{y}_\ell[n]}{\mathbf{y}_\ell[1]}\right)_{\mathbf{H}^*} \\ &= (\langle S', \mathbf{y}_\ell\rangle - \langle \Delta S, \mathbf{y}_\ell\rangle - \mu\rho,\ \langle U', \mathbf{y}_\ell\rangle - \langle \Delta U, \mathbf{y}_\ell\rangle + \rho,\ a_{\ell,0} \cdot z, \\ &\quad \left(r'_{\ell,0} + \frac{b}{\mathbf{y}_\ell[1]}\right)\mathbf{y}_\ell[1], \ldots,\ \left(r'_{\ell,0} + \frac{b}{\mathbf{y}_\ell[1]}\right)\mathbf{y}_\ell[n])_{\mathbf{H}^*} \ .\end{aligned}$$

The randomness $r'_{\ell,0}$ is updated to $r'_{\ell,0} + b/\mathbf{y}_\ell[1]$. If $\rho = \langle \Delta U, \mathbf{y}_\ell\rangle$ we are cleaning the noises using the relation (4), otherwise we are not. Finally, we switch back ω' to $\mu\omega$ in the challenge ciphertext to arrive at G_3. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}(\mathsf{G}_2)| \leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

The challenge ciphertext in G_3 does not depend on b anymore and as a result $\mathsf{Adv}(\mathsf{G}_3) = 0$. We have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}(1^\lambda) &= \mathsf{Adv}(\mathsf{G}_0) \\ &= |\mathsf{Adv}(\mathsf{G}_0) - \mathsf{Adv}(\mathsf{G}_3)| \\ &\leq \sum_{i=1}^{3} |\mathsf{Adv}(\mathsf{G}_i) - \mathsf{Adv}(\mathsf{G}_{i-1})| \\ &\leq 2nK \cdot (P(6P+3)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \\ &\quad + \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda) \\ &\leq (2nK \cdot (P \cdot (6P+3)+2)+5) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .\end{aligned}$$

The proof is concluded. □

Lemma 24. *Assuming the* SXDH *assumption for* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, the difference between advantages* $|\mathsf{Adv}(\mathsf{G}_1) - \mathsf{Adv}(\mathsf{G}_0)|$ *in Theorem 7 is negligible.*

Proof. We recall the form of ciphertext and functional key components:

$$\begin{aligned}
\mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ 0, \cdots,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j} \cdot (j, 1),\ a_{\ell,j} \cdot z,\ 0, \cdots,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ 0, \ldots,\ 0)_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S, \mathbf{y}_\ell \rangle,\ \langle U, \mathbf{y}_\ell \rangle,\ a_{\ell,0} \cdot z, 0, \ldots, 0)_{\mathbf{H}^*}
\end{aligned}$$

We use a sequence of games indexed by $\ell \in \{0, \ldots, K\}$ corresponding to the ordered list of K functional key queries. In $\mathsf{G}_{0.\ell}$, the first ℓ key queries are responded with the semi-functional form of G_1 and it holds that $\mathsf{G}_{0.0} = \mathsf{G}_0$ while $\mathsf{G}_{0.K} = \mathsf{G}_1$. Consequently, for $\ell \in [K]$ and without any confusion, the game $\mathsf{G}_{0.\ell-1}$ is understood as the predecessor of $\mathsf{G}_{0.\ell}$ in the sequence of hybrids $(\mathsf{G}_{0.0}, \mathsf{G}_{0.1}, \ldots, \mathsf{G}_{0.K})$. The sequence of games from $\mathsf{G}_{0.\ell-1}$ to $\mathsf{G}_{0.\ell}$ is depicted in Figure 10. The details are given below:

Game $\mathsf{G}_{0.\ell-1.0}$: This is the game $\mathsf{G}_{0.\ell-1}$.

Game $\mathsf{G}_{0.\ell-1.1}$: We first apply Lemma 4 to the vectors $((\mathbf{c}_j)_j, \mathbf{c}_{\mathsf{ipfe}}), ((\mathbf{k}^*_j)_j, \mathbf{k}^*_{\mathsf{ipfe}})$, where for each $i \in [n]$, we introduce $\tau z_j \Delta\mathbf{x}[i]$ in the coordinate $(i+3)$ of $\mathbf{c}_{\mathsf{ipfe}}$ as well as $a'_0 \mathbf{y}_\ell[i]$ and $a'_j \mathbf{y}_\ell[i]/z_j$ in the coordinate $(i+3)$ of $\mathbf{k}_{\ell,\mathsf{ipfe}}$ and $\mathbf{k}_{\ell,j}$, respectively. The values $z_j \xleftarrow{\$} \mathbb{Z}_q$ are sampled uniformly at random and indexed by attributes j. The application of the lemma makes use of the $(n+4, n+5, n+6, n+7)$-th *hidden* vectors in the bases $(\mathbf{F}, \mathbf{F}^*)$. More precisely, we use a sequence of hybrids $\mathsf{G}_{0.\ell-1.0.i}$ where i runs over $\{0, \ldots, n\}$ so that $\mathsf{G}_{0.\ell-1.0.0} = \mathsf{G}_{0.\ell-1.0}$ and for $i \geq 1$ the indices $(4, 5, \ldots, i+3)$ in $((\mathbf{c}_j)_j, (\mathbf{k}^*_j)_j)$ as well as $(\mathbf{c}_{\mathsf{ipfe}}, \mathbf{k}^*_{\mathsf{ipfe}})$ are masked according to Lemma 4. In the end $\mathsf{G}_{0.\ell-1.0.n} = \mathsf{G}_{0.\ell-1.1}$. The ciphertext and functional key components in $\mathsf{G}_{0.\ell-1.0.i}$, where $i \in [n]$, are:

$$\begin{aligned}
\mathbf{c}_j &= (\sigma_j \cdot (1, -j),\ \psi,\ \tau z_j \Delta\mathbf{x}[1], \ldots,\ \tau z_j \Delta\mathbf{x}[i],\ \overbrace{0, \ldots, 0}^{n-i \text{ coord.'s}}, 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j} \cdot (j, 1),\ a_{\ell,j} \cdot z,\ a'_j \mathbf{y}_\ell[1]/z_j, \ldots,\ a'_j \mathbf{y}_\ell[i]/z_j,\ 0, \ldots, 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1], \ldots,\ \tau\Delta\mathbf{x}[i],\ \overbrace{0, \ldots, 0}^{n-i \text{ coord.'s}})_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S, \mathbf{y}_\ell \rangle,\ \langle U, \mathbf{y}_\ell \rangle,\ a_{\ell,0} \cdot z,\ a'_0 \mathbf{y}_\ell[1], \ldots,\ a'_0 \mathbf{y}_\ell[i],\ 0, \ldots, 0)_{\mathbf{H}^*} \ .
\end{aligned}$$

For each $i \in [n]$, in order to go from $\mathsf{G}_{0.\ell-1.0.i-1}$ to $\mathsf{G}_{0.\ell-1.0.i}$, Lemma 4 is applied on coordinates $(1, 2, n+4, n+5, i+3, n+6, n+7)$ of $(\mathbf{F}, \mathbf{F}^*)$ together with coordinates $(3, i+3)$ of $(\mathbf{H}, \mathbf{H}^*)$. We note that due to the reuse of the auxiliary coordinates $(n+4, n+5, n+6, n+7)$ in $\mathbf{F}, \mathbf{F}^*$, which will be affected by the basis changes of Lemma 4, we cannot introduce all n masks at once but rather one by one while cleaning after masking the $(i+3-1)$-th coordinate before using them to mask the $(i+3)$-th. Throughout the hybrids, the functional key is still capable of decrypting the challenge ciphertext if the key's policy is satisfied, thanks to the fact that the masks $(a'_j)_j$ is a random labeling of a'_0. For each $i \in [n]$, we have

$$|\mathsf{Adv}(\mathsf{G}_{0.\ell-1.0.i}) - \mathsf{Adv}(\mathsf{G}_{0.\ell-1.0.i-1})| \leq (P \cdot (6P+3) + 2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$$

and hence

$$|\mathsf{Adv}(\mathsf{G}_{0.\ell-1.1}) - \mathsf{Adv}(\mathsf{G}_{0.\ell-1.0})| \leq n \cdot (P \cdot (6P+3) + 2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .$$

Game $\mathsf{G}_{0.\ell-1.0}$: $a_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a_{\ell,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_{\ell,0}}(\mathbb{A})$, $\mathsf{pk} := \left(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\right)$, $\mathbf{F} \in \mathbb{G}_1^{(n+7)\times(n+7)}$, $\mathbf{H} \in \mathbb{G}_1^{(n+3)\times(n+3)}$

$$\begin{array}{cc|c|c|c|c|c|c|c|c|c|c} \mathbf{c}_j & (& \sigma_j\cdot(1,-j) & \psi & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}^*_{\ell,j} & (& \pi_{\ell,j}\cdot(j,1) & a_{\ell,j}\cdot z & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{cc|c|c} \mathbf{t}_i & & [\![\omega\cdot(s_i+\mu u_i)+\mathbf{x}^*_b[i]]\!]_1 \\ \mathbf{m}^*_{\ell,i} & & [\![\mathbf{y}_\ell[i]]\!]_2 \end{array}$$

$$\begin{array}{cc|c|c|c|c|c|c} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & 0 & \cdots & 0 &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}z & 0 & \cdots & 0 &)_{\mathbf{H}^*} \end{array}$$

The hybrids $\mathsf{G}_{0.\ell-1.0.i}$ indexed by $i \in [n]$ to go from $\mathsf{G}_{0.\ell-1.0}$ to $\mathsf{G}_{0.\ell-1.1}$

$$\begin{array}{cc|c|c|c|c|c|c|c|c|c|c|c} \mathbf{c}_j & (& \cdots & \psi & \tau\Delta\mathbf{x}[1]z_j & \cdots & \tau\Delta\mathbf{x}[i]z_j & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}^*_{\ell,j} & (& \cdots & a_{\ell,j}\cdot z & a'_{\ell,j}\mathbf{y}_\ell[1]/z_j & \cdots & a'_{\ell,j}\mathbf{y}_\ell[i]/z_j & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{cc|c|c|c|c|c|c|c|c} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[i] & 0 & \cdots & 0 &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & a'_{\ell,0}\mathbf{y}_\ell[1] & \cdots & a'_{\ell,0}\mathbf{y}_\ell[i] & 0 & \cdots & 0 &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_{0.\ell-1.1}$:

$$\begin{array}{cc|c|c|c|c|c|c|c|c|c|c} \mathbf{c}_j & (& \sigma_j\cdot(1,-j) & \psi & \tau\Delta\mathbf{x}[1]z_j & \cdots & \tau\Delta\mathbf{x}[n]z_j & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}^*_{\ell,j} & (& \pi_{\ell,j}\cdot(j,1) & a_{\ell,j}\cdot z & a'_{\ell,j}\mathbf{y}_\ell[1]/z_j & \cdots & a'_{\ell,j}\mathbf{y}_\ell[n]/z_j & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{cc|c|c|c|c|c|c|c} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & a'_{\ell,0}\mathbf{y}_\ell[1] & \cdots & a'_{\ell,0}\mathbf{y}_\ell[n] &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_{0.\ell-1.2}$: $r'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{cc|c|c|c|c|c|c|c|c|c} \mathbf{c}_j & (& \sigma_j\cdot(1,-j) & \psi & \tau\Delta\mathbf{x}[1]z_j & \cdots & \tau\Delta\mathbf{x}[n]z_j & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}^*_{\ell,j} & (& \pi_{\ell,j}\cdot(j,1) & a_{\ell,j}\cdot z & a'_{\ell,j}\mathbf{y}_\ell[1]/z_j & \cdots & a'_{\ell,j}\mathbf{y}_\ell[n]/z_j & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{cc|c|c|c|c|c|c|c} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & (a'_{\ell,0}+r'_{\ell,0})\mathbf{y}_\ell[1] & \cdots & (a'_{\ell,0}+r'_{\ell,0})\mathbf{y}_\ell[n] &)_{\mathbf{H}^*} \end{array}$$

The hybrids $\mathsf{G}_{0.\ell-1.2.i}$ indexed by $i \in [n]$ to go from $\mathsf{G}_{0.\ell-1.2}$ to $\mathsf{G}_{0.\ell-1.3}$

$$\begin{array}{cc|c|c|c|c|c|c|c|c|c|c|c} \mathbf{c}_j & (& \ldots & \psi & 0 & \ldots & 0 & \tau\Delta\mathbf{x}[i+1]z_j & \ldots & \tau\Delta\mathbf{x}[n]z_j & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}^*_{\ell,j} & (& \ldots & a_{\ell,j}z & 0 & \ldots & 0 & a'_{\ell,j}\mathbf{y}_\ell[i+1]/z_j & \ldots & a'_{\ell,j}\mathbf{y}_\ell[n]/z_j & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{cc|c|c|c|c|c|c|c} \mathbf{c}_{\mathsf{ipfe}} & (& \ldots & \psi & \tau\Delta\mathbf{x}[1] & \ldots & \tau\Delta\mathbf{x}[i] & \tau\Delta\mathbf{x}[i+1] & \ldots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \ldots & a_{\ell,0}z & r'_{\ell,0}\mathbf{y}_\ell[1] & \ldots & r'_{\ell,0}\mathbf{y}_\ell[i] & (a'_{\ell,0}+r'_{\ell,0})\mathbf{y}_\ell[i+1] & \ldots & (a'_{\ell,0}+r'_{\ell,0})\mathbf{y}_\ell[n] &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_{0.\ell-1.3}$: $r'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$

$$\begin{array}{cc|c|c|c|c|c|c|c|c|c|c} \mathbf{c}_j & (& \sigma_j\cdot(1,-j) & \psi & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\ \mathbf{k}^*_{\ell,j} & (& \pi_{\ell,j}\cdot(j,1) & a_{\ell,j}\cdot z & 0 & \cdots & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*} \end{array}$$

$$\begin{array}{cc|c|c|c|c|c|c|c} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \mu\omega & \psi & \tau\Delta\mathbf{x}[1] & \cdots & \tau\Delta\mathbf{x}[n] &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle\mathbf{s},\mathbf{y}_\ell\rangle & \langle\mathbf{u},\mathbf{y}_\ell\rangle & a_{\ell,0}\cdot z & r'_{\ell,0}\mathbf{y}_\ell[1] & \cdots & r'_{\ell,0}\mathbf{y}_\ell[n] &)_{\mathbf{H}^*} \end{array}$$

Fig. 10: Games for Lemma 24. The index i runs in $\{1,\ldots,n\}$. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries.

Game $\mathsf{G}_{0.\ell-1.2}$: After masking all the key components and ciphertext components with another random labeling, the vectors become:

$$\begin{aligned}
\mathbf{c}_j &= (\sigma_j \cdot (1,-j),\ \psi,\ \tau z_j \Delta\mathbf{x}[1],\ldots,\ \tau z_j \Delta\mathbf{x}[n],\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j} \cdot (j,1),\ a_{\ell,j} \cdot z,\ a'_j \mathbf{y}_\ell[1]/z_j,\ldots,\ a'_j \mathbf{y}_\ell[n]/z_j,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ldots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S,\mathbf{y}_\ell\rangle,\ \langle U,\mathbf{y}_\ell\rangle,\ a_{\ell,0}\cdot z,\ a'_0\mathbf{y}_\ell[1],\ldots,\ a'_0\mathbf{y}_\ell[n])_{\mathbf{H}^*}
\end{aligned}$$

In this game we randomize $a'_{\ell,0}$ in $\mathbf{k}^*_{\ell,\mathsf{ipfe}}$ by a uniform mask $r_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$:

$$\mathbf{k}^*_{\ell,\mathsf{ipfe}} = (\langle S,\mathbf{y}_\ell\rangle,\ \langle U,\mathbf{y}_\ell\rangle,\ a_{\ell,0}\cdot z,\ \boxed{(a'_0 + r'_{\ell,0})}\cdot\mathbf{y}_\ell[1],\ldots,\ \boxed{(a'_0 + r'_{\ell,0})}\cdot\mathbf{y}_\ell[n])_{\mathbf{H}^*}\ .$$

This *all-at-once* change is done for every functional key responded to the adversary. We consider two cases:

- If $\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle \neq 0$, the security model implies that $\mathbb{A}(\mathsf{S}) = 0$ where $\mathbb{A}$ is the access structure embedded in the key and S contains the attributes in the challenge ciphertext. Hence, for all $i \in [n]$, there is no way to find a reconstruction vector $(c_j)_j$ for an authorized set $A \subseteq \mathsf{S}$, i.e. there are not enough $a'_{\ell,j}\cdot\mathbf{y}_\ell[i]/z_j$ from the ℓ-th functional key to recover

$$\sum_{j\in A}\frac{c_j a'_{\ell,j}\cdot\mathbf{y}_\ell[i]}{z_j}\cdot\tau z_j\Delta\mathbf{x}[i] = \tau a'_{\ell,0}\mathbf{y}_\ell[i]\Delta\mathbf{x}[i]\ .$$

  Furthermore, because $(a'_{\ell,j})_j$ is a random labeling of $a'_{\ell,0}$ using the LSSS of $\mathbb{A}$ and $\tau, z_j \xleftarrow{\$} \mathbb{Z}_q$, then all $(a'_j)_j$ are randomized into $(a'_j/z_j)_j$ and become independent uniformly random values, it holds that in this case, masking a'_0 by r'_0 is perfectly indistinguishable under the adversary's view, even an unbounded one.
- If $\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle = 0$, changing $a'_{\ell,0}$ to $a'_0 + r'_{\ell,0}$ does not affect the view of the adversary. The case of functional keys that are not satisfied by the challenge attributes is argued as above. We now concentrate on the keys that can decrypt correctly the challenge ciphertext. Firstly, the vectors of the dual bases are all hidden from the adversary. Even when multiplying the key with the ciphertext vectors, the best an (even unbounded) adversary can learn is:

$$\begin{aligned}
&\log_{g_\mathsf{t}}(\mathbf{k}^*_{\ell,\mathsf{ipfe}} \times \mathbf{c}_{\mathsf{ipfe}}) \\
&= \omega\langle S,\mathbf{y}_\ell\rangle + \mu\omega\langle U,\mathbf{y}_\ell\rangle + \psi a_{\ell,0} z + \textstyle\sum_{i=1}^n \tau(a'_{\ell,0} + r'_{\ell,0})\mathbf{y}_\ell[i]\Delta\mathbf{x}[i] \\
&= \omega\langle S,\mathbf{y}_\ell\rangle + \mu\omega\langle U,\mathbf{y}_\ell\rangle + \psi a_{\ell,0} z + \tau(a'_{\ell,0} + r'_{\ell,0})\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle \\
&= \omega\langle S,\mathbf{y}_\ell\rangle + \mu\omega\langle U,\mathbf{y}_\ell\rangle + \psi a_{\ell,0} z \\
&\log_{g_\mathsf{t}}\left(\textstyle\sum_{j\in A}(c_j\cdot\mathbf{k}^*_{\ell,j}) \times \mathbf{c}_j\right) \\
&= \textstyle\sum_{j\in A}\psi c_j a_{\ell,j} z + \sum_{i=1}^n\left(\sum_{j\in A}\frac{c_j a'_{\ell,j}\cdot\mathbf{y}_\ell[i]}{z_j}\cdot\tau z_j\Delta\mathbf{x}[i]\right) \\
&= \psi a_{\ell,0} z + a'_{\ell,0}\tau\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle \\
&= \psi a_{\ell,0} z
\end{aligned}$$

  where $A \subseteq \mathsf{S}$ is an authorized set and $(c_j)_j$ is its reconstruction vector obtained from LSSS. The result does not depend on $a'_{\ell,0}$ anymore.

In total, changing $(a'_{\ell,0}\mathbf{y}_\ell[i])_{i\in[n]}$ to $((a'_{\ell,0} + r'_{\ell,0})\mathbf{y}_\ell[i])_{i\in[n]}$, for all i at once, is perfectly indistinguishable under the adversary's view, even an unbounded one. Thus, we have $\mathsf{Adv}(\mathsf{G}_{0.\ell-1.2}) = \mathsf{Adv}(\mathsf{G}_{0.\ell-1.1})$.

Game $\mathsf{G}_{0.\ell-1.3}$: In this game we clean the masks in the vectors $\mathbf{c}_j$ and $\mathbf{k}_{\ell,j}$. This process of cleaning is done via basis changes on $(\mathbf{F},\mathbf{F}^*)$ similar to what is done from $\mathsf{G}_{0.\ell-1.0}$ to $\mathsf{G}_{0.\ell-1.1}$

but in reverse order:

$$\begin{aligned}
\mathbf{c}_j &= (\sigma_j \cdot (1,-j),\ \psi,\ \boxed{0},\ldots,\ \boxed{0},\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j} \cdot (j,1),\ a_{\ell,j} \cdot z,\ \boxed{0},\ldots,\ \boxed{0},\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ldots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S,\mathbf{y}_\ell\rangle,\ \langle U,\mathbf{y}_\ell\rangle,\ a_{\ell,0} \cdot z,\ \boxed{r'_{\ell,0}} \cdot \mathbf{y}_\ell[1],\ldots,\ \boxed{r'_{\ell,0}} \cdot \mathbf{y}_\ell[n])_{\mathbf{H}^*}\ .
\end{aligned}$$

Similar to what we do to go from $\mathsf{G}_{0.\ell-1.0}$ to $\mathsf{G}_{0.\ell-1.1}$, we proceed by a sequence of hybrids, indexed by $i \in \{0,1,\ldots,n\}$. We recall the reason for this sequence of n hybrids is the fact that there are only 4 hidden vectors in the basis that we can use, so we cannot apply Lemma 4 for 2 indices $(i+3) \neq (j+3)$ at the same time over the same basis changes w.r.t $(\mathbf{H},\mathbf{H}^*),(\mathbf{F},\mathbf{F}^*)$.

Game $\mathsf{G}_{0.\ell-1.2.i}$: the ciphertext and key components has the form:

$$\begin{aligned}
\mathbf{c}_j &= (\sigma_j \cdot (1,-j),\ \psi,\ \overbrace{\boxed{0},\ldots,\ \boxed{0}}^{i \text{ coordinates}}, \\
&\qquad \tau z_j\Delta\mathbf{x}[i+1],\ldots,\ \tau z_j\Delta\mathbf{x}[n],\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \\
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j} \cdot (j,1),\ a_{\ell,j} \cdot z,\ \overbrace{\boxed{0},\ldots,\ \boxed{0}}^{i \text{ coordinates}}, \\
&\qquad a'_j\mathbf{y}_\ell[i+1]/z_j,\ldots,\ a'_j\mathbf{y}_\ell[n]/z_j,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau\Delta\mathbf{x}[1],\ldots,\ \tau\Delta\mathbf{x}[i],\ \tau\Delta\mathbf{x}[i+1],\ldots,\ \tau\Delta\mathbf{x}[n])_{\mathbf{H}} \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle S,\mathbf{y}_\ell\rangle,\ \langle U,\mathbf{y}_\ell\rangle,\ a_{\ell,0} \cdot z,\ \overbrace{\boxed{r'_{\ell,0}} \cdot \mathbf{y}_\ell[1],\ldots,\ \boxed{r'_{\ell,0}} \cdot \mathbf{y}_\ell[i]}^{i \text{ coordinates}}, \\
&\qquad (a'_0 + r'_{\ell,0})\mathbf{y}_\ell[i+1],\ldots,\ (a'_{\ell,0} + r'_{\ell,0})\mathbf{y}_\ell[n])_{\mathbf{H}^*}\ .
\end{aligned}$$

We note that the decryption using the ℓ-th functional key still works if the attributes of the challenge ciphertext satisfy the key's policy because: let $A \subseteq \mathsf{S}$ be an authorized set and $(c_j)_j$ be its reconstruction vector from LSSS

$$\begin{aligned}
&\log_{g_\mathsf{t}}(\mathbf{k}^*_{\ell,\mathsf{ipfe}} \times \mathbf{c}_{\mathsf{ipfe}}) \\
&= \omega\langle S,\mathbf{y}_\ell\rangle + \mu\omega\langle U,\mathbf{y}_\ell\rangle + \psi a_{\ell,0}z + \textstyle\sum_{k=1}^{i} \tau r'_{\ell,0}\mathbf{y}_\ell[k]\Delta\mathbf{x}[k] \\
&\quad + \textstyle\sum_{k=i+1}^{n} \tau(a'_{\ell,0} + r'_{\ell,0})\mathbf{y}_\ell[k]\Delta\mathbf{x}[k] \\
&= \omega\langle S,\mathbf{y}_\ell\rangle + \mu\omega\langle U,\mathbf{y}_\ell\rangle + \psi a_{\ell,0}z + \tau r'_{\ell,0}\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle + \textstyle\sum_{k=i+1}^{n} \tau a'_{\ell,0}\mathbf{y}_\ell[k]\Delta\mathbf{x}[k] \\
&= \omega\langle S,\mathbf{y}_\ell\rangle + \mu\omega\langle U,\mathbf{y}_\ell\rangle + \psi a_{\ell,0}z + \textstyle\sum_{k=i+1}^{n} \tau a'_{\ell,0}\mathbf{y}_\ell[k]\Delta\mathbf{x}[k] \\
&\log_{g_\mathsf{t}}\left(\textstyle\sum_{j\in A}(c_j \cdot \mathbf{k}^*_{\ell,j}) \times \mathbf{c}_j\right) \\
&= \textstyle\sum_{j\in A} \psi c_j a_{\ell,j}z + \sum_{k=i+1}^{n}\left(\sum_{j\in A} \frac{c_j a'_{\ell,j}\cdot\mathbf{y}_\ell[k]}{z_j} \cdot \tau z_j\Delta\mathbf{x}[k]\right) \\
&= \psi a_{\ell,0}z + \textstyle\sum_{k=i+1}^{n} \tau a'_{\ell,0}\mathbf{y}_\ell[k]\Delta\mathbf{x}[k]
\end{aligned}$$

and the security model requires that $\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle = 0$ in this case.

For each $i \in [n]$, in order to go from the hybrid $\mathsf{G}_{0.\ell-1.2.i-1}$ to $\mathsf{G}_{0.\ell-1.2.i}$, we apply Lemma 4 for the coordinates $(1,2,3,n+4,n+5,i+3,n+6,n+7)$ of $(\mathbf{F},\mathbf{F}^*)$ together with coordinates $(3,i+3)$ of $(\mathbf{H},\mathbf{H}^*)$. Finally, the difference in advantages is

$$|\mathsf{Adv}(\mathsf{G}_{0.\ell-1.3}) - \mathsf{Adv}(\mathsf{G}_{0.\ell-1.2})| \leq n \cdot (P \cdot (6P+3)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)\ .$$

We perform the above sequence of games for each ℓ-th functional key and in the end we arrive at $\mathsf{G}_{0.K} = \mathsf{G}_1$. The difference in advantages is

$$|\mathsf{Adv}(\mathsf{G}_1) - \mathsf{Adv}(\mathsf{G}_0)| \leq 2nK \cdot (P(6P+3)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$$

and the proof is completed. □

B.4 Proof of Theorem 14

Theorem 14. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be a multi-client IPFE scheme with fine-grained access control via LSSS, constructed in Section 5.2 in the setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$. Then, $\mathcal{E}$ is one-time IND-secure if the* SXDH *assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$. More specifically, let K denote the number of functional key queries, P denote the maximum number of attributes in the access structure $\mathbb{A}$ queried for functional keys, and Q denote the number of random oracle (RO) queries. We have the following bound:*

$$\mathsf{Adv}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}(1^\lambda) \leq (2KP \cdot (6P+3) + 2K + 2Q + 5) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$$

and in the reduction there is an additive loss $\mathcal{O}(Q \cdot t_{\mathbb{G}_1})$ in time, where $t_{\mathbb{G}_1}$ is the cost for one addition in $\mathbb{G}_1$.

Proof (Of Theorem 14). The sequence of games can be found in Figure 11 and 12. The full-domain hash function $\mathsf{H} : \mathsf{Tag} \times 2^{\mathsf{Att}} \to \mathbb{G}_1^2$ is modeled as a random oracle and we denote by Q the number of random oracle queries by the adversary. The changes that make the transitions between games are highlighted in gray. The advantage of an adversary $\mathcal{A}$ in a game G_i is denoted by

$$\mathsf{Adv}(\mathsf{G}_i) \coloneqq |\Pr[\mathsf{G}_i = 1] - 1/2|$$

where the probability is taken over the random choices of $\mathcal{A}$ and coins of G_i.

The details of the games are given below. We start from the adaptive security game. In the subsequent games, we give details of the basis change and explain how they can be done in parallel, in the spirit of our *duplicate-and-compress* technique.

Game G_0: This is the adaptive security game. The simulator generates all dual basis pairs

$$\mathbf{H}_i = (\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \mathbf{h}_{i,3}, \mathbf{h}_{i,4}) \qquad \mathbf{H}^*_i = (\mathbf{h}^*_{i,1}, \mathbf{h}^*_{i,2}, \mathbf{h}^*_{i,3}, \mathbf{h}^*_{i,4})$$
$$\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_6, \mathbf{f}_7, \mathbf{f}_8) \qquad \mathbf{F}^* = (\mathbf{f}^*_1, \mathbf{f}^*_2, \mathbf{f}^*_3, \mathbf{f}^*_4, \mathbf{f}^*_5, \mathbf{f}^*_6, \mathbf{f}^*_7, \mathbf{f}^*_8)$$

and sets

$$\begin{cases} \mathsf{msk} \coloneqq (S,\ U,\ \mathbf{f}^*_1,\ \mathbf{f}^*_2,\ \mathbf{f}^*_3,\ (\mathbf{h}^*_{i,1}, \mathbf{h}^*_{i,2}, \mathbf{h}^*_{i,3})_{i\in[n]}) \\ \mathsf{ek}_i \coloneqq (s_i,\ u_i,\ p_i \cdot H^{(1)}_i,\ p_i \cdot H^{(2)}_i,\ \mathbf{h}_{i,3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3) \text{ for } i \in [n] \end{cases}$$

where $H^{(k)}_i$ denotes the k-th row of H_i.

Extract$(\mathbb{A}, \mathbf{y}^{(\ell)})$: For the ℓ-th functional key query w.r.t an access structure $\mathbb{A}$ and a vector $\mathbf{y}^{(\ell)} \in \mathbb{Z}_q^n$ that specifies the inner product function $F_{\mathbf{y}^{(\ell)}}$, for each $i \in [n]$ the simulator samples $a^{(\ell)}_0 \xleftarrow{\$} \mathbb{Z}_q$, constructs the associated LSSS and runs the labeling algorithm to obtain the labels $(a^{(\ell)}_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a^{(\ell)}_0}(\mathbb{A})$. It then returns

$$\mathbf{k}^{(\ell)}_{i,j} \coloneqq (\pi^{(\ell)}_{i,j} \cdot (j, 1),\ a^{(\ell)}_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$
$$\mathbf{m}^{(\ell)}_i \coloneqq [\![\mathbf{y}^{(\ell)}[i]]\!]_2 \text{ for } i \in [n]$$
$$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} \coloneqq (\langle S, \mathbf{y}^{(\ell)}\rangle,\ \langle U, \mathbf{y}^{(\ell)}\rangle,\ a^{(\ell)}_0 \cdot z,\ 0)_{\mathbf{H}^*_i}$$

where $\pi^{(\ell)}_{i,j} \xleftarrow{\$} \mathbb{Z}_q$.

LoR$(i, \mathbf{x}^*_0, \mathbf{x}^*_1, \mathsf{tag}, \mathsf{S})$: As described in Figure 2, the $(\mathsf{tag}, \mathsf{S})$ of the first **LoR** query will determine the only challenge tag from then on. Upon receiving a set $\mathsf{S} \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ of attributes, the simulator samples $\psi_i \xleftarrow{\$} \mathbb{Z}_q$, flips a coin $b \xleftarrow{\$} \{0,1\}$, compute $\mathsf{H}(\mathsf{tag}, \mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1) \in \mathbb{G}_1^2$ and

$$\mathbf{t}_i \coloneqq [\![\omega_{\mathsf{tag},\mathsf{S}} \cdot s_i + \omega'_{\mathsf{tag},\mathsf{S}} u_i + \mathbf{x}^*_b[i]]\!]_1$$
$$\mathbf{c}_{i,\mathsf{ipfe}} \coloneqq (\omega_{\mathsf{tag},\mathsf{S}},\ \omega'_{\mathsf{tag},\mathsf{S}},\ \psi_i,\ 0)_{\mathbf{H}_i}$$

Game G_0 : $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$, $a_0^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $(a_j^{(\ell)})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0^{(\ell)}}(\mathbb{A})$

$$\forall\, i \in \mathcal{C}\cup\mathcal{H}\ \ \mathsf{ek}_i\ \ (s_i,\quad u_i,\quad p_i\cdot H_i^{(1)},\quad p_i\cdot H_i^{(2)},\quad \mathbf{h}_{i,3},\quad \mathbf{f}_1,\quad \mathbf{f}_2,\quad \mathbf{f}_3)$$

$$\begin{array}{lll|c|c|c|c|c|c|l}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{c}_{i,j} & (\ \sigma_{i,j}\cdot(1,-j) & \psi_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}}\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{c}_{i,j} & (\ \sigma'_{i,j}\cdot(1,-j) & \psi_i^{(k)} & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}}\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{k}_{i,j}^{(\ell)} & (\ \pi_{i,j}^{(\ell)}\cdot(j,1) & a_j^{(\ell)}\cdot z_\ell & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{lll|c|c}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{t}_i & [\![\omega_{\mathsf{tag},\mathsf{S}}\cdot s_i+\omega'_{\mathsf{tag},\mathsf{S}}\cdot u_i+\mathbf{x}_b^*[i]]\!]_1\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},x_i,\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{t}_i & [\![\chi_{\mathsf{tag}',\mathsf{S}'}\cdot s_i+\chi'_{\mathsf{tag}',\mathsf{S}'}\cdot u_i+x_i]\!]_1\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{m}_i^{(\ell)} & [\![\mathbf{y}^{(\ell)}[i]]\!]_2
\end{array}$$

$$\begin{array}{lll|c|c|c|c}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\omega_{\mathsf{tag},\mathsf{S}} & p_i\omega'_{\mathsf{tag},\mathsf{S}} & \psi_i & 0 &)_{\mathbf{H}_i}\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},x_i,\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\chi_{\mathsf{tag}',\mathsf{S}'} & p_i\chi'_{\mathsf{tag}',\mathsf{S}'} & \psi_i^{(k)} & 0 &)_{\mathbf{H}_i}\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \langle\mathbf{s},\mathbf{y}^{(\ell)}\rangle & \langle\mathbf{u},\mathbf{y}^{(\ell)}\rangle & a_0^{(\ell)}z_\ell & 0 &)_{\mathbf{H}_i^*}
\end{array}$$

Game G_1 : $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$, $a_0^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $(a_j^{(\ell)})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0^{(\ell)}}(\mathbb{A})$; $\Delta\mathbf{x} = \mathbf{x}_0^* - \mathbf{x}_1^*$

$$\begin{array}{lll|c|c|c|c|c|c|l}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{c}_{i,j} & (\ \sigma_{i,j}\cdot(1,-j) & \psi_i & \tau\Delta\mathbf{x}[i] & 0 & \tau\Delta\mathbf{x}[i]z_j & 0 & 0 &)_{\mathbf{F}}\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{c}_{i,j} & (\ \sigma'_{i,j}\cdot(1,-j) & \psi_i^{(k)} & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}}\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{k}_{i,j}^{(\ell)} & (\ \pi_{i,j}^{(\ell)}\cdot(j,1) & a_j^{(\ell)}\cdot z_\ell & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{lll|c|c|c|c|c}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\omega_{\mathsf{tag},\mathsf{S}} & p_i\omega'_{\mathsf{tag},\mathsf{S}} & \psi_i & \tau\Delta\mathbf{x}[i] &)_{\mathbf{H}_i}\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},x_i,\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\chi_{\mathsf{tag}',\mathsf{S}'} & p_i\chi'_{\mathsf{tag}',\mathsf{S}'} & \psi_i^{(k)} & 0 &)_{\mathbf{H}_i}\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \langle\mathbf{s},\mathbf{y}^{(\ell)}\rangle & \langle\mathbf{u},\mathbf{y}^{(\ell)}\rangle & a_0^{(\ell)}z_\ell & 0 &)_{\mathbf{H}_i^*}
\end{array}$$

Game G_2 : $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$, $a_0'^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $(a_j'^{(\ell)})_j \leftarrow \Lambda_{a_0'^{(\ell)}}(\mathbb{A})$; $\Delta\mathbf{x} = \mathbf{x}_0^* - \mathbf{x}_1^*$

$$\begin{array}{lll|c|c|c|c|c|c|l}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{c}_{i,j} & (\ \sigma_{i,j}\cdot(1,-j) & \psi_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}}\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{c}_{i,j} & (\ \sigma'_{i,j}\cdot(1,-j) & \psi_i^{(k)} & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}}\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{k}_{i,j}^{(\ell)} & (\ \pi_{i,j}^{(\ell)}\cdot(j,1) & a_j^{(\ell)}\cdot z_\ell & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{lll|c|c|c|c|c}
\mathbf{LoR}(i\in\mathcal{H},\mathbf{x}_0^*[i],\mathbf{x}_1^*[i],\mathsf{tag},\mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\omega_{\mathsf{tag},\mathsf{S}} & p_i\omega'_{\mathsf{tag},\mathsf{S}} & \psi_i & \tau\Delta\mathbf{x}[i] &)_{\mathbf{H}_i}\\
\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H},x_i,\mathsf{tag}'\neq\mathsf{tag},\mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\chi_{\mathsf{tag}',\mathsf{S}'} & p_i\chi'_{\mathsf{tag}',\mathsf{S}'} & \psi_i^{(k)} & 0 &)_{\mathbf{H}_i}\\
\forall\, i\in\mathcal{C}\cup\mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \langle\mathbf{s},\mathbf{y}^{(\ell)}\rangle & \langle\mathbf{u},\mathbf{y}^{(\ell)}\rangle & a_0^{(\ell)}z_\ell & r_0^{(\ell)}\cdot\mathbf{y}^{(\ell)}[i] &)_{\mathbf{H}_i^*}
\end{array}$$

Game G_3 : $\mathsf{H}(\mathsf{tag},\mathsf{S}) = [\![\mathsf{RF}(\mathsf{tag},\mathsf{S})]\!]_1 := ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $[\![\mathsf{H}(\mathsf{tag}',\mathsf{S}')]\!]_1 = [\![\mathsf{RF}(\mathsf{tag}',\mathsf{S}')]\!]_1 := ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$

Fig. 11: Games $\mathsf{G}_0, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3$ for Theorem 14. The transition from G_1 to G_2 is given in Figure 13 in Appendix B.4. The sets $\mathcal{H}$ and $\mathcal{C}$ contain honest and corrupted $i \in [n]$, respectively. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries. We are considering the one-time IND-CPA game, and the non-challenge ciphertexts from **Enc** are indexed by $k \in \mathbb{N}$. The function H is modeled as a random oracle. In G_3 we use a random function $\mathsf{RF} : \mathsf{Tag} \times 2^{\mathsf{Att}} \to (\mathbb{Z}_q^*)^2$.

where for each $j \in \mathsf{S}$

$$\mathbf{c}_{i,j} := (\sigma_{i,j}\cdot(1,-j),\ \psi_i,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}}$$

and $\sigma_{i,j} \xleftarrow{\$} \mathbb{Z}_q$ for $j \in \mathsf{S}'$ and $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega]\!]_1, [\![\omega']\!]_1)$ is modeled as a random oracle (RO).

Enc$(i, x_i, \mathsf{tag}', \mathsf{S}')$: As dictated by the security model in Figure 2, the adversary can only query for encryptions of messages under tag' different from the challenge tag. The ciphertext

Game G_4 : $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag},\mathsf{S}) := [\![\mathsf{RF}(\mathsf{tag},\mathsf{S})]\!]_1 := ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') := [\![\mathsf{RF}'(\mathsf{tag}')\cdot(1,\ \mu)]\!]_1 = ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$

$$\forall i \in \mathcal{C}\cup\mathcal{H}\ \ \mathsf{ek}_i\ \ (s_i,\quad u_i,\quad p_i\cdot H_i^{(1)},\quad p_i\cdot H_i^{(2)},\quad \mathbf{h}_{i,3},\quad \mathbf{f}_1,\quad \mathbf{f}_2,\quad \mathbf{f}_3)$$

$\mathbf{LoR}(i\in\mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{t}_i$	$[\![\omega_{\mathsf{tag},\mathsf{S}}\cdot s_i + \omega'_{\mathsf{tag},\mathsf{S}}\cdot u_i + \mathbf{x}_b^*[i]]\!]_1$
$\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H}, x_i, \mathsf{tag}'\neq\mathsf{tag}, \mathsf{S}')$	$\mathbf{t}_i$	$[\![\chi_{\mathsf{tag}',\mathsf{S}'}\cdot s_i + \mu\chi_{\mathsf{tag}',\mathsf{S}'}\cdot u_i + x_i]\!]_1$
$\forall i\in\mathcal{C}\cup\mathcal{H}$	$\mathbf{m}_i^{(\ell)}$	$[\![\mathbf{y}^{(\ell)}[i]]\!]_2$

$\mathbf{LoR}(i\in\mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	$($	$p_i\omega_{\mathsf{tag},\mathsf{S}}$	$p_i\omega'_{\mathsf{tag},\mathsf{S}}$	ψ_i	$\tau\Delta\mathbf{x}[i]$ $)_{\mathbf{H}_i}$
$\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H}, x_i, \mathsf{tag}'\neq\mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	$($	$p_i\chi_{\mathsf{tag}',\mathsf{S}'}$	$p_i\mu\chi_{\mathsf{tag}',\mathsf{S}'}$	$\psi_i^{(k)}$	0 $)_{\mathbf{H}_i}$
$\forall i\in\mathcal{C}\cup\mathcal{H}$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$	$($	$\langle S, \mathbf{y}^{(\ell)}\rangle$	$\langle U, \mathbf{y}^{(\ell)}\rangle$	$a_0^{(\ell)} z_\ell$	$r_0^{(\ell)}\mathbf{y}^{(\ell)}[i]$ $)_{\mathbf{H}_i^*}$

Game G_5 : $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag},\mathsf{S}) := ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') := ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$, $R_i \xleftarrow{\$} \mathrm{span}(H_i^{(3)}, H_i^{(4)}) \subseteq \mathbb{Z}_q^4$

$$\forall i\in\mathcal{C}\cup\mathcal{H}\ \ \mathsf{ek}_i\ \ (s_i,\quad u_i,\quad p_i\cdot (H_i^{(1)} - \mu R_i),\quad p_i\cdot (H_i^{(2)} + R_i),\quad \mathbf{h}_{i,3},\quad \mathbf{f}_1,\quad \mathbf{f}_2,\quad \mathbf{f}_3)$$

Game G_6 : $R_i \xleftarrow{\$} \mathrm{span}(H_i^{(4)})$, $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag},\mathsf{S}) := ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') := ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$, $\Delta\mathbf{x} = \mathbf{x}_0^* - \mathbf{x}_1^*$. We also define $S' = S + \Delta S, U' = U + \Delta U$, where $\Delta S, \Delta U \in \mathbb{Z}_q^n$ s.t. $\Delta S + \mu\Delta U = 0$ and $\omega_{\mathsf{tag},\mathsf{S}}\cdot\Delta S + \omega'_{\mathsf{tag},\mathsf{S}}\cdot\Delta U = \mathbf{x}_b^* - \mathbf{x}_0^*$

$$i\in\mathcal{C}\cup\mathcal{H}\ \ \mathsf{ek}_i\ \ (\ s_i',\quad u_i'\quad p_i\cdot(H_i^{(1)} - \mu R_i),\quad p_i\cdot(H_i^{(2)} + R_i),\quad \mathbf{h}_{i,3},\quad \mathbf{f}_1,\quad \mathbf{f}_2,\quad \mathbf{f}_3)$$

$\mathbf{LoR}(i\in\mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{t}_i$	$[\![\omega_{\mathsf{tag},\mathsf{S}}\cdot s_i' + \omega'_{\mathsf{tag},\mathsf{S}}\cdot u_i' + \mathbf{x}_0^*[i]]\!]_1$
$\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H}, x_i, \mathsf{tag}'\neq\mathsf{tag}, \mathsf{S}')$	$\mathbf{t}_i$	$[\![\chi_{\mathsf{tag}',\mathsf{S}'}\cdot s_i' + \mu\chi_{\mathsf{tag}',\mathsf{S}'}\cdot u_i' + x_i]\!]_1$
$\forall i$	$\mathbf{m}_i^{(\ell)}$	$[\![\mathbf{y}^{(\ell)}[i]]\!]_2$

$\mathbf{LoR}(i\in\mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	$($	$p_i\omega_{\mathsf{tag},\mathsf{S}}$	$p_i\omega'_{\mathsf{tag},\mathsf{S}}$	ψ_i	$\tau'\Delta\mathbf{x}[i]$ $)_{\mathbf{H}_i}$
$\mathbf{Enc}(i\in\mathcal{C}\cup\mathcal{H}, x_i, \mathsf{tag}'\neq\mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	$($	$p_i\chi_{\mathsf{tag}',\mathsf{S}'}$	$p_i\mu\chi_{\mathsf{tag}',\mathsf{S}'}$	$\psi_i^{(k)}$	0 $)_{\mathbf{H}_i}$
$\forall i\in\mathcal{C}\cup\mathcal{H}$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$	$($	$\langle S', \mathbf{y}^{(\ell)}\rangle$	$\langle U', \mathbf{y}^{(\ell)}\rangle$	$a_0^{(\ell)} z_\ell$	$r_0^{(\ell)}\mathbf{y}^{(\ell)}[i]$ $)_{\mathbf{H}_i^*}$

Fig. 12: Games $\mathsf{G}_4, \mathsf{G}_5, \mathsf{G}_6$ for Theorem 14. The sets $\mathcal{H}$ and $\mathcal{C}$ contain honest and corrupted $i \in [n]$, respectively. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries. We are considering the one-time IND-CPA game, and the non-challenge ciphertexts from **Enc** are indexed by $k \in \mathbb{N}$. In G_4 we use a random function $\mathsf{RF}' : \mathsf{Tag} \to \mathbb{Z}_q^*$.

is returned:

$$\mathbf{c}_{i,j} := (\sigma_{i,j}\cdot(1,-j),\ \psi_i^{(k)},\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \text{ for } j\in\mathsf{S}'$$
$$\mathbf{t}_i := [\![\chi_{\mathsf{tag}',\mathsf{S}'} s_i + \chi'_{\mathsf{tag}',\mathsf{S}'} u_i + x_i]\!]_1$$
$$\mathbf{c}_{i,\mathsf{ipfe}} := (\chi_{\mathsf{tag}',\mathsf{S}'},\ \chi'_{\mathsf{tag}',\mathsf{S}'},\ \psi_i',\ 0)_{\mathbf{H}_i}$$

where $\sigma_{i,j}, \psi_i^{(k)} \xleftarrow{\$} \mathbb{Z}_q$ for $j\in\mathsf{S}'$ and $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$ is modeled as a random oracle (RO).

Corrupt(i) : Return $\mathsf{ek}_i = (s_i,\ u_i,\ p_i\cdot H_i^{(1)},\ p_i\cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3)$.

Eventually the adversary outputs a bit b'. The simulator then runs and outputs **Finalise**(b'). We have $\mathsf{Adv}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}(1^\lambda) = \mathsf{Adv}(\mathsf{G}_0)$.

Game G_1: We first introduce the masks in the challenge ciphertext components. The basis changes are done in a manner similar to the proof of Lemma 4. The ciphertext components

are computed as below:

$$\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{S}): \ \mathbf{c}_{i,j} = (\sigma_{i,j} \cdot (1, -j), \psi_i, \tau\Delta\mathbf{x}[i], 0, \tau\Delta\mathbf{x}[i]z_j, 0, 0)_{\mathbf{F}}$$
$$\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{S}): \ \mathbf{c}_{i,\mathsf{ipfe}} = (\omega_{\mathsf{tag},\mathsf{S}} p_i, \ \omega'_{\mathsf{tag},\mathsf{S}} p_i, \ \psi_i, \ \tau\Delta\mathbf{x}[i])_{\mathbf{H}_i} \ .$$

The basis changes of $(\mathbf{H}_i, \mathbf{H}_i^*)$ can be done in parallel, while the change for $(\mathbf{F}, \mathbf{F}^*)$ does not depend on i and we will write the vectors $(\mathbf{c}_{i,j}, \mathbf{c}_{i,\mathsf{ipfe}})$ with appropriate coordinates for each i under the basis change's effect. For corrupted i, we cannot enforce the appearance of $\tau\Delta\mathbf{x}[i]$ because the adversary can create their own ciphertext components using ek_i. We note that if $i \in \mathcal{C}$ and $\mathbf{y}^{(\ell)}[i] \neq 0$ and $\mathbb{A}^{(\ell)}(\mathsf{S}) = 1$, condition 3 of Definition 9 implies that $\Delta\mathbf{x}[i] = 0$ and the transition is trivial. Hence, the fact that we cannot control the ciphertext components for $i \in \mathcal{C}$ is not problematic. The difference in advantages is bounded by $2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$.

Game G_2: To reach this game we proceed key by key, indexed by $\ell \in \{0, \ldots, K\}$, from $\mathsf{G}_{1.0} = \mathsf{G}_1$ to $\mathsf{G}_{1.K} = \mathsf{G}_2$. The game $\mathsf{G}_{1.\ell}$ has the first ℓ functional keys switched to semi-functional as described in G_2.

In order to go from $\mathsf{G}_{1.\ell-1}$ to $\mathsf{G}_{1.\ell}$, we employ the following sequence of games, which is depicted in Figure 13.

Game $\mathsf{G}_{1.\ell-1.0}$: This is $\mathsf{G}_{1.\ell-1}$.

Game $\mathsf{G}_{1.\ell-1.1}$: We apply Lemma 4 for each $i \in [n]$ whose $\mathbf{y}[i] \neq 0$ to mask the vectors

$$\{(\mathbf{k}^{(\ell)}_{i,j})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}\}$$

with another random labeling $(a_j'^{(\ell)})_j \leftarrow \Lambda_{a_0'^{(\ell)}}(\mathbb{A})$ where $a_0'^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$ and the ciphertext components are those returned from **LoR**. For all $i \in [n]$, Lemma 4 is applied in parallel using the same random labeling for random labeling $(a_j'^{(\ell)})_j \leftarrow \Lambda_{a_0'^{(\ell)}}(\mathbb{A})$. The affected coordinates are $(3, 4)$ of $(\mathbf{H}_i, \mathbf{H}_i^*)$ and all coordinates of $(\mathbf{F}, \mathbf{F}^*)$. The constants are $x := \Delta\mathbf{x}[i]$ and $y := \mathbf{y}^{(\ell)}[i]$. We remark that if $\mathbf{y}^{(\ell)}[i] = 0$ then for $i \in \mathcal{H}$, whose ciphertext components $\mathbf{c}_{i,j}, \mathbf{c}_{i,\mathsf{ipfe}}$ must be queried to **LoR** under condition 2, the transition is trivial. More precisely, the challenge ciphertext and the ℓ-th functional key components will be:

$$\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{S}): \ \mathbf{c}_{i,j} = (\sigma_{i,j} \cdot (1, -j), \psi_i, \tau\Delta\mathbf{x}[i], 0, \tau\Delta\mathbf{x}[i]z_j, 0, 0)_{\mathbf{F}}$$
$$\forall\, i \in \mathcal{C} \cup \mathcal{H}: \ \mathbf{k}^{(\ell)}_{i,j} = (\pi^{(\ell)}_{i,j} \cdot (j, 1), a^{(\ell)}_j \cdot z, 0, 0, a'^{(\ell)}_j \mathbf{y}^{(\ell)}[i]/z_j, 0, 0)_{\mathbf{F}^*}$$
$$\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{S}): \ \mathbf{c}_{i,\mathsf{ipfe}} = (\omega_{\mathsf{tag},\mathsf{S}} p_i, \ \omega'_{\mathsf{tag},\mathsf{S}} p_i, \ \psi_i, \tau\Delta\mathbf{x}[i])_{\mathbf{H}_i}$$
$$\forall\, i \in \mathcal{C} \cup \mathcal{H}: \ \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} = (\langle S, \mathbf{y}^{(\ell)} \rangle, \ \langle U, \mathbf{y}^{(\ell)} \rangle, \ a^{(\ell)}_0 z, a'^{(\ell)}_0 \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \ .$$

By Lemma 4, the difference in advantages is:

$$|\mathsf{Adv}(\mathsf{G}_{0.\ell-1.1}) - \mathsf{Adv}(\mathsf{G}_{0.\ell-1.0})| \leq (P \cdot (6P + 3) + 1) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1, \mathbb{G}_2}(1^\lambda) \ .$$

If the attributes of the challenge ciphertext satisfy the ℓ-th key's policy, the n ciphertext components can still be combined and decrypted to obtain $\langle \mathbf{x}_b^*, \mathbf{y}^{(\ell)} \rangle$ using $\mathsf{dk}_{\mathbb{A}, \mathbf{y}^{(\ell)}}$. The reasons why we can apply the lemma in parallel can be summarized below:

- Each client i has the vectors $\mathbf{c}_{i,\mathsf{ipfe}}$ and $\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$ lying in separate dual bases $(\mathbf{H}_i, \mathbf{H}_i^*)$ so the basis changing matrices can be written independently for each client. We note that the basis changes of $(\mathbf{F}, \mathbf{F}^*)$ does not depend on i but only on the attributes j. The computation over $\mathbf{c}$-vectors in Lemma 4 can be done for the vectors $(\mathbf{c}_{i,j})_j$ and $(\mathbf{k}^{(\ell)}_{i,j})_j$ at the same time for all $i \in [n]$, by setting the appropriate coordinates in $(\mathbf{W}, \mathbf{W}^*)$ and seeing how they are affected under these basis changes to produce the final vectors in $(\mathbf{F}, \mathbf{F}^*)$.
- When we perform a sequence of hybrids indexed by an attribute m, e.g. to introduce the factor $1/z_j$ in $\mathbf{k}^{(\ell)}_{i,j}$ where $j \neq m$, only the vectors $\mathbf{k}^{(\ell)}_{i,m}$ have non-zero coordinate at $\mathbf{f}^*_5$ and $\mathbf{c}_{i,m}$ have non-zero coordinate at $\mathbf{f}_5$. Hence, the relating basis changes will affect only those $\mathbf{k}^{(\ell)}_{i,m}, \mathbf{c}_{i,m}$ for all $i \in [n]$ at once.

Game $\mathsf{G}_{1.\ell-1.0} = \mathsf{G}_{1.\ell-1}$

Game $\mathsf{G}_{1.\ell-1.1}$: $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$, $a_0'^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $(a_j'^{(\ell)})_j \leftarrow \Lambda_{a_0'^{(\ell)}}(\mathbb{A})$, $\Delta\mathbf{x} := \mathbf{x}_b^*[i] - \mathbf{x}_0^*[i]$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1,-j)$	ψ_i	$\tau\Delta\mathbf{x}[i]$	0	$\tau\Delta\mathbf{x}[i]z_j$	0	0	$)_{\mathbf{F}}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,j}$	(	$\sigma'_{i,j} \cdot (1,-j)$	$\psi_i^{(k)}$	0	0	0	0	0	$)_{\mathbf{F}}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' \neq \ell$	$\mathbf{k}_{i,j}^{(\ell')}$	(	$\pi_{i,j}^{(\ell')} \cdot (j,1)$	$a_j^{(\ell')} \cdot z_\ell$	0	0	0	0	0	$)_{\mathbf{F}^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,j}^{(\ell)}$	(	$\pi_{i,j}^{(\ell)} \cdot (j,1)$	$a_j^{(\ell)} \cdot z_\ell$	0	0	$a_j'^{(\ell)}\mathbf{y}^{(\ell)}[i]/z_j$	0	0	$)_{\mathbf{F}^*}$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\omega_{\mathsf{tag},\mathsf{S}} p_i$	$\omega'_{\mathsf{tag},\mathsf{S}} p_i$	ψ_i	$\tau\Delta\mathbf{x}[i]$	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\chi_{\mathsf{tag}',\mathsf{S}'} p_i$	$\chi'_{\mathsf{tag}',\mathsf{S}'} p_i$	$\psi_i^{(k)}$	0	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' < \ell$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell')}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell')} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell')} \rangle$	$a_0^{(\ell')} z_\ell$	$r_0^{(\ell')} \cdot \mathbf{y}^{(\ell)}[i]$	$)_{\mathbf{H}_i^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle$	$a_0^{(\ell)} z_\ell$	$a_0'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i]$	$)_{\mathbf{H}_i^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' > \ell$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell')}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell')} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell')} \rangle$	$a_0^{(\ell')} z_\ell$	0	$)_{\mathbf{H}_i^*}$

Game $\mathsf{G}_{1.\ell-1.2}$: $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1,-j)$	ψ_i	$\tau\Delta\mathbf{x}[i]$	0	$\tau\Delta\mathbf{x}[i]z_j$	0	0	$)_{\mathbf{F}}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,j}$	(	$\sigma'_{i,j} \cdot (1,-j)$	$\psi_i^{(k)}$	0	0	0	0	0	$)_{\mathbf{F}}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' \neq \ell$	$\mathbf{k}_{i,j}^{(\ell')}$	(	$\pi_{i,j}^{(\ell')} \cdot (j,1)$	$a_j^{(\ell')} \cdot z_\ell$	0	0	0	0	0	$)_{\mathbf{F}^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,j}^{(\ell)}$	(	$\pi_{i,j}^{(\ell)} \cdot (j,1)$	$a_j^{(\ell)} \cdot z_\ell$	0	0	$a_j'^{(\ell)}\mathbf{y}^{(\ell)}[i]/z_j$	0	0	$)_{\mathbf{F}^*}$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\omega_{\mathsf{tag},\mathsf{S}} p_i$	$\omega'_{\mathsf{tag},\mathsf{S}} p_i$	ψ_i	$\tau\Delta\mathbf{x}[i]$	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\chi_{\mathsf{tag}',\mathsf{S}'} p_i$	$\chi'_{\mathsf{tag}',\mathsf{S}'} p_i$	$\psi_i^{(k)}$	0	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' < \ell$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell')}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell')} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell')} \rangle$	$a_0^{(\ell')} z_\ell$	$r_0^{(\ell')} \cdot \mathbf{y}^{(\ell)}[i]$	$)_{\mathbf{H}_i^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle$	$a_0^{(\ell)} z_\ell$	$r_0^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i]$	$)_{\mathbf{H}_i^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' > \ell$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell')}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell')} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell')} \rangle$	$a_0^{(\ell')} z_\ell$	0	$)_{\mathbf{H}_i^*}$

Game $\mathsf{G}_{1.\ell-1.3} = \mathsf{G}_{1.\ell}$: $\mathsf{H}(\mathsf{tag},\mathsf{S}) \to ([\![\omega_{\mathsf{tag},\mathsf{S}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{S}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{S}') \to ([\![\chi_{\mathsf{tag}',\mathsf{S}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{S}'}]\!]_1)$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1,-j)$	ψ_i	$\tau\Delta\mathbf{x}[i]$	0	$\tau\Delta\mathbf{x}[i]z_j$	0	0	$)_{\mathbf{F}}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,j}$	(	$\sigma'_{i,j} \cdot (1,-j)$	$\psi_i^{(k)}$	0	0	0	0	0	$)_{\mathbf{F}}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,j}^{(\ell)}$	(	$\pi_{i,j}^{(\ell)} \cdot (j,1)$	$a_j^{(\ell)} \cdot z_\ell$	0	0	0	0	0	$)_{\mathbf{F}^*}$

$\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\omega_{\mathsf{tag},\mathsf{S}} p_i$	$\omega'_{\mathsf{tag},\mathsf{S}} p_i$	ψ_i	$\tau\Delta\mathbf{x}[i]$	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\chi_{\mathsf{tag}',\mathsf{S}'} p_i$	$\chi'_{\mathsf{tag}',\mathsf{S}'} p_i$	$\psi_i^{(k)}$	0	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' \leq \ell$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell')}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell')} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell')} \rangle$	$a_0^{(\ell')} z_\ell$	$r_0^{(\ell')} \cdot \mathbf{y}^{(\ell)}[i]$	$)_{\mathbf{H}_i^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' > \ell$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell')}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell')} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell')} \rangle$	$a_0^{(\ell')} z_\ell$	0	$)_{\mathbf{H}_i^*}$

Fig. 13: The sequence of hybrids to go from $\mathsf{G}_{1.\ell-1}$ to $\mathsf{G}_{1.\ell}$, where $\ell \in [K]$. We have $\mathsf{G}_{1.0} = \mathsf{G}_1$ and $\mathsf{G}_{1.K} = \mathsf{G}_2$ in the proof of Theorem 14. The sets $\mathcal{H}$ and $\mathcal{C}$ contain honest and corrupted $i \in [n]$, respectively. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries. The non-challenge ciphertexts from **Enc** are indexed by $k \in \mathbb{N}$. The function H is modeled as a random oracle.

– Finally, considering the ciphertexts from **Enc**, whose coordinates w.r.t the basis changes are 0 all the time, they stay invariant and the correctness is preserved.

We remark that it is this possibility to parallelize the basis changes and the application that makes our *duplicate-and-compress* technique work, by compressing the access control parts into one pair of bases $(\mathbf{F}, \mathbf{F}^*)$ while duplicating the "root" part $(\mathbf{c}_{i,\mathsf{ipfe}}, \mathbf{k}^*_{i,\mathsf{ipfe}})$ into $(\mathbf{H}_i, \mathbf{H}^*_i)$ for each i.

Game $\mathsf{G}_{1.\ell-1.2}$: We now change all the masks $a_0'^{(\ell)}$ to $r_0^{(\ell)}$ in the vectors $\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$, for all $i \in [n]$:

$$\forall\, i \in \mathcal{C} \cup \mathcal{H} :\ \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} = (\langle S, \mathbf{y}^{(\ell)}\rangle,\ \langle U, \mathbf{y}^{(\ell)}\rangle,\ a_0^{(\ell)} z,\ r_0'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}^*_i}\ .$$

We recall that the construction impedes the use of different sets of attributes among clients $i \in [n]$ by hashing S together with tag at the time of encryption. That is, the encryption receives the same set S for all challenge ciphertext components, for all $i \in [n]$. We have to consider two cases:

– If $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle \neq 0$, the security model implies that $\mathbb{A}(\mathsf{S}) = 0$ where $\mathbb{A}$ is the access structure embedded in the key and S contains the attributes in the challenge ciphertext. Hence, for all $i \in [n]$, there exists no authorized set $A \subseteq \mathsf{S}$ for which we can find the reconstruction vector $(c_j)_j$ from the LSSS. That is, for all $i \in [n]$, there are *not* enough $a_j'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i]/z_j$ from the components $(\mathbf{k}_{i,j})_j$ of ℓ-th functional key to combine with $(\mathbf{c}_{i,j})_j$ and recover $\tau a_0'^{(\ell)} \mathbf{y}^{(\ell)}[i]\Delta\mathbf{x}[i]$. We recall that $(a_j'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i]/z_j)_j$ is randomized from the random labeling $(a_j'^{(\ell)})_j$ of $a_0'^{(\ell)}$ and becomes independent uniformly random values, i.e. *not* a random labeling of $a_0'^{(\ell)}$ anymore. The only relation between $a_0'^{(\ell)}$ and $(a_j'^{(\ell)})_j$ is

$$\sum_{j\in A} \frac{c_j a_j'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i]}{z_j} \cdot \tau z_j \Delta\mathbf{x}[i] = \tau a_0'^{(\ell)} \mathbf{y}_\ell[i]\Delta\mathbf{x}[i] \tag{5}$$

where $(z_j)_j$ appear only in the unique ciphertexts $(\mathbf{c}_{i,j})_j$ from the one-time **LoR**. Hence, in this case $a_0'^{(\ell)}$ is perfectly indistinguishable from a uniformly random value under the adversary's view.

– If $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle = 0$, the sum over i, j during decryption makes sure that the ℓ-th key is still capable of decrypting the challenge ciphertext from **LoR** if the policy is satisfied. More specifically, let $A \subseteq \mathsf{S}$ be an authorized set for which we can find the reconstruction vector $(c_j)_j$ from the LSSS. Then, for all $i \in [n]$, $(c_j)_j$ can be used with $(\mathbf{k}_{i,j})_j$ of ℓ-th functional key as well as the ciphertext components $(\mathbf{c}_{i,j})_j$ to recover $a_0^{(\ell)}$. The calculation leads to:

$$\begin{aligned}
\sum_{i=1}^{n} \left(\sum_{j\in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}^{(\ell)}_{i,j}) \right) &= \sum_{j\in A} \left(\sum_{i=1}^{n} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}^{(\ell)}_{i,j}) \right) \\
&= \sum_{j\in A} \left(\sum_{i=1}^{n} \psi_i c_j a_j^{(\ell)} z + \tau c_j \cdot a_j'^{(\ell)} \Delta\mathbf{x}[i]\mathbf{y}^{(\ell)}[i] \right) \\
&= \tau a_0'^{(\ell)} \langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle + \sum_{i=1}^{n} \psi_i \left(\sum_{j\in A} c_j a_j^{(\ell)} z \right) \\
&= \sum_{i=1}^{n} \psi_i a_0^{(\ell)} z \\
\sum_{i=1}^{n} \left(\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} \right) &= \sum_{i=1}^{n} \left(\psi_i a_0^{(\ell)} z + \tau a_0'^{(\ell)} \mathbf{y}_\ell[i]\Delta\mathbf{x}[i] \right) \\
&= \sum_{i=1}^{n} \psi_i a_0^{(\ell)} z
\end{aligned}$$

and it does not depend on $a_0'^{(\ell)}$ anymore. This is also the only relation w.r.t $a_0'^{(\ell)}$ that an (even unbounded) adversary can deduce.

Totally, the change from $a_0'^{(\ell)}$ to $r_0^{(\ell)}$ is perfectly indistinguishable and $\mathsf{Adv}(\mathsf{G}_{1.\ell-1.1}) = \mathsf{Adv}(\mathsf{G}_{1.\ell-1.2})$.

Game $\mathsf{G}_{1.\ell-1.3}$: In this game, we apply Lemma 4 for each $i \in [n]$ to the families

$$\{(\mathbf{c}_{i,j})_j, \mathbf{c}_{i,\mathsf{ipfe}}\} \text{ and } \{(\mathbf{k}_{i,j}^{(\ell)})_j, \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}\}$$

so as to clean the vectors $\{(\mathbf{c}_{i,j})_j\}$. All the family of vectors for $i \in [n]$ are treated in parallel, thanks to the same reasons when we go from $\mathsf{G}_{1.\ell-1.0}$ to $\mathsf{G}_{1.\ell-1.1}$. We remark that the basis changes are done for $(\mathbf{F}, \mathbf{F}^*)$, not depending on $\Delta\mathbf{x}[i]$ nor $\mathbf{y}^{(\ell)}[i]$, and for each $i \in [n]$ the vectors $(\mathbf{c}_{i,j})_j, \mathbf{k}_{i,j}^{(\ell)}$ are written with appropriate coordinates.

There is a difference in comparison to the adaptive single-client proof. Because we are rewriting all vectors $\mathbf{k}_{i,j}$ for *all* $i \in [n]$ at the same time and all client i must use the same S, having a reconstruction vector $(c_j)_j$, either all n summations

$$\sum_j \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}^{(\ell)})$$

have the term $\tau a_j'^{(\ell)} \mathbf{y}^{(\ell)}[i]\Delta\mathbf{x}[i]$ for $i \in [n]$ or none of them has. If none of them has, it does not affect the correctness of decryption. We recall that $\Delta\mathbf{x}[i] = 0$ for all $i \in \mathcal{C}$ and thus even though we cannot introduce $\tau\Delta\mathbf{x}[i]$ for the i-th corrupted ciphertext, we can still recover:

$$\sum_j c_j \cdot \left(\sum_{i=1}^{n} \left[\!\!\left[\tau a_j'^{(\ell)} \mathbf{y}^{(\ell)}[i]\Delta\mathbf{x}[i] \right]\!\!\right]_{\mathsf{t}} \right) = \sum_j \left[\!\!\left[\tau c_j a_j'^{(\ell)} \langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \right]\!\!\right]_{\mathsf{t}}$$

in $\sum_{i=1}^{n} \left(\sum_j \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}^{(\ell)}) \right)$ for decryption.

Hence, if $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$ and the policy is satisfied, the masks $a_j'^{(\ell)}$ does not affect the decryption. Otherwise, if $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \neq 0$ then the policy is not satisfied and lacking $a_0'^{(\ell)}$ in $\mathbf{k}_{i,\mathsf{ipfe}}$ does not affect the incapability of the key. We recall that in the adaptive single-client proof, we can only clean the mask $a_j'^{(\ell)} \mathbf{y}^{(\ell)}[i]/z_j$ *one by one* and that prevents us from completing the value $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle$. Following Lemma 4, the difference in advantages is:

$$|\mathsf{Adv}(\mathsf{G}_{0.\ell-1.3}) - \mathsf{Adv}(\mathsf{G}_{0.\ell-1.2})| \leq (P \cdot (6P+3)+1) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .$$

The game $\mathsf{G}_{1.\ell-1.3}$ is identical to $\mathsf{G}_{1.\ell}$.

We perform the transition from $\mathsf{G}_1 = \mathsf{G}_{1.0}$ to $\mathsf{G}_{1.K}$, whose total difference in advantages is:

$$|\mathsf{Adv}(\mathsf{G}_{1.K}) - \mathsf{Adv}(\mathsf{G}_1)| \leq K \cdot (2P \cdot (6P+3)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .$$

We then use DDH in $\mathbb{G}_1$ to clean the coordinate $(3,6)$ of $\mathbf{c}_{i,j}$, similar to the beginning of the games for Lemma 4 but in reverse order, and finally arrive at G_2. We have

$$|\mathsf{Adv}(\mathsf{G}_2) - \mathsf{Adv}(\mathsf{G}_1)| \leq K(2P \cdot (6P+3)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) \ .$$

Game G_3: We simulate any new random oracle query $\mathsf{H}(\mathsf{tag}, \mathsf{S})$ by a random pair of elements in $\mathbb{G}_1$. The distribution is identical and thus $\mathsf{Adv}(\mathsf{G}_3) = \mathsf{Adv}(\mathsf{G}_2)$.

Game G_4: In this game, the simulator first guesses the challenged tag among the Q queries to the random oracle, which should be fixed for all queries to **LoR**. If the guess is not correct, the simulator aborts and outputs 0. Then, for any new random oracle query $\mathsf{H}(\mathsf{tag}', \mathsf{S}')$ where $\mathsf{tag}' \neq \mathsf{tag}$, we respond by a random vector lying in $\mathrm{span}((1,\ \mu)) \subseteq \mathbb{G}_1^2$, for $\mu \xleftarrow{\$} \mathbb{Z}_q$. On the other hand, the RO query $\mathsf{H}(\mathsf{tag}, \mathsf{S})$ is still responded by $\left[\!\!\left[(\omega_{\mathsf{tag},\mathsf{S}}, \omega'_{\mathsf{tag},\mathsf{S}}) \right]\!\!\right]_1$ where

$(\omega_{\mathsf{tag},\mathsf{S}}, \omega'_{\mathsf{tag},\mathsf{S}})$ is a pair of independent random elements in $\mathbb{Z}_q$. If the challenged tag is not guessed correctly, among the Q RO queries, the simulation is aborted and outputs 0.
We use the random self-reducibility of DDH, where the running time of the simulator increases by an additive factor $O(Q \cdot t_{\mathbb{G}_1})$ with $t_{\mathbb{G}_1}$ being the time for one addition in $\mathbb{G}_1$ and Q being the number of random oracle queries. We define $\mathsf{Event}(\mathsf{tag})$ to denote the event where the challenged tag is guessed correctly, with probability $1/Q$. We have

$$|\Pr[\mathsf{G}_3 = 1 \mid \mathsf{Event}(\mathsf{tag})] - \Pr[\mathsf{G}_4 = 1 \mid \mathsf{Event}(\mathsf{tag})]| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) \ .$$

Notice that $\Pr[\mathsf{G}_4 = 1 \mid \neg\mathsf{Event}(\mathsf{tag})] = 0$ and the output of G_3 is independent of $\mathsf{Event}(\mathsf{tag})$. Therefore, we have

$$\begin{aligned}\mathsf{Adv}(\mathsf{G}_4) &= \frac{1}{Q} \cdot \Pr[\mathsf{G}_4 = 1 \mid \mathsf{Event}(\mathsf{tag})] \\ &\quad + \Pr[\neg\mathsf{Event}(\mathsf{tag})] \Pr[\mathsf{G}_4 = 1 \mid \neg\mathsf{Event}(\mathsf{tag})] - \frac{1}{2} \\ &\geq \frac{1}{Q} \cdot \Big(\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)\Big)\end{aligned}$$

and thus the difference in advantages is

$$\mathsf{Adv}(\mathsf{G}_3) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + Q \cdot \mathsf{Adv}(\mathsf{G}_4) \ .$$

Game G_5: In this game, we change the way the encryption keys ek_i are generated: for $R_i \xleftarrow{\$} \mathrm{span}(H_i^{(4)}) \subseteq \mathbb{Z}_q^4$

$$\mathsf{ek}_i = (s_i,\ u_i,\ p_i \cdot (H_i^{(1)} - \mu R_i),\ p_i \cdot (H_i^{(2)} + R_i),\ \mathbf{h}_{i,3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3) \ .$$

Similar to the selective proof, we use a basis change. From the beginning, the dual bases $(\mathbf{H}_i, \mathbf{H}_i^*)$ are specified by $H_i \xleftarrow{\$} \mathbb{Z}_q^{4\times 4}$ as part of msk and all H_i are kept hidden from the adversary:

$$\mathbf{H}_i = H_i \cdot \mathbf{T}; \quad \mathbf{H}_i^* = H'_i \cdot \mathbf{T}^*$$

where $H'_i := \left(H_i^{-1}\right)^\top$.
Then, before answering any query, the simulator samples $\theta_3, \theta_4 \xleftarrow{\$} \mathbb{Z}_q^*$, defines $R_i := \theta_3 H_i^{(3)} + \theta_4 H_i^{(4)}$, and performs a basis change on (H_i, H'_i) to obtain:

$$K_i := \begin{bmatrix} 1 & 0 & -\mu\theta_3 & -\mu\theta_4 \\ 0 & 1 & \theta_3 & \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot H_i = \left[\begin{array}{c} H_i^{(1)} - \mu R_i \\ \hline H_i^{(2)} + R_i \\ \hline H_i^{(3)} \\ \hline H_i^{(4)} \end{array}\right] \ .$$

The K_i is invertible and is indeed a basis changing matrix. For each corruption query $\mathbf{Corrupt}(i)$, the simulator returns:

$$\mathsf{ek}_i = (s_i,\ u_i,\ p_i \cdot K_i^{(1)},\ p_i \cdot K_i^{(2)},\ \mathbf{h}_{i,3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3) \ .$$

First, we note that the ciphertext vectors are still written in $(\mathbf{H}_i, \mathbf{H}_i^*)$:

$$\begin{aligned}&\textbf{LoR} && \mathbf{c}_{i,\mathsf{ipfe}} = (\omega_{\mathsf{tag},\mathsf{S}} p_i, \omega'_{\mathsf{tag},\mathsf{S}} p_i,\ \psi_i + p_i\theta_3(\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}}), \\ &&& \qquad \tau\Delta\mathbf{x}[i] + p_i\theta_4(\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}}))_{\mathbf{H}_i} \\ &\textbf{Enc} && \mathbf{c}_{i,\mathsf{ipfe}} = (\chi_{\mathsf{tag}',\mathsf{S}'} p_i, \mu\chi_{\mathsf{tag}',\mathsf{S}'} p_i,\ \psi'_i,\ 0)_{\mathbf{H}_i}\end{aligned}$$

and the functional key components are

$$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} = (\langle S, \mathbf{y}^{(\ell)}\rangle, \langle U, \mathbf{y}^{(\ell)}\rangle, a_0^{(\ell)} z_\ell, r_0^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \ .$$

To begin with, the correctness of decryption for ciphertexts queried to **Enc** is preserved, when the key's policy is satisfied by the ciphertext's attributes, thanks to the programing of the RO $\mathsf{H}(\mathsf{tag}', \mathsf{S}') \to [\![(\chi_{\mathsf{tag}',\mathsf{S}'}, \mu\chi_{\mathsf{tag}',\mathsf{S}'})]\!]_1$. It remains to show how the simulator simulates the functional key components and responses for **LoR**.
During **Setup**, the simulator sets $p_i := 1/n$ and samples $\tilde{r}_0^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^*$, which is supposed to be used as the mask in $\mathbf{k}_{i,\mathsf{ipfe}}$. Upon receiving the **Extract** query for $\mathbf{y}^{(\ell)}$, the simulator sets $r_0^{(\ell)} := \tilde{r}_0^{(\ell)} + \frac{-\theta_3 \cdot n a_0^{(\ell)} z_\ell}{\theta_4 \langle \mathbf{y}^{(\ell)}, \mathbf{1}\rangle}$. The key components for all $i \in [n]$ will be simulated as follows:

$$\begin{aligned}
\mathbf{k}_{i,j}^{(\ell)} &:= (\pi_{i,j}^{(\ell)} \cdot (j,1),\ a_j^{(\ell)} \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A}) \\
\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} &:= \left(\langle S, \mathbf{y}^{(\ell)}\rangle,\ \langle U, \mathbf{y}^{(\ell)}\rangle,\ a_0^{(\ell)} \cdot z_\ell,\ r_0^{(\ell)} \mathbf{y}[i]\right)_{\mathbf{H}_i^*} \\
&= \left(\langle S, \mathbf{y}^{(\ell)}\rangle,\ \langle U, \mathbf{y}^{(\ell)}\rangle,\ a_0^{(\ell)} \cdot z_\ell,\ \frac{-\theta_3 \cdot n \cdot a_0^{(\ell)} z_\ell}{\theta_4 \langle \mathbf{y}^{(\ell)}, \mathbf{1}\rangle} \cdot \mathbf{y}[i]\right)_{\mathbf{H}_i^*}
\end{aligned}$$

where $(a_j^{(\ell)})_j \leftarrow \Lambda_{a_0^{(\ell)}}(\mathbb{A})$.
Concerning the ciphertext components, we note that for all $i \in [n]$

– If i is queried to **LoR**:

$$\begin{aligned}
\mathbf{c}_{i,\mathsf{ipfe}} = (&\omega_{\mathsf{tag},\mathsf{S}} p_i, \omega'_{\mathsf{tag},\mathsf{S}} p_i,\ \psi_i + p_i \theta_3 (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}}), \\
&\tau \Delta\mathbf{x}[i] + \theta_4 (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})/n)_{\mathbf{H}_i} \ .
\end{aligned}$$

This includes all $i \in \mathcal{H}$ due to constraint 2 in Definition 9 of admissibility.

– Otherwise, the adversary uses a corrupted ek_i to compute

$$\begin{aligned}
\mathbf{c}_{i,\mathsf{ipfe}} = (&\omega_{\mathsf{tag},\mathsf{S}} p_i, \omega'_{\mathsf{tag},\mathsf{S}} p_i,\ \psi_i + \theta_3 (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})/n, \\
&\theta_4 (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})/n)_{\mathbf{H}_i} \ .
\end{aligned}$$

In the end, suppose that the challenge attributes satisfy the access structure of the ℓ-th key, i.e. $\mathbb{A}(\mathsf{S}) = 1$. The computation for decryption gives us:

$$\begin{aligned}
& \sum_{i=1}^{n} \left(\left(\sum_{j \in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}) \right) - (\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}}) + \mathbf{e}(\mathbf{t}_i, \mathbf{m}_i) \right) \\
\overset{(*)}{=}\ & \sum_{i=1}^{n} \left[\!\!\left[\psi_i a_0^{(\ell)} z + \frac{\theta_3 \cdot a_0^{(\ell)} z_\ell \cdot (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})}{n} \right]\!\!\right]_{\mathsf{t}} \\
& - \sum_{i=1}^{n} \left[\!\!\left[\frac{\omega}{n} \cdot \langle S, \mathbf{y}^{(\ell)}\rangle + \frac{\omega'}{n} \cdot \langle U, \mathbf{y}^{(\ell)}\rangle + \psi_i a_0^{(\ell)} z - \frac{\theta_3 \cdot a_0^{(\ell)} z_\ell \cdot (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}}) \mathbf{y}^{(\ell)}[i]}{\langle \mathbf{y}^{(\ell)}, \mathbf{1}\rangle} \right]\!\!\right]_{\mathsf{t}} \\
& + \sum_{i=1}^{n} [\![(\omega s_i + \omega' u_i + \mathbf{x}_b^*[i]) \mathbf{y}^{(\ell)}[i]]\!]_{\mathsf{t}} \\
=\ & [\![\langle \mathbf{x}_b^*, \mathbf{y}^{(\ell)}\rangle]\!]_{\mathsf{t}} \ .
\end{aligned}$$

Equality $(*)$ comes from the fact that $\sum_{i \in \mathcal{H}} \Delta\mathbf{x}[i]\mathbf{y}[i] = 0$ and $\Delta\mathbf{x}[i]\mathbf{y}[i] = 0$ for all $i \in \mathcal{C}$, by Definition 9, which then implies:

$$\tau r_0^{(\ell)} \cdot \sum_{i=1}^{n} \Delta\mathbf{x}[i]\mathbf{y}[i] = \tau r_0^{(\ell)} \cdot \sum_{i \in \mathcal{H}} \Delta\mathbf{x}[i]\mathbf{y}[i] = 0 \ .$$

Therefore, the correctness w.r.t the challenge ciphertext is preserved. The basis change is statistically indistinguishable and we have $\mathsf{Adv}(\mathsf{G}_5) = \mathsf{Adv}(\mathsf{G}_4)$.

Game G_6: In this game, we change the challenge ciphertext from using (s_i, u_i) to encrypt $\mathbf{x}_b^*[i]$ to using (s_i', u_i') to encrypt $\mathbf{x}_0^*[i]$ for $i \in [n]$. The new vectors $S' = S + \Delta S$ and $U' = U + \Delta U$ satisfy

$$S' \coloneqq S + \Delta S$$
$$U' \coloneqq U + \Delta U$$

where $(\Delta S, \Delta U)$ satisfies

$$\begin{cases} \Delta S + \mu \cdot \Delta U = 0 \\ \omega_{\mathsf{tag},\mathsf{S}} \cdot \Delta S + \omega'_{\mathsf{tag},\mathsf{S}} \cdot \Delta U = \mathbf{x}_b^* - \mathbf{x}_0^* \ . \end{cases} \tag{6}$$

We observe that if $\langle \Delta \mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$, then $\langle \Delta S, \mathbf{y}^{(\ell)} \rangle = \langle \Delta U, \mathbf{y}^{(\ell)} \rangle = 0$.
The simulator works as follows:

1. During **Setup**, all secret information are generated for msk, including the vectors (S, U) and the dual bases $(H_i, H_i')_{i\in[n]}, (F, F')$.
2. If the adversary queries **Corrupt**(i) and i is *not yet* queried to **LoR**: return ek_i that is

$$(s_i, u_i, p_i \cdot (H_i^{(1)} - \mu R_i), p_i \cdot (H_i^{(2)} + R_i), \mathbf{h}_{i,3}, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3) \ .$$

   The response to the query of i to **LoR** later will be:

LoR	$\mathbf{t}_i$	$\left[\!\left[\omega_{\mathsf{tag},\mathsf{S}} \cdot s_i' + \omega'_{\mathsf{tag},\mathsf{S}} \cdot u_i' + \mathbf{x}_0^*[i] + r_i v_i\right]\!\right]_1$

3. If i is queried to **LoR** then **Corrupt**(i) happens afterwards, return:

LoR	$\mathbf{t}_i$	$\left[\!\left[\omega_{\mathsf{tag},\mathsf{S}} \cdot s_i' + \omega'_{\mathsf{tag},\mathsf{S}} \cdot u_i' + \mathbf{x}_0^*[i]\right]\!\right]_1$
Corrupt(i)	ek_i	$(s_i', u_i', p_i \cdot (H_i^{(1)} - \mu R_i), p_i \cdot (H_i^{(2)} + R_i), \mathbf{h}_{i,3}, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$

   where $s_i' \coloneqq s_i + \Delta S[i], u_i' \coloneqq u_i + \Delta U[i]$ defined by (6).
4. For all $i \in \mathcal{H}$ whose ek_i is never revealed, in the end the simulator virtually sets

$$\mathsf{ek}_i = (s_i', u_i', p_i \cdot (H_i^{(1)} - \mu R_i), p_i \cdot (H_i^{(2)} + R_i), \mathbf{h}_{i,3}, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3) \ .$$

5. For all functional key queries, the vectors $(\mathbf{k}_{i,j})_j$ stay the same as in G_5. Meanwhile the vectors $\mathbf{k}_{i,\mathsf{ipfe}}$ are answered by

$$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} = (\langle S, \mathbf{y}^{(\ell)} \rangle, \ \langle U, \mathbf{y}^{(\ell)} \rangle, \ a_0^{(\ell)} z, \ r_0^{(\ell)} \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \ .$$

   Moreover, all queries to **Enc** are answered by

Enc	$\mathbf{t}_i$	$\left[\!\left[\chi_{\mathsf{tag}',\mathsf{S}'} \cdot s_i + \mu \chi_{\mathsf{tag}',\mathsf{S}'} \cdot u_i + x_i\right]\!\right]_1$
$\forall\, i$	$\mathbf{m}_i^{(\ell)}$	$\left[\!\left[\mathbf{y}^{(\ell)}[i]\right]\!\right]_2$.

It is straightforward to verify that the correctness of decryption is preserved w.r.t **Enc** virtually w.r.t (S', U'), due to the programmation of the RO and the relation $\Delta S + \mu \cdot \Delta U = 0$ from (6). We now turn our attention to the challenge ciphertext from **LoR**. To begin with, the group elements $\mathbf{t}_i = \left[\!\left[\omega_{\mathsf{tag},\mathsf{S}} \cdot s_i' + \omega'_{\mathsf{tag},\mathsf{S}} \cdot u_i' + \mathbf{x}_0^*[i]\right]\!\right]_1 = \left[\!\left[\omega_{\mathsf{tag},\mathsf{S}} \cdot s_i + \omega_{\mathsf{tag},\mathsf{S}} \cdot u_i + \mathbf{x}_b^*[i]\right]\!\right]_1$ is invariant under the usage of (s_i', u_i'). The only case that might be problematic is case 2, where ek_i is corrupted before $\Delta\mathbf{x}[i]$ is known. However, constraint 3 in Definition 9 of admissibility means the secret scalars $(s_i, u_i) = (s_i', u_i')$ and therefore the adversary's view over ek_i remains consistent.

The final step to complete the transition from G_5 to G_6 is proving that under SXDH, we can put the vectors $\mathbf{k}_{i,\mathsf{ipfe}}$, which are currently computed using (S, U), back to the consistent form having (S', U'). It suffices to consider the case $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \neq 0$; Otherwise $(\langle S', \mathbf{y}^{(\ell)} \rangle, \langle U', \mathbf{y}^{(\ell)} \rangle) = (\langle S, \mathbf{y}^{(\ell)} \rangle, \langle U, \mathbf{y}^{(\ell)} \rangle)$ and the keys are already in correct forms under (S', U'). However, because we are in the *adaptive* setting The functional key that are using (S, U) can be rewritten as:

$$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} = (\langle S', \mathbf{y}^{(\ell)} \rangle - \langle \Delta S, \mathbf{y}^{(\ell)} \rangle,\ \langle U', \mathbf{y}^{(\ell)} \rangle - \langle \Delta U, \mathbf{y}^{(\ell)} \rangle,\ a_0^{(\ell)} z,\ r_0^{(\ell)} \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \ .$$

We use a basis change to "correct" the extra terms $\langle \Delta S, \mathbf{y} \rangle$ and $\langle \Delta U, \mathbf{y} \rangle$. Given a DSDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $\rho := c - ab$ is either 0 or $\langle \Delta U, \mathbf{y}^{(\ell)} \rangle / 2$, the matrices (H_i, H_i') are defined as below:

$$H_i := \begin{bmatrix} 1 & 0 & \mu a & \mu a \\ 0 & 1 & -a & -a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{1,2,3,4} \qquad H_i' := \left(H_i^{-1}\right)^\top = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\mu a & a & 1 & 0 \\ -\mu a & a & 0 & 1 \end{bmatrix}_{1,2,3,4}$$

$$\mathbf{H}_i = H_i \cdot \mathbf{T}; \qquad \mathbf{H}_i^* = H_i' \cdot \mathbf{T}^*$$

This will change $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}$ and $\mathbf{h}^*_{i,3}, \mathbf{h}^*_{i,4}$. However, even for a corrupted i, all the adversary knows from ek_i is

$$p_i \cdot (H_i^{(1)} - \mu R_i),\ p_i \cdot (H_i^{(2)} + R_i)$$

where $R_i := \theta_3 H_i^{(3)} + \theta_4 H_i^{(4)} \xleftarrow{\$} \mathrm{span}(H_i^{(3)}, H_i^{(4)})$. Hence, the changes remain indistinguishable from the adversary's view. In addition, we do not have $[\![a]\!]_1$ to compute each new vector $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}$ but the simulation of the encryption oracles concerns solely the combination $\mathbf{h}_{i,1} + \mu\mathbf{h}_{i,2}$ which indeed does not involve $[\![a]\!]_1$. In fact $\mathbf{h}_{i,1} + \mu\mathbf{h}_{i,2}$ stays invariant under this basis change. Therefore the simulation can still be performed and from the corrupted ek_i, the adversary can still compute the corresponding ciphertext components (for a non-challenge tag tag).

We demonstrate how the simulator simulates the ciphertext components, as $[\![a]\!]_1$ is unknown and thus not all vectors of $\mathbf{H}_i$ can be computed. During **Setup**, the value p_i is set to $1/n$. The ciphertexts component from **LoR** can be written in $\mathbf{T}$ to see how they will be affected:

$$\begin{aligned} \mathbf{c}_{i,\mathsf{ipfe}} &= (\omega_{\mathsf{tag},\mathsf{S}} \cdot p_i,\ \omega'_{\mathsf{tag},\mathsf{S}} \cdot p_i,\ \psi_i,\ \tau\Delta\mathbf{x}[i])_{\mathbf{T}} \\ &= \big(\omega_{\mathsf{tag},\mathsf{S}} \cdot p_i,\ \omega'_{\mathsf{tag},\mathsf{S}} \cdot p_i, \\ &\qquad \psi_i + (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})a/n,\ \tau\Delta\mathbf{x}[i] + (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})a/n\big)_{\mathbf{H}_i} \ . \end{aligned}$$

If the adversary uses a corrupted ek_i to generate the i-th challenge ciphertext component, it will results in:

$$\begin{aligned} \mathbf{c}_{i,\mathsf{ipfe}} &= (\omega_{\mathsf{tag},\mathsf{S}} \cdot p_i,\ \omega'_{\mathsf{tag},\mathsf{S}} \cdot p_i,\ \psi_i,\ 0)_{\mathbf{T}} \\ &= \big(\omega_{\mathsf{tag},\mathsf{S}} \cdot p_i,\ \omega'_{\mathsf{tag},\mathsf{S}} \cdot p_i, \\ &\qquad \psi_i + (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})a/n,\ (\omega'_{\mathsf{tag},\mathsf{S}} - \mu\omega_{\mathsf{tag},\mathsf{S}})a/n\big)_{\mathbf{H}_i} \ . \end{aligned}$$

On the other hand, the ciphertexts component from **Enc** can be written in $\mathbf{T}$:

$$\begin{aligned} \mathbf{c}_{i,\mathsf{ipfe}} &= (\chi_{\mathsf{tag}',\mathsf{S}'} \cdot p_i,\ \mu\chi_{\mathsf{tag}',\mathsf{S}'} \cdot p_i,\ \psi_i^{(k)},\ 0)_{\mathbf{T}} \\ &= (\chi_{\mathsf{tag}',\mathsf{S}'} \cdot p_i,\ \mu\chi_{\mathsf{tag}',\mathsf{S}'} \cdot p_i,\ \psi_i^{(k)} - \mu a\chi_{\mathsf{tag}',\mathsf{S}'}p_i + \mu a\chi_{\mathsf{tag}',\mathsf{S}'}p_i, \\ &\qquad 0 - \mu a\chi_{\mathsf{tag}',\mathsf{S}'}p_i + \mu a\chi_{\mathsf{tag}',\mathsf{S}'}p_i)_{\mathbf{H}_i} \\ &= (\chi_{\mathsf{tag}',\mathsf{S}'} \cdot p_i,\ \mu\chi_{\mathsf{tag}',\mathsf{S}'} \cdot p_i,\ \psi_i^{(k)},\ 0)_{\mathbf{H}_i} \ , \end{aligned}$$

which retains their normal form required for the **Enc** oracle. We stress that even if the adversary uses a corrupted ek_i to craft their own ciphertext components for a non-challenge $\mathsf{tag}' \neq \mathsf{tag}$, the normal form of **Enc** will not be violated.
Upon receiving the **Extract** query for $\mathbf{y}^{(\ell)}$, first the simulator computes $\tilde{r}_0^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^*$ as in G_5 then updates $r_0^{(\ell)} := \tilde{r}_0^{(\ell)} - \frac{n a_0^{(\ell)} z^{(\ell)}}{\langle \mathbf{y}^{(\ell)}, \mathbf{1} \rangle}$. We will also use a secret sharing $(d_j^{(\ell)})_j$ of 1. We now consider the correction of the key components:

$$\begin{aligned}
\mathbf{k}_{i,j}^{(\ell)} &:= (\pi_{i,j}^{(\ell)} \cdot (j,1),\ a_j^{(\ell)} \cdot z^{(\ell)} - d_j^{(\ell)} b,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A}) \\
\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} &= (\langle S', \mathbf{y}^{(\ell)} \rangle - \langle \Delta S, \mathbf{y} \rangle,\ \langle U', \mathbf{y}^{(\ell)} \rangle - \langle \Delta U, \mathbf{y}^{(\ell)} \rangle,\ a_0^{(\ell)} z^{(\ell)},\ r_0^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \\
&\quad + (-\mu c,\ c,\ b,\ b)_{\mathbf{T}^*} \\
&= (\langle S', \mathbf{y}^{(\ell)} \rangle - \langle \Delta S, \mathbf{y}^{(\ell)} \rangle,\ \langle U', \mathbf{y}^{(\ell)} \rangle - \langle \Delta U, \mathbf{y}^{(\ell)} \rangle,\ a_0^{(\ell)} z^{(\ell)},\ r_0^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \\
&\quad + (-2\mu\rho,\ 2\rho,\ b,\ b)_{\mathbf{H}_i^*} \\
&= (\langle S', \mathbf{y}^{(\ell)} \rangle - \langle \Delta S, \mathbf{y}^{(\ell)} \rangle - 2\mu\rho,\ \langle U', \mathbf{y}^{(\ell)} \rangle - \langle \Delta U, \mathbf{y}^{(\ell)} \rangle + 2\rho,\ a_0^{(\ell)} z^{(\ell)} + b,\ r_0^{(\ell)} \mathbf{y}^{(\ell)}[i] + b)_{\mathbf{H}_i^*}.
\end{aligned}$$

Notice that $r_0^{(\ell)} \mathbf{y}^{(\ell)}[i] \cdot \mathbf{h}_{i,4}^*$ can be computed using $[\![a]\!]_2$ and $T^{(4)}, T^{(1)}, T^{(2)}$, while $b \cdot \mathbf{f}_3^*$ can be computed using $[\![b]\!]_2$ and $F^{(3)}$. It can be verified that the correctness of decryption when $\mathbb{A}(\mathsf{S}) = 1$ is preserved, using the fact that $\langle \Delta \mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$. If $\rho = 0$ then we are not correcting the key components. Otherwise, if $\rho = \langle \Delta U, \mathbf{y}^{(\ell)} \rangle / 2$, using the property $\Delta S + \mu \Delta U = 0$, the vectors $\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$ are corrected to the form they have in G_5. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_6) - \mathsf{Adv}(\mathsf{G}_5)| \leq 2 \cdot \mathsf{Adv}_{\mathbb{G}_2}^{\mathsf{DDH}}(1^\lambda)$.

The challenge ciphertext in G_6 does not depend on b anymore and thus $\mathsf{Adv}(\mathsf{G}_6) = 0$. We have the bound:

$$\mathsf{Adv}_{\mathcal{E},\mathcal{F},\mathcal{A}}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}(1^\lambda) \leq (2KP \cdot (6P+3) + 2K + 2Q + 5) \cdot \mathsf{Adv}_{\mathbb{G}_1,\mathbb{G}_2}^{\mathsf{SXDH}}(1^\lambda)$$

and the proof is concluded. □

B.5 Security Theorem for Section 5.4

Theorem 25. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be a multi-client IPFE scheme with fine-grained access control via LSSS, resulted from Section 5.4 in a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$. Then, $\mathcal{E}$ is one-time IND-secure if the SXDH assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$. More specifically, let K denote the number of functional key queries, P denote the maximum number of attributes in the access structure $\mathbb{A}$ queried for functional keys, and Q denote the number of random oracle (RO) queries. We have the following bound:*

$$\mathsf{Adv}_{\mathcal{E},\mathcal{F}^{\mathsf{IP}},\mathsf{LSSS},\mathcal{A}}^{\mathsf{mc\text{-}ind\text{-}cpa\text{-}1\text{-}chal}}(1^\lambda) \leq (Q \cdot (2KP \cdot (6P+3) + 4K + 2) + 3) \cdot \mathsf{Adv}_{\mathbb{G}_1,\mathbb{G}_2}^{\mathsf{SXDH}}(1^\lambda)$$

and in the reduction there is an additive loss $\mathcal{O}(Q \cdot t_{\mathbb{G}_1})$ in time, where $t_{\mathbb{G}_1}$ is the cost for one addition in $\mathbb{G}_1$.

Proof. We will prove the one-time IND-security. By applying Lemma 13, the standard IND-security follows. The main sequence of games is similar to that used in the proof of Theorem 14 and is depicted in Figure 14 and Figure 15. The transition from G_0 to G_1 is similar to what we have done in the proof of Theorem 14. Then, we need to program the RO in games $\mathsf{G}_2, \mathsf{G}_3$ in a similar manner as we have done for the proof of Theorem 14.

The sequence of games to go from G_3 to G_4 is given in Figure 16. Proceeding key by key, we again rely on Lemma 4 to mask the key components by another random labeling $(a_{i,j}'^{(\ell)})_j \leftarrow \Lambda_{a_{i,0}'^{(\ell)}}(\mathbb{A})$, indexed by $i \in [n]$. There is a difference comparing with the proof of Theorem 14: we use different a new random labeling $(a_{i,j}'^{(\ell)})_j \leftarrow \Lambda_{a_{i,0}'^{(\ell)}}(\mathbb{A})$ for each i, meanwhile in the less flexible construction's proof, we can use the same new labeling during the parallel application of Lemma 4 for all i. In contrast to the less flexible scheme in Section 5.2, the step to replace $a_{i,0}'^{(\ell)}$ is more delicate. The

fact that we have to use an independent new labeling for each client i comes from the current situation where each client can have a ciphertext component encrypted under different $(\mathsf{tag}, \mathsf{S}_i)$. As a result, for different $i \neq i'$, we cannot treat all $(a_{i,0}^{\prime(\ell)}, a_{i',0}^{\prime(\ell)})$ in a unified manner because if $\mathbb{A}(\mathsf{S}_i) \neq \mathbb{A}(\mathsf{S}_{i'})$, during decryption one can be removed by the KP-ABE part but the other cannot. We emphasize that even though in this case the functional key under $\mathbb{A}$ is *not* allowed to decrypt the challenge ciphertext, the adversary's view over $(a_{i,0}^{\prime(\ell)}, a_{i',0}^{\prime(\ell)})$ is already different.

Next, to go from G_3 to G_4, we use a sequence of hybrids indexed by $\ell \in [K]$ for the ℓ-th functional key. The transition from $\mathsf{G}_{3.\ell-1.0}$ to $\mathsf{G}_{3.\ell-1.1}$ is the parallel applications of Lemma 4. We recall the points that allow us to apply the lemma in parallel, similarly as in the proof of Theorem 14:

- Each client i has the vectors $\mathbf{c}_{i,\mathsf{ipfe}}$ and $\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$ lying in separate dual bases $(\mathbf{H}_i, \mathbf{H}_i^*)$ so the basis changing matrices can be written independently for each client. We note that the basis changes of $(\mathbf{F}, \mathbf{F}^*)$ does not depend on i but only on the attributes j. The computation over $\mathbf{c}$-vectors in Lemma 4 can be done for the vectors $(\mathbf{c}_{i,j})_j$ and $(\mathbf{k}_{i,j}^{(\ell)})_j$ at the same time for all $i \in [n]$, by setting the appropriate coordinates in $(\mathbf{W}, \mathbf{W}^*)$ and seeing how they are affected under these basis changes to produce the final vectors in $(\mathbf{F}, \mathbf{F}^*)$.
- When we perform a sequence of hybrids indexed by an attribute m, e.g. to introduce the factor $1/z_j$ in $\mathbf{k}_{i,j}^{(\ell)}$ where $j \neq m$, only the vectors $\mathbf{k}_{i,m}^{(\ell)}$ have non-zero coordinate at $\mathbf{f}_5^*$ and $\mathbf{c}_{i,m}$ have non-zero coordinate at $\mathbf{f}_5$. For all other $\ell' \neq \ell$, the coordinates are 0 and thus even under formal basis change, the vectors $(\mathbf{k}_{i,j}^{(\ell')})_j, \mathbf{k}_{i,m}^{(\ell')}$ are not affected. Hence, the relating basis changes will affect only those $\mathbf{k}_{i,m}^{(\ell)}, \mathbf{c}_{i,m}$ for all $i \in [n]$ at once.
- Finally, considering the ciphertexts from **Enc**, whose coordinates w.r.t the basis changes are 0 all the time, they stay invariant and the correctness is preserved.

In the proof of Theorem 14, the transition from $\mathsf{G}_{3.\ell-1.1}$ to $\mathsf{G}_{3.\ell-1.2}$ is a *statistical* transition because the same S is used for all ciphertext components of client i, which means for all $i \in [n]$ either the new labels added from Lemma 4 cannot be removed using LSSS, thus indistinguishable from a totally random value, or they regroups together to obtain the shared secret multiplied by $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$ due to the security model, where $\Delta\mathbf{x} := \mathbf{x}_1^* - \mathbf{x}_0^*$. In this new, more flexible construction, because of the potential different view w.r.t $(a_{i,0}^{\prime(\ell)}, a_{i',0}^{\prime(\ell)})$ we explained above, the transition is not statistical anymore. We proceed by a sequence of hybrids:

- $\mathsf{G}_{3.\ell-1.1.0} = \mathsf{G}_{3.\ell-1.1}$ to $\mathsf{G}_{3.\ell-1.1.1}$: We perform a formal basis change using

$$H_i := \begin{bmatrix} 1 & -\alpha_i \\ 0 & 1 \end{bmatrix}_{4,5} \qquad H_i' := \left(H_i^{-1}\right)^\top = \begin{bmatrix} 1 & 0 \\ \alpha_i & 1 \end{bmatrix}_{4,5}$$
$$\mathbf{H}_i = H_i \cdot \mathbf{T}; \qquad \mathbf{H}_i^* = H_i' \cdot \mathbf{T}^* \ .$$

  for $\alpha_i \xleftarrow{\$} \mathbb{Z}_q$ generated at **Setup**. The basis change affects $\mathbf{h}_{i,4}, \mathbf{h}_{i,5}^*$ that are kept hidden from the adversary. Under this formal basis change, the **Enc** components $\mathbf{c}_{i,\mathsf{ipfe}}$ stay the same as their 4-th coordinates are always 0. For the **LoR** components, thanks to the constraint that $\Delta\mathbf{x}[i] = 0$ for all $i \in \mathcal{C}$ all components $\mathbf{c}_{i,\mathsf{ipfe}}$ will be in the correct form after the basis change. The correctness can be easily verified if the attributes satisfy the key's policy, in both **LoR** and **Enc**. We have $\mathsf{Adv}(\mathsf{G}_{3.\ell-1.1}) = \mathsf{Adv}(\mathsf{G}_{3.\ell-1.1.1})$.
- $\mathsf{G}_{3.\ell-1.1.1}$ to $\mathsf{G}_{3.\ell-1.1.2}$: Given a DDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $\rho := c - ab$ is either 0 or a uniformly random value, we use the basis changes for $(\mathbf{H}_i, \mathbf{H}_i^*)$, in parallel for all $i \in [n]$, to mask $a_{i,0}^{\prime(\ell)}$ with a random value $r_0^{(\ell)}$. The matrices in use are:

$$H_i := \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}_{4,5} \qquad H_i' := \left(H_i^{-1}\right)^\top = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix}_{4,5}$$
$$\mathbf{H}_i = H_i \cdot \mathbf{T}; \qquad \mathbf{H}_i^* = H_i' \cdot \mathbf{T}^* \ .$$

The affected vectors are $\mathbf{h}_{i,4}, \mathbf{h}^*_{i,5}$ but they are hidden from the adversary. During **Setup**, the simulator additionally samples $\theta \xleftarrow{\$} \mathbb{Z}_q^*$ and sets:

$$\begin{aligned}\mathsf{ek}_i &:= (s_i,\ u_i,\ p_i \cdot H_i^{(1)},\ p_i \cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ \theta\mathbf{h}_{i,5},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3)\\ \mathsf{msk} &:= (S,\ U,\ (\theta_i)_i,\ \mathbf{f}_1^*,\ \mathbf{f}_2^*,\ \mathbf{f}_3^*,\ (\mathbf{h}^*_{i,1}, \mathbf{h}^*_{i,2}, \mathbf{h}^*_{i,3})_{i\in[n]})\ ,\end{aligned}$$

which has the same distribution as the real msk and ek_i and θ_i is virtually having the same value θ. Because this is a computational basis change, for all other ℓ'-th functional key, we can write them directly in $\mathbf{H}_i^*$ following the form from $\mathsf{G}_{3.\ell-1.1.1}$. For the ℓ-th key components, for all i, $\mathbf{k}_{i,\mathsf{ipfe}}$ can be written as follows:

$$\begin{aligned}\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} &= (\langle S, \mathbf{y}^{(\ell)}\rangle, \langle U, \mathbf{y}^{(\ell)}\rangle, a^{(\ell)}_{i,0} z^{(\ell)}, a'^{(\ell)}_{i,0} \cdot \mathbf{y}^{(\ell)}[i],\ 0)_{\mathbf{H}_i^*}\\ &\quad + \left(-\frac{b\theta\langle \mathbf{1}, \mathbf{y}\rangle}{\omega_{\mathsf{tag}^*}}, 0, 0, c\mathbf{y}^{(\ell)}[i], (b + d^{(\ell)}_{\mathbb{A},i}) \cdot \mathbf{y}^{(\ell)}[i]\right)_{\mathbf{T}^*}\\ &= \left(\langle S - \frac{b\theta \cdot \mathbf{1}}{\omega_{\mathsf{tag}^*}}, \mathbf{y}^{(\ell)}\rangle, \langle U, \mathbf{y}^{(\ell)}\rangle, a^{(\ell)}_{i,0} z^{(\ell)}, (a'^{(\ell)}_{i,0} + \rho - a d^{(\ell)}_{\mathbb{A},i}) \cdot \mathbf{y}^{(\ell)}[i], (b + d^{(\ell)}_{\mathbb{A},i}) \cdot \mathbf{y}^{(\ell)}[i]\right)_{\mathbf{H}_i^*}\ ,\end{aligned}$$

where $(d^{(\ell)}_{\mathbb{A},i})_i$ are chosen such that $\sum_i^n \mathbf{y}^{(\ell)}[i] d^{(\ell)}_{\mathbb{A},i} = 0$ and $\mathbf{1} = (1, 1, \ldots, 1)$. We use $[\![b]\!]_2$ to virtually simulate a new master secret key $S' := S - \frac{b\theta\mathbf{1}}{\omega_{\mathsf{tag}^*}}$, and the challenge tag tag^* is known by guessing, following G_3. At the same time, despite the fact that we cannot compute $\mathbf{h}_{i,4}$, the ciphertext components can be written directly in $\mathbf{T}$ to observe the impact of this basis change:

$$\begin{aligned}\textbf{LoR}:\ \mathbf{c}_{i,\mathsf{ipfe}} &= (\omega_{\mathsf{tag}} p_i, \omega'_{\mathsf{tag}} p_i, \psi_i, \tau\Delta\mathbf{x}[i], \theta)_{\mathbf{T}}\\ &= (\omega_{\mathsf{tag}} p_i, \omega'_{\mathsf{tag}} p_i, \psi_i, \tau\Delta\mathbf{x}[i], \theta + a\tau\Delta\mathbf{x}[i])_{\mathbf{H}_i}\\ \textbf{Enc}:\ \mathbf{c}_{i,\mathsf{ipfe}} &= (\omega_{\mathsf{tag}} p_i, \omega'_{\mathsf{tag}} p_i, \psi_i, 0, \theta)_{\mathbf{T}}\\ &= (\omega_{\mathsf{tag}} p_i, \omega'_{\mathsf{tag}} p_i, \psi_i, 0, \theta)_{\mathbf{H}_i}\ .\end{aligned}$$

We note that we are crucially using the fact the values $(d^{(\ell)}_{\mathbb{A},i})_i$ are independently generated from one functional key query to another, and in particular they depend directly on $\mathbf{y}^{(\ell)}$ in our simulation. Concerning the ciphertext components from **LoR**, we first remark that for all $i \in [n]$, the simulator virtually sets $\theta_i := \theta$ and $\alpha_i := a$. It is worth noting that $\alpha_i := a$ is independent of the key queries and thus our formal basis change's argument from $\mathsf{G}_{3.\ell-1.1.1}$ still holds. Even for $i \in \mathcal{C}$ and the adversary might use ek_i to craft their own challenge ciphertext, we still have:

$$\begin{aligned}\mathbf{c}_{i,\mathsf{ipfe}} &= (\omega_{\mathsf{tag}} p_i, \omega'_{\mathsf{tag}} p_i, \psi_i, 0, \theta)_{\mathbf{T}}\\ &= (\omega_{\mathsf{tag}} p_i, \omega'_{\mathsf{tag}} p_i, \psi_i, 0, \theta)_{\mathbf{H}_i}\ .\end{aligned}$$

because it is constrained that $\Delta\mathbf{x}[i] = 0$ for $i \in \mathcal{C}$. If $\mathbb{A}(\mathsf{S}_i) = 1$ for all $i \in [n]$, i.e. the ℓ-th functional key should work with the challenge ciphertext, one can verify from the decryption

computation that:

$$
\begin{aligned}
& \sum_{i=1}^{n} \left(\left(\sum_{j \in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}) \right) - (\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}}) + \mathbf{e}(\mathbf{t}_i, \mathbf{m}_i) \right) \\
= \; & \sum_{i=1}^{n} \left[\!\!\left[\psi_i a_0^{(\ell)} z + \tau a_{i,0}^{\prime(\ell)} \Delta\mathbf{x}[i]\mathbf{y}[i] \right]\!\!\right]_{\mathsf{t}} \\
& - \sum_{i=1}^{n} \Big[\!\!\Big[\omega p_i \cdot \langle S, \mathbf{y}^{(\ell)} \rangle + \omega' p_i \cdot \langle U, \mathbf{y}^{(\ell)} \rangle - b\theta \langle \mathbf{1}, \mathbf{y} \rangle p_i + \psi_i a_0^{(\ell)} z \\
& + \tau(a_{i,0}^{\prime(\ell)} + \rho) \Delta\mathbf{x}[i]\mathbf{y}[i] - a\tau d_{\mathbb{A},i}^{(\ell)} \Delta\mathbf{x}[i]\mathbf{y}[i] + \theta d_{\mathbb{A},i}^{(\ell)} \mathbf{y}[i] - a\tau d_{\mathbb{A},i}^{(\ell)} \Delta\mathbf{x}[i]\mathbf{y}[i] + b\theta \mathbf{y}^{(\ell)}[i] \Big]\!\!\Big]_{\mathsf{t}} \\
& + \sum_{i=1}^{n} \left[\!\!\left[(\omega s_i + \omega' u_i + \mathbf{x}_b^*[i]) \mathbf{y}^{(\ell)}[i] \right]\!\!\right]_{\mathsf{t}} \\
= \; & - \sum_{i=1}^{n} [\![\omega p_i \cdot \langle S, \mathbf{y}^{(\ell)} \rangle + \omega' p_i \cdot \langle U, \mathbf{y}^{(\ell)} \rangle + \tau\rho \Delta\mathbf{x}[i]\mathbf{y}[i] + \theta d_{\mathbb{A},i}^{(\ell)} \mathbf{y}[i]]\!]_{\mathsf{t}} \\
& + \left[\!\!\left[\omega \cdot \langle S, \mathbf{y}^{(\ell)} \rangle + \omega' \cdot \langle U, \mathbf{y}^{(\ell)} \rangle + \langle \mathbf{x}_b^*, \mathbf{y}^{(\ell)} \rangle \right]\!\!\right]_{\mathsf{t}} \\
= \; & - \left[\!\!\left[\omega \cdot \langle S, \mathbf{y}^{(\ell)} \rangle + \omega' \cdot \langle U, \mathbf{y}^{(\ell)} \rangle + \tau\rho \langle \Delta\mathbf{x}, \mathbf{y} \rangle \right]\!\!\right]_{\mathsf{t}} - \sum_{i=1}^{n} \left[\!\!\left[\theta d_{\mathbb{A},i}^{(\ell)} \mathbf{y}[i] \right]\!\!\right]_{\mathsf{t}} \\
& + \left[\!\!\left[\omega \cdot \langle S, \mathbf{y}^{(\ell)} \rangle + \omega' \cdot \langle U, \mathbf{y}^{(\ell)} \rangle + \langle \mathbf{x}_b^*, \mathbf{y}^{(\ell)} \rangle \right]\!\!\right]_{\mathsf{t}} \\
\overset{(*)}{=} \; & \left[\!\!\left[\langle \mathbf{x}_b^*, \mathbf{y}^{(\ell)} \rangle \right]\!\!\right]_{\mathsf{t}}
\end{aligned}
$$

and $(*)$ comes from the fact that $\langle \Delta\mathbf{x}, \mathbf{y} \rangle = 0$ under the adversary's admissibility as well as $\sum_{i=1}^{n} \theta d_{\mathbb{A},i}^{(\ell)} \mathbf{y}[i] = \theta \sum_{i=1}^{n} d_{\mathbb{A},i}^{(\ell)} \mathbf{y}[i] = 0$ and $\sum p_i = 1$. Hence, the correctness is also preserved w.r.t the **LoR** and **Enc** ciphertext.

If $\rho = 0$ we are in the previous hybrid $\mathsf{G}_{3.\ell-1.1.1}$, else we are in $\mathsf{G}_{3.\ell-1.1.2}$ for $r_0^{(\ell)} := \rho, \alpha_i := a, \theta_i := \theta$ and $(d_{\mathbb{A},i}^{(\ell)} \mathbf{y}^{(\ell)}[i])_i$ is used for the secret sharing in $(\mathbf{k}_{i,\mathsf{ipfe}})_i$. A final remark is that we use the same ρ for all i, so as to have the same random mask and later it can be factored out to ensure decryption's correctness. We have $|\mathsf{Adv}(\mathsf{G}_{3.\ell-1.1.1}) - \mathsf{Adv}(\mathsf{G}_{3.\ell-1.1.2})| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

- $\mathsf{G}_{3.\ell-1.1.2}$ to $\mathsf{G}_{3.\ell-1.1.3} = \mathsf{G}_{3.\ell-1.2}$: This is the reverse transition of the one from $\mathsf{G}_{3.\ell-1.1}$ to $\mathsf{G}_{3.\ell-1.1.1}$ using a formal basis change depending only on $\alpha_i \xleftarrow{\$} \mathbb{Z}_q$ generated at **Setup**. We have $\mathsf{Adv}(\mathsf{G}_{3.\ell-1.2}) = \mathsf{Adv}(\mathsf{G}_{3.\ell-1.1.3}) = \mathsf{Adv}(\mathsf{G}_{3.\ell-1.1.2})$.

The transition from $\mathsf{G}_{3.\ell-1.2}$ to $\mathsf{G}_{3.\ell-1.3} = \mathsf{G}_{3.\ell}$ is another parallel application of Lemma 4 in order to "redo" the new labeling of $a_{i,0}^{\prime(\ell)}$ for all i. In the end, after masking all K functional key queries having an extra basis change that depends on DDH in $\mathbb{G}_2$, an additional $K \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$ in the security reduction will ensue. After all K functional keys are turned semi-functional, we note that a similar argument as in Theorem 14, using the games $(\mathsf{G}_5, \mathsf{G}_6)$ in Figure 15, will work *idem* because we do not need further intervention from the 5-th coordinates of $(\mathbf{H}_i, \mathbf{H}_i^*)$. □

Game G_0 : $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $a^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a^{(\ell)}_{i,j})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a^{(\ell)}_{i,0}}(\mathbb{A})$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta_i \xleftarrow{\$} \mathbb{Z}_q$

$$\forall\, i \in \mathcal{C} \cup \mathcal{H}\ \mathsf{ek}_i \quad (s_i, \quad u_i, \quad p_i \cdot H_i^{(1)}, \quad p_i \cdot H_i^{(2)}, \quad \mathbf{h}_{i,3}, \quad \theta_i \mathbf{h}_{i,5}, \quad \mathbf{f}_1, \quad \mathbf{f}_2, \quad \mathbf{f}_3)$$

$$\begin{array}{lll|c|c|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,j} & (\ \sigma_{i,j} \cdot (1, -j) & \psi_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,j} & (\ \sigma'_{i,j} \cdot (1, -j) & \psi'_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,j} & (\ \pi^{(\ell)}_{i,j} \cdot (j, 1) & a^{(\ell)}_{i,j} \cdot z^{(\ell)} & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{lll|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{t}_i & [\![\omega_{\mathsf{tag}} \cdot s_i + \omega'_{\mathsf{tag}} \cdot u_i + \mathbf{x}_b^*[i]]\!]_1 \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{t}_i & [\![\chi_{\mathsf{tag}'} \cdot s_i + \chi'_{\mathsf{tag}'} \cdot u_i + x_i]\!]_1 \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{m}_i^{(\ell)} & [\![\mathbf{y}^{(\ell)}[i]]\!]_2
\end{array}$$

$$\begin{array}{lll|c|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i & 0 & \theta_i &)_{\mathbf{H}_i} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\chi_{\mathsf{tag}'} & p_i\chi'_{\mathsf{tag}'} & \psi'_i & 0 & \theta_i &)_{\mathbf{H}_i} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (\ \langle S, \mathbf{y}^{(\ell)} \rangle & \langle U, \mathbf{y}^{(\ell)} \rangle & a^{(\ell)}_{i,0} z^{(\ell)} & 0 & d^{(\ell)}_{\mathbb{A},i} &)_{\mathbf{H}_i^*}
\end{array}$$

Game G_1 : $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $a^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a^{(\ell)}_{i,j})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a^{(\ell)}_{i,0}}(\mathbb{A})$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} = \mathbf{x}_0^* - \mathbf{x}_1^*$

$$\begin{array}{lll|c|c|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,j} & (\ \sigma_{i,j} \cdot (1, -j) & \psi_i & \tau\Delta\mathbf{x}[i] & 0 & \tau\Delta\mathbf{x}[i] z_j & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,j} & (\ \sigma'_{i,j} \cdot (1, -j) & \psi'_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,j} & (\ \pi^{(\ell)}_{i,j} \cdot (j, 1) & a^{(\ell)}_{i,j} \cdot z^{(\ell)} & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}$$

$$\begin{array}{lll|c|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i & \tau\Delta\mathbf{x}[i] & \theta_i &)_{\mathbf{H}_i} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\chi_{\mathsf{tag}'} & p_i\chi'_{\mathsf{tag}'} & \psi'_i & 0 & \theta_i &)_{\mathbf{H}_i} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (\ \langle S, \mathbf{y}^{(\ell)} \rangle & \langle U, \mathbf{y}^{(\ell)} \rangle & a^{(\ell)}_{i,0} z^{(\ell)} & 0 & d^{(\ell)}_{\mathbb{A},i} &)_{\mathbf{H}_i^*}
\end{array}$$

Game $\mathsf{G}_2 : \mathsf{H}(\mathsf{tag}) = [\![\mathsf{RF}(\mathsf{tag})]\!]_1 := ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $[\![\mathsf{H}(\mathsf{tag}')]\!]_1 = [\![\mathsf{RF}(\mathsf{tag}')]\!]_1 := ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$

Game G_3 : $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag}) := [\![\mathsf{RF}(\mathsf{tag})]\!]_1 := ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') := [\![\mathsf{RF}'(\mathsf{tag}') \cdot (1,\ \mu)]\!]_1 = ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}'}]\!]_1), \sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta'_i, \theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} = \mathbf{x}_0^* - \mathbf{x}_1^*$

$$\forall\, i \in \mathcal{C} \cup \mathcal{H}\ \mathsf{ek}_i \quad (s_i, \quad u_i, \quad p_i \cdot H_i^{(1)}, \quad p_i \cdot H_i^{(2)}, \quad \mathbf{h}_{i,3}, \quad \theta_i \mathbf{h}_{i,5}, \quad \mathbf{f}_1, \quad \mathbf{f}_2, \quad \mathbf{f}_3)$$

$$\begin{array}{lll|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{t}_i & [\![\omega_{\mathsf{tag}} \cdot s_i + \omega'_{\mathsf{tag}} \cdot u_i + \mathbf{x}_b^*[i]]\!]_1 \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{t}_i & [\![\chi_{\mathsf{tag}'} \cdot s_i + \mu\chi_{\mathsf{tag}'} \cdot u_i + x_i]\!]_1 \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{m}_i^{(\ell)} & [\![\mathbf{y}^{(\ell)}[i]]\!]_2
\end{array}$$

$$\begin{array}{lll|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i & \tau\Delta\mathbf{x}[i] & \theta'_i &)_{\mathbf{H}_i} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (\ p_i\chi_{\mathsf{tag}'} & p_i\mu\chi_{\mathsf{tag}'} & \psi'_i & 0 & \theta_i &)_{\mathbf{H}_i} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (\ \langle S, \mathbf{y}^{(\ell)} \rangle & \langle U, \mathbf{y}^{(\ell)} \rangle & a^{(\ell)}_{i,0} z^{(\ell)} & r_0^{(\ell)} \mathbf{y}^{(\ell)}[i] & d^{(\ell)}_{\mathbb{A},i} &)_{\mathbf{H}_i^*}
\end{array}$$

Fig. 14: Games $\mathsf{G}_0, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3$ for Theorem 25. The sets $\mathcal{H}$ and $\mathcal{C}$ contain honest and corrupted $i \in [n]$, respectively. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1, \ldots, K\}$ for the functional key queries. The function H is modeled as a random oracle. In G_2 we use a random function $\mathsf{RF} : \mathsf{Tag} \to (\mathbb{Z}_q^*)^2$.

Game G_4 : $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $a'^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta'_i, \theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} = \mathbf{x}^*_0 - \mathbf{x}^*_1$

$$
\begin{array}{lll|c|c|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,j} & (& \sigma_{i,j} \cdot (1,-j) & \psi_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,j} & (& \sigma'_{i,j} \cdot (1,-j) & \psi'_i & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,j} & (& \pi^{(\ell)}_{i,j} \cdot (j,1) & a^{(\ell)}_{i,j} \cdot z_\ell & 0 & 0 & 0 & 0 & 0 &)_{\mathbf{F}^*}
\end{array}
$$

$$
\begin{array}{lll|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (& p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i & \tau\Delta\mathbf{x}[i] & \theta'_i &)_{\mathbf{H}_i} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (& p_i\chi_{\mathsf{tag}'} & p_i\chi'_{\mathsf{tag}'} & \psi'_i & 0 & \theta_i &)_{\mathbf{H}_i} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (& \langle S, \mathbf{y}^{(\ell)} \rangle & \langle U, \mathbf{y}^{(\ell)} \rangle & a^{(\ell)}_{i,0} z_\ell & r^{(\ell)}_0 \cdot \mathbf{y}^{(\ell)}[i] & d^{(\ell)}_{\mathbb{A},i} &)_{\mathbf{H}^*_i}
\end{array}
$$

Game G_5 : $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag}, \mathsf{S}) := ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}', \mathsf{S}') := ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}'}]\!]_1)$, $R_i \xleftarrow{\$} \mathrm{span}(H^{(3)}_i, H^{(4)}_i) \subseteq \mathbb{Z}^4_q$

$$\forall\, i\ \mathsf{ek}_i\ \ (s_i,\quad u_i,\quad p_i \cdot (H^{(1)}_i - \mu R_i),\quad p_i \cdot (H^{(2)}_i + R_i),\quad \mathbf{h}_{i,3},\quad \theta_i\mathbf{h}_{i,5},\quad \mathbf{f}_1,\quad \mathbf{f}_2,\quad \mathbf{f}_3)$$

Game G_6 : $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}^4_q$, $\mu, v^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag}) := ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') := ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}'}]\!]_1)$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta'_i, \theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} = \mathbf{x}^*_0 - \mathbf{x}^*_1$. We also define $S' = S + \Delta S, U' = U + \Delta U$, where $\Delta S, \Delta U \in \mathbb{Z}^n_q$ s.t. $\Delta S + \mu\Delta U = 0$ and $\omega_{\mathsf{tag}} \cdot \Delta S + \omega'_{\mathsf{tag}} \cdot \Delta U = \mathbf{x}^*_b - \mathbf{x}^*_0$

$$\forall\, i\ \mathsf{ek}_i\ \ (s'_i,\quad u'_i\quad p_i \cdot (H^{(1)}_i - \mu R_i),\quad p_i \cdot (H^{(2)}_i + R_i),\quad \mathbf{h}_{i,3},\quad \theta_i\mathbf{h}_{i,5},\quad \mathbf{f}_1,\quad \mathbf{f}_2,\quad \mathbf{f}_3)$$

$$
\begin{array}{lll|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S}) & \mathbf{t}_i & & \left[\!\!\left[\omega_{\mathsf{tag}} \cdot s'_i + \omega'_{\mathsf{tag}} \cdot u'_i + \mathbf{x}^*_0[i]\right]\!\!\right]_1 \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{t}_i & & \left[\!\!\left[\chi_{\mathsf{tag}'} \cdot s'_i + \mu\chi_{\mathsf{tag}'} \cdot u'_i + x_i\right]\!\!\right]_1 \\
\forall\, i & \mathbf{m}^{(\ell)}_i & & [\![\mathbf{y}^{(\ell)}[i]]\!]_2
\end{array}
$$

$$
\begin{array}{lll|c|c|c|c|l}
\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S}) & \mathbf{c}_{i,\mathsf{ipfe}} & (& p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i & \tau'\Delta\mathbf{x}[i] & \theta'_i &)_{\mathbf{H}_i} \\
\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}') & \mathbf{c}_{i,\mathsf{ipfe}} & (& p_i\chi_{\mathsf{tag}'} & p_i\mu\chi_{\mathsf{tag}'} & \psi'_i & 0 & \theta_i &)_{\mathbf{H}_i} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (& \langle S', \mathbf{y}^{(\ell)} \rangle & \langle U', \mathbf{y}^{(\ell)} \rangle & a^{(\ell)}_{i,0} z_\ell & r^{(\ell)}_0 \mathbf{y}^{(\ell)}[i] & d^{(\ell)}_{\mathbb{A},i} &)_{\mathbf{H}^*_i}
\end{array}
$$

Fig. 15: Games $\mathsf{G}_4, \mathsf{G}_5, \mathsf{G}_6$ for Theorem 25. The transition from G_3 to G_4 is given in Figure 16. The sets $\mathcal{H}$ and $\mathcal{C}$ contain honest and corrupted $i \in [n]$, respectively. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1, \ldots, K\}$ for the functional key queries.

Game $\mathsf{G}_{3.\ell-1.0} = \mathsf{G}_{3.\ell-1}$

Game $\mathsf{G}_{3.\ell-1.1}$: $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $a'^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_1[i] - \mathbf{x}^*_0[i]$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1,-j)$	ψ_i	$\tau\Delta\mathbf{x}[i]$	0	$\tau\Delta\mathbf{x}[i]z_j$	0	0	$)_{\mathbf{F}}$
$\mathbf{Enc}$	$\mathbf{c}_{i,j}$	(	$\sigma'_{i,j} \cdot (1,-j)$	ψ'_i	0	0	0	0	0	$)_{\mathbf{F}}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' \neq \ell$	$\mathbf{k}^{(\ell')}_{i,j}$	(	$\pi^{(\ell')}_{i,j} \cdot (j,1)$	$a^{(\ell')}_{i,j} \cdot z^{(\ell')}$	0	0	0	0	0	$)_{\mathbf{F}^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}^{(\ell)}_{i,j}$	(	$\pi^{(\ell)}_{i,j} \cdot (j,1)$	$a^{(\ell)}_{i,j} \cdot z^{(\ell)}$	0	0	$a'^{(\ell)}_{i,j}\mathbf{y}^{(\ell)}[i]/z_j$	0	0	$)_{\mathbf{F}^*}$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\omega_{\mathsf{tag}}p_i$	$\omega'_{\mathsf{tag}}p_i$	ψ_i	$\tau\Delta\mathbf{x}[i]$	θ_i	$)_{\mathbf{H}_i}$
$(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\chi_{\mathsf{tag}'}p_i$	$\chi'_{\mathsf{tag}'}p_i$	ψ'_i	0	θ_i	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' < \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell')}\rangle$	$\langle U, \mathbf{y}^{(\ell')}\rangle$	$a^{(\ell')}_{i,0}z^{(\ell')}$	$r^{(\ell')}_0 \cdot \mathbf{y}^{(\ell')}[i]$	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell)}\rangle$	$\langle U, \mathbf{y}^{(\ell)}\rangle$	$a^{(\ell)}_{i,0}z^{(\ell)}$	$a'^{(\ell)}_{i,0} \cdot \mathbf{y}^{(\ell)}[i]$	$d^{(\ell)}_{\mathbb{A},i}$	$)_{\mathbf{H}^*_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' > \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell')}\rangle$	$\langle U, \mathbf{y}^{(\ell')}\rangle$	$a^{(\ell')}_{i,0}z^{(\ell')}$	0	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$

Game $\mathsf{G}_{3.\ell-1.1.1}$: $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta'_i, \theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_1[i] - \mathbf{x}^*_0[i]$

$\mathbf{LoR}$	$\mathbf{c}_{i,\mathsf{ipfe}}$	($\cdots$	ψ_i	$\tau\Delta\mathbf{x}[i]$	$\theta_i + \alpha_i\tau\Delta\mathbf{x}[i]$	$)_{\mathbf{H}_i}$
$\mathbf{Enc}$	$\mathbf{c}_{i,\mathsf{ipfe}}$	($\cdots$	ψ'_i	0	θ_i	$)_{\mathbf{H}_i}$
$\forall\, i,\ \ell' < \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	($\cdots$	$a^{(\ell')}_{i,0}z^{(\ell')}$	$r^{(\ell')}_0 \cdot \mathbf{y}^{(\ell')}[i] - \alpha_i d^{(\ell')}_i$	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$
$\forall\, i$	$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$	($\cdots$	$a^{(\ell)}_{i,0}z^{(\ell)}$	$a'^{(\ell)}_{i,0}\mathbf{y}^{(\ell)}[i] - \alpha_i d^{(\ell)}_{\mathbb{A},i}$	$d^{(\ell)}_{\mathbb{A},i}$	$)_{\mathbf{H}^*_i}$
$\forall\, i,\ \ell' > \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	($\cdots$	$a^{(\ell')}_{i,0}z^{(\ell')}$	$- \alpha_i d^{(\ell')}_i$	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$

Game $\mathsf{G}_{3.\ell-1.1.2}$: $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta'_i, \theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_1[i] - \mathbf{x}^*_0[i]$

$\mathbf{LoR}$	$\mathbf{c}_{i,\mathsf{ipfe}}$	($\cdots$	ψ_i	$\tau\Delta\mathbf{x}[i]$	$\theta_i + \alpha_i\tau\Delta\mathbf{x}[i]$	$)_{\mathbf{H}_i}$
$\mathbf{Enc}$	$\mathbf{c}_{i,\mathsf{ipfe}}$	($\cdots$	ψ'_i	0	θ_i	$)_{\mathbf{H}_i}$
$\forall\, i,\ \ell' < \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	($\cdots$	$a^{(\ell')}_{i,0}z^{(\ell')}$	$r^{(\ell')}_0 \cdot \mathbf{y}^{(\ell')}[i] - \alpha_i d^{(\ell')}_i$	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$
$\forall\, i$	$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$	($\cdots$	$a^{(\ell)}_{i,0}z^{(\ell)}$	$(a'^{(\ell)}_{i,0} + r^{(\ell)}_0)\mathbf{y}^{(\ell)}[i] - \alpha_i d^{(\ell)}_{\mathbb{A},i}$	$d^{(\ell)}_{\mathbb{A},i}$	$)_{\mathbf{H}^*_i}$
$\forall\, i,\ \ell' > \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	($\cdots$	$a^{(\ell')}_{i,0}z^{(\ell')}$	$- \alpha_i d^{(\ell')}_i$	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$

Game $\mathsf{G}_{3.\ell-1.2} = \mathsf{G}_{3.\ell-1.1.3}$: $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_1[i] - \mathbf{x}^*_0[i]$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1,-j)$	ψ_i	$\tau\Delta\mathbf{x}[i]$	0	$\tau\Delta\mathbf{x}[i]z_j$	0	0	$)_{\mathbf{F}}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,j}$	(	$\sigma'_{i,j} \cdot (1,-j)$	ψ'_i	0	0	0	0	0	$)_{\mathbf{F}}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' \neq \ell$	$\mathbf{k}^{(\ell')}_{i,j}$	(	$\pi^{(\ell')}_{i,j} \cdot (j,1)$	$a^{(\ell')}_{i,j} \cdot z^{(\ell')}$	0	0	0	0	0	$)_{\mathbf{F}^*}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}^{(\ell)}_{i,j}$	(	$\pi^{(\ell)}_{i,j} \cdot (j,1)$	$a^{(\ell)}_{i,j} \cdot z^{(\ell)}$	0	0	$a'^{(\ell)}_j\mathbf{y}^{(\ell)}[i]/z_j$	0	0	$)_{\mathbf{F}^*}$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\omega_{\mathsf{tag}}p_i$	$\omega'_{\mathsf{tag}}p_i$	ψ_i	$\tau\Delta\mathbf{x}[i]$	θ_i	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\chi_{\mathsf{tag}'}p_i$	$\chi'_{\mathsf{tag}'}p_i$	ψ'_i	0	θ_i	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' < \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell')}\rangle$	$\langle U, \mathbf{y}^{(\ell')}\rangle$	$a^{(\ell')}_{i,0}z^{(\ell')}$	$r^{(\ell')}_0 \cdot \mathbf{y}^{(\ell')}[i]$	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell)}\rangle$	$\langle U, \mathbf{y}^{(\ell)}\rangle$	$a^{(\ell)}_{i,0}z^{(\ell)}$	$(a'^{(\ell)}_{i,0} + r^{(\ell)}_0) \cdot \mathbf{y}^{(\ell)}[i]$	$d^{(\ell)}_{\mathbb{A},i}$	$)_{\mathbf{H}^*_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H},\ \ell' > \ell$	$\mathbf{k}^{(\ell')}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell')}\rangle$	$\langle U, \mathbf{y}^{(\ell')}\rangle$	$a^{(\ell')}_{i,0}z^{(\ell')}$	0	$d^{(\ell')}_i$	$)_{\mathbf{H}^*_i}$

Game $\mathsf{G}_{3.\ell-1.3} = \mathsf{G}_{3.\ell}$: $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i}\theta_i = 0$, $\theta'_i, \theta_i \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_1[i] - \mathbf{x}^*_0[i]$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1,-j)$	ψ_i	$\tau\Delta\mathbf{x}[i]$	0	$\tau\Delta\mathbf{x}[i]z_j$	0	0	$)_{\mathbf{F}}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,j}$	(	$\sigma'_{i,j} \cdot (1,-j)$	ψ'_i	0	0	0	0	0	$)_{\mathbf{F}}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}^{(\ell)}_{i,j}$	(	$\pi^{(\ell)}_{i,j} \cdot (j,1)$	$a^{(\ell)}_{i,j} \cdot z^{(\ell)}$	0	0	0	0	0	$)_{\mathbf{F}^*}$

$\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{S})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\omega_{\mathsf{tag}}p_i$	$\omega'_{\mathsf{tag}}p_i$	ψ_i	$\tau\Delta\mathbf{x}[i]$	θ_i	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{S}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$\chi_{\mathsf{tag}'}p_i$	$\chi'_{\mathsf{tag}'}p_i$	ψ'_i	0	θ_i	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$	(	$\langle S, \mathbf{y}^{(\ell)}\rangle$	$\langle U, \mathbf{y}^{(\ell)}\rangle$	$a^{(\ell)}_{i,0}z^{(\ell)}$	$r^{(\ell)}_0 \cdot \mathbf{y}^{(\ell)}[i]$	$d^{(\ell)}_{\mathbb{A},i}$	$)_{\mathbf{H}^*_i}$

Fig. 16: The sequence of hybrids to go from $\mathsf{G}_{3.\ell-1}$ to $\mathsf{G}_{3.\ell}$, where $\ell \in [K]$. We have $\mathsf{G}_{3.0} = \mathsf{G}_3$ and $\mathsf{G}_{3.K} = \mathsf{G}_4$ in the proof of Theorem 25. The sets $\mathcal{H}$ and $\mathcal{C}$ contain honest and corrupted $i \in [n]$, respectively. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in S for ciphertext components. The index ℓ runs in $\{1, \ldots, K\}$ for the functional key queries. The function H is modeled as a random oracle.