Multi-Client Functional Encryption with Fine-Grained Access Control

Ky Nguyen[1,2], Duong Hieu Phan[3], and David Pointcheval[1,2]

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

Abstract. Multi-Client Functional Encryption (MCFE) has been considered as an important primitive for making functional encryption useful in practice. It covers the ability to compute joint function over data from multiple parties similar to Multi-Input Functional Encryption (MIFE) but it handles information leakage better than MIFE. Both the MCFE and MIFE primitives are aimed at applications in multi-user settings where decryption can be correctly output for legitimate users only. In such a setting, the problem of dealing with access control in a fine-grained manner is particularly relevant. In this paper, we introduce a framework for MCFE with fine-grained access control and propose constructions for both single-client and multi-client settings, 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 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, while MCFE is more difficult to achieve than MIFE (one can obtain MIFE from MCFE by making all the labels in MCFE a fixed public constant).

Keywords: Multi-client functional encryption, access control, adaptive security.

1 Introduction

1.1 Functional Encryption

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) and Attribute-Based Encryption (ABE), 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. 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]. 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.

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 $(x_1, \ldots, x_n)$ where the components are independent. An index i for each client and, in the case of MCFE, a (typically time-based) label ℓ are used for every encryption: $(c_1 = \mathsf{Enc}(1, x_1, \ell), \ldots, c_n = \mathsf{Enc}(n, x_n, \ell))$. Anyone owning a functional decryption key dk_f, for an n-ary function f and multiple ciphertexts (for the same label ℓ, in the case of MCFE) can compute $f(x_1, \ldots, x_n)$ but nothing else about the individual x_i's. The difference between MIFE and MCFE seems minor (MCFE is essentially MIFE with labels, that limit combinations into vectors) but we will see that this leads to very different constructions, as clients have to be able to implicitly coordinate together on the label, and different usability in practice. In particular, in MCFE, the combination of ciphertexts generated for different labels does not give a valid global ciphertext and the adversary learns nothing from it. 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 theses keys might get corrupted. And one has to deal with corruptions of encryption keys in multi-user settings.

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). Finally, as already explained, FE is a generalization of IBE and ABE, and after having been illustrated with IBE and ABE, linear and quadratic evaluations 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).

1.2 Related Work

Abdalla *et al.* [ACGU20] have been the first (and this is the unique paper) to address this problem, for FE and MIFE. 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: labels can be used as specific attributes, and labels can be embedded in policies to automatically obtain multi-client settings. While this appears to be natural at first glance, we do not see how to implement it efficiently because a label value is generated during the encryption process: if we embed a label as an attribute in the ciphertext, we must generate a key for each label value for each user, which becomes infeasible. It thus remains a challenging open problem to construct an efficient MCFE supporting access-control structure. In this paper, we take a completely different approach than in [ACGU20] to answer this question. Interestingly, our schemes are more efficient than theirs: we build MCFE for Inner-Product, with any LSSS access-structure and adaptive-security

in the random-oracle model, that is more efficient than their MIFE. In addition, removing labels leads to an MIFE scheme for Inner-Product, with any LSSS access-structure, adaptively-secure in the standard model, still more efficient than their scheme.

1.3 Contributions

Single-client setting. We propose new schemes in which the 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 thus this 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 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 label, 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 of order $O(nd)$, where d is the number of attributes specified during encryption, and does not suffer a quadratic blow-up of n^2 group elements.

Comparisons and discussions. We now focus on the FE schemes for Inner-Product, with fine-grained access control, in the pairing-based setting, with comparisons with the schemes from [ACGU20] in Table 1. There are other schemes for single-client IPFE with fine-grained access control based on other assumptions such as LWE, *e.g.* [LLW21, PD21], but because they were not generalized to the multi-input or multi-client settings, which are our main objectives, we do not consider them.

1.4 Technical Overview

As shown in Table 1, our selectively-secure construction (in Section 4.1) suffers a slight deterioration in efficiency and security because the ciphertexts are larger and the model is indistinguishability-based rather than simulation-based as considered in [ACGU20, Section 3.1]. Despite slightly larger ciphertexts, we are able to lift our selectively-secure construction to an adaptively-secure construction in Section 4.2 whose ciphertexts are smaller than the ones in [ACGU20, Section 3.2], while achieving the same level of security. Moreover, the adaptively-secure construction is a straightforward generalization of the selectively-secure one, as the computation stays the same in both (see Figure 3). The adaptively-secure construction in [ACGU20, Section 3.2] uses *function encodings* to handle the access control, and instantiations of function encodings are provided in [ACGU20, Appendix B], for various predicate classes, among which the *read-once monotone span programs* are the most expressive ones. On the other hand, we can also express the predicate class in our adaptively-secure construction using LSSS-realizable access structures, which is equivalent to MSP.

Another approach to achieve single-client adaptive security. One of our main ideas to construct a functional encryption scheme with fine-grained access control is to use a secret sharing scheme for creating shares of a secret value $a_0 \xleftarrow{\$} \mathbb{Z}_q$, which acts as a mask for the IPFE-related ciphertext of Agrawal *et al.*'s type [ALS16], following a linear secret sharing scheme implementing a monotone access structure $\mathbb{A}$ over a set Att of attributes. The shares will then be embedded in the

Scheme	$\mathcal{P}$	$\lvert\mathsf{ct}\rvert$	Security	Type
[ACGU20, Sect. 3.1]	MSP	$n + 2d + 2$	sel-sim	Single-client
[ACGU20, Sect. 3.2]	roMSP	$3nd + 3d + 2$	ad-ind	
Sect. 4.1	LSSS	$n + 8d + 4$	sel-ind	
Sect. 4.2	LSSS	$nd + 2n + 7d + 3$	ad-ind	
[ACGU20, Sect. 6.2] applied to [ACGU20, Sect. 3.1]	MSP	$n^2 + 2nd + 2n$	ad-ind	MIFE, generic transformation
Sect. 5.2	LSSS	$8nd + 5n$	ad-ind	MCFE

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 most expressive predicate class that can be handled (column $\mathcal{P}$), the achieved level of security (column **Security**), and whether we are in the single-client or multi-client/multi-input setting (column **Type**). We denote by n the dimension of vectors to compute inner-products and by d the number of attributes used in the key's policy. The abbreviations MSP, roMSP and LSSS stands for *monotone span programs, read-once monotone span programs* and *linear secret sharing schemes*, respectively. The abbreviations $\{\mathsf{sel}, \mathsf{ad}, \mathsf{ind}, \mathsf{sim}\}$ denote selective security, adaptive security, indistinguishability-based, and simulation-based, respectively. All schemes require the inner-products to be polynomially small and their security relies on SXDH.

functional secret key components $(\mathbf{k}_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ where $\mathsf{List\text{-}Att}(\mathbb{A})$ is the list of attributes appearing in the access structure $\mathbb{A}$. 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 $\mathbf{k}_{\mathsf{root}}$, and allow functional decryption. The key components are constructed as vectors in a *Dual Pairing Vector Space* (DPVS). Roughly speaking, a DPVS is a (prime-order) bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, \mathbf{e})$ that we enhance with the notion of vector addition, scalar multiplication, and dual orthogonal bases. We also define the product of two vectors over $\mathbb{G}_1$ and $\mathbb{G}_2$ in DPVS, which uses the pairing $\mathbf{e}$ and results in an element in $\mathbb{G}_\mathsf{t}$ whose exponent in g_t is the inner-product of the vectors of exponents from the two initial vectors. The access control is now handled by these vectors in DPVS, where 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 R of attributes are embedded in the ciphertext using vectors $\{(\mathbf{c}_j)_{j \in \mathsf{R}}, \mathbf{c}_{\mathsf{root}}\}$ over $\mathbb{G}_1$. We use the techniques for adaptively-secure ABE introduced in the original work of Okamoto and Takashima [OT10, OT12a, OT12b] to argue the security of this KP-ABE part in our scheme. Interestingly, there is a twist stemming from the security model when integrating ABE into FE: during the security game, an adversary can additionally query for keys that work with the challenge ciphertext, i.e. the key's policy is satisfied. In vein of the *dual-system methodology* 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. To circumvent this obstacle, we resort to a slight variant of the technique in [OT10, OT12a, OT12b] (see Section 3), while accepting an anathema to use DPVSes of dimensions linear in the dimension n of vectors for inner-products. Our single-client constructions are presented in Section 4.

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. In the multi-client setting, each client must use different DPVSes for the KP-ABE part. To recall, from the single-client construction, the DPVSes are already

of dimension linear in the number of clients to achieve adaptive security and a naive duplication would add a quadratic factor in the communication. In general, we want to avoid using n different pairs of bases for the ciphertext components $\{(\mathbf{c}_{i,j})_{j\in\mathsf{R}}, \mathbf{c}_{i,\mathsf{root}}\}$, for each $i \in [n]$, where each basis is of dimension n resulting from the adaptively secure single-client construction. Since we are in the pairing-based setting and using DPVS, it is equivalently effective to concentrate on compressing the dual bases used for key components. The intriguing point we observe is as long as each client uses an independent DPVS, the technique we use to take care of those 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 all key components so as to answer the adversary's adaptive key queries, rather than centralizing such information in few vectors of big dimension.

This idea might seem counter-intuitive, especially when we ponder the original techniques introduced by Okamoto and Takashima [OT10, OT12a, OT12b] trying at all cost to escape duplicating the bases. Technically, they used a *hidden* part of the dual bases in DPVS, which is never used in real life and can be enlarged conveniently, to amplify randomness in multiple vectors of the functional key that are indexed by attributes of the policy. This quantity of amplified randomness will be later used to turn the keys semi-functional. At first glance, it seems obligatory to duplicate the bases for each $i \in [n]$ to express the KP-ABE related key components $\{(\mathbf{k}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}_{i,\mathsf{root}}\}$ in order to control the decryption over all n clients' ciphertexts, while keeping the dimension *linear in* n for adaptive security. It is true that the vectors $(\mathbf{k}_{i,\mathsf{root}})_i$ must be put in n independent bases for n clients. Surprisingly, our main insight is that the other key components $(\mathbf{k}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ that are indexed by attributes and serve the randomness amplification can be put in *the same basis* for all clients $i \in [n]$. Indeed, the argument by Okamoto and Takashima depends crucially but only on the attributes $j \in \mathsf{List\text{-}Att}(\mathbb{A})$. Moreover, the process of making $\mathbf{k}_{i,\mathsf{root}}$ semi-functional using $(\mathbf{k}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ afterwards employs the access structure specified in the key and the attributes in the ciphertext but *not* depending on i. Hence, if we have n collections of vectors $((\mathbf{k}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}_{i,\mathsf{root}})_i$ the whole process can be applied in parallel for those n collections. We emphasize that this parallel process is feasible thanks to an indispensable smooth control, as low as the level of the vectors' coordinates, in DPVS. This potential of parallelization helps us spread the necessary information for answering adaptive key queries, which accounts for the linearly large dimension, into n collections $\{(\mathbf{k}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}, \mathbf{k}_{i,\mathsf{root}}\}_{i\in[n]}$. 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.

From single-client to multi-client. In Section 5, we explain in details how we obtain an adaptively-secure multi-client version without tremendous modifications in the single-client adaptively secure construction's mechanism. The IPFE-related part of Agrawal *et al.*'s type [ALS16] can be dealt with in a similar manner Chotard *et al.* [CDG+18a] did. We use the "duplicate-and-compress" technique to leverage the KP-ABE part from single-client to multi-client setting. We have to duplicate the DPVSes for all clients, but at the same time this duplication helps us save a linear factor in the dimension, leading to ciphertext's size being roughly the same as the one in the single-client adaptively-secure scheme. Our MCFE scheme needs a *random oracle* (RO) and in Section 5.4 we discuss equally how one can obtain an MIFE in the *standard* model from our scheme. Putting the main ideas forward, the removal of the RO from MCFE to achieve MIFE is not trivial due to the delicacies of access control for combining ciphertexts and decrypting them with functional keys, besides the labels used when encrypting. Our solution to obtain an MCFE in ROM that leads to an MIFE scheme without RO exploits another layer of secret sharing in the keys, with minimal changes to the MCFE scheme from Section 5.2.

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. 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 $(\mathsf{G}_0, \mathsf{G}_1, \ldots, \mathsf{G}_i, \ldots, \mathsf{G}_L)$, which is 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 construction.

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 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.*

Definition 3. *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. We recall below the necessary definitions and techniques used in DPVS.

We use $\mathbb{G}_1$ as a running example for the definitions. Let us fix $N \in \mathbb{N}$ and consider $\mathbb{G}_1^N$ having N copies of $\mathbb{G}_1$ in the following manner:

– 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 $-\mathbf{x} := [\![(-x_1, \ldots, -x_N)]\!]_1$, which is well-defined as $\mathbb{G}_1$ is a group written additively. We note that $\mathbb{G}_1^N$ equipped with these addition and scalar multiplications satisfies the axioms of a vector space.
- 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 $B \in \mathbb{Z}_q^{N \times N}$ 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. 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:

$$\mathbf{x} \times \mathbf{y} := \prod_{i=1}^N \mathbf{e}(\mathbf{x}[i], \mathbf{y}[i]) = \left[\!\!\left[\sum_{i=1}^N x_i y_i \right]\!\!\right]_{\mathsf{t}} = [\![\langle (x_1, \ldots, x_N), (y_1, \ldots, y_N) \rangle]\!]_{\mathsf{t}} \ .$$

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 for every $i, j \in [N]$:

$$\mathbf{b}_i \times \mathbf{b}_j^* = \left[\!\!\left[\langle B^{(i)}, B'^{(j)} \rangle \right]\!\!\right]_{\mathsf{t}} = [\![\delta_{i,j}]\!]_{\mathsf{t}}$$

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 $B \xleftarrow{\$} \mathbb{Z}_q^{N \times N}$, 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 security of our constructions. The details of such basis changes can be found in Appendix A.1.

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.

Definition 4 (Access Structure). *Let* $\mathsf{Att} = \{\mathsf{att}_1, \mathsf{att}_2, \ldots, \mathsf{att}_m\}$ *be a finite set 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*.*

Given a set of attributes $\mathsf{R} \subseteq \mathsf{Att}$, we write $\mathbb{A}(\mathsf{R}) = 1$ if and only if there exists $A \subseteq \mathsf{R}$ such that A is authorized.

Definition 5 (Secret Sharing Scheme). *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:*

- *Any authorized set in* $\mathbb{A}$ *can be used to reconstruct* s *from the shares of its elements.*

– *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 6 (LSSS [Bei96]). *Let K be a field and $\mathsf{Att} = \{\mathsf{att}_1, \mathsf{att}_2, \ldots, \mathsf{att}_d\}$ be a set 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 $\mathbf{s} \coloneqq (s, v_2, v_3, \ldots, v_f) \cdot \mathbf{A}^\top$. 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 $\mathbb{A}$.

In this work, we consider mainly the fine-grained access control given by a *monotone access structure* $\mathbb{A}$ over a set of attributes Att, which satisfies: if $\mathsf{R}_1 \subseteq \mathsf{R}_2 \subseteq \mathsf{Att}$ and $\mathsf{R}_1 \in \mathbb{A}$, then $\mathsf{R}_2 \in \mathbb{A}$. Both access structures described by f-out-of-d threshold gates and monotone formulae are monotone.

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:

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

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 Functional Encryption with Fine-Grained Access Control

We first present the syntax of functional encryption with a fine-grained access control using policy over attributes following the works in [ACGU20, LLW21, PD21]. We consider the *key-policy* setting where policies are embedded into the functional decryption key, and attributes are embedded in ciphertexts. The function class $\mathcal{F} \coloneqq \{F_\lambda : \mathcal{D}_\lambda \to \mathcal{R}_\lambda\}_\lambda$ is a family of functions indexed by security parameters $\lambda \in \mathbb{N}$. The class of predicates $\mathcal{P} \coloneqq \{\mathsf{P} : \mathsf{Att} \to \{0, 1\}\}$ expresses the attribute-based control over the usage of functional decryption keys. 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. A plaintext consists of $(\mathsf{att}, x) \in \mathsf{Att} \times \mathcal{D}_\lambda$, whose corresponding ciphertext can be decrypted to $F_\lambda(x)$ using the functional key $\mathsf{sk}_{\mathsf{P},F_\lambda}$ iff $\mathsf{P}(\mathsf{att}) = 1$. In a straightforward manner, we extend the syntax to multi-ary predicates and thus a plaintext can contain multiple attributes. The syntax of such functional encryption schemes is given below:

Definition 7 (Functional encryption with fine-grained access control). *A* functional encryption scheme with fine-grained access control *consists of the four algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$*:*

$\mathsf{Setup}(1^\lambda)$**:** *Given as input a security parameter λ, output a pair* $(\mathsf{pk}, \mathsf{msk})$.

$\mathsf{Extract}(\mathsf{msk}, \mathsf{P}, F_\lambda)$**:** *Given a predicate* P*, a function description $F_\lambda \in \mathcal{F}$, and the master secret key* msk*, output a secret key* $\mathsf{sk}_{\mathsf{P},F_\lambda}$.

$\mathsf{Enc}(\mathsf{pk}, x, \mathsf{R})$**:** *Given as inputs the public key* pk*, a message $x \in \mathcal{D}_\lambda$, and a set* R *of attributes, output a ciphertext* ct.

$\mathsf{Dec}(\mathsf{sk}_{F_\lambda,\mathsf{P}}, \mathsf{ct})$**:** *Given the functional secret key* $\mathsf{sk}_{\mathsf{P},F_\lambda}$*, and a ciphertext* ct*, output an element in $\mathcal{R}_\lambda$ or an invalid symbol $\perp$.*

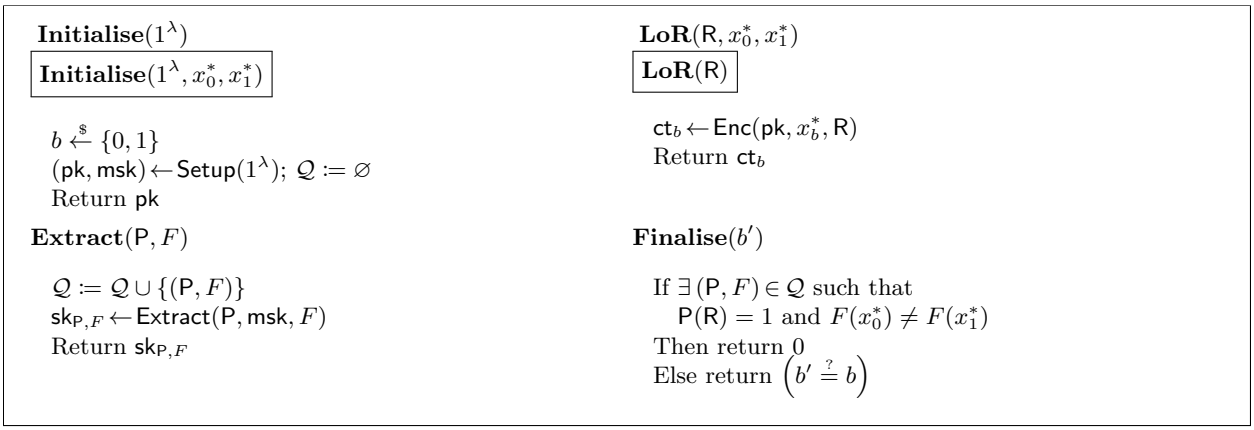

Fig. 1: The security games $\mathsf{Expr}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ and $\boxed{\mathsf{Expr}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)}$ for Definition 8

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

$$\mathsf{Dec}(\mathsf{sk}_{\mathsf{P},F_\lambda}, \mathsf{Enc}(\mathsf{pk}, x, \mathsf{R})) = F_\lambda(x) \text{ whenever } F_\lambda(x) \neq \perp^{4} \ ,$$

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

Security. Definition 8 considers 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.

Definition 8 (IND-CPA security). *An IPFE scheme with fine-grained access control* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the function class* $\mathcal{F}$ *is* secure against chosen-plaintext attacks *if for all* ppt *adversaries* $\mathcal{A}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*, the following probability is negligible*

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

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

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

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

There are other approaches to formulate the security notion, notably in [ACGU20] the authors considered a *simulation-based* notion in a selective setting. For completeness, we give the definition for this notion in Appendix A.2.

[4] See [BO13, ABN10] for discussions about this condition.

3 The Masking Lemma

We state an important lemma that will be used throughout the proofs of our constructions. The techniques were introduced in [OT10, OT12a, OT12b] and very recently used in [DGP21]. The context is that we are considering an LSSS-realizable access structure $\mathbb{A}$ and perform a random labeling of $a_0 \xleftarrow{\$} \mathbb{Z}_q$ for $\mathbb{A}$. The labels of this labeling are embedded in the vectors $(\mathbf{k}_j)_j$ and a_0 is embedded in $\mathbf{k}_{\mathsf{root}}$. We are given additionally some vectors $\{(\mathbf{c}_j)_j, \mathbf{c}_{\mathsf{root}}\}$ derived from some set R of attributes. Our goal is to mask the value a_0 in $\mathbf{k}_{\mathsf{root}}$ by introducing a non-zero value in the coordinate of hidden basis vectors, while the facing coordinate in $\mathbf{c}_{\mathsf{root}}$ is also made non-zero to mask τ. Consequently, this will mask τa_0 when performing the products in DPVS. When using in a security proof of an IPFE scheme with fine-grained access control, the vectors $\{(\mathbf{k}_j)_j, \mathbf{k}_{\mathsf{root}}\}$ constitute the functional key for the access structure $\mathbb{A}$, where $\mathbb{A}$ is contained in the key query of the adversary. At the same time, the vectors $\{(\mathbf{c}_j)_j, \mathbf{c}_{\mathsf{root}}\}$ make up the ciphertext under some challenge attributes, also defined by the adversary. Hence, after applying the lemma, the key as well as the challenge ciphertext will become readily semi-functional for later steps in the proof.

The idea of introducing a mask in auxiliary coordinates of a vector (or a dual vector) in DPVS is not new and was used to great success in [OT10, OT12a, OT12b] as well as their follow-up works for proving the notoriously hard notion of adaptive security w.r.t ABE schemes. However, the constraints that present during the masking step are largely different between our upcoming proofs and the previous setting employed in the works of Okamoto and Takashima. More specifically, in their adaptive proofs for ABE schemes, the simulation masks the key and turns it semi-functional where all keys responded to the adversary cannot decrypt the challenge ciphertext, as usually modeled in the security notion of ABE. Because the functional keys are already useless for the decrypting purpose, we have more freedom to make them semi-functional without worrying that the simulation might fail. In other words, the masks in the key can be made totally random and independent of the vectors.

On the other hand, as it is showed from our security model in Definition 8, the simulation now has to deal with both types of functional keys: those whose access structure is not satisfied, and the others that allow decrypting the challenge ciphertext. If we are working towards the goal of adaptive security, where both the challenge messages and challenge attributes are chosen adaptively, the simulator must be much more careful about what key to switch to semi-functional (after the challenge ciphertext is made semi-functional). Our lemma does not go as far as the technique by Okamoto and Takashima to introduce a totally random mask, but we rather introduce new masks $\tau x z_j$ in all $\mathbf{c}_j$ vectors and τx in the $\mathbf{c}_{\mathsf{root}}$ vector, where $x \in \mathbb{Z}_q$ is some known constant and $z_j \xleftarrow{\$} \mathbb{Z}_q$. At the same time, we also add the masks $a'_j y / z_j$ in all $\mathbf{k}_j$ vectors as well as $a'_0 y$ in $\mathbf{k}_{\mathsf{root}}$, where $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ is a new random labeling for $a'_0 \xleftarrow{\$} \mathbb{Z}_q$ and $y \in \mathbb{Z}_q$ is a known constant. This will induce a value $\tau a'_0 xy$ masking $\psi a_0 z$ when performing the product $\mathbf{c}_{\mathsf{root}} \times \mathbf{k}_{\mathsf{root}}$. In the end, if $\mathbb{A}(\mathsf{R}) = 1$, from $\mathbf{c}_j$ and $\mathbf{k}_j$ it is possible to resconstruct $\tau a'_0 xy$ and recover $\psi a_0 z$. If $\mathbb{A}(\mathsf{R}) = 0$, the fact that our labeling is derived from an LSSS helps us argue the statistical indistinguishability between $\psi a_0 z$ and a totally random value. Finally, in our main security proofs of the specific schemes of (MC)FE for inner products, we will set the constants x, y appropriately when invoking the lemma for an automatic removal of the new labeling if $\mathbb{A}(\mathsf{R}) = 1$, which then enables adaptive security.

Lemma 1. *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 $\mathsf{R} \subseteq \mathsf{Att}$ 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:*

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

and

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

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

We restate as well the weaker version, where we know in advance $\mathbb{A}(\mathsf{R}) = 0$, in Lemma 2 in Appendix B.1. The main sequence of games for proving Lemma 1, with two additional games for Lemma 2, is depicted in Figure 2. The detailed proof can be found in Appendix B.1.

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. The function class of interests is $\mathcal{F}^{\mathsf{IP}} = \{F_{\mathbf{y}}\}$ and $F_{\mathbf{y}} : \left(\mathbb{Z}_q^*\right)^n \to \mathbb{Z}_q$ is defined as $F_{\mathbf{y}}(\mathbf{x}) \coloneqq \langle \mathbf{x}, \mathbf{y} \rangle$. We consider the access control provided by LSSS-realizable monotone access structures. We are in the 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 written additively. In order to facilitate the understanding and the motivation of our construction following Definition 8, we present both selectively-secure and adaptively-secure constructions in Figure 3. 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.

4.1 Selective Security

The *correctness* is ensured by construction:

$$\begin{aligned} [\![\mathsf{out}]\!]_{\mathsf{t}} &= \sum_{j \in A} \mathbf{c}_j \times (c_j \cdot \mathbf{k}^*_j) + \sum_{i=1}^{n} (\mathbf{e}(\mathbf{t}_i, \mathbf{m}^*_i)) - \left(\mathbf{c}_{\mathsf{ipfe}} \times \mathbf{k}^*_{\mathsf{ipfe}}\right) \\ &= [\![\psi a_0 z]\!]_{\mathsf{t}} + \sum_{i=1}^{n} \left([\![(\omega \cdot (s_i + \mu u_i) + \mathbf{x}[i])\, \mathbf{y}[i]]\!]_{\mathsf{t}}\right) + [\![-\omega \cdot \langle \mathbf{s} + \mu \mathbf{u}, \mathbf{y} \rangle - \psi a_0 z]\!]_{\mathsf{t}} \\ &= [\![\psi a_0 z + \omega \langle \mathbf{s} + \mu \mathbf{u}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle]\!]_{\mathsf{t}} + [\![-\psi a_0 z - \omega \langle \mathbf{s} + \mu \mathbf{u}, \mathbf{y} \rangle]\!]_{\mathsf{t}} \\ &= [\![\langle \mathbf{x}, \mathbf{y} \rangle]\!]_{\mathsf{t}} \quad . \end{aligned}$$

We now turn our attention to the *selective security* property. The full proof can be found in Appendix B.2.

Game G_0 :

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Two additional games for Lemma 2, if we know in advance $\mathbb{A}(\mathsf{R}) = 0$:

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

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

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

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

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

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

Fig. 2: Games $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3, \mathsf{G}_4$ for the proof of Lemma 1. The index j runs over the list $\mathsf{List\text{-}Att}(\mathbb{A})$ for the $\mathbf{k}$-vectors and runs over the attributes in R 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 $\mathbb{A}(\mathsf{R}) = 0$, and thus we regain the totally random masking from the works of Okamoto-Takashima (see Lemma 2 in Appendix B.1).

$\underline{\mathsf{Setup}(1^\lambda)}$**:** Choose three 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 $(\mathbb{G}_1^4, \mathbb{G}_2^4)$, and $(\mathbf{F}, \mathbf{F}^*)$ are dual bases of $(\mathbb{G}_1^8, \mathbb{G}_2^8)$. 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 $\mu, z \xleftarrow{\$} \mathbb{Z}_q^*, \mathbf{s}, \mathbf{u} \xleftarrow{\$} (\mathbb{Z}_q^*)^n$ and write $\mathbf{s} = (s_1, \ldots, s_n)$, $\mathbf{u} = (u_1, \ldots, u_n)$. Output the public key and the master secret key as

$$\begin{cases} \mathsf{pk} \coloneqq \left(\mathbf{h}_1 + \mu \mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ (\llbracket s_i + \mu \cdot u_i \rrbracket_1)_{i\in[n]}\right) \\ \mathsf{msk} \coloneqq (z,\ \mathbf{s},\ \mathbf{u},\ (\mathbf{f}_i^*)_{i\in[3]},\ (\mathbf{h}_i^*)_{i\in[3]})\ . \end{cases}$$

$\underline{\mathsf{Extract}(\mathsf{msk}, \mathbb{A}, \mathbf{y} \in \mathbb{Z}_q^n)}$**:** 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 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 $\mathbf{c}_A = (c_j)_{j\in A}$ 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 $\mathsf{msk} = (z,\ \mathbf{s},\ \mathbf{u},\ (\mathbf{f}_i^*)_{i\in[3]},\ (\mathbf{h}_i^*)_{i\in[3]})$. Compute:

$$\mathbf{k}_j^* \coloneqq (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$

$$\boxed{\mathbf{k}_j^* \coloneqq (\pi_j \cdot (j, 1),\ a_j \cdot z, \overbrace{0, \ldots, 0}^{n \text{ times}},\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})}$$

$$\mathbf{m}_i^* \coloneqq \llbracket \mathbf{y}[i] \rrbracket_2 \text{ for } i \in [n]$$

$$\mathbf{k}_{\mathsf{ipfe}}^* \coloneqq (\langle \mathbf{s}, \mathbf{y} \rangle, \langle \mathbf{u}, \mathbf{y} \rangle,\ a_0 \cdot z,\ 0)_{\mathbf{H}^*} \qquad \boxed{\mathbf{k}_{\mathsf{ipfe}}^* \coloneqq (\langle \mathbf{s}, \mathbf{y} \rangle, \langle \mathbf{u}, \mathbf{y} \rangle,\ a_0 \cdot z, \overbrace{0, \ldots, 0}^{n \text{ times}})_{\mathbf{H}^*}}$$

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

$\underline{\mathsf{Enc}(\mathsf{pk}, \mathbf{x}, \mathsf{R})}$**:** Parse the public key $\mathsf{pk} = \left(\mathbf{h}_1 + \mu \mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ (\llbracket s_i + \mu \cdot u_i \rrbracket_1)_{i\in[n]}\right)$ and $\mathsf{R} \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ as the set of attributes, then sample $\omega, \psi \xleftarrow{\$} \mathbb{Z}_q$. Compute

$$\mathbf{c}_j = \sigma_j \cdot \mathbf{f}_1 - j \cdot \sigma_j \cdot \mathbf{f}_2 + \psi \cdot \mathbf{f}_3 = (\sigma_j \cdot (1, -j),\ \psi,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \text{ for each } j \in \mathsf{R}$$

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

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

$$\mathbf{t}_i = \omega \cdot \llbracket s_i + \mu \cdot u_i \rrbracket_1 + \llbracket \mathbf{x}[i] \rrbracket_1 = \llbracket \omega \cdot (s_i + \mu u_i) + \mathbf{x}[i] \rrbracket_1 \text{ for } i \in [n]$$

$$\mathbf{c}_{\mathsf{ipfe}} = \omega \cdot (\mathbf{h}_1 + \mu \mathbf{h}_2) + \psi \cdot \mathbf{h}_3 = (\omega,\ \mu\omega,\ \psi,\ 0)_{\mathbf{H}} \qquad \boxed{\mathbf{c}_{\mathsf{ipfe}} = (\omega,\ \mu\omega,\ \psi,\ \overbrace{0, \ldots, 0}^{n \text{ times}})_{\mathbf{H}}}$$

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

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

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

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

Fig. 3: The selectively-secure and adaptively-secure constructions for IPFE with fine-grained access control via LSSS. The proofs for selective security and adaptive security can be found in Section 4.1 and Section 4.2, respectively.

Theorem 1. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an inner-product functional encryption scheme with fine-grained access control via LSSS presented in Figure 3 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, selectively in the challenge messages and adaptively in the attributes, if the* SXDH *assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$. More precisely, let 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:*

$$\mathsf{Adv}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda) \leq (2KP \cdot (6P+3) + 2K + 9) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .$$

Proof (Main ideas). We proceed by a sequence of games, starting from the selective chosen-plaintext security game described in Figure 1. We will make use of the dual-system methodology introduced by Waters [Wat09] to prove the security for our scheme, following the similar ideas successfully employed in revocation systems and ABE schemes, e.g. those in [AL10, OT12a, OT12b, DGP21]. Intuitively, the simulator, which generates by itself all the secret information, should be able to respond to the functional secret key query $(\mathbb{A}, F)$ by the adversary. This is required to be the case even *before* the adversary declares the set of attributes for which it gets the challenge ciphertext. Our idea is to switch gradually the ciphertexts to semi-functional, then the keys corresponding to the policies that are not satisfied by the adversary's attributes to semi-functional, while keeping in mind that we cannot know those non-satisfied policies until the adversary declares the set R. Moreover, the key for a satisfied policy should be still usable to decrypt the challenge ciphertext.

We deal with this problem of two types of keys by temporarily considering the weaker notion of security where the challenge message is declared upfront, and the attribute set R is still adaptively chosen. Then, given the challenge messages at hand, the simulator will try to decide which key to be switched to semi-functional depending on the key query $(\mathbb{A}, \mathbf{y})$. As it turns out, for the declared challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$, whenever $\langle \mathbf{y}, \mathbf{x}_0^* \rangle \neq \langle \mathbf{y}, \mathbf{x}_1^* \rangle$, the functional key linked to $(\mathbb{A}, \mathbf{y})$ will be switched to semi-functional. Intuitively, in a valid attack, the aforementioned inequality implies that the policy in $\mathsf{sk}_{\mathbb{A},\mathbf{y}}$ is not satisfied by R and generally the associated functional key can be turned semi-functional without affecting its useless nature (for decrypting challenge ciphertexts under R). The remaining key queries, where the two inner-products are equal, will be kept normal. Because we are in the selective setting and can decide which key cannot decrypt the challenge ciphertext, to turn the keys semi-functional, we apply the extended version of Lemma 1 (i.e. going until G_6 in Figure 2) to introduce a totally random $r'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle$ in the ℓ-th functional key having $\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle \neq 0$. The need of $\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle$ when masking the key requires selective challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$.

The last step is to modify the master secret key $(\mathbf{s}, \mathbf{u})$ so that the challenge ciphertext is now encrypting $\mathbf{x}_0^*[i]$ and is no longer depending on b. The new $(\mathbf{s}', \mathbf{u}')$ 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 $(\mathbf{s}, \mathbf{u})$ is identical to doing so using $(\mathbf{s}', \mathbf{u}')$. 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 $(\mathbf{s}, \mathbf{u})$ introduces errors when we update $(\mathbf{s}, \mathbf{u})$ to $(\mathbf{s}', \mathbf{u}')$. 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 $(\mathbf{s}', \mathbf{u}')$. Finally, because the challenge ciphertext no longer depends on b, the advantage becomes 0 and we conclude. □

4.2 Adaptive Security

The main difference between the adaptive version and the selectively-secure version in Section 4.1 is the increase in the dimension of dual bases, from constant dimensions to dimensions linear in n. The details can be found in Figure 3. The computation for encrypting and decrypting stays essentially

the same. In the proof of security, we will explain why using bigger DPVSes allows us to achieve the stronger adaptive notion.

The *correctness* follows the correctness of the selectively-secure construction. The following theorem proves the adaptive security as defined in Definition 8. Figure 13 in Appendix B.3 describes the main ideas including the sequence of games employed in the proof. Full details can also be found in Appendix B.3.

Theorem 2. *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 3 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 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:*

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

Proof (Main ideas). We first recall the main reason why we need the challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$ to be sent in advance in the proof of Theorem 1. 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

$$\langle \mathbf{x}_0^*, \mathbf{y} \rangle \neq \langle \mathbf{x}_1^*, \mathbf{y} \rangle \ . \tag{2}$$

According to the model of security, condition (2) implies that the access structure $\mathbb{A}$ is not satisfied by the attributes R 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 $\Delta\mathbf{x} := \mathbf{x}_b^* - \mathbf{x}_0^*$, it must 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 (2) 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 1. Similar to what we have done in the selective proof, 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 $j \in \mathsf{R}$ and $z_j \xleftarrow{\$} \mathbb{Z}_q$. This explains why we need n more basis vectors over $(\mathbf{F}, \mathbf{F}^*)$ for $\tau z_j \cdot \Delta\mathbf{x}[i]$, where $i \in [n]$, 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 1 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 R. In other words, we cannot find a reconstruction vector $(c_j)_j$ from LSSS so as to recover $a'_{\ell,0}$. Thanks to the property of LSSS and the fact that we are using random labelings, $a'_{\ell,0}$ is statistically indistinguishable from a totally random value. Otherwise, if $\mathbb{A}(\mathsf{R}) = 1$, 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.

After successfully masking the keys, we use a similar argument as in the selective proof of Theorem 1 to make the challenge ciphertext not depend on b anymore. We recall that for the functional key queries where $\langle \Delta\mathbf{x}, \mathbf{y}\rangle = 0$, simulating them using $(\mathbf{s}, \mathbf{u})$ stays identical when we update $(\mathbf{s}, \mathbf{u})$ to $(\mathbf{s}', \mathbf{u}')$ in the challenge ciphertext. Otherwise, under the SXDH assumption, we can correct the keys where $\langle \Delta\mathbf{x}, \mathbf{y}\rangle \neq 0$ to the correct form w.r.t $(\mathbf{s}', \mathbf{u}')$ using the randomness $r'_{\ell,0}$ introduced from previous steps. □

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

In this section, we 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, while treating the tags separately from the predicates. Furthermore, the security of the transformation by Abdalla *et al.* [ACGU20, Theorem 6.3] necessitates a multiplicative degradation of factor n in the security reduction for multi-input constructions. The security of our MCFE construction, on the other hand, does not depend on the number of clients n, see Theorem 3. Finally, in Section 5.4 we discuss further our construction and revisit the MIFE regime for comparison with [ACGU20].

First of all, we define and give the model of security for *multi-client functional encryption for inner-products with fine-grained access control* in Section 5.1. The adaptive proof is given in Theorem 3. The function class of interests is $\mathcal{F}^{\mathsf{IP}} = \{F_{\mathbf{y}}\}$ and $F_{\mathbf{y}} : \left(\mathbb{Z}_q^*\right)^n \to \mathbb{Z}_q$ is defined as $F_{\mathbf{y}}(\mathbf{x}) := \langle \mathbf{x}, \mathbf{y}\rangle$. We consider the access control provided by LSSS-realizable monotone access structures. We are in the 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 written additively. The Tag space contains the tags, which are also referred to as "labels" interchangeably in Section 1, that accompany plaintext components at the time of encryption.

5.1 Definitions

We extend the notion of functional encryption with fine-grained access control to the multi-client setting. The syntax is given below.

Definition 9 (Multi-client functional encryption with fine-grained access control). *A* multi-client functional encryption (MCFE) scheme with fine-grained access control *consists of the 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.*

$\mathsf{Extract}(\mathsf{msk}, \mathsf{P}, F_\lambda)$**:** *Given a predicate* P*, a function description* $F_\lambda \in \mathcal{F}$*, and the master secret key* msk*, output a decryption key* $\mathsf{dk}_{\mathsf{P},F_\lambda}$*.*

$\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{R})$**:** *Given as inputs an encryption key* ek_i*, a message* $x_i \in \mathcal{D}_\lambda$*, a tag* tag*, and a set* R *of attributes, output a ciphertext* $\mathsf{ct}_{\mathsf{tag},i}$*.*

$\mathsf{Dec}(\mathsf{dk}_{\mathsf{P},F_\lambda}, \mathsf{tag}, \mathbf{c})$**:** *Given the decryption key* $\mathsf{dk}_{\mathsf{P},F_\lambda}$*, a tag* tag*, 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 $(\mathsf{msk}, (\mathsf{ek}_i)_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(F_\lambda, \mathsf{P}) \in \mathcal{F} \times \mathcal{P}$ and $\mathsf{dk}_{\mathsf{P},F_\lambda} \leftarrow \mathsf{Extract}(\mathsf{P}, \mathsf{msk}, F_\lambda)$, for all tag and R satisfying $\mathsf{P}(\mathsf{R}) = 1$, for all $(x_i)_{i\in[n]} \in \mathcal{D}_\lambda^n$, the following holds with overwhelming probability:

$$\mathsf{Dec}\left(\mathsf{dk}_{\mathsf{P},F_\lambda}, \mathsf{tag}, (\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{R}))_{i\in[n]}\right) = F_\lambda(x_1, \ldots, x_n) \text{ if } F_\lambda(x_1, \ldots, x_n) \neq \perp$$

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 key's policy, which also complicates the way we model the security notion. In the plain setting without attribute-based control, interested readers can refer to [CDG+18a] or [LT19] for more details.

Definition 10 (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. In the security game given in Figure 4 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:*

- *There exists $i \in \mathcal{C}$ such that $(i, x_i^{(0)}, x_i^{(1)}, \mathsf{tag}, \mathsf{R})$ is queried to* **LoR** *and $x_i^{(0)} \neq x_i^{(1)}$.*
- *There exist a tag tag and $i, j \in \mathcal{H}$ such that $i \neq j$, there exists a query $(i, x_i^{(0)}, x_i^{(1)}, \mathsf{tag}, \mathsf{R})$ to* **LoR** *but there exist no query $(j, x_j^{(0)}, x_j^{(1)}, \mathsf{tag}, \mathsf{R})$ to* **LoR**.
- *There exist a tag tag, a set R of attributes, a function $F \in \mathcal{F}$, a predicate $\mathsf{P} \in \mathcal{P}$, and there exist two input vectors $(x_1^{(0)}, \ldots, x_n^{(0)})$ and $(x_1^{(1)}, \ldots, x_n^{(1)})$ such that*
  - *The attributes in R satisfy the predicate P and $(\mathsf{P}, F) \in \mathcal{Q}$.*
  - *$F\left(x_1^{(0)}, \ldots, x_n^{(0)}\right) \neq F\left(x_1^{(1)}, \ldots, x_n^{(1)}\right)$.*
  - *For all $i \in \mathcal{H}$, there exist ciphertexts w.r.t $(\mathsf{tag}, \mathsf{R})$ for $(x_i^{(0)}, x_i^{(1)})$ received from* **LoR**.
  - *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 1. 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 $(\mathsf{tag}, \mathsf{R}_i)$. 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. In Section 5.2, our concrete constructions enforce the same tag and same attributes in the ciphertext components by hashing them during encryption. In Section 5.4, we discuss how to relax the constraint and achieve the flexible notion where each client i can use a different $(\mathsf{tag}, \mathsf{R}_i)$ 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.

Remark 2. 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 $(i, \mathsf{tag}, \mathsf{R})$. 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. Finally, the oracle **LoR** might be queried for only honest clients i, because the adversary can already use ek_i of any corrupted client i for encryption. In addition, for every honest clients i, there must be a ciphertext query to **LoR** under the challenge $(\mathsf{tag}, \mathsf{R})$. 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 slot $i \in [n]$ are queried to **LoR**.

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

Definition 11 (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 function class* $\mathcal{F} = \{F_\lambda\}_{\lambda \in \mathbb{N}}$ *is* IND-secure *if for all* ppt *adversaries* $\mathcal{A}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*, the following probability is negligible*

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

The game $\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ *is depicted in Figure 4. 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 secure against chosen-plaintext attacks *if the following probability is negligible*

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

5.2 Construction

This section presents a multi-client FE scheme with fine-grained access control, as defined in Section 5.1, for the function class $\mathcal{F}^{\mathsf{IP}}$ of inner-products. The access control is expressed via LSSS-realizable monotone access structures. 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 5. 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 3 we detail how the basis changes in Lemma 1 can be done in parallel for $(\mathbf{H}_i, \mathbf{H}_i^*), (\mathbf{F}, \mathbf{F}^*)$ for all $i \in [n]$.

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

Enc$(i, x_i, \mathsf{tag}, \mathsf{R})$

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

Finalise(b')

If $\mathcal{A}$ is NOT admissible w.r.t $(\mathcal{C}, \mathcal{Q}, \mathcal{H})$
or there exists $i \in \mathcal{C}$ among the queries to **LoR**:
  return 0
Else return $\left(b' \stackrel{?}{=} b\right)$

LoR$(i, x_i^{(0)}, x_i^{(1)}, \mathsf{tag}^*, \mathsf{R}^*)$ | **LoR**$(i, \mathsf{tag}^*, \mathsf{R}^*)$

If $(i, \mathsf{tag}^*, \mathsf{R}^*)$ appears previously
or a different $(i, \mathsf{tag}', \mathsf{R}')$ was queried:
  Ignore
Else: $\mathsf{ct}^{(b)}_{\mathsf{tag}^*,i} \leftarrow \mathsf{Enc}(\mathsf{ek}_i, x_i^{(b)}, \mathsf{tag}^*, \mathsf{R}^*)$
Return $\mathsf{ct}^{(b)}_{\mathsf{tag}^*,i}$

Corrupt(i)

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

Extract(P, F)

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

Fig. 4: The security game $\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ and $\boxed{\mathsf{Expr}^{\mathsf{mc\text{-}sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)}$ for Definition 11

The *correctness* of the scheme is verified by:

$$\begin{aligned}
[\![\mathsf{out}]\!]_\mathsf{t} &= \sum_{i=1}^{n} \left(\left(\sum_{j\in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}) \right) - (\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}}) + \mathbf{e}(\mathbf{t}_i, \mathbf{m}_i) \right) \\
&= \sum_{i=1}^{n} \left([\![\psi_i a_{i,0} z]\!]_\mathsf{t} - [\![\omega p_i \cdot \langle \mathbf{s}, \mathbf{y}\rangle + \omega' p_i \cdot \langle \mathbf{u}, \mathbf{y}\rangle + \psi_i a_{i,0} z]\!]_\mathsf{t} + [\![(\omega s_i + \omega' u_i + x_i) y_i]\!]_\mathsf{t} \right) \\
&= [\![\langle \mathbf{x}, \mathbf{y}\rangle]\!]_\mathsf{t} \quad .
\end{aligned}$$

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 Appendix B.4. The sequence of games can be found in Figures 6 and 7.

Theorem 3. *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 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 secure against chosen-plaintext attacks 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:*

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

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 $\mathbb{G}_1$.

$\underline{\mathsf{Setup}(1^\lambda)}$**:** 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 $(F, F'), (H_i, H_i')$ respectively (see Appendix A.1 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 $\mu \xleftarrow{\$} \mathbb{Z}_q^*, \mathbf{s}, \mathbf{u} \xleftarrow{\$} (\mathbb{Z}_q^*)^n$ and write $\mathbf{s} = (s_1, \ldots, s_n)$, $\mathbf{u} = (u_1, \ldots, u_n)$. 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

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

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

$\underline{\mathsf{Extract}(\mathsf{msk}, \mathbb{A}, \mathbf{y} \in \mathbb{Z}_q^n)}$**:** 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 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 $\mathsf{msk} = (\mathbf{s},\ \mathbf{u},\ \mathbf{f}_1^*,\ \mathbf{f}_2^*,\ \mathbf{f}_3^*,\ (\mathbf{h}_{i,1}^*, \mathbf{h}_{i,2}^*, \mathbf{h}_{i,3}^*)_{i\in[n]})$ and write $\mathbf{y} = (y_1, \ldots, y_n)$. For each $i \in [n]$, compute $\mathbf{m}_i \coloneqq [\![y_i]\!]_2$ and

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

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

$\underline{\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{R})}$**:** 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 $\mathsf{R} \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ as the set of attributes, compute $\mathsf{H}(\mathsf{tag}, \mathsf{R}) \to ([\![\omega]\!]_1, [\![\omega']\!]_1) \in \mathbb{G}_1^2$ 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 $j \in \mathsf{R}$, 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 $\mathsf{ct}_{\mathsf{tag},i} \coloneqq \left((\mathbf{c}_{i,j})_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}}\right)$.

$\underline{\mathsf{Dec}(\mathsf{dk}_{\mathbb{A},\mathbf{y}}, \mathsf{tag}, \mathbf{c} \coloneqq (\mathsf{ct}_{\mathsf{tag},i}))}$**:** Parse $\mathsf{ct}_{\mathsf{tag},i} = ((\mathbf{c}_{i,j})_{j\in\mathsf{R}}, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}})$ 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 $A \subseteq \mathsf{R}$ 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. 5: The construction for multi-client IPFE with fine-grained access control via LSSS from Section 5.2.

Proof (Main ideas). Recall that in the selective single-client proof of Theorem 1, we switch the ℓ-th functional key to semi-functional if $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \neq 0$. On the other hand, in the single-client adaptive proof of Theorem 2, to get rid of the constant $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle$ in the semi-functional key, we augment 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 $\tau, r_0^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$. 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.

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 ciphertext returned from **LoR** of client i and only $r_{i,0}^{(\ell)}\mathbf{y}^{(\ell)}[i]$ in the corresponding key component. 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. There are some further technical tweaks to be done. 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 **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 1 as we are in the adaptive setting and not knowing what policy will the ciphertext's attributes 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 R, 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 R for all i is guaranteed by hashing $(\mathsf{tag}, \mathsf{R})$ together. The more complicated and flexible case with possibly different R_i for each i is discussed in Section 5.4. The application of Lemma 1 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 **LoR** is encryption $\mathbf{x}_0^*$ instead of $\mathbf{x}_b^*$. However, the situation is more complicated than the single-client construction because the oracle **Enc** is using (s_i, u_i) as well. Therefore, in order to be able to perform the correction step on the functional key, we have to program the full-domain hash function such that for all queries $(\mathsf{tag}', \mathsf{R}')$ different from the challenge $(\mathsf{tag}, \mathsf{R})$, the value $\mathsf{H}(\mathsf{tag}', \mathsf{R}')$ belongs to $\mathrm{span}([\![(1,\ \mu)]\!]_1) \subseteq \mathbb{G}_1^2$, for $\mu \xleftarrow{\$} \mathbb{Z}_q$. For the challenge $(\mathsf{tag}, \mathsf{R})$, the value $\mathsf{H}(\mathsf{tag}, \mathsf{R})$ remains a pair of random group elements. The main reason behind this is that our correction step requires $\mathsf{H}(\mathsf{tag}', \mathsf{R}')$ belongs to $\mathrm{span}([\![(1,\ \mu)]\!]_1)$ so that it will not affect the normal ciphertext returned from **Enc**. This implies a linear relation between $\Delta\mathbf{s} \coloneqq \mathbf{s}' - \mathbf{s}$ and $\Delta\mathbf{u} \coloneqq \mathbf{u}' - \mathbf{u}$. However, if we put $\mathsf{H}(\mathsf{tag}, \mathsf{R})$ 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 **LoR** will create another linear relation, which reduces significantly the degree of

freedom to choose $(\Delta\mathbf{s}, \Delta\mathbf{u})$ 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 $(\mathsf{tag}, \mathsf{R})$ when encrypting for all client $i \in [n]$, by hashing them using the full-domain hash function. In practice, this could render a significant cost for synchronisation among clients so as to agree on the tag *and* the attributes at the time of encryption. In addition, by fixing one public tag, one can only obtain an MIFE scheme on the ROM because we still need the random oracle to process R.

If we allow different $(\mathsf{tag}, \mathsf{R}_i)$ 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 $i_1 \neq i_2$ encrypting $\mathbf{x} = (x_1, x_2)$ under different sets $(\mathsf{R}_1, \mathsf{R}_2)$ of attributes, the ℓ-th and ℓ'-th key queries have access structures $\mathbb{A}$ and $\mathbb{A}'$ where $\mathbb{A}(\mathsf{R}_1) = \mathbb{A}'(\mathsf{R}_2) = 1$ and $\mathbb{A}'(\mathsf{R}_1) = \mathbb{A}(\mathsf{R}_2) = 0$, 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, an adversary can 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 $p_1\omega_{\mathsf{tag}}\langle \mathbf{s}, \mathbf{y}\rangle + p_1\omega'_{\mathsf{tag}}\langle \mathbf{u}, \mathbf{y}\rangle$. Similar computation allows the same adversary to obtain $p_2\omega_{\mathsf{tag}}\langle \mathbf{s}, \mathbf{y}\rangle + p_2\omega'_{\mathsf{tag}}\langle \mathbf{u}, \mathbf{y}\rangle$ 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 R 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 $\ell \neq \ell'$ associated by 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 R. 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 $\theta\mathbf{h}_{i,5}$ for some $\theta \xleftarrow{\$} \mathbb{Z}_q$. 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 $\mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag}, \mathsf{R}_i)$ 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 $(d_{\mathbb{A},i})_i$ is a random n-out-of-n secret sharing of 0, independent among functional keys. The vectors are essentially the same as in Figure 5, except $\mathbf{c}_{i,\mathsf{ipfe}}, \mathbf{k}_{i,\mathsf{ipfe}}$ for each i as follows:

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

The decryption calculation stays invariant because $\sum_{i=1}^n d_{\mathbb{A},i} = 0$. 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 $d_{\mathbb{A},1} + d_{\mathbb{A}',2} = 0$ only with negligible probability. More formally, the security proof for this modified scheme can be obtained with recourse to the proof of Theorem 3. We sketch the proof and highlight the main differences compared to the less flexible scheme in Appendix B.5.

Remark 3. 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

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

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

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

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

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

Game G_1 : $\mathsf{H}(\mathsf{tag},\mathsf{R}) \to ([\![\omega_{\mathsf{tag},\mathsf{R}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{R}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') \to ([\![\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{R}'}]\!]_1)$, $a^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a^{(\ell)}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a^{(\ell)}_{i,0}}(\mathbb{A})$

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

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

Game G_2 : $\mathsf{H}(\mathsf{tag},\mathsf{R}) \to ([\![\omega_{\mathsf{tag},\mathsf{R}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{R}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') \to ([\![\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1, [\![\chi'_{\mathsf{tag}',\mathsf{R}'}]\!]_1)$, $a'^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$

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

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

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

Fig. 6: Games $\mathsf{G}_0, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3$ for Theorem 3. The transition from G_1 to G_2 is given in Figure 15 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 R 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_3 we use a random function $\mathsf{RF} : \mathsf{Tag} \times 2^{\mathsf{Att}} \to (\mathbb{Z}_q^*)^2$.

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

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

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

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

Game G_5 : $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag},\mathsf{R}) := ([\![\omega_{\mathsf{tag},\mathsf{R}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{R}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') := ([\![\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1)$, $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_q^4$

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

Game G_6 : $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_q^4$, $\mu, v_{i,0}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $\mathsf{H}(\mathsf{tag},\mathsf{R}) := ([\![\omega_{\mathsf{tag},\mathsf{R}}]\!]_1, [\![\omega'_{\mathsf{tag},\mathsf{R}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') := ([\![\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1)$. We also define $\mathbf{s}' = \mathbf{s} + \Delta\mathbf{s}, \mathbf{u}' = \mathbf{u} + \Delta\mathbf{u}$, where $\Delta\mathbf{s}, \Delta\mathbf{u} \in \mathbb{Z}_q^n$ s.t. $\Delta\mathbf{s} + \mu\Delta\mathbf{u} = 0$ and $\omega_{\mathsf{tag},\mathsf{R}} \cdot \Delta\mathbf{s} + \omega'_{\mathsf{tag},\mathsf{R}} \cdot \Delta\mathbf{u} = \Delta\mathbf{x}$

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

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

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

Fig. 7: Games $\mathsf{G}_4, \mathsf{G}_5, \mathsf{G}_6$ for Theorem 3. 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 R for ciphertext components. The index ℓ runs in $\{1, \ldots, K\}$ for the functional key queries. In G_4 we use a random function $\mathsf{RF}' : \mathsf{Tag} \times 2^{\mathsf{Att}} \to \mathbb{Z}_q^*$.

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.

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) 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.

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.

AL10. Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, Heidelberg, May 2010.

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.

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

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.

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.

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`.

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.

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 Patrick Longa and Carla Ràfols, editors, *LATIN 2021*, LNCS, pages 127–148. Springer, Heidelberg, October 2021.

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

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.

A Additional Definitions

A.1 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 $(\mathbf{U} = (\mathbf{u}_i)_i, \mathbf{U}^* = (\mathbf{u}_i^*)_i)$ 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 $\mathbf{B} \coloneqq B \cdot \mathbf{U}$, which means:

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

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.2 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.

$\mathsf{Real}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$:

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathsf{R} \leftarrow \mathcal{A}^{\mathsf{Extract}(\mathsf{msk},\cdot,\cdot)}(1^\lambda, \mathsf{pk})$
$\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, x^*, \mathsf{R})$
$b \leftarrow \mathcal{A}^{\mathsf{Extract}(\mathsf{msk},\cdot,\cdot)}(\mathsf{pk}, \mathsf{ct}^*)$
Return b

$\mathsf{Sim}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$:

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Sim.Setup}(1^\lambda)$
$\mathsf{R} \leftarrow \mathcal{A}^{\mathsf{Sim.Extract}(\mathsf{msk},\cdot,\cdot)}(1^\lambda, \mathsf{pk})$
$\mathsf{ct}^* \leftarrow \mathsf{Sim.Enc}(\mathsf{pk}, x^*, \mathsf{R})$
$b \leftarrow \mathcal{A}^{\mathsf{Sim.Extract}(\mathsf{msk},\cdot,\cdot)}(\mathsf{pk}, \mathsf{ct}^*)$
Return b

Fig. 8: The security games $\mathsf{Real}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ and $\mathsf{Sim}^{\mathsf{sel\text{-}sim}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ for Definition 12

Definition 12 (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* $\mathcal{F}$ *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:*

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

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 8. The probability is taken over the random coins of* $\mathcal{A}$ *and the algorithms.*

B Proofs of the Main Body

B.1 Proof of Lemma 2

Lemma 2. *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* $\mathsf{R} \subseteq \mathsf{Att}$ *such that* $\mathbb{A}(\mathsf{R}) = 0$*, i.e.* R *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* $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ *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:*

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

and

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

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

Remark 4. The proof of Lemma 1 is a direct subsequence of the games we use to prove Lemma 2, i.e. from G_0 to G_4 in Figure 2, 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 $\mathbb{A}(\mathsf{R}) = 0$, which is used only for going from G_4 to G_5 and from G_5 to G_6 in Figure 2.

Proof (Of Lemma 2). 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 2.

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:

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

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

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

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

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

and $\psi = b \cdot x$. 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 $\mathbf{c}_j$:

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

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:

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

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 $j \in \mathsf{R}$, 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:

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

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 ciphertext components in $\mathbf{T}$ and $\mathbf{W}$. More precisely, the simulator can virtually set:

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

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 induced by $\mathbb{A}$. The simulator can virtually set:

$$
\begin{aligned}
\mathbf{k}^*_{\mathsf{root}} &= (a_0 z,\ 0)_{\mathbf{H}^*} + (b \cdot y,\ c \cdot y)_{\mathbf{T}^*} \\
&= (a_{\ell,0} z + b \cdot y,\ \rho \cdot y)_{\mathbf{H}^*} \\
\mathbf{k}^*_j &= (\pi_j \cdot (j, 1),\ a_j \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \\
&\quad + (0,\ 0,\ bd'_j \cdot y,\ cd'_j \cdot y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{W}^*} \\
&= (\pi_j \cdot (j, 1),\ a_{\ell,j} \cdot z + b \cdot y \cdot d'_j,\ \rho \cdot d'_j \cdot y,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathcal{J}_{\mathsf{policy}}\ .
\end{aligned}
$$

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:

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

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 $|\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}(\mathsf{G}_2)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

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 $1/z_j$:

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

This transition is discussed separately in Lemma 3. In the end, the difference in advantages is

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

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

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

We first observe that the attributes in $(\mathbf{c}_j)_{j\in\mathsf{R}}$ 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 $A \subseteq \mathsf{R}$ 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 $\mathbb{A}$, 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 unbounnded 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 $\mathbf{c}_j, \mathbf{k}_j$:

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

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 $\mathsf{P}(\mathsf{R}) = 0$ 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

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

and the proof is concluded. □

Lemma 3. *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 2 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 $\mathsf{List\text{-}Att}(\mathbb{A})$, leading to $\mathsf{G}_{3.0} = \mathsf{G}_3$ and $\mathsf{G}_{3.P} = \mathsf{G}_4$. The current form of other vectors is:

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

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. 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

$$\begin{aligned}
a'_0 &= \rho y \cdot a''_0; & a_0 &= \tilde{a}_0 + \delta \cdot a''_0 \\
a'_j &= \rho y \cdot a''_j; & a_j &= \tilde{a}_j + \delta \cdot a''_j \\
\pi_j &= \Pi_j + a''_j \cdot \tilde{\pi}_j\ . & &
\end{aligned} \tag{3}$$

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 (3), 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 9.

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

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

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. 9: Games for Lemma 3. 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 (3) 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 10.

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:

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

and the simulator can set

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

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 $\mathsf{G}_{3.m-1.2}$: 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:

$$F \coloneqq \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ -a & 0 & 1 \end{bmatrix}_{2,4,5} \qquad F' \coloneqq \left(F^{-1}\right)^{\top} = \begin{bmatrix} 1 & -a & a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{2,4,5}$$
$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^*$$

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

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

On the other hand, the simulator can set the keys as below: if $i = m$

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

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 $\tau x z_j / z_m$:

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

We apply Lemma 4 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 $(1, 2, 4, 7, 8)$ 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 $\mathbf{k}^*_{\ell,m}$ are taken into account and affected by the basis changes. For other vectors, the concerning coordinates can be written directly in the target bases because they are all 0. We proceed by a sequence of games depicted in Figure 10. 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 at the **Setup** phase, where a ppt simulator can first sample a value $z \xleftarrow{\$} \mathbb{Z}_q$ and use z in the basis change. Afterwards, when all attributes are declared (in an adaptive functional key query), z would be the mask at the attribute m corresponding to the current hybrid.
We have

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

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

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

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

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

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

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

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

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

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

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

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

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

Fig. 10: The hybrids to go from $\mathsf{G}_{3.m-1.2}$ to $\mathsf{G}_{3.m-1.3}$, by applying Lemma 4. 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 (3) for the rewriting of $\mathbf{k}^*_j$ into $\mathbf{h}^*_j$.

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 $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ 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 $\mathsf{G}_{3.K} = \mathsf{G}_4$ 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 4. *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:*

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

and

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

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 $\rho \coloneqq c - ab$ is either 0 or uniformly random, we use the following basis changing matrices (F, F'):

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

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

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 $\tau \coloneqq c - ab$ is either 0 or uniformly random, we use the following basis changing matrices (F, F'):

$$F \coloneqq \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{1,2,4,5} \qquad F' \coloneqq \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & 1 & 0 \\ 0 & -a & 0 & 1 \end{bmatrix}_{1,2,4,5}$$
$$\mathbf{F} = F \cdot \mathbf{W}; \qquad \mathbf{F}^* = F' \cdot \mathbf{W}^*$$

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):

$$\begin{cases} \tau(z_4 + jz_3) = \mu_1 \\ -\tau(z_2 + jz_1) = \mu_2 \\ \rho(mz_1 + z_2) = \theta_1 \\ \rho(mz_3 + z_4) = \theta_2 \end{cases} \Leftrightarrow \begin{cases} z_4 + jz_3 = \mu_1/\tau \\ mz_3 + z_4 = \theta_2/\rho \\ z_2 + jz_1 = -\mu_2/\tau \\ mz_1 + z_2 = \theta_1/\rho \end{cases}$$
$$\Leftrightarrow \begin{cases} (j-m)z_3 = \mu_1/\tau - \theta_2/\rho \\ mz_3 + z_4 = \theta_2/\rho \\ (j-m)z_1 = -\mu_2/\tau - \theta_1/\rho \\ mz_1 + z_2 = \theta_1/\rho \end{cases} .$$

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:

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

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 $\rho \coloneqq c - ab$ is either γ or the constant α, we use the following basis changing matrices (F, F'):

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

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:

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

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:

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

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.2 Proof of Theorem 1

Proof (Of Theorem 1). The security games and their transitions are given in Figure 11. The transition from G_4 to G_5 is given in Figure 12.

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. We let K denote the number of functional key queries $(\mathsf{P}, \mathbf{y})$ and index the functional key by $\ell \in \{1, \ldots, K\}$.

Game G_0: This is the selective security game as given in Figure 1. The adversary first declares its challenge messages $\mathbf{x}_0^*$ and $\mathbf{x}_1^*$. The simulator generates all private information, including the three dual basis pairs

$$\begin{aligned}\mathbf{H} &= (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4) & \mathbf{H}^* &= (\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*, \mathbf{h}_4^*) \\ \mathbf{F} &= (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_6, \mathbf{f}_7, \mathbf{f}_8) & \mathbf{F}^* &= (\mathbf{f}_1^*, \mathbf{f}_2^*, \mathbf{f}_3^*, \mathbf{f}_4^*, \mathbf{f}_5^*, \mathbf{f}_6^*, \mathbf{f}_7^*, \mathbf{f}_8^*)\end{aligned}$$

as well as $\mu, z \xleftarrow{\$} \mathbb{Z}_q^*, \mathbf{s}, \mathbf{u} \xleftarrow{\$} (\mathbb{Z}_q^*)^n$. It sets

$$\mathsf{msk} := (z,\ \mathbf{s},\ \mathbf{u},\ (\mathbf{f}_i^*)_{i\in[3]},\ (\mathbf{h}_i^*)_{i\in[3]})$$

and sends $\mathsf{pk} = \big((\mathbf{h}_1 + \mu\mathbf{h}_2),\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ (\llbracket s_i + \mu \cdot u_i \rrbracket_1)_{i\in[n]}\big)$ to the adversary.

Extract$(\mathbb{A}, \mathbf{y})$: For the ℓ-th functional key query w.r.t an LSSS-realizable monotone access structure $\mathbb{A}$ and a vector $\mathbf{y}_\ell \in \mathbb{Z}_q^n$ that denotes the inner product function $F_{\mathbf{y}}$, the simulator samples $a_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, run the labeling algorithm $\Lambda_{a_{\ell,0}}(\mathbb{A})$ (see Definition 1) to obtain the labels $(a_{\ell,j})_{\ell,j}$ where j runs over the attributes in Att, possibly with repetitions. It then returns

$$\begin{aligned}\mathbf{k}_{\ell,j}^* &:= (\pi_{\ell,j} \cdot (j, 1),\ a_{\ell,j} \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathcal{J}_{\mathsf{policy}} \\ \mathbf{m}_{\ell,i}^* &:= \llbracket \mathbf{y}_\ell[i] \rrbracket_2 \text{ for } i \in [n] \\ \mathbf{k}_{\ell,\mathsf{ipfe}}^* &:= (\langle \mathbf{s}, \mathbf{y}_\ell \rangle,\ \langle \mathbf{u}, \mathbf{y}_\ell \rangle,\ a_{\ell,0} \cdot z,\ 0)_{\mathbf{H}^*}\end{aligned}$$

where $\pi_{\ell,j} \xleftarrow{\$} \mathbb{Z}_q$.

LoR(R) : Upon receiving a set R of attributes, the simulator samples $\omega, \psi \xleftarrow{\$} \mathbb{Z}_q$, flips a coin $b \xleftarrow{\$} \{0, 1\}$, and computes

$$\begin{aligned}\mathbf{t}_i &:= \llbracket \omega \cdot (s_i + \mu u_i) + \mathbf{x}_b^*[i] \rrbracket_1 \text{ for } i \in [n] \\ \mathbf{c}_{\mathsf{ipfe}} &:= (\omega,\ \mu\omega,\ \psi,\ 0)_{\mathbf{H}}\end{aligned}$$

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

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

and $\sigma_i \xleftarrow{\$} \mathbb{Z}_q$ for every $i \in [n]$.

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

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

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

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

Game G_1 : $\tau \xleftarrow{\$} \mathbb{Z}_q$, $\mathsf{pk} := \big(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\big)$

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

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

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

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

Game G_3 : $r'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x} := \mathbf{x}^*_b - \mathbf{x}^*_0$, $\mathsf{pk} := \big(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\big)$

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

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

Game G_4 : $\omega', r''_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $\mathsf{pk} := \big(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\big)$

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

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

Game G_5 : $\omega', r''_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, $\mathbf{s}' = \mathbf{s}+\Delta\mathbf{s}, \mathbf{u}' = \mathbf{u}+\Delta\mathbf{u}$, 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 $\Delta\mathbf{s}+\mu\cdot\Delta\mathbf{u} = 0$, $\mathsf{pk} = \big(\mathbf{h}_1 + \mu\mathbf{h}_2,\ \mathbf{h}_3,\ (\mathbf{f}_i)_{i\in[3]},\ ([\![s_i + \mu\cdot u_i]\!]_1)_i\big)$

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

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

Fig. 11: Games for Theorem 1. The index i runs in $\{1,\ldots,n\}$. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in R for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries. The hybrids for G_4 to G_5 are given in Figure 12 in Appendix B.2.

Game G_4 :

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

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

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

Game $\mathsf{G}_{4.1}$: $r''_{\ell,0}, \omega' \xleftarrow{\$} \mathbb{Z}_q$, $\mathbf{s}' = \mathbf{s} + \Delta\mathbf{s}, \mathbf{u}' = \mathbf{u} + \Delta\mathbf{u}$, 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 $\Delta\mathbf{s} + \mu \cdot \Delta\mathbf{u} = 0$

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

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

$$\begin{array}{lcc|c|c|c|l} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \boxed{\mu\omega} & \psi & \tau &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \boxed{\langle \mathbf{s}', \mathbf{y}_\ell \rangle} & \boxed{\langle \mathbf{u}', \mathbf{y}_\ell \rangle} & a_{\ell,0} z & \boxed{r''_{\ell,0}} \langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle &)_{\mathbf{H}^*} \end{array}$$

Game $\mathsf{G}_{4.2} = \mathsf{G}_5$:

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

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

$$\begin{array}{lcc|c|c|c|l} \mathbf{c}_{\mathsf{ipfe}} & (& \omega & \boxed{\omega'} & \psi & \tau &)_{\mathbf{H}} \\ \mathbf{k}^*_{\ell,\mathsf{ipfe}} & (& \langle \mathbf{s}', \mathbf{y}_\ell \rangle & \langle \mathbf{u}', \mathbf{y}_\ell \rangle & a_{\ell,0} z & r''_{\ell,0} \langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle &)_{\mathbf{H}^*} \end{array}$$

Fig. 12: Games $\mathsf{G}_{4.1}, \mathsf{G}_{4.2}$ for the transition G_4 to G_5 in the proof of Theorem 1. We are in the case $\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle \neq 0$. The changes are made for the ℓ-th functional key query. The index i runs in $\{1, \ldots, n\}$. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in R for ciphertext components.

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

Game G_1: In this game, we change the normal ciphertexts to semi-functional ciphertexts. In particular, the challenge ciphertext will change in the following components:

$$\begin{aligned} \mathbf{c}_{\mathsf{ipfe}} &\coloneqq (\omega,\ \mu\omega,\ \psi,\ \boxed{\tau}\,)_{\mathbf{H}} \\ \mathbf{c}_j &\coloneqq (\sigma_j \cdot (1,-j),\ \psi,\ \boxed{\tau},\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \text{ for } j \in \mathsf{R} \end{aligned}$$

for a uniformly random value $\tau \xleftarrow{\$} \mathbb{Z}_q$.

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:

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

Note that we can compute all the basis vectors except $\mathbf{h}_2^*$ and $\mathbf{f}_5^*$ but currently they are not needed because their coordinates are 0 in all the keys. The simulator can virtually set

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

and $\psi = b$. 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 another mask in the ciphertext, namely:

$$\mathbf{c}_j \coloneqq (\sigma_j \cdot (1, -j),\ \psi,\ \tau,\ 0,\ \tau z_j,\ 0,\ 0)_{\mathbf{F}} \text{ for } j \in \mathsf{R}$$

for uniformly random values $z_j \xleftarrow{\$} \mathbb{Z}_q$.

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:

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

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 $j \in \mathsf{R}$, 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

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

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, if the ℓ-th key query $(\mathbb{A}, \mathbf{y}_\ell)$ satisfies that $\langle \mathbf{y}_\ell, \mathbf{x}^*_0 \rangle \neq \langle \mathbf{y}_\ell, \mathbf{x}^*_1 \rangle$ we switch this ℓ-th functional secret key to semi-functional. In the spirit of the dual system method, they will not be useful in the decrypting the (already) semi-functional challenge ciphertexts. On the contrary, if $\langle \mathbf{y}_\ell, \mathbf{x}^*_0 \rangle = \langle \mathbf{y}_\ell, \mathbf{x}^*_1 \rangle$ we respond the ℓ-th key query with a normal functional key. Indeed, the above condition helps us preserve the functionality of such keys, because the set of all queried access structure $\mathbb{A}$ whose $\mathbf{y}_\ell$ satisfies $\langle \mathbf{y}_\ell, \mathbf{x}^*_0 \rangle = \langle \mathbf{y}_\ell, \mathbf{x}^*_1 \rangle$ will contain all identities whose policy is satisfied by the attributes of the challenge ciphertext. When given the ℓ-th key query $(\mathbb{A}, \mathbf{y}_\ell)$ the simulator samples $r'_{\ell,0} \xleftarrow{\$} \mathbb{Z}_q$, and the functional key will change in the following components:

$$\begin{aligned}
\mathbf{k}^*_{\ell,j} &= (\pi_{\ell,j} \cdot (j, 1),\ a_{\ell,j} \cdot z,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A}) \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &= (\langle \mathbf{s}, \mathbf{y}_\ell \rangle,\ \langle \mathbf{u}, \mathbf{y}_\ell \rangle,\ a_{\ell,0} z,\ \boxed{r'_{\ell,0} \cdot \langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle})_{\mathbf{H}^*}
\end{aligned}$$

and $\pi_j \xleftarrow{\$} \mathbb{Z}_q$. If $\langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle = 0$, then this does not change the functional key. We also clean the τ and τz_j masks in $\mathbf{c}_j$ at the end of this game:

$$\mathbf{c}_j \coloneqq (\sigma_j \cdot (1, -j),\ \psi,\ \boxed{0},\ 0,\ \boxed{0},\ 0,\ 0)_{\mathbf{F}} \text{ for } j \in \mathsf{R} \ .$$

If $\langle \mathbf{x}^*_0 - \mathbf{x}^*_1, \mathbf{y}_\ell \rangle \neq 0$, then the 3-rd component of $\mathbf{k}^*_{\ell,\mathsf{ipfe}}$ is a random value, otherwise it stays 0 as in G_2. The transition from G_2 to G_3 is discussed in Lemma 5 and the difference in advantages is

$$|\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}(\mathsf{G}_2)| \leq (2K \cdot (P(6P+3)+2)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda) \ .$$

Game G_4: The key and ciphertext components are changed to

$$\begin{aligned}
\mathbf{c}_{\mathsf{ipfe}} &\coloneqq (\omega,\ \boxed{\omega'},\ \psi,\ \tau)_{\mathbf{H}} \\
\mathbf{t}_i &\coloneqq \left[\!\!\left[\omega \cdot s_i + \boxed{\omega' \cdot u_i} + \mathbf{x}^*_b[i] \right]\!\!\right]_1 \text{ for } i \in [n] \\
\mathbf{m}^*_{\ell,i} &\coloneqq [\![\mathbf{y}_\ell[i]]\!]_2 \\
\mathbf{k}^*_{\ell,\mathsf{ipfe}} &\coloneqq (\langle \mathbf{s}, \mathbf{y}_\ell \rangle,\ \langle \mathbf{u}, \mathbf{y}_\ell \rangle,\ a_{\ell,0} z,\ r'_{\ell,0} \cdot \langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle)_{\mathbf{H}^*} \ ,
\end{aligned}$$

where $\omega' \xleftarrow{\$} \mathbb{Z}_q$ is chosen uniformly at random.
Given a DDH instance $([\![\mu]\!]_1, [\![\omega]\!]_1, [\![\omega']\!]_1)$ where either $\omega' - \mu\omega = 0$ or $\omega' - \mu\omega$ is a uniformly random value in $\mathbb{Z}_q$, the simulator can simulate the ciphertext components $\mathbf{t}_i$ and $\mathbf{c}_{\mathsf{ipfe}}$ as follows:

$$\begin{aligned}
\mathbf{t}_i &= \left[\!\!\left[\omega \cdot s_i + \omega' \cdot u_i + \mathbf{x}^*_b[i] \right]\!\!\right]_1 \text{ for } i \in [n] \\
\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \omega',\ \psi,\ \tau)_{\mathbf{H}}
\end{aligned}$$

which is possible using $g_1^{\omega'}$ together with the master secret vectors $\mathbf{s}, \mathbf{u}$ as well as the basis changing matrices $(\mathbf{H}, \mathbf{H}^*)$. If $\omega' - \mu\omega = 0$ we are in G_3, otherwise we are in the current game. Hence, the difference in advantages is $|\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$.

Game G_5: In this game we rewrite the master secret vectors $\mathbf{s}, \mathbf{u}$:

$$\mathbf{s}' \coloneqq \mathbf{s} + \Delta\mathbf{s}$$
$$\mathbf{u}' \coloneqq \mathbf{u} + \Delta\mathbf{u}$$

where $(\Delta\mathbf{s}, \Delta\mathbf{u})$ satisfies:

$$\Delta\mathbf{s} + \mu\Delta\mathbf{u} = 0 \tag{4}$$
$$\omega \cdot \Delta\mathbf{s} + \omega' \cdot \Delta\mathbf{u} = \mathbf{x}_b - \mathbf{x}_0 \ . \tag{5}$$

The key and ciphertext components $\mathbf{t}_i$ and $\mathbf{k}^*_{\mathsf{ipfe}}$ will be changed to:

$$\mathbf{t}_i = \left[\!\left[\omega s'_i + \omega' u'_i + \mathbf{x}^*_0[i]\right]\!\right]_1$$
$$\mathbf{k}^*_{\ell,\mathsf{ipfe}} = (\langle \mathbf{s}', \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}', \mathbf{y}_\ell\rangle,\ a_{\ell,0}z,\ r''_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}$$

We have to make sure that the variable change does not affect the public key output to the adversary, i.e. it should hold that $\mathbf{s} + \mu \cdot \mathbf{u} = \mathbf{s}' + \mu\mathbf{u}'$ and hence $(\Delta\mathbf{s}[i], \Delta\mathbf{u}[i])$ must satisfy the equation (4). Moreover, since we are rewriting variables and replacing $\mathbf{x}^*_b[i]$ by $\mathbf{x}^*_0[i]$ in the ciphertext, $\mathbf{s}'[i]$ and $\mathbf{u}'[i]$ must also satisfy $\omega\mathbf{s}'[i] + \omega'\mathbf{u}'[i] + \mathbf{x}^*_0[i] = \omega\mathbf{s}[i] + \omega'\mathbf{u}[i] + \mathbf{x}^*_b[i]$, or equivalently the equation (5). Because ω, ω', μ are uniformly random and independent, the system has a solution $(\Delta\mathbf{s}[i], \Delta\mathbf{u}[i])$ and the simulation will succeed when $\omega' - \mu\omega \neq 0$, which happens with overwhelming probability.
The ciphertext components $\mathbf{t}_i$ become:

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

and the key component $\mathbf{k}_{\ell,\mathsf{ipfe}}$ becomes

$$\mathbf{k}_{\ell,\mathsf{ipfe}} = (\langle \mathbf{s}' - \Delta\mathbf{s}, \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}' - \Delta\mathbf{u}, \mathbf{y}_\ell\rangle,\ a_{\ell,0}z,\ r'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*} \ .$$

Thanks to the systems equations (4) and (5), it holds that if $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle = 0$ then $\langle \Delta\mathbf{s}, \mathbf{y}_\ell\rangle = \langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle = 0$. Therefore, in that case the key component $\mathbf{k}_{\ell,\mathsf{ipfe}}$ has the desired form of G_5

$$\mathbf{k}_{\ell,\mathsf{ipfe}} = (\langle \mathbf{s}', \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}', \mathbf{y}_\ell\rangle,\ a_{\ell,0}z,\ r''_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*} \ ,$$

and $r''_{\ell,0} \coloneqq r'_{\ell,0}$.
It remains to consider the case when $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$. It is then obligatory we remove the additive terms $\langle \Delta\mathbf{s}, \mathbf{y}_\ell\rangle$ and $\langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle$ from the first 2 coordinates of $\mathbf{k}_{\ell,\mathsf{ipfe}}$. The main reason is that the adversary might test this ℓ-th key for decrypting a normal ciphertext, whose attributes *satisfy* the key's policy. We note that concerning the challenge ciphertext, $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$ already implies that the key's policy is not satisfied by the challenge attributes R and thus it is not decryptable using this ℓ-th key. We describe a sequence of hybrids to go from G_4 to G_5 in the case $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$ in Figure 12.

Game $\mathsf{G}_{4.1}$: First, we change ω' back to $\mu\omega$ in both $\mathbf{t}_i$ and $\mathbf{c}_{\mathsf{ipfe}}$:

$$\mathbf{t}_i = \left[\!\left[\omega s'_i + \mu\omega u'_i + \mathbf{x}^*_0[i]\right]\!\right]_1 \text{ for } i \in [n]$$
$$\mathbf{c}_{\mathsf{ipfe}} = (\omega,\ \mu\omega,\ \psi,\ \tau)_{\mathbf{H}} \ .$$

This change is negligible under the adversary's view under the DDH assumption in $\mathbb{G}_1$, in the same manner as we have done to go from G_3 to G_4.
At the same time, we move $\mu\langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle$ from the second component to the first component of $\mathbf{k}_{\mathsf{ipfe}}$. That is, the key and ciphertext components will become:

$$\begin{aligned}\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau)_{\mathbf{H}}\\ \mathbf{k}_{\ell,\mathsf{ipfe}} &= (\langle \mathbf{s}', \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}', \mathbf{y}_\ell\rangle,\ a_{\ell,0}z,\ r''_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\ .\end{aligned}$$

We stress that the swap is done using a different DSDH instance as below, independent of the DDH we use to switch ω' to $\mu\omega$ above. It is important to note that we are currently in the case $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$, which implies the policy in the ℓ-th key is not satisfied by the ciphertext's attributes and the decryption of the challenge ciphertext does not play an important role. In contrast, the functional key should still be able to decrypt *normal* ciphertexts, which can be computed by the adversary using pk. This is why we are considering the swap in order to "correct" the key and get rid of the noises introduced by $\Delta\mathbf{s}$ and $\Delta\mathbf{u}$.
Given a DSDH instance $([\![a]\!]_2, [\![b]\!]_2, [\![c]\!]_2)$ where $c - ab = \rho$ for either $\rho = \langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle$ or $\rho = 0$, the simulator perform the following basis change using:

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

This basis change affects $\mathbf{h}_1$ and $\mathbf{h}_2$, but it is perfectly indistinguishable because the adversary knows only $\mathbf{h}_1 + \mu\mathbf{h}_2$, where μ is a uniformly random value and $\mathbf{h}_1, \mathbf{h}_2$ are two random basis vectors. Note that this basis change does not affect the public information $\mathbf{h}_1 + \mu\mathbf{h}_2$ known by the adversary. The vector $\mathbf{h}_4^*$ will be changed as well but it is already hidden from the adversary. We cannot compute $\mathbf{h}_1$ and $\mathbf{h}_2$ because we do not have $[\![a]\!]_1$ but the ciphertext component can be written directly in $\mathbf{T}$:

$$\begin{aligned}\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau)_{\mathbf{T}}\\ &= (\omega,\ \mu\omega,\ \psi,\ \tau - a\mu\tau + a\mu\tau)_{\mathbf{H}}\\ &= (\omega,\ \mu\omega,\ \psi,\ \tau)_{\mathbf{H}}\ .\end{aligned}$$

The key component can be virtually set:

$$\begin{aligned}\mathbf{k}_{\ell,\mathsf{ipfe}} &= (\langle \mathbf{s}' - \Delta\mathbf{s}, \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}' - \Delta\mathbf{u}, \mathbf{y}_\ell\rangle,\ a_{\ell,0}z,\ r'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\\ &\quad + (-c\mu,\ c,\ 0,\ b)_{\mathbf{T}^*}\\ &= (\langle \mathbf{s}' - \Delta\mathbf{s}, \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}' - \Delta\mathbf{u}, \mathbf{y}_\ell\rangle,\ a_{\ell,0}z,\ r'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\\ &\quad + (-\mu\rho,\ \rho,\ 0,\ b)_{\mathbf{T}^*}\\ &= (\langle \mathbf{s}' - \Delta\mathbf{s}, \mathbf{y}_\ell\rangle - \mu\rho,\ \langle \mathbf{u}' - \Delta\mathbf{u}, \mathbf{y}_\ell\rangle + \rho,\\ &\qquad a_{\ell,0}z,\ (r'_{\ell,0} + b/\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\ ,\end{aligned}$$

where $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$ in the current case and $r'_{\ell,0}$ is updated to $r''_{\ell,0} \coloneqq r'_{\ell,0} + b/\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle$. If $\rho = \langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle$ then we are swapping, otherwise we are not. After the swap, the key and

ciphertext components are as follows:

$$\begin{aligned}\mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \mu\omega,\ \psi,\ \tau)_{\mathbf{H}}\\ \mathbf{k}_{\ell,\mathsf{ipfe}} &= (\langle \mathbf{s}' - (\Delta\mathbf{s} + \mu\Delta\mathbf{u}), \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}', \mathbf{y}_\ell\rangle,\ a_{\ell,0} z,\ r''_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\\ &\overset{(*)}{=} (\langle \mathbf{s}', \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}', \mathbf{y}_\ell\rangle,\ a_{\ell,0} z,\ r''_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\ ,\end{aligned}$$

where $(*)$ comes from the fact that $\Delta\mathbf{s}[i] + \mu\Delta\mathbf{u}[i] = 0$ for all $i \in [n]$.
Totally, the difference of advantages is

$$|\mathsf{Adv}(\mathsf{G}_{4.1}) - \mathsf{Adv}(\mathsf{G}_4)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)\ .$$

Game $\mathsf{G}_{4.2}$: We use DDH to switch $\mu\omega$ to a uniformly random value ω', as from G_3 to G_4:

$$\begin{aligned}\mathbf{t}_i &= \left[\!\!\left[\omega s'_i + \omega' u'_i + \mathbf{x}^*_0[i]\right]\!\!\right]_1 \text{ for } i \in [n]\\ \mathbf{c}_{\mathsf{ipfe}} &= (\omega,\ \omega',\ \psi,\ \tau)_{\mathbf{H}}\\ \mathbf{k}_{\ell,\mathsf{ipfe}} &= (\langle \mathbf{s}', \mathbf{y}_\ell\rangle,\ \langle \mathbf{u}', \mathbf{y}_\ell\rangle,\ a_{\ell,0} z,\ r''_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)_{\mathbf{H}^*}\ .\end{aligned}$$

The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_{4.2}) - \mathsf{Adv}(\mathsf{G}_{4.1})| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda)$.

In the end, we have $\mathsf{G}_{4.2}$ being identical to G_5.

In G_5, the challenge bit b is not involved in the computation anymore. Hence, the advantage becomes $\mathsf{Adv}(\mathsf{G}_5) = 0$ and we obtain:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda) &= \mathsf{Adv}(\mathsf{G}_0)\\ &= |\mathsf{Adv}(\mathsf{G}_0) - \mathsf{Adv}(\mathsf{G}_5)|\\ &\leq \sum_{i=1}^{5} |\mathsf{Adv}(\mathsf{G}_i) - \mathsf{Adv}(\mathsf{G}_{i-1})|\\ &\leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + (K \cdot (2P \cdot (6P+3) + 2) + 2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)\\ &\quad + \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}(1^\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)\\ &\leq (2KP \cdot (6P+3) + 2K + 9) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)\ .\end{aligned}$$

and the proof is concluded. □

Lemma 5. *Assuming the* SXDH *assumption for* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, the difference between advantages* $|\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}(\mathsf{G}_2)|$ *in Theorem 1 is negligible.*

Proof. If $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle = 0$, then the ℓ-th functional key is identical in both games. Otherwise, it is a direct application of the masking lemma in Section 3, because the security model ensures that if $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$, the policy in the ℓ-th functional key is not satisfied by the attributes in the challenge ciphertext. We note that games $\mathsf{G}_0, \mathsf{G}_1$ of Theorem 1 already introduces the masks in $(\mathbf{c}_j)_j, \mathbf{c}_{\mathsf{ipfe}}$, we only need to apply the Lemma 2 to perform the masking of $(\mathbf{k}_{\ell,j})_j, \mathbf{k}_{\ell,\mathsf{ipfe}}$, for each ℓ-th functional key. The effected coordinates are $(3,4)$ of $(\mathbf{H}, \mathbf{H}^*)$ and all coordinates of $(\mathbf{F}, \mathbf{F}^*)$. In the matrices, only the relating indices in (H, H') are used, the others are kept as in the identity matrix $I_4 \in \mathbb{Z}_q^{4\times 4}$. The two constants to be used in the lemma are 1 and $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle$. One important remark here is the obligation to apply the lemma key by key, not simultaneously. The reason is that for two different functional keys queried by the adversary, the two policies in question might depend on the same

attribute in the challenge ciphertext, and thus the KP-ABE part of the keys might reveal information mutually. In the end, after all the functional keys are masked, there can be at most K keys being changed, we clean the masks in $(\mathbf{c}_j)_j, \mathbf{c}_{\mathsf{ipfe}}$. The difference in advantages is

$$|\mathsf{Adv}(\mathsf{G}_3) - \mathsf{Adv}(\mathsf{G}_2)| \leq (K \cdot (2P(6P+3)+2)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(1^\lambda)$$

and the proof is completed. □

B.3 Proof of Theorem 2

Proof (Of Theorem 2). We give the sequence of games in Figure 13. 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: This is the adaptive security game as given in Figure 1. We have $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}} = \mathsf{Adv}(\mathsf{G}_0)$.
Game G_1: In this game we introduce the masks in the key components $\mathbf{k}_{\ell,\mathsf{ipfe}}$:

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

The transition from G_0 to G_1 is discussed separately in Lemma 6. 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'

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

We note that if the attributes in the ciphertext satisfy some ℓ-th key's policy, it is still decryptable using this key. 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 $(\mathbf{s},\mathbf{u})$, i.e. the component $\mathbf{k}_{\ell,\mathsf{ipfe}}$ is:

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

Last but not least, we can require the adversary to query all functional keys conforming to the coondition 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.

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. 13: Games for Theorem 2. The index i runs in $\{1,\ldots,n\}$. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in R 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 6 in Appendix B.3, which will make use of the auxiliary vectors in $(\mathbf{F},\mathbf{F}^*)$ and $(\mathbf{H},\mathbf{H}^*)$.

When the adversary declares the challenge messages $(\mathbf{x}^*_0,\mathbf{x}^*_1)$, the simulator updates the master secret vectors $\mathbf{s},\mathbf{u}$ to:

$$\begin{aligned} \mathbf{s}' &:= \mathbf{s} + \Delta\mathbf{s} \\ \mathbf{u}' &:= \mathbf{u} + \Delta\mathbf{u} \end{aligned}$$

where $(\Delta\mathbf{s},\Delta\mathbf{u})$ satisfies:

$$\begin{aligned} &\Delta\mathbf{s} + \mu\Delta\mathbf{u} = 0 \\ &\omega\cdot\Delta\mathbf{s} + \omega'\cdot\Delta\mathbf{u} = \mathbf{x}_b - \mathbf{x}_0 \ . \end{aligned} \tag{6}$$

It is straightforward to see that this change does not affect the public information pk that the adversary possesses, because $\mathbf{s}+\mu\mathbf{u} = \mathbf{s}'+\mu\mathbf{u}'$. The challenge ciphertext is now encrypting $\mathbf{x}^*_0$

under $(\mathbf{s}', \mathbf{u}')$, 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:

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

We have to consider two cases:

- In the case $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle = 0$, there is no further changes to do because $\mathbf{s}' = \mathbf{s}$ and $\mathbf{u}' = \mathbf{u}$. 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 $\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$, we need to remove the noises $\langle \Delta\mathbf{s}, \mathbf{y}_\ell\rangle$ and $\langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle$ so that the functional keys have the correct form w.r.t the new master secret vectors $(\mathbf{s}', \mathbf{u}')$ 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.

We use the same approach as in the proof of Theorem 1, where 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 $\rho = \langle \Delta\mathbf{u}, \mathbf{y}_\ell\rangle$ 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:

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

The randomness $r'_{\ell,0}$ is updated to $r'_{\ell,0} + b/\mathbf{y}_\ell[1]$. If $\rho = \langle \Delta\mathbf{u}, \mathbf{y}_\ell \rangle$ we are cleaning the noises using the relation (6), 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

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

The proof is concluded. □

Lemma 6. *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 2 is negligible.*

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

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

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 14. 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 1 for $i \in [n]$, where at each step, 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 $\{0,\ldots,n\}$. In the end $\mathsf{G}_{0.\ell-1.0.0} = \mathsf{G}_{0.\ell-1.0}$ and $\mathsf{G}_{0.\ell-1.0.n} = \mathsf{G}_{0.\ell-1.1}$. The ciphertext and functional key components in $\mathsf{G}_{0.\ell-1.0.i}$, where $i \in [n]$, are:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Fig. 14: Games for Lemma 6. The index i runs in $\{1,\ldots,n\}$. The index j runs in $\mathsf{List\text{-}Att}(\mathbb{A})$ for key components and in R for ciphertext components. The index ℓ runs in $\{1,\ldots,K\}$ for the functional key queries.

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

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

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

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

This 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 $\mathbb{A}(\mathsf{R}) = 0$ where $\mathbb{A}$ is the access structure embedded in the key and R 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 $A \subseteq \mathsf{R}$, 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 $\tau, z_j \xleftarrow{\$} \mathbb{Z}_q$, 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:

  $$\begin{aligned}
  &\log_{g_t}(\mathbf{k}^*_{\ell,\mathsf{ipfe}} \times \mathbf{c}_{\mathsf{ipfe}}) \\
  &= \omega\langle \mathbf{s}, \mathbf{y}_\ell\rangle + \mu\omega\langle \mathbf{u}, \mathbf{y}_\ell\rangle + \psi a_{\ell,0} z + \textstyle\sum_{i=1}^n \tau(a'_{\ell,0} + r'_{\ell,0}) \mathbf{y}_\ell[i] \Delta\mathbf{x}[i] \\
  &= \omega\langle \mathbf{s}, \mathbf{y}_\ell\rangle + \mu\omega\langle \mathbf{u}, \mathbf{y}_\ell\rangle + \psi a_{\ell,0} z + \tau(a'_{\ell,0} + r'_{\ell,0})\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \\
  &= \omega\langle \mathbf{s}, \mathbf{y}_\ell\rangle + \mu\omega\langle \mathbf{u}, \mathbf{y}_\ell\rangle + \psi a_{\ell,0} z \\
  &\log_{g_t}\left(\textstyle\sum_{j \in A}(c_j \cdot \mathbf{k}^*_{\ell,j}) \times \mathbf{c}_j\right)
  \end{aligned}$$

$= \sum_{j\in A} \psi c_j a_{\ell,j} z + \sum_{i=1}^{n} \left(\sum_{j\in A} \frac{c_j a'_{\ell,j} \cdot \mathbf{y}_\ell[i]}{z_j} \cdot \tau z_j \Delta\mathbf{x}[i] \right)$
$= \psi a_{\ell,0} z + a'_{\ell,0} \tau \langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle$
$= \psi a_{\ell,0} z$

where $A \subseteq \mathsf{R}$ 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:

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

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 1 for 2 indices $i \in [n]$ at the same time over the same bases $(\mathbf{H}, \mathbf{H}^*), (\mathbf{F}, \mathbf{F}^*)$.

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

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

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 $A \subseteq \mathsf{R}$ be an authorized set and $(c_j)_j$ be its reconstruction vector from LSSS

$\log_{g_t}(\mathbf{k}^*_{\ell,\mathsf{ipfe}} \times \mathbf{c}_{\mathsf{ipfe}})$
$= \omega\langle \mathbf{s}, \mathbf{y}_\ell \rangle + \mu\omega\langle \mathbf{u}, \mathbf{y}_\ell \rangle + \psi a_{\ell,0} z + \sum_{k=1}^{i} \tau r'_{\ell,0} \mathbf{y}_\ell[k] \Delta\mathbf{x}[k]$
$\quad + \sum_{k=i+1}^{n} \tau (a'_{\ell,0} + r'_{\ell,0}) \mathbf{y}_\ell[k] \Delta\mathbf{x}[k]$
$= \omega\langle \mathbf{s}, \mathbf{y}_\ell \rangle + \mu\omega\langle \mathbf{u}, \mathbf{y}_\ell \rangle + \psi a_{\ell,0} z + \tau r'_{\ell,0} \langle \Delta\mathbf{x}, \mathbf{y}_\ell \rangle + \sum_{k=i+1}^{n} \tau a'_{\ell,0} \mathbf{y}_\ell[k] \Delta\mathbf{x}[k]$
$= \omega\langle \mathbf{s}, \mathbf{y}_\ell \rangle + \mu\omega\langle \mathbf{u}, \mathbf{y}_\ell \rangle + \psi a_{\ell,0} z + \sum_{k=i+1}^{n} \tau a'_{\ell,0} \mathbf{y}_\ell[k] \Delta\mathbf{x}[k]$

$\log_{g_t} \left(\sum_{j\in A} (c_j \cdot \mathbf{k}^*_{\ell,j}) \times \mathbf{c}_j \right)$

$= \sum_{j\in A} \psi c_j a_{\ell,j} z + \sum_{k=i+1}^{n} \left(\sum_{j\in A} \frac{c_j a'_{\ell,j} \cdot \mathbf{y}_\ell[k]}{z_j} \cdot \tau z_j \Delta\mathbf{x}[k] \right)$
$= \psi a_{\ell,0} z + \sum_{k=i+1}^{n} \tau a'_{\ell,0} \mathbf{y}_\ell[k] \Delta\mathbf{x}[k]$
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 1 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 3

Proof (Of Theorem 3). The sequence of games can be found in Figure 6 and 7. 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

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

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

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

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

Extract$(\mathbb{A}, \mathbf{y}^{(\ell)})$: 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 $a^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, constructs the associated LSSS and runs the labeling algorithm to obtain the labels $(a^{(\ell)}_{i,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a^{(\ell)}_{i,0}}(\mathbb{A})$. It then returns

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

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

LoR$(i, \mathsf{tag}, \mathsf{R})$: As described in Figure 4, the $(\mathsf{tag}, \mathsf{R})$ of the first **LoR** query will determine the only challenge tag from then on. Upon receiving a set $\mathsf{R} \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ of attributes, the simulator samples $\psi_i \xleftarrow{\$} \mathbb{Z}_q$, flips a coin $b \xleftarrow{\$} \{0,1\}$, compute $\mathsf{H}(\mathsf{tag}, \mathsf{R}) \to (\llbracket \omega_{\mathsf{tag},\mathsf{R}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag},\mathsf{R}} \rrbracket_1) \in \mathbb{G}_1^2$ and

$$\begin{aligned}\mathbf{t}_i &\coloneqq \llbracket \omega_{\mathsf{tag},\mathsf{R}} \cdot s_i + \omega'_{\mathsf{tag},\mathsf{R}} u_i + \mathbf{x}_b^*[i] \rrbracket_1 \\ \mathbf{c}_{i,\mathsf{ipfe}} &\coloneqq (\omega_{\mathsf{tag},\mathsf{R}},\ \omega'_{\mathsf{tag},\mathsf{R}},\ \psi_i,\ 0)_{\mathbf{H}_i}\end{aligned}$$

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

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

and $\sigma_{i,j} \xleftarrow{\$} \mathbb{Z}_q$ for $j \in \mathsf{R}'$ and $\mathsf{H}(\mathsf{tag}, \mathsf{R}) \to (\llbracket \omega \rrbracket_1, \llbracket \omega' \rrbracket_1)$ is modeled as a random oracle (RO).

Enc$(i, x_i, \mathsf{tag}', \mathsf{R}')$: As dictated by the security model in Figure 4, the adversary can only query for encryptions of messages under tag' different from the challenge tag. The ciphertext is returned:

$$\begin{aligned}\mathbf{c}_{i,j} &\coloneqq (\sigma_{i,j} \cdot (1, -j),\ \psi'_i,\ 0,\ 0,\ 0,\ 0,\ 0)_{\mathbf{F}} \text{ for } j \in \mathsf{R}' \\ \mathbf{t}_i &\coloneqq \llbracket \chi_{\mathsf{tag}',\mathsf{R}'} s_i + \chi'_{\mathsf{tag}',\mathsf{R}'} u_i + x_i \rrbracket_1 \\ \mathbf{c}_{i,\mathsf{ipfe}} &\coloneqq (\chi_{\mathsf{tag}',\mathsf{R}'},\ \chi'_{\mathsf{tag}',\mathsf{R}'},\ \psi'_i,\ 0)_{\mathbf{H}_i}\end{aligned}$$

where $\sigma_{i,j}, \psi'_i \xleftarrow{\$} \mathbb{Z}_q$ for $j \in \mathsf{R}'$ and $\mathsf{H}(\mathsf{tag}', \mathsf{R}') \to (\llbracket \chi_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1)$ 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 $\mathsf{Adv}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda) = \mathsf{Adv}(\mathsf{G}_0)$.

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 1. The ciphertext components are computed as below:

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

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. 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 15.

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

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

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

with another random labeling $(a'^{(\ell)}_j)_j \leftarrow \Lambda_{a'^{(\ell)}_0}(\mathbb{A})$ where $a'^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$ and the ciphertext components are those returned from **LoR**. For all $i \in [n]$, Lemma 1 is applied in parallel using the same random labeling for random labeling $(a'^{(\ell)}_j)_j \leftarrow \Lambda_{a'^{(\ell)}_0}(\mathbb{A})$. The affected coordinates are $(3, 4)$ of

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

Game $\mathsf{G}_{1.\ell-1.1}$: $\mathsf{H}(\mathsf{tag},\mathsf{R}) \to (\llbracket \omega_{\mathsf{tag},\mathsf{R}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag},\mathsf{R}} \rrbracket_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') \to (\llbracket \chi_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1)$, $a_0'^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $(a_j'^{(\ell)})_j \leftarrow \Lambda_{a_0'^{(\ell)}}(\mathbb{A})$, $\Delta\mathbf{x} := \mathbf{x}_b^*[i] - \mathbf{x}_0^*[i]$

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

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

Game $\mathsf{G}_{1.\ell-1.2}$: $\mathsf{H}(\mathsf{tag},\mathsf{R}) \to (\llbracket \omega_{\mathsf{tag},\mathsf{R}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag},\mathsf{R}} \rrbracket_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') \to (\llbracket \chi_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1)$

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

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

Game $\mathsf{G}_{1.\ell-1.3} = \mathsf{G}_{1.\ell}$: $\mathsf{H}(\mathsf{tag},\mathsf{R}) \to (\llbracket \omega_{\mathsf{tag},\mathsf{R}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag},\mathsf{R}} \rrbracket_1)$, $\mathsf{H}(\mathsf{tag}',\mathsf{R}') \to (\llbracket \chi_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}',\mathsf{R}'} \rrbracket_1)$

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

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

Fig. 15: 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 3. 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 R 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.

$(\mathbf{H}_i, \mathbf{H}_i^*)$ and all coordinates of $(\mathbf{F}, \mathbf{F}^*)$. The constants are $x := \Delta\mathbf{x}[i]$ and $y := \mathbf{y}^{(\ell)}[i]$. More precisely, the challenge ciphertext and the ℓ-th functional key components will be:

$$\begin{aligned}
\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{R}): \ \mathbf{c}_{i,j} &= (\sigma_{i,j} \cdot (1, -j), \psi_i, \tau\Delta\mathbf{x}[i], 0, \tau\Delta\mathbf{x}[i] z_j, 0, 0)_{\mathbf{F}} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H}: \ \mathbf{k}_{i,j}^{(\ell)} &= (\pi_{i,j}^{(\ell)} \cdot (j, 1), a_{i,j}^{(\ell)} \cdot z, 0, 0, a_j'^{(\ell)} \mathbf{y}^{(\ell)}[i]/z_j, 0, 0)_{\mathbf{F}^*} \\
\mathbf{LoR}(i \in \mathcal{C} \cup \mathcal{H}, \mathsf{tag}, \mathsf{R}): \ \mathbf{c}_{i,\mathsf{ipfe}} &= (\omega_{\mathsf{tag},\mathsf{R}} p_i, \ \omega'_{\mathsf{tag},\mathsf{R}} p_i, \ \psi_i, \tau\Delta\mathbf{x}[i])_{\mathbf{H}_i} \\
\forall\, i \in \mathcal{C} \cup \mathcal{H}: \ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} &= (\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle, \ \langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle, \ a_{i,0}^{(\ell)} z, a_0'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \ .
\end{aligned}$$

By Lemma 1, 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:

- 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 1 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 $1/z_j$, 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 $\mathbf{f}_5$. 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.
- 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 remark that it is this possibility to parallelize the basis changes and the application that makes our *duplicate-and-compress* technique work.

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

$$\forall\, i \in \mathcal{C} \cup \mathcal{H}: \ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} = (\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle, \ \langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle, \ a_{i,0}^{(\ell)} z, \ r_0'^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} \ .$$

We recall that the model of security impedes the use of different sets of attributes among clients $i \in [n]$. That is, the encryption receives the same set R 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 $\mathbb{A}(\mathsf{R}) = 0$ where $\mathbb{A}$ is the access structure embedded in the key and R contains the attributes in the challenge ciphertext. Hence, for all $i \in [n]$, there exists no authorized set $A \subseteq \mathsf{R}$ 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

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

  Furthermore, because $(a_j'^{(\ell)})_j$ is a random labeling of $a_0'^{(\ell)}$ using the underlying LSSS and $\tau, z_j \xleftarrow{\$} \mathbb{Z}_q$, it holds that 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 $A \subseteq \mathsf{R}$ 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:

$$\begin{aligned}
\sum_{i=1}^{n} \left(\sum_{j \in A} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}^{(\ell)}) \right) &= \sum_{j \in A} \left(\sum_{i=1}^{n} \mathbf{c}_{i,j} \times (c_j \cdot \mathbf{k}_{i,j}^{(\ell)}) \right) \\
&= \sum_{j \in A} \left(\sum_{i=1}^{n} \psi_i c_j a_{i,j}^{(\ell)} z + \tau c_j \cdot a_j'^{(\ell)} \Delta\mathbf{x}[i] \mathbf{y}^{(\ell)}[i] \right) \\
&= \sum_{i=1}^{n} \psi_i \left(\sum_{j \in A} c_j a_{i,j}^{(\ell)} z \right) \\
&= \sum_{i=1}^{n} \psi_i a_{i,0}^{(\ell)} z \\
\sum_{i=1}^{n} \left(\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} \right) &= \sum_{i=1}^{n} \left(\psi_i a_{i,0}^{(\ell)} z + \tau a_0'^{(\ell)} \mathbf{y}_\ell[i] \Delta\mathbf{x}[i] \right) \\
&= \sum_{i=1}^{n} \psi_i a_{i,0}^{(\ell)} z
\end{aligned}$$

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 1 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 $(\mathbf{F}, \mathbf{F}^*)$ 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 changing all $i \in [n]$ at the same time, if $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$ and the policy is satisfied having a reconstruction vector $(c_j)_j$, the summation

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

always has the term

$$\sum_{j} c_j \cdot \left(\sum_{i=1}^{n} \tau a_j'^{(\ell)} \mathbf{y}_\ell[i] \Delta\mathbf{x}[i] \right) = \sum_{j} \tau c_j a_j'^{(\ell)} \langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle \ .$$

Hence, 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 1, 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 the subspace indistinguishability to clean the coordinate $(3,6)$ of $\mathbf{c}_{i,j}$ 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 $\mathsf{H}(\mathsf{tag},\mathsf{R})$ 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 $\mathsf{H}(\mathsf{tag}',\mathsf{R}')$ where $\mathsf{tag}' \neq \mathsf{tag}$, we respond by a random vector lying in $\mathrm{span}((1,\ \mu)) \subseteq \mathbb{Z}_q^2$, for $\mu \xleftarrow{\$} \mathbb{Z}_q$. On the other hand, the RO query $\mathsf{H}(\mathsf{tag},\mathsf{R})$ is still responded by $\left[\!\left[(\omega_{\mathsf{tag},\mathsf{R}}, \omega'_{\mathsf{tag},\mathsf{R}})\right]\!\right]_1$ where $(\omega_{\mathsf{tag},\mathsf{R}}, \omega'_{\mathsf{tag},\mathsf{R}})$ 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

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

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 $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_q^4$

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

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 $H_i' \coloneqq \left(H_i^{-1}\right)^\top$.
Then, before answering any query, the simulator perform a basis change on (H_i, H_i') to obtain:

$$K_i \coloneqq \begin{bmatrix} 1 & \mu & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot H_i + \left[\begin{array}{c} -\mu \cdot \mathbf{r}_i \\ \hline \mu \cdot \mathbf{r}_i - H_i^{(2)} \\ \hline \mathbf{0} \\ \hline \mathbf{0} \end{array}\right] = \left[\begin{array}{c} H_i^{(1)} + \mu H_i^{(2)} - \mu \mathbf{r}_i \\ \hline \mu \mathbf{r}_i \\ \hline H_i^{(3)} \\ \hline H_i^{(4)} \end{array}\right] .$$

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

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

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

$$\begin{aligned} &\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{R}) && \mathbf{c}_{i,\mathsf{ipfe}} = (\omega_{\mathsf{tag},\mathsf{R}} p_i, \omega'_{\mathsf{tag},\mathsf{R}} p_i,\ \psi_i,\ \tau \Delta\mathbf{x}[i])_{\mathbf{H}_i} \\ &\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{R}') && \mathbf{c}_{i,\mathsf{ipfe}} = (\chi_{\mathsf{tag}',\mathsf{R}'} p_i, \mu\chi_{\mathsf{tag}',\mathsf{R}'} p_i,\ \psi_i',\ 0)_{\mathbf{H}_i} \ . \end{aligned}$$

We briefly recall the argument that was used in the selective scenario and is still applicable here. This basis change from (H_i, H_i') to (K_i, K_i') is indistinguishable for the corrupted i because the distribution of ek_i stays the same and even though ek_i behaves inconsistently w.r.t $\mathsf{H}(\mathsf{tag}, \mathsf{R})$, i.e.

$$p_i K_i^{(1)} \cdot [\![\omega_{\mathsf{tag}}]\!]_1 + p_i K_i^{(2)} \cdot [\![\omega'_{\mathsf{tag}}]\!]_1 \neq \omega_{\mathsf{tag}} p_i \cdot \mathbf{h}_{i,1} + \omega'_{\mathsf{tag}} p_i \cdot \mathbf{h}_{i,2}$$

the security model requires that a corrupted i will not be queried to the challenge oracle **LoR**. This means that the inconsistency is unknown to the adversary. Moreover, all ciphertexts from **Enc**, which must be under $\mathsf{tag}' \neq \mathsf{tag}$ and are thus consistent with the new ek_i, will behave as usual if one tries to decrypt them later.
For an honest i, the encryption key ek_i is never revealed and even if queried to **Enc** for a ciphertext at index i, it must be under the a different $\mathsf{tag}' \neq \mathsf{tag}$. Consequently, even an unbounded adversary will only obtain

$$p_i K_i^{(1)} \cdot [\![\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1 + p_i K_i^{(2)} \cdot [\![\mu\chi_{\mathsf{tag}',\mathsf{R}'}]\!]_1 = \chi_{\mathsf{tag}',\mathsf{R}'} p_i \cdot \mathbf{h}_{i,1} + \mu\chi_{\mathsf{tag}',\mathsf{R}'} p_i \cdot \mathbf{h}_{i,2} \ ,$$

where $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}$ are totally hidden. This means that the (even unbounded) adversary's view stays the same for this honest client i. Finally, 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 $\mathbf{s}' = \mathbf{s} + \Delta\mathbf{s}$ and $\mathbf{u}' = \mathbf{u} + \Delta\mathbf{u}$ satisfy

$$\begin{aligned} \mathbf{s}' &\coloneqq \mathbf{s} + \Delta\mathbf{s} \\ \mathbf{u}' &\coloneqq \mathbf{u} + \Delta\mathbf{u} \end{aligned}$$

where $(\Delta\mathbf{s}, \Delta\mathbf{u})$ satisfies

$$\begin{cases} \Delta\mathbf{s} + \mu \cdot \Delta\mathbf{u} = 0 \\ \omega_{\mathsf{tag},\mathsf{R}} \cdot \Delta\mathbf{s} + \omega'_{\mathsf{tag},\mathsf{R}} \cdot \Delta\mathbf{u} = \Delta\mathbf{x} \end{cases} .$$

The challenge ciphertext does not change and so do the encryption keys for corrupted i, due to the constraint $\Delta\mathbf{x}[i] = 0$. Moreover, we observe that if $\langle \Delta\mathbf{x}, \mathbf{y}^{(\ell)} \rangle = 0$, then $\Delta\mathbf{s} = \Delta\mathbf{u} = 0$.
The functional key that are using $(\mathbf{s}, \mathbf{u})$ will be changed to:

$$\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} = (\langle \mathbf{s}', \mathbf{y}^{(\ell)} \rangle - \langle \Delta\mathbf{s}, \mathbf{y}^{(\ell)} \rangle,\ \langle \mathbf{u}', \mathbf{y}^{(\ell)} \rangle - \langle \Delta\mathbf{u}, \mathbf{y}^{(\ell)} \rangle,\ a^{(\ell)}_{i,0} z,\ r^{(\ell)}_0 \mathbf{y}^{(\ell)}[i])_{\mathbf{H}^*_i} .$$

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

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

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

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

$$p_i \cdot (H^{(1)}_i + \mu H^{(2)}_i - \mu \mathbf{r}_i),\ p_i \cdot \mathbf{r}_i$$

where $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_q^4$. 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 $[\![a]\!]_1$. Therefore the simulation can still be performed.
The ciphertexts component from **LoR**, which are queried only for honest $i \in \mathcal{H}$, can be written in $\mathbf{T}$ to see how they will be affected:

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

where the simulator can set $p_i \coloneqq 1/n$ at the **Setup** phase for all $i \in \mathcal{C} \cup \mathcal{H}$ and define

$$\epsilon \coloneqq \langle \Delta\mathbf{x}, \mathbf{1} \rangle = \sum_{i \in \mathcal{H}} \Delta\mathbf{x}[i] .$$

We note that ϵ can be known at the time the adversary send the challenge ciphertexts and $(p_i \cdot \Delta\mathbf{x}[i]/\epsilon)_{i \in \mathcal{H}}$ together with $(p_i)_{i \in \mathcal{C}}$ still satisfy

$$\sum_{i \in \mathcal{H}} p_i \cdot \frac{\Delta\mathbf{x}[i]}{\epsilon} + \sum_{i \in \mathcal{C}} p_i = 1$$

which is required for decryption. The change from p_i to $p_i \Delta\mathbf{x}[i]/\epsilon$ for all ciphertext returned from **LoR** is indistinguishable under the adversary's view because the encryption key ek_i is not revealed to the adversary. Under this basis change, the scalar τ is updated to

$$\tau' \coloneqq \tau + \frac{a(\omega'_{\mathsf{tag},\mathsf{R}} - \mu\omega_{\mathsf{tag},\mathsf{R}})}{n\epsilon}$$

and stays the same for all challenge ciphertext components $\mathbf{c}_{i,\mathsf{ipfe}}$ as desired because it does not depend on i. On the other hand, the ciphertexts component from **Enc** can be written in $\mathbf{T}$:

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

which retains their normal form required for the **Enc** oracle. We now consider the correction of the key components:

$$\begin{aligned}\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} &= (\langle \mathbf{s}', \mathbf{y}^{(\ell)}\rangle - \langle \Delta\mathbf{s}, \mathbf{y}\rangle,\ \langle \mathbf{u}', \mathbf{y}^{(\ell)}\rangle - \langle \Delta\mathbf{u}, \mathbf{y}^{(\ell)}\rangle,\ a^{(\ell)}_{i,0}z,\ r^{(\ell)}_0 \cdot \mathbf{y}^{(\ell)}[i] - b)_{\mathbf{H}^*_i} \\ &\quad + (-\mu c,\ c,\ 0,\ b)_{\mathbf{T}^*} \\ &= (\langle \mathbf{s}', \mathbf{y}^{(\ell)}\rangle - \langle \Delta\mathbf{s}, \mathbf{y}^{(\ell)}\rangle,\ \langle \mathbf{u}', \mathbf{y}^{(\ell)}\rangle - \langle \Delta\mathbf{u}, \mathbf{y}^{(\ell)}\rangle,\ a^{(\ell)}_{i,0}z,\ r^{(\ell)}_0 \cdot \mathbf{y}^{(\ell)}[i] - b)_{\mathbf{H}^*_i} \\ &\quad + (-\mu\rho,\ \rho,\ 0,\ b)_{\mathbf{H}^*_i} \\ &= (\langle \mathbf{s}', \mathbf{y}^{(\ell)}\rangle - \langle \Delta\mathbf{s}, \mathbf{y}^{(\ell)}\rangle - \mu\rho,\ \langle \mathbf{u}', \mathbf{y}^{(\ell)}\rangle - \langle \Delta\mathbf{u}, \mathbf{y}^{(\ell)}\rangle + \rho,\ a^{(\ell)}_{i,0}z,\ r^{(\ell)}_0\mathbf{y}^{(\ell)}[i])_{\mathbf{H}^*_i}\ .\end{aligned}$$

Notice that $(-b) \cdot \mathbf{h}^*_{i,4}$ can be computed using $[\![b]\!]_2$ and $H^{(4)}_i$. If $\rho = 0$ then we are not correcting the key components. Otherwise, if $\rho = \langle \Delta\mathbf{u}, \mathbf{y}^{(\ell)}\rangle$, using the property $\Delta\mathbf{s} + \mu\Delta\mathbf{u} = 0$, the vectors $\mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}}$ are corrected to the form they have in G_5. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_6) - \mathsf{Adv}(\mathsf{G}_5)| \leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2}(1^\lambda)$.

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

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

and the proof is concluded. □

B.5 Security Theorem for Section 5.4

Theorem 4. *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 secure against chosen-plaintext attacks 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:*

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

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 (Sketch). The main sequence of games is similar to that used in the proof of Theorem 3. The main difference is depicted in Figure 16 and Figure 17. The transition from G_0 to G_1 is similar to what we have done in the proof of Theorem 3. The sequence of games to go from G_1 to G_2 is given in Figure 18. Proceeding key by key, we again rely on Lemma 1 to mask the key components by another random labeling $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$, indexed by $i \in [n]$. There is a difference comparing with the proof of Theorem 3: we use different a new random labeling $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$ for each i,

meanwhile in the less flexible construction's proof, we can use the same new labeling during the parallel application of Lemma 1 for all i.

In contrast to the less flexible scheme in Section 5.2, the step to replace $a_{i,0}^{\prime(\ell)}$ 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 $(\mathsf{tag}, \mathsf{R}_i)$. 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 $\mathbb{A}(\mathsf{R}_i) \neq \mathbb{A}(\mathsf{R}_{i'})$, 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.

To go from G_1 to G_2, we use a sequence of hybrids indexed by $\ell \in [K]$ for the ℓ-th functional key. The transition from $\mathsf{G}_{1.\ell-1.0}$ to $\mathsf{G}_{1.\ell-1.1}$ is the parallel applications of Lemma 1. We recall the points that allow us to apply the lemma in parallel, similarly as in the proof of Theorem 3:

- 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 1 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 $1/z_j$, 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 $\mathbf{f}_5$. 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.
- 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.

In the proof of Theorem 3, the transition from $\mathsf{G}_{1.\ell-1.1}$ to $\mathsf{G}_{1.\ell-1.2}$ is a *statistical* transition because the same R is used for all ciphertext components of client i, which means for all $i \in [n]$ either the new labels added from Lemma 1 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. 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. 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 working matrices are:

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

The affected vectors are $\mathbf{h}_{i,4}, \mathbf{h}_{i,5}^*$ but they are hidden from the adversary. For all i, the key components can be written as follows:

$$\begin{aligned} \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} &= (\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle, \langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle, a_{i,0}^{(\ell)} z_\ell, a_{i,0}^{(\ell)} \cdot \mathbf{y}^{(\ell)}[i], d_{\mathbb{A},i}^{(\ell)} - b \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{H}_i^*} + (0, 0, 0, c \cdot \mathbf{y}^{(\ell)}[i], b \cdot \mathbf{y}^{(\ell)}[i])_{\mathbf{T}^*} \\ &= (\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle, \langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle, a_{i,0}^{(\ell)} z_\ell, (a_{i,0}^{(\ell)} + \rho) \cdot \mathbf{y}^{(\ell)}[i], d_{\mathbb{A},i}^{(\ell)})_{\mathbf{H}_i^*} \ . \end{aligned}$$

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:

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

The change in the 5-th component of $\mathbf{c}_{i,\mathsf{ipfe}}$ returned from **LoR**, which is needed to preserve their "decryptability" in case the key's policy is satisfied, is unrecognizable because $\theta\mathbf{h}_{i,5}$ is not revealed to the adversary, for all honest i queried to **LoR**. The value $\theta\Delta\mathbf{x}[i]$ is updated to $\theta'_i := (\theta + a\tau)\Delta\mathbf{x}[i]$. Now, there is a subtle point that we need to preserve the correction of the key so that it can decrypt the challenge ciphertext if all $\mathbb{A}(\mathsf{R}_i) = 1$. In other words, the sharing $(d^{(\ell)}_{\mathbb{A},i})_{i\in[n]}$ should still work with the new values $(\theta_i)_i$ in the challenge ciphertext from **LoR**. Actually the 5-th coordinate in $\mathbf{c}_{i,\mathsf{ipfe}}$ is different among different i from **LoR**, but if the correction is preserved w.r.t **LoR** then the simulation still succeeds as the ciphertexts from **Enc** are always kept normal. Moreover, to avoid the mix-and-match attack mentioned in Section 5.4, the sharing $(d^{(\ell)}_{\mathbb{A},i})_{i\in[n]}$ should be randomised for each ℓ. To resolve this, we can sample $\alpha \xleftarrow{\$} \mathbb{Z}_q$ and then define for $i \in [n]$:

$$d^{(\ell)}_{\mathbb{A},i} := \alpha\mathbf{y}^{(\ell)}[i] \ .$$

It can be verified that the new $(\theta_i \cdot d^{(\ell)}_{\mathbb{A},i})_{i\in[n]}$ is still a randomized n-out-of-n sharing of 0 if $\langle\Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle = 0$. When performing the product in DPVS, we have

$$\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} = \langle\mathbf{s}, \mathbf{y}^{(\ell)}\rangle\omega_{\mathsf{tag}}p_i + \langle\mathbf{u}, \mathbf{y}^{(\ell)}\rangle\omega'_{\mathsf{tag}}p_i + a^{(\ell)}_{i,0}z_\ell\psi_i + (a^{(\ell)}_{i,0} + \rho)\tau\Delta\mathbf{x}[i]\mathbf{y}^{(\ell)}[i] + (\theta + a\tau)\Delta\mathbf{x}[i] \cdot d^{(\ell)}_{\mathbb{A},i}$$

and summing over $i \in [n]$ will leads to the correct result for decryption, when $\mathbb{A}(\mathsf{R}_i) = 1$ for all i (which implies $\langle\Delta\mathbf{x}, \mathbf{y}^{(\ell)}\rangle = 0$).

If $\rho = 0$ we are in the previous hybrid $\mathsf{G}_{1.\ell-1.1}$, else we are in $\mathsf{G}_{1.\ell-1.2}$. 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. The transition from $\mathsf{G}_{1.\ell-1.2}$ to $\mathsf{G}_{1.\ell-1.3} = \mathsf{G}_{1.\ell}$ is another parallel application of Lemma 1 in order to "redo" the new labeling of $a'^{(\ell)}_{i,0}$ for all i. In the end, after maksing 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 3, using G_3 in Figure 16 and the games in Figure 17, 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 : $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $a^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a^{(\ell)}_{i,j})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a^{(\ell)}_{i,0}}(\mathbb{A})$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i} = 0$, $\theta \xleftarrow{\$} \mathbb{Z}_q$

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

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

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

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

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

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

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

Game G_2 : $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, $a'^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$, $\sum_{i=1}^n d^{(\ell)}_{\mathbb{A},i} = 0$, $\theta \xleftarrow{\$} \mathbb{Z}_q$

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

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

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

Fig. 16: Games $\mathsf{G}_0, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3$ for Theorem 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 R 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_3 we use a random function $\mathsf{RF} : \mathsf{Tag} \to (\mathbb{Z}_q^*)^2$.

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

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

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

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{R})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$p_i\omega_{\mathsf{tag}}$	$p_i\omega'_{\mathsf{tag}}$	ψ_i	$\tau\Delta\mathbf{x}[i]$	θ	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{R}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$p_i\chi_{\mathsf{tag}'}$	$\boxed{p_i\mu\chi_{\mathsf{tag}'}}$	ψ'_i	0	θ	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$	(	$\langle \mathbf{s}, \mathbf{y}^{(\ell)} \rangle$	$\langle \mathbf{u}, \mathbf{y}^{(\ell)} \rangle$	$a_{i,0}^{(\ell)} z_\ell$	$r_0^{(\ell)} \mathbf{y}^{(\ell)}[i]$	$d_{\mathbb{A},i}^{(\ell)}$	$)_{\mathbf{H}_i^*}$

Game G_5 : $\mu \xleftarrow{\$} \mathbb{Z}_q, \mathsf{H}(\mathsf{tag}, \mathsf{R}) := ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}', \mathsf{R}') := ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}'}]\!]_1)$, $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_q^4$

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

Game G_6 : $\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_q^4$, $\mu, v_{i,0}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $\mathsf{H}(\mathsf{tag}) := ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') := ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\mu\chi_{\mathsf{tag}'}]\!]_1)$. We also define $\mathbf{s}' = \mathbf{s} + \Delta\mathbf{s}, \mathbf{u}' = \mathbf{u} + \Delta\mathbf{u}$, where $\Delta\mathbf{s}, \Delta\mathbf{u} \in \mathbb{Z}_q^n$ s.t. $\Delta\mathbf{s} + \mu\Delta\mathbf{u} = 0$ and $\omega_{\mathsf{tag}} \cdot \Delta\mathbf{s} + \omega'_{\mathsf{tag}} \cdot \Delta\mathbf{u} = \Delta\mathbf{x}$

$$i \in \mathcal{C} \cup \mathcal{H}\ \mathsf{ek}_i\ \ (\boxed{s'_i}, \quad \boxed{u'_i}, \quad p_i \cdot (H_i^{(1)} + \mu H_i^{(2)} - \mu\mathbf{r}_i), \quad p_i \cdot \mathbf{r}_i, \quad \mathbf{h}_{i,3}, \quad \mathbf{h}_{i,5}, \quad \mathbf{f}_1, \quad \mathbf{f}_2, \quad \mathbf{f}_3)$$

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

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}_0^*[i], \mathbf{x}_1^*[i], \mathsf{tag}, \mathsf{R})$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$p_i\omega_{\mathsf{tag}}$	$p_i\omega'_{\mathsf{tag}}$	ψ_i	$\boxed{\tau'\Delta\mathbf{x}[i]}$	θ	$)_{\mathbf{H}_i}$
$\mathbf{Enc}(i \in \mathcal{C} \cup \mathcal{H}, x_i, \mathsf{tag}' \neq \mathsf{tag}, \mathsf{R}')$	$\mathbf{c}_{i,\mathsf{ipfe}}$	(	$p_i\chi_{\mathsf{tag}'}$	$p_i\mu\chi_{\mathsf{tag}'}$	ψ'_i	0	θ	$)_{\mathbf{H}_i}$
$\forall\, i \in \mathcal{C} \cup \mathcal{H}$	$\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$	(	$\langle \boxed{\mathbf{s}'}, \mathbf{y}^{(\ell)} \rangle$	$\langle \boxed{\mathbf{u}'}, \mathbf{y}^{(\ell)} \rangle$	$a_{i,0}^{(\ell)} z_\ell$	$r_0^{(\ell)} \mathbf{y}^{(\ell)}[i]$	$d_{\mathbb{A},i}^{(\ell)}$	$)_{\mathbf{H}_i^*}$

Fig. 17: Games $\mathsf{G}_4, \mathsf{G}_5, \mathsf{G}_6$ for Theorem 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 R for ciphertext components. The index ℓ runs in $\{1, \ldots, K\}$ for the functional key queries. In G_4 we use a random function $\mathsf{RF}' : \mathsf{Tag} \to \mathbb{Z}_q^*$.

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

Game $\mathsf{G}_{1.\ell-1.1}$: $\mathsf{H}(\mathsf{tag}) \to (\llbracket \omega_{\mathsf{tag}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag}} \rrbracket_1)$, $\mathsf{H}(\mathsf{tag}') \to (\llbracket \chi_{\mathsf{tag}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}'} \rrbracket_1)$, $a'^{(\ell)}_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'^{(\ell)}_{i,j})_j \leftarrow \Lambda_{a'^{(\ell)}_{i,0}}(\mathbb{A})$, $\Delta\mathbf{x} := \mathbf{x}^*_b[i] - \mathbf{x}^*_0[i]$

$\mathbf{LoR}(i \in \mathcal{H}, \mathbf{x}^*_0[i], \mathbf{x}^*_1[i], \mathsf{tag}, \mathsf{R})$	$\mathbf{c}_{i,j}$	(	$\sigma_{i,j} \cdot (1, -j)$	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	$)_{\mathbf{F}^*}$

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

Game $\mathsf{G}_{1.\ell-1.2}$: $\mathsf{H}(\mathsf{tag}) \to (\llbracket \omega_{\mathsf{tag}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag}} \rrbracket_1)$, $\mathsf{H}(\mathsf{tag}') \to (\llbracket \chi_{\mathsf{tag}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}'} \rrbracket_1)$

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

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

Game $\mathsf{G}_{1.\ell-1.3} = \mathsf{G}_{1.\ell}$: $\mathsf{H}(\mathsf{tag}) \to (\llbracket \omega_{\mathsf{tag}} \rrbracket_1, \llbracket \omega'_{\mathsf{tag}} \rrbracket_1)$, $\mathsf{H}(\mathsf{tag}') \to (\llbracket \chi_{\mathsf{tag}'} \rrbracket_1, \llbracket \chi'_{\mathsf{tag}'} \rrbracket_1)$

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

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

Fig. 18: 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 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 R 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.