Multi-Client Functional Encryption with Public Inputs and Strong Security

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

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

Abstract. Recent years have witnessed a significant development for functional encryption (FE) in the multi-user setting, particularly with multi-client functional encryption (MCFE). The challenge becomes more important when combined with access control, such as attribute-based encryption (ABE), which was actually not covered syntactically by the public-key FE nor semantically by the secret-key MCFE frameworks. On the other hand, as for complex primitives, many works have studied the admissibility of adversaries to ensure that the security model encompasses all real threats of attacks.

- At a conceptual level, by adding a public input to FE/MCFE, we cover many previous primitives, notably attribute-based function classes. Furthermore, with the strongest admissibility for inner-product functionality, our framework is quite versatile, as it encrypts multiple sub-vectors, allows repetitions and corruptions, and eventually also encompasses public-key FE and classical ABE, bridging the *private* setting of MCFE with the *public* setting of FE and ABE.
- Finally, we propose an MCFE with public inputs with the class of functions that combines inner-products (on private inputs) and attribute-based access-control (on public inputs) for LSSS policies. We achieve the first AB-MCFE for inner products with strong admissibility (from Nguyen *et al.*, ACNS'23) and with adaptive security.

In the end, our concrete MCFE leads to MIFE for inner products, public-key single-input inner-product FE with LSSS key-policy, and KP-ABE for LSSS, with adaptive security. Previous AB-MCFE constructions is either restricted in terms of weaker admissibility (Nguyen *et al.*, ASIACRYPT'22) or considers a slightly larger functionality of attribute-weighted sum but with only selective security (Agrawal *et al.*, CRYPTO'23).

1 Introduction and Motivation

Functional Encryption (FE). To overcome the *all-or-nothing* limitation of traditional encryption, Functional Encryption [16] has been introduced to allow the sender to control access to their encrypted data in a fine-grained manner through *functional decryption keys*. It was considered as a generalization of Attribute-Based Encryption (ABE) and Identity-Based Encryption (IBE), when the evaluated function is the identity function under some conditions. But as the inputs are all encrypted in the ciphertext (*e.g.* in [16, Definition 2]), this does not really cover ABE and IBE syntactically. We discuss more on this point in a subsequent paragraph *Final Syntactical Point: Public Inputs* in this section.

Multi-User Settings for FE. In practice, the number of useful functions may not be so large, and they can even be known in advance: Public Key Encryption (PKE) can then be transformed into FE by encrypting the evaluations of each function under different keys. But this covers the so-called *single-input* setting where one player knows the whole input at encryption time. Functional Encryption becomes more interesting in multi-user/input settings. Multi-Input Functional Encryption (MIFE) and Multi-Client Functional Encryption (MCFE) have thus been introduced in [25, 26], where the function evaluates on a list of inputs. In the former setting, a *single* user encrypts the various inputs at different times, while in the latter setting, *multiple* users (called *clients*) independently encrypt their inputs. Evaluation of the function performed on the joint-inputs, using a functional decryption key generated by a trusted authority. Another remark is that the public-key setting only makes sense for single-input FE. When considering multi-client or multi-input settings, because of possible combinations of the inputs, security requires secret-key encryption.

Since the introduction of MCFE/MIFE in [25, 26] and later in [18], growing interests can be found in recent studies on enriching the concrete function class that can be realized MCFE/MIFE or their generalizations, including inner products (over one inner produc of scalars, or sum of inner products of vectors at each ciphertext) [19, 2, 1, 29, 20, 40, 32, 34], quadratic functions [6, 9], or attribute-weighted sums [11, 30], and more. Along the way, other works also investigate the questions of security notions for MCFE/MIFE, such as looking into function privacy [7, 40, 30, 34] or the *admissibility* of the adversary [32].

A Reminder on Admissibility. To begin with, one important difference between MCFE and MIFE [25, 26, 18] concerns the entity who encrypts the data. In MIFE the encryptor is unique, and the adversary cannot corrupt the encryptor. This unique encryptor is free to *repeatedly* encrypt new messages at the same slot i and combine the resultant ciphertexts with other ciphertexts. This repeated encryption is called *repetitions* in the literature and the general combination of ciphertexts in MIFE clearly demonstrates a *mix-and-match* nature among these ciphertext repetitions, thus leaking a large amount of information on the plaintexts. Meanwhile, in MCFE, the encryptors consist of multiple clients, and the adversary can corrupt some of them to get the clients' secret encryption keys. This ensues the introduction of *tags* at the time of encyrption in MCFE, which is defined in [18], so that a decryption key dk_F can decrypt jointly ciphertexts $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{ek}_i, x_i, \mathsf{tag})$ if the same tag are used for all clients i. The usage of tags provide a finer mecanism to control the combination of ciphertexts, as only ciphertexts with the same tag can be combined in the function evaluation. The problem of repetitions is, however, not addressed in the original definition of MCFE [18], whose underlying reason is that it is up to the resposibility of client i to avoid encrypting repeatedly their data under the same i, tag. In practice, there is potentially scenario where repetitions under the same (i, tag) happen due to the client's misbehavior or the adversary's corruption. Therefore, recent works [11, 32] have considered the repetitions in the MCFE setting, in the security model as well as resolutions towards concrete constructions.

Similar to the quest of defining security for public-key advanced encryption notions such as IBE/ABE/FE, the central principle that should be captured by a security definition for MCFE/MIFE is that no (feasible) adversary can learn more that what is allowed by the function evaluation, even with the corruption of some clients in MCFE, and even with ciphertext repetitions at some slot/client i. Putting it differently, it is necessary to include in the security definition some constraints that exclude trivial attacks, *e.g.* by requiring that the adversary cannot choose the challenge ciphertexts $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$ in such a way that they can trivially win the game by evaluating the function $F(\mathbf{x}^{(0)}) \neq F(\mathbf{x}^{(1)})$ on chosen messages. For example, back to the simpler identity function of IBE as a special case of FE, the decryption key should not allow to decrypt the challenge identity-based ciphertext. The constraints that exclude trivial attacks are usually referred to as *admissibility* conditions. With respect to the function class to compute inner products, where the function evaluation computes the inner products, the seminal work of [18] put forth an admissiblity condition that then becomes widely used in other works on MCFE [2, 1, 29, 20, 7]. Until very recently, this first-generation admissibility condition was revisited in [32] that proposes a stronger and optimal notion of admissibility for MCFE, still within the class to compute inner products. We review the progress below.

From Secret-key MCFE to Public-key FE. We now focus on a viewpoint that is more on the multi-client setting. Following the first formalization in [18], many follow-up studies on MCFE, for instance [2, 1, 29, 20, 7], set down an *admissibility condition* in order to exclude trivial attacks: for any corrupted client i and challenge message-pair $(x_i^{(0)}, x_i^{(1)})$ for i, it requires that $x_i^{(0)} = x_i^{(1)}$. This is indeed the right condition if the secret-key encryption is *deterministic*, which was considered on the first period of development of MCFE, as with the corruption of the encryption key ek_i, the adversary could re-encrypt $x_i^{(0)}$ and compare with the challenge ciphertext. However, if the encryption is *probabilistic*, this condition is not well justified and appears too restrictive.

Indeed, we observe that it is this condition where for all $i \in \mathcal{C}$ all challenge pairs $x_i^{(0)} = x_i^{(1)}$ that prevents to go from the *secret-key* MCFE to the *public-key* FE. To obtain a public-key FE from a secret-key MCFE, the natural approach is to instantiate the MCFE with $n = 1$ client, then to publish the only client's encryption key ek as the public key. Under the early admissibility condition *as per* [18] of the underlying MCFE, in order to base the security of the public-key FE on the security of the MCFE, the only queries that the reduction can forward to its MCFE challenger are the *trivial* one from the FE adversary where $x^{(0)} = x^{(1)}$, and this is far weaker than the standard CPA-security of public-key FE. It is now clear that a less restrictive notion of admissibility, equivalently a stronger notion of security, is needed to capture the security of public-key FE from the security of MCFE. The ensued question is then: *how strong should the admissibility of secret-key MCFE be to capture the security of public-key FE?*

Our first contribution is to show that MCFE with the strong admissibility notion [32] also covers public-key single-input FE. Secondly, defining FE and MCFE with public inputs, we additionally cover ABE and IBE, where the attributes and identities can be public. Last but far from the least, adding public inputs to MCFE is complementary to their existing advantages, notably towards the conversion from MCFE to MIFE: guaranteeing security against *repetitions on private inputs* is sufficient for our notion of MCFE to imply MIFE. The following paragraph elaborates more about this relationship.

On the reconciliation between MCFE and MIFE. At first glance, MIFE appears to be just MCFE with a constant label. However, the distinction is more significant because in MIFE, there is only one encryptor, while in MCFE, there are multiple encryptors (clients). Therefore, whereas there is no corruption of users in MIFE, dealing with corruptions in MCFE is a main concern. In summary, recent works [31, 32, 9, 11] agree on two principal advantages of MCFE over MIFE:

- with a label associated to each encrypted input, one can limit the combinations of the inputs for each evaluation
- as inputs can be encrypted by different clients, multiple independent secrets are involved, for each client, then one can deal with corruption of individual keys in MCFE, whereas in MIFE there is a unique encryptor and no corruption can be allowed.

At this point, it seems that MCFE is strictly stronger than MIFE. However, again, the situation is more complicated because, as pointed out by [19], in the original definition of MCFE [25, 18], the clients were assumed not to encrypt two messages under the same label (namely [25, 18] do not allow repetitions). Under this restriction, one *cannot* turn a MCFE to a MIFE. In short, MIFE, when augmented with labels, can be seen as an MCFE *with* repetitions but *without* corruption. This connection is studied in preceding works [2, 1] and recently confirmed in [11].

But the story is not at the end yet, especially when one wants to combine MCFE and MIFE with other functionalities, such as attribute-based access-control [31, 11], where the conversion MCFE to MIFE is highly non-trivial as mentioned in [11]. As a final remark, our context of multi-client/multi-input setting for FE with access control is different from the setting of *multi-authority* ABE, *e.g.* as studied in [22], where in our case there is always only *one* authority generating the functional decryption keys.

Final Syntactical Point: Public Inputs. When reviewing the existing initial definitions of FE [16], MCFE [18], and MIFE [25], we observe that the syntax of encryption in these definitions themselves *a priori* does not allow parts of the plaintext to be public. When denoting encryption keys ek_i (in the secret-key MCFE/MIFE setting) or public key pk (in the public-key single-client FE), specifically the MIFE syntax in [25, Section 2.1] is written $c \leftarrow \mathsf{Enc}(\mathsf{ek}_i, x)$ given the i-th plaintext x, the MCFE syntax in [18, Definition 1] is written $c \leftarrow \mathsf{Enc}(\mathsf{ek}_i, x, \mathsf{tag})$ given the i-th plaintext x and the tag tag, and the FE syntax in [16, Definition 2] is written $c \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$ given the plaintext x.

First of all, having the encryption as they are listed above, the IND-CPA security alone implies that no partial information about the plaintext is leaked. This applies to the case $x = (m, \mathsf{S})$ where m is the contents of the message and S is some attribute/index in the context of KP-ABE or IBE. As such, without further specifications, how to derive *non-attribute/index-hiding* KP-ABE/IBE from the existing definitions of FE, MCFE, and MIFE is not clear. It then necessarily requires more properties on the *function class* so as to capture the non-attribute-hiding property. We emphasize that this is also the approach that was taken in [16], where the authors introduced the notion of *empty key* that defines a function such that "anyone can [...] obtain all the information about x that intentionally leaks from c", *cf.* [16, Page 3]. A similar notion to this empty-key function is indeed what we need to capture the non-attribute-hiding property when expressing KP-ABE/IBE in the syntax of FE. With respect to [16], when describing how to capture KP-ABE or Ciphertext-Policy ABE [16, Page 5], the empty-key function however is not made clear in the key space of all poly-sized boolean formula in the former, nor in the key space of all poly-long bitstrings of variables in the latter. In the multi-user setting of MCFE/MIFE, no such property of empty-key function is mentioned in the introduced definitions [25, 18].

1.1 This Paper

A Simple Extension of MCFE to also Cover MIFE, public-key FE and ABE. From the above discussion, a crucial question arises:

> *How can we extend MCFE, syntactically and security-wise via admissibility, in a minimal way to encompass all the settings, from MIFE to public-key FE and ABE?*

Thus appears the motivation behind proposing MCFE with public inputs, wherein we simply augment the ciphertexts of the MCFE with a public inputs. This part of public inputs is taken care by the function evaluation. However, in order to cover public-key FE, we need to consider the stronger notion of admissibility with the possibility to handle sub-vectors.

Conceptual Contribution. In a nutshell, we propose a simple extension of MCFE with not only private but also public data to be input to the function evaluation. Combining with the consideration of a stronger admissibility for adversary and sub-vectors in encryption, we cover previous primitives such as MCFE, MIFE, public-key single-input FE and ABE. When only private inputs are considered, if the function involves attributes for access-control, this is necessarily with the attribute-hiding property, while this is not always required. Hence, we will show this is quite relevant for MCFE and MIFE with attribute-based access-control. We also describe, and achieve, a very high security notion that, while considering the multi-client setting with secret-key encryption, also covers public-key attribute-based encryption.

Strong Admissibility and Public-Key Setting. Recently, a stronger and optimal notion of the admissibility of an attack was introduced [32]. Intuitively, to recall, when there is a unique client, with the initial admissibility from [18, 2, 1, 29, 20, 7], when the encryption key of this single user is corrupted, the only queries that the reduction can forward are the *trivial* one from the FE adversary where $x^{(0)} = x^{(1)}$, hence it is not sufficient to capture the reduction from MCFE to FE with meaningful CPA-security. With strong admissibility from [32], *i.e. without* the requirement that $x_i^{(0)} = x_i^{(1)}$ for corrupted $i \in \mathcal{C}$, we show that the reduction can capture the security of the public-key FE by making public the encryption key. In particular, within the function class to compute inner products, addressing strong admissibility also necessitates the ability to manage sub-vectors in encryption. That is, each client i encrypts a vector $\mathbf{x}_i$, and the function evaluation returns the sum of inner products $\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{y}_i \rangle$ with respect to a function associated to $(\mathbf{y}_i)_i$. This is a technically relevant issue because under this stronger admissibility, moving to public-key FE gives the usual functionality of inner products, when $n = 1$, and not just scalar products.

Our work extends the work from [32], and we will develop more the conceptual implications in the next paragraph, as well as the concrete case in **Constructive Contributions** below. A discussion on our strong admissiblity is given in paragraph **Discussion on admissibility** after the formal definition in Definition 5. The aforementioned implications can be summarized with the following simplified diagram (more details are given in Theorem 8), where

$$\begin{array}{ccc} \mathsf{MCFE}^{\mathsf{rep\text{-}priv}} & \longrightarrow & \mathsf{MIFE} \\ \downarrow & & \\ \mathsf{FE}^{\mathsf{w\text{-}pub}} & \longrightarrow & \mathsf{KP\text{-}ABE} \end{array}$$

- $\mathsf{MCFE}^{\mathsf{rep\text{-}priv}}$ is our new notion of MCFE, with strong admissibility and public inputs, but repetitions are only allowed on the private inputs (multiple encryption queries with the same tag must be with the same public input);
- MIFE is the usual definition, with private inputs only, with repetitions, without tags nor corruptions. The implication comes from the allowed *repetitions on private inputs* in our MCFE;
- $\mathsf{FE}^{\mathsf{w\text{-}pub}}$ is the classical public-key single-input FE definition enhanced with *public inputs*. Implication comes from the *strong-admissibility* that allows to deal with public-key encryption when there is a unique client;
- KP-ABE denotes the usual definition of key-policy ABE. The implication comes from the *public inputs* in $\mathsf{FE}^{\mathsf{w\text{-}pub}}$, that can be used to encode the attributes in a non-hiding way.

It is very interesting that MCFE with the *strong admissibility* from [32] leads to public-key single-input FE, when there is a unique client, and even to Key-Policy ABE when allowing public inputs (to provide the attributes in the ciphertext).

Constructive Contributions. These implications depend on the actual classes of functions. As a constructive result, we propose an MCFE with the class of functions that combines inner-products (on private inputs) and attribute-based access control (on public inputs) for LSSS policies. It achieves the strong admissibility notion, in the adaptive setting (whereas [11] only provides selective security), with repetitions on the private inputs and static corruptions. It also deals with sub-vectors (whereas [31, 32] only consider scalars). As a consequence, removing the tags, the corruptions and the public inputs, we obtain an MIFE for inner products, with strong admissibility and adaptive security; limiting to one client, one gets public-key single-input inner-product FE and KP-ABE for LSSS, with adaptive security. Our construction uses pairings in the ROM, and we note that there exists other approaches to tackle IPFE with access control using lattices, *e.g.* [27, 38], though they are only single-client to our knowledge.

We would like to emphasize that strong admissibility is not only theoretical (as it allows us to cover public-key single-input inner-product FE) but also more intuitive: the only restriction we impose on the adversary is to prevent them from choosing challenge messages in such a way that, with their corrupted keys and the function evaluation, they *cannot trivially win* the game by evaluating the function on chosen messages. Requiring the adversary to use the same message for corrupted users as in the previous admissibility now seems somewhat artificial to us. Achieving strong admissibility is also more challenging as it requires the encryption to be probabilistic and any deterministic encryption cannot meet strong admissibity as we already explained. Consequently, the only two existing AB-IP-MCFE schemes [31, 11] are not secure when considering strong admissibility as the encryptions in these schemes are deterministic. Of course, we do not claim to break the schemes [31, 11], because we consider a stronger security level. We would propose that strong admissibility should be considered in the multi-user setting of FE.

In summary, we propose the first AB-IP-MCFE with strong admissibility and with adaptive security for inner-product functionality while [11] considers a slightly larger functionality of average weighted sum but with selective security on the challenge messages. In term of efficiency,

Scheme	$\mathcal{F}$	(priv,pub)-inp	Eval	Security
[11, Sect. 5]	$\mathcal{F}^{\mathsf{ABP}}_{n,(N_i)_i}$	$((\mathbf{z}_{i,j})_j, (\mathbf{x}_{i,j})_j)$	$\sum_{i\in[n]}\sum_{j\in[N_i]}\langle f_i(\mathbf{x}_{i,j}), \mathbf{z}_{i,j}\rangle$	sel, <u>w-rep</u>, stat, wk-adm
[31, Sect. 5.4]	$\mathcal{F}^{\mathsf{IP,B}}_{n,q} \times \mathsf{LSSS}$	$(\mathbf{x}[i], \mathsf{S}_i)$	$(\bigwedge_i \mathsf{LSSS}(\mathsf{S}_i)) \cdot \langle \mathbf{x}, \mathbf{y}\rangle$	<u>ad</u>, no-rep, <u>dyn</u>, wk-adm
[32, Sect. 6]	$\mathcal{F}^{\mathsf{IP,B}}_{n,q}$	$(\mathbf{x}[i], \perp)$	$\langle \mathbf{x}, \mathbf{y}\rangle$	<u>ad</u>, <u>w-rep</u>, stat, <u>s-adm</u>
Corol. 13	$\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$	$(\mathbf{x}_i, \mathsf{S}_i)$	$(\bigwedge_i \mathsf{LSSS}(\mathsf{S}_i)) \cdot \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i\rangle$	<u>ad</u>, rep-priv‡, stat, <u>s-adm</u>

‡ This intermediate notion only provides security against repetitions on private inputs, and not on public inputs.

Table 1: We compare our MCFE with existing MCFE, casting the function class into the syntax with *public inputs*. The MCFE in [11, Sect. 5] is defined for the function class of *attribute-weighted sums* $\mathcal{F}^{\mathsf{ABP}}_{n,(N_i)_i}$ containing $(f_i)_{i\in[n]}$ of *arithmetic branching programs*. that evaluates on public inputs $(\mathbf{x}_{i,j})_j \in (\mathbb{Z}_q^{N_0})^{N_i}$ and private inputs $(\mathbf{z}_{i,j})_j \in (\mathbb{Z}_q^{N_1})^{N_i}$, where N_i is some parameter for slot i, and $N_0, N_1 \in \mathbb{N}$. The MCFE schemes in [31, Sect. 5.4] are defined for the functionality class $\mathcal{F}^{\mathsf{IP,B}}_{n,q} \times \mathsf{LSSS}$, that evaluates on public attributes S_i and private inputs $\mathbf{x}[i] \in \mathbb{Z}_q$, where $n, q \in \mathbb{N}$, q is prime, $\max_i(|\mathbf{x}[i]|) < B$, and $B = \mathsf{poly}(\lambda) \in \mathbb{N}$ is a polynomial; Meanwhile MCFE from [32, Sect. 6] is implied by the DMCFE therein and is for $\mathcal{F}^{\mathsf{IP,B}}_{n,q}$ without access control. Sect. 4.2 extends MCFE to the class $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$ and $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}}$ that evaluates on public attributes S_i and private inputs $\mathbf{x}_i \in (\mathbb{Z}_q)^{N_i}$ where for all i, $\max(\|\mathbf{x}_i\|_\infty, \|\mathbf{y}_i\|_\infty) < B$, where $B = \mathsf{poly}(\lambda) \in \mathbb{N}$ is a polynomial. We also use the shorthand (no-rep, w-rep) to indicate whether *repetitions* at positions under a challenge tag are allowed, and (sel, ad) to indicate whether *adaptive* challenge ciphertext are allowed. The shorthand (stat, dyn) indicates whether the corruption is *static* or *dynamic*. Finally, (wk-adm, s-adm) indicates whether the *weak* [18] or *strong* admissibility [32] is considered. The preferred properties are <u>underlined</u>.

we have the same asymptotic efficiency as [11]: each client sends a ciphertext of linear size in the size of its subvector message, independent of the total number of clients. In Table 1, we compare our construction with existing works.

Relation with Multi-Party Functional Encryption. Our MCFE with Public Input can be seen as a special case of Multi-Party Functional Encryption (MPFE) [7]. However, our goal is not to define yet another new and more general primitive, but only to add the minimal extension to an existing well-studied primitive to reconcile with other primitives. By simply considering public inputs for MCFE with a stronger admissibility notion, we cover not only attribute-based access control, but also public-key single-input FE. While MPFE is very general, it only considers the secret-key encryption setting and does not cover public-key single-input FE. Up to the notions of MCFE, our results complete the picture of unifying MCFE/MIFE/FE/ABE, by considering the public inputs and the strong admissibility notion. The strong admissibility is necessary following our discussion in the paragraph *From Secret-key MCFE to Public-key FE* above, in order to capture the security of public-key FE from the security of MCFE, and is then proven sufficient in our Theorem 8. The public inputs are necessary to capture the non-attribute-hiding property of KP-ABE/IBE in the syntax of FE, as discussed in the paragraph *Final Syntactical Point: Public Inputs* on page 3 above, inherits the same spirit of empty-key function in [16], and is cleanly demonstrated in our Theorem 8. Finally, our concret final AB-IP-MCFE in Corollary 13 is the first to achieve adaptive security for inner-product functionality in the multi-client setting, with public inputs, and with strong admissibility.

1.2 Technical Overview

Given the above conceptual overview, we now highlight the technical points for our concrete construction of MCFE to compute inner products under access control in Section 4.2. All relevant points that relate to the security proof are given by a dedicated overview in paragraph **Proof Strategy** before the proof of Theorem 12. The functionality of interests is $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$ and $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}}$ contains $F_{\mathbf{y}_1,\ldots,\mathbf{y}_n} : \prod_{i\in[n]}(\mathbb{Z}_q^{N_i}) \to \mathbb{Z}_q$ that is defined as $F_{\mathbf{y}_1,\ldots,\mathbf{y}_n}(\mathbf{x}_1,\ldots,\mathbf{x}_n) := \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i\rangle$, where for all i, $\max(\|\mathbf{x}_i\|_\infty, \|\mathbf{y}_i\|_\infty) < B$, where $B = \mathsf{poly}(\lambda) \in \mathbb{N}$ is a polynomial. For the ease of notation, we can assume the subvectors are of length $N = \max_i(N_i)$. The access control is given by $\mathsf{Rel} : \mathsf{LSSS} \times (\prod_{i=1}^n 2^{\mathsf{Att}}) \to \{0,1\}$, where $\mathsf{Rel}(\mathbb{A}, (\mathsf{S}_i)_i) = \prod_i \mathbb{A}(\mathsf{S}_i)$, the class LSSS contains Linear Secret Sharing Schemes over Att, and 2^{Att} denotes the superset of an attribute space $\mathsf{Att} \subseteq \mathbb{Z}_q$.

First Technical Obstacle: Admissibility with vectors and probabilistic encryption. Our goal is to handle the less restrictive admissibility condition w.r.t the function calculating $F_{\mathbf{y}_1,\ldots,\mathbf{y}_n}(\mathbf{x}_1,\ldots,\mathbf{x}_n) := \sum_{i=1}^{n}\langle \mathbf{x}_i, \mathbf{y}_i\rangle$, under access control from Rel. Each of the n clients in our MCFE scheme are encrypting a vector $\mathbf{x}_i$, together with a tag tag and their set of attributes S_i. The fact that we are working with vectors is generalizing first and foremost the construction by Nguyen *et al.* [31] that only supports scalar inputs $x_i \in \mathbb{Z}_q$. Moreover, under the new admissibility that is studied in [32], conditions for the challenge ciphertexts in terms of corrupted clients i become less restrictive. To recall, the admissibility in [31] is inherited from the orginal one introduced in [18] and will require that for any corrupted $i \in \mathcal{C}$, it holds that $x_i^{(0)} = x_i^{(1)}$. Following the motivation that is put forth in [32] so as to relax the foregoing condition, in the case of scalars where inputs to clients i have dimension 1, the *stronger* admissibility condition is that for any corrupted $i \in \mathcal{C}$, for any key queries with y_i as the i-th parameter for inner products, it must hold $(x_i^{(0)} - x_i^{(1)}) \cdot y_i = 0$. In our case, having the goal of generalizing [31] to encrypt vectors $\mathbf{x}_i$ by client i under the *stronger* admissibility, the condition becomes: for any corrupted $i \in \mathcal{C}$, for any key queries with $\mathbf{y}_i$ as the i-th parameter for inner products, it must hold $\langle \mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)}, \mathbf{y}_i\rangle = 0$. This opens up much more liberty to the adversary in terms of what they can challenge. That is, as soon as the dimension of the vectors $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}, \mathbf{y}_i)$ is at least 2, the adversary can choose $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)})$ such that $\mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)}$ is orthogonal to $\mathbf{y}_i$, where both $\mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)} \neq \mathbf{0}, \mathbf{y}_i \neq \mathbf{0}$. In retrospective, the scalar version of [32] implies already that either $(x_i^{(0)} - x_i^{(1)}) = 0$ or $y_i = 0$, which is a special case of the vector version. Last but not least, regarding honest clients $i \in \mathcal{H}$ where $\mathcal{H} := [n] \setminus \mathcal{C}$, it must hold that for all key queries with $(\mathbf{y}_i)_{i\in\mathcal{H}}$ as the parameters corresponding to honest slots, $\sum_{i\in\mathcal{H}}\langle \mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)}, \mathbf{y}_i\rangle = 0$. Particularly, the condition $\langle \mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)}, \mathbf{y}_i\rangle = 0$ for any $i \in \mathcal{C}$ and any $(i, \mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}, \mathsf{tag})$ to $\mathbf{LoR}$ already implies that encryption of our MCFE must be necessarily probabilistic, because the adversary is allowed to makes challenge queries $\mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)} \neq \mathbf{0}$. This is highlighted in paragraph *Strong Admissibility and Public-Key Setting* (on page 4) of our introduction.

Solution to the First Obstacle: Probabilistic Vectorization of the Scheme of [31]- Overview of Our Scheme. Our starting point is the scalar construction of [31], in the bilinear 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 crux of our vectorization is to use the *dual pairing vector spaces* (DPVSes) to encode the vectors $\mathbf{x}_i$ and $\mathbf{y}_i$. In particular, each client encrypt their vector $\mathbf{x}_i$ by $\mathbf{c}$-vectors in $\mathbb{G}_1$, and the functional key for $\mathbf{y}_i$ is encoded in $\mathbf{k}^*$-vectors in $\mathbb{G}_2$. The importance is *fresh* randomness must be added to the $\mathbf{c}$-vectors *independently* by each i, which cannot be founded on RO or *pseudorandom functions* as in previous works [31, 11]. To implement such randomness and ensure that *correctness* is preserved, we make use of the concrete fact of DPVS that it provides linear combinations of vectors in $\mathbb{G}_1$ and $\mathbb{G}_2$. This can be verified when viewing a DPVS as a $\mathbb{Z}_q$-algebra, sastisfying $\mathbb{Z}_q$-linearity and being equipped with an product operation that is provided by the bilinear map $\mathbf{e}(\cdot,\cdot)$. Principal ideas are given below:

1. In order to deal with the fact that we need to embed fresh randomness into *each* i-th component of the ciphertext, while maintaining the *correctness* in regards to the access control, our main idea is to distribute a random mask $[\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_\mathsf{t}$, where ν_i is the randomness for the i-th client, and $a_{i,0}$ is the i-th secret to be shared in the key component by the LSSS.
   - Each ν_i is embedded into attribute-indexed ciphertext components (see $\widetilde{\mathbf{t}}_{i,j}$ in the below), each $a_{i,0}$ is shared at the time of key extraction into the key components (see $\widetilde{\mathbf{m}}_{i,j}$ in the below).
   - Firstly, $\mathbb{Z}_q$-linearity of the DPVS allows regrouping the randomness ν_i into $\sum_i \nu_i$ in aggregated attribute-indexed ciphertext components (see $\widetilde{\mathbf{t}}_{0,j}$ in the below), then LSSS-reconstruction allows obtaining $[\![(\sum_i \nu_i) \cdot a_{i,0}]\!]_\mathsf{t}$ for each i.
   - Relying on $\mathbb{Z}_q$-linearity again, while noting that $\sum_i \nu_i$ is independent from i, permits obtaining the mask $[\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_\mathsf{t}$ by summing over i the terms $[\![(\sum_i \nu_i) \cdot a_{i,0}]\!]_\mathsf{t}$.

- Because the subvectors $\mathbf{x}_i$ is of some length $N \in \mathbb{N}$, the functional parts $\mathbf{c}_{i,\mathsf{ipfe}}$ in i-th ciphertext components as well as the key components $\mathbf{k}_{i,\mathsf{ipfe}}$ contain another layer of masking over coordinates of the subvector. Thanks to the fact the key extraction in MCFE is centralized, this layer of masking is done by secret coefficients $(\theta_{i,k})_{k\in[N]}$ that are given in encryption key ek_i, and fresh random shares $d_{\mathbb{A},i,k}$ such that $\sum_{i=1}^{n}\sum_{k=1}^{N}\theta_{i,k}d_{\mathbb{A},i,k} = 0$ is generated at the time of key extraction.

We now describe the form of our ciphertexts and key components in items 2 and 3. The complexity of the decryption algorithm is explained in item 4.

2. The encryption by each client i is defined by $\mathsf{Enc}(\mathsf{ek}_i, \mathbf{x}_i \in \mathbb{Z}_q^N, z_i := (\mathsf{tag}, \mathsf{S}_i))$ where $\mathsf{S}_i \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ is the set of attributes to be used. The tags is treated by hashing $\mathsf{H}(\mathsf{tag}) \to ([\![\omega]\!]_1, [\![\omega']\!]_1) \in \mathbb{G}_1^2$. For the access control part, each attribute $j \in \mathsf{S}_i$ gives rise to components $\mathbf{c}_{i,j}$ as follows, where $\psi_i, \nu_i, \widetilde{\sigma}_{i,j}, \sigma_{i,j} \xleftarrow{\$} \mathbb{Z}_q$ are randomness

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

For the inner product computation, $\mathbf{c}_{i,\mathsf{ipfe}}$ serves as parts for unmasking later on:

$$\mathbf{c}_{i,\mathsf{ipfe}} = (\underbrace{\omega p_i,\ \omega' p_i}_{\text{functional coord.'s}},\ \psi_i,\ 0^N,\ \underbrace{(\theta_{i,k})_{k=1}^{N}}_{\text{masks for subvectors of } i \text{ (length } N)},\ 0)_{\mathbf{H}_i}$$

where $\theta_{i,k} \xleftarrow{\$} \mathbb{Z}_q$. The main masking components for $\mathbf{x}_i$ are $\mathbf{t}_i$:

$$\mathbf{t}_i = (\omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \mathbf{x}_i,\ \overbrace{\nu_i}^{\text{specific randomness for each } i},\ 0, \rho_i)_{\mathbf{B}_i};$$

The new *probabilisitic vectorization* is implemented via

$$\widetilde{\mathbf{t}}_{i,j} = (\overbrace{\widetilde{\sigma}_{i,j}\cdot(1,-j)}^{\text{index by } j},\ \underbrace{\nu_i}_{\text{specific randomness for } i},\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{G}}$$

The ciphertext of client i contains $\mathsf{ct}_{\mathsf{tag},i} := \left(\left(\mathbf{c}_{i,j}, \widetilde{\mathbf{t}}_{i,j}\right)_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}}\right)$.

3. The key extraction of the MCFE is defined to be $\mathsf{Extract}(\mathsf{msk}, (\mathbf{y}_i)_{i\in[n]} \in \prod_{i=1}^{n}\mathbb{Z}_q^N, \mathsf{ac\text{-}k} := \mathbb{A})$ where $\mathbb{A}$ is an LSSS-realizable monotone access structure 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 $\mathbb{A}$, with possible repetitions. The function is specified by n vectors $\mathbf{y}_1, \ldots, \mathbf{y}_n$ in $\mathbb{Z}_q^N$. First of all, regarding the access control, we generate a random secret $a_{i,0}$ and share it into the labels $(a_{i,j})_j$ where j runs over the attributes in Att (see Definition 8).

The shares $(a_{i,j})_j$ as well as the secret $a_{i,0}$ are embedded into, where $z, \pi_{i,j}, \mathsf{rnd}_i, \mathsf{rnd}_{i,\mathsf{ipfe}} \xleftarrow{\$} \mathbb{Z}_q$,

$$\mathbf{k}_{i,j} := \left(\overbrace{\pi_{i,j}\cdot(j,1)}^{\text{index vectors by } j},\ a_{i,j}\cdot z,\ 0^N,\ 0,\ 0^N,\ 0,\ 0\right)_{\mathbf{F}^*} \quad \text{for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$

$$\mathbf{k}_{i,\mathsf{ipfe}} := \left(\underbrace{\sum_{i=1}^{n}\langle\mathbf{s}_i,\mathbf{y}_i\rangle, \sum_{i=1}^{n}\langle\mathbf{u}_i,\mathbf{y}_i\rangle}_{\text{functional coord's}},\ a_{i,0}\cdot z,\ 0^N,\ \underbrace{(d_{\mathbb{A},i,k})_{k=1}^{N}}_{\text{masks for subvectors of } i \text{ (length } N)},\ \overbrace{\mathsf{rnd}_{i,\mathsf{ipfe}}}^{\text{randomness specific to } i}\right)_{\mathbf{H}_i^*}.$$

It is important to note that different client i will correspond to a different ensemble of $\mathbf{k}_{i,\mathsf{ipfe}}, (\mathbf{k}_{i,j})_j$. The "functional coord.'s" are sums of inner products $\sum_{i=1}^{n}\langle\mathbf{s}_i,\mathbf{y}_i\rangle, \sum_{i=1}^{n}\langle\mathbf{u}_i,\mathbf{y}_i\rangle$ of the $\mathbf{y}_i$'s with the master secret keys $\mathbf{s}_i, \mathbf{u}_i$ that will later intervenes when decrypting the ciphertexts (following the same blueprint by Agrawal-Libert-Stehlé [10] for adaptive secure IPFE). Last but not least, for each $i \in [n]$, each $k \in [N]$, the masking at level of each client's subvector (length N) is done by $d_{\mathbb{A},i,k} \xleftarrow{\$} \mathbb{Z}_q$ such that $\sum_{i=1}^{n}\sum_{k=1}^{N}\theta_{i,k}d_{\mathbb{A},i,k} = 0$. Finally, the

new technical idea that differs from [31] reflected in the key extraction by For each $i \in [n]$, compute

$$\mathbf{m}_i := \left(\mathbf{y}_i, \sum_{i=1}^{n} a_{i,0}, \mathsf{rnd}_i, 0, 0\right)_{\mathbf{B}_i^*} ; \widetilde{\mathbf{m}}_{i,j} := (\tilde{\pi}_{i,j} \cdot (j,1),\ a_{i,j},\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{G}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$

where $\tilde{\pi}_{i,j}, \mathsf{rnd}_i \xleftarrow{\$} \mathbb{Z}_q$. Then $\mathsf{dk}_{\mathbb{A},\mathbf{y}} := \left((\mathbf{k}_{i,j}, \widetilde{\mathbf{m}}_{i,j})_{i,j}, (\mathbf{m}_i, \mathbf{k}_{i,\mathsf{ipfe}})_{i\in[n]}\right)$ is output as the functional key.

4. For *correctness*, Algorithm 2 shows how the decryption is done. In the follwing we suppose an LSSS-reconstruction set A_i is used that gives coefficients $(c_{i,j})_j$ for each i.
   - (See step 1 in Algorithm 2) More specficicially, the treat of probabilisitic vectorization starts by regrouping the $\widetilde{\mathbf{t}}_{i,j}$'s into $\widetilde{t}_{0,j}$ that homomorphically groups specific randomness over i into $\sum_i \widetilde{\sigma}_{i,j}$ for each attribute j. At the same time, $\widetilde{t}_{0,j}$ also holds $\sum_i \nu_i$ that groups fresh random values ν_i that is particular for each i.
   - (See step 2 in Algorithm 2) Next, LSSS-reconstruction is carried out using the coefficients $(c_{i,j})_j$. The aforementioned $\widetilde{t}_{0,j}$'s, over attribute j with approriate coefficients and products over DPVS with key components $(\widetilde{\mathbf{m}}_{i,j})_j$, gives $[\![(\sum_i \nu_i) \cdot a_{i,0}]\!]_{\mathsf{t}}$ at each i. It is important to notice that summing over all i will recover $[\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_{\mathsf{t}}$, which is needed for unmasking the function evaluation below. Meanwhile the access control part of the ciphertext $(\mathbf{c}_{i,j})_j$ together with the key components $(\mathbf{k}_{i,j})_j$ also reconstructs $[\![\psi_i a_{i,0} z]\!]_{\mathsf{t}}$.
   - (See step 3 in Algorithm 2) The subsequent step consists of performing DPVS products over the functional components $(\mathbf{t}_i, \mathbf{m}_i)_i$ of the ciphertext and the key, so as to obtain the evaluation $[\![\sum_i \langle \mathbf{x}_i, \mathbf{y}_i\rangle]\!]_{\mathsf{t}}$ that is masked by $[\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_{\mathsf{t}} + \sum_i [\![\omega \cdot \langle \mathbf{s}_i, \mathbf{y}_i\rangle + \omega' \cdot \langle \mathbf{u}_i, \mathbf{y}_i\rangle]\!]_{\mathsf{t}}$. We recall that ω, ω' come form the hashing of tags, and $\mathbf{s}_i, \mathbf{u}_i$ are the master secret keys.
   - (See step 4 in Algorithm 2) Finally, the auxiliary functional components $\mathbf{c}_{i,\mathsf{ipfe}}, \mathbf{k}_{i,\mathsf{ipfe}}$ are used, in conjonction with the reconstructed $[\![\psi_i a_{i,0} z]\!]_{\mathsf{t}}$ from above, in order to remove $\sum_i [\![\omega \cdot \langle \mathbf{s}_i, \mathbf{y}_i\rangle + \omega' \cdot \langle \mathbf{u}_i, \mathbf{y}_i\rangle]\!]_{\mathsf{t}}$. The aggregated $[\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_{\mathsf{t}}$ from step 2 is used to recover in the end $[\![\sum_i \langle \mathbf{x}_i, \mathbf{y}_i\rangle]\!]_{\mathsf{t}}$. Thanks to the polynomially large ranges of $\mathcal{F}_{\mathsf{subvec}}^{\mathsf{IP},B}$, one can then recover the inner product $\sum_i \langle \mathbf{x}_i, \mathbf{y}_i\rangle$ via discrete logarithm computations.

We refer to Section 4.2 for formal details and Algorithm 2 to see how the decryption is done. The underlying idea with probabilisitic vectorization is to distribute the masking term $[\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_{\mathsf{t}}$ over attribute-indexed components of the ciphertext (for the fresh ν_i's) and the key components (for the secret $a_{i,0}$'s). Then, $\mathbb{Z}_q$-linearity and products over the DPVS vectors are heavily employed. The fresh ν_i's that appear at each encryption of $\mathbf{x}_i$ by client i is also used to handle the *repetitions* of challenge ciphertexts, as we will see in the below paragraph. We remark that the below obstacles and their solutions are relating to the proof of security, whose main lines of arguments and ideas are complemented by the paragraph **Proof Strategy** from page 21.

Second Technical Obstacle: Repetitions and Access Control. We have mentioned in the introduction that tolerating repetitions of challenge messages $\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}$ is a crucial requirement for MCFE, in order for MCFE to imply MIFE in terms of provably secure cryptographic primitives. In our setting with *both private and public inputs*, the challenge ciphertexts given private $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)})$ are encrypted with public parts comprising of a tag and the set of attributes S_i. This means that repetitions are now must be *vis-à-vis* the public parts, in particular S_i. The latter complicates significantly the situation, which is already observed in a very recent work by Agrawal *et al.* [11]. Indeed on one hand, for a specific slot $i \in [n]$ and tag, *full* repetitions of $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)})$ and $\mathsf{S}_i^{(j_i)}$ mean that the MCFE should be resilient against attacks that try combining different attribute set $\mathsf{S}_i^{(j_i)} \neq \mathsf{S}_i^{(\tilde{j}_i)}$ at slot i, where $\mathbb{A}(\mathsf{S}_i^{(j_i)}) \neq \mathbb{A}(\mathsf{S}_i^{(\tilde{j}_i)})$. On the other hand, in terms of the inner product calculation, allowing repetitions on the private inputs $\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}$ for a fixed (i, tag) needs being taken into account by the admissibility: for all repetitions j_i

$$\sum_{i\in\mathcal{H}} \langle \mathbf{x}_i^{(0,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i\rangle = 0 \tag{1}$$

This implies for each $i \in \mathcal{H}$, over all repetitions j_i, the term $\langle \mathbf{x}_i^{(0,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i \rangle$ is constant. At the same time, for all $i \in \mathcal{C}$ that are corrupted, under repetitions j_i, it must be

$$\langle \mathbf{x}_i^{(0,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i \rangle = 0 \tag{2}$$

This makes sense even in the case of *static corruption*, since we do *not* prohibit such queries even after the set $\mathcal{C}$ is fixed. Finally, condition (2) does *not* need to cover private inputs of corrupted $i \in \mathcal{C}$ that are not queried to the oracle **LoR** because there exists no challenge bit b in those self-crafted ciphertexts using $(i \in \mathcal{C}, \mathsf{ek}_i, \mathsf{tag})$ on some $\mathbf{z}_i$, and decrypting jointly with others challenge ciphertexts under some key $\mathsf{dk}_{\mathbb{A},(\mathbf{y}_i)_{i\in[n]}}$ always gives the same i-th component $\langle \mathbf{z}_i, \mathbf{y}_i \rangle$ regardless of b. Formally, the concrete detailed admissibility for inner products is given on page 20.

Solution to the Second Obstacle: Masking with (Private-only) Repetitions. In this work we restrain our focus to the case where the repetitions are only allowed for the private inputs $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)})$. That is, the adversary is allowed to query multiple $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)})$, indexed by j_i, for a fixed $(i, \mathsf{tag}, \mathsf{S}_i)$. Dealing with *private-input* repetitions is handled by our generalization of the masking lemma from [31]. The formal statement of the lemma can be found in Lemma 1. At a high level, the setting of Lemma 1 contains a set of $\mathbf{c}$-vectors in which attributes j are encoded, and a set of $\mathbf{k}^*$-vectors that encode a policy $\mathbb{A}$ by secret shares $(a_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ w.r.t the policy $\mathbb{A}$. The lemma proves that for any given repetitive $x^{(\mathsf{rep})}$ and $y \in \mathbb{Z}_q$, where $\mathsf{rep} \in [J]$, we can randomize the $\mathbf{c}$-vectors by random $z_j \xleftarrow{\$} \mathbb{Z}_q^*$, at the same time encoding $(a'_j/z_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ in the $\mathbf{k}$-vectors. Particularly $(a'_j/z_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ is a *decorrelated* set of shares $(a'_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ w.r.t the policy $\mathbb{A}$ to share $a'_0 \xleftarrow{\$} \mathbb{Z}_q$. In the proof of the MCFE, we alllow repetitions of the challenge ciphertexts while fixing $(i, \mathsf{tag}, \mathsf{S}_i)$. After applying Lemma 1[3], as soon as $\mathbb{A}(\mathsf{S}_i) = 0$, there is an attribute j whose $z_j \xleftarrow{\$} \mathbb{Z}_q^*$ never appears in the $\mathbf{c}$-vectors returned to the adversary, thanks to the fact that $(i, \mathsf{tag}, \mathsf{S}_i)$ is fixed once for all repetitions of private inputs at i. That implies the decorrelated $(a'_j/z_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ cannot be related together, in an *information theoretical* sense, to recover $(a'_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ and reconstruct the shared value. We are then allowed to switch a'_0 into a uniformly random value for further steps in the MCFE proof. Finally, as demonstrated in Theorem 8, even in this setting of private-only repetitions, our MCFE with public inputs still cover MIFE, and the concrete scheme for inner products with access control in Section 4.2 gives MIFE for inner products.

Third Technical Obstacle: Adaptive Security. Another technical hurdle with which we successfully deal in our MCFE is the adaptive security of the challenge queries $\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)}$ indexed by repetitions j_i along with public inputs $(i, \mathsf{tag}, \mathsf{S}_i)$. Existing comparable schemes either achieves selective security [11], or considers the simpler scalar case [31].

Solution to the Third Obstacle: Adaptive Security via Perfect Indistinguishability and Complexity Leveraging. Aiming at adaptive security w.r.t $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)})$ with public inputs $(i, \mathsf{tag}, \mathsf{S}_i)$, we employ a *complexity leveraging* technique that is based on *formal* basis changes in the dual pairing vector spaces. More specifically, in order to prove two hybrids $\mathsf{G}_i, \mathsf{G}_{i+K}$ for some fixed K, are indistinguishable in the adaptive security proof, we define an event E that happens with fixed probability and whose probability space depends on the data that can be *adaptively* chosen by the adversary. Then, condition on E we move to the *selective* version $\mathsf{G}_i^*, \mathsf{G}_{i+1}^*, \ldots, \mathsf{G}_{i+K}^*$. If we can prove the sequence of *perfect indistinguishability* involving

$$\{\mathsf{G}_i^* \mid E\} \equiv \{\mathsf{G}_{i+1}^* \mid E\} \equiv \cdots \equiv \{\mathsf{G}_{i+K}^* \mid E\}$$

where E happens with fixed probability and is *independent* of the view of the adversary during the reductions $\{\mathsf{G}_{i+t}^* \mid E\} \equiv \{\mathsf{G}_{i+t+1}^* \mid E\}$ in the sequence, for all $t \in [K-1]$, then a probabilistic

[3] The randomness that is needed for the masking also comes from our above probabilistic vectorization, wherever we need individual randomness.

argument concludes that $\{\mathsf{G}_i\} \equiv \{\mathsf{G}_{i+1}\} \equiv \cdots \equiv \{\mathsf{G}_{i+K}\}$. The formal basis changes are used to achieve perfect indistinguishability between these selective versions $\{\mathsf{G}^*_{i+t} \mid E\}$ of the game. In the MCFE adaptive proof, the adaptive data include $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)})$ indexed by multiple repetitions j_i. We extensively use admissibility conditions (1) as well as (2) to define the basis changes. Details can be found in the proof of Theorem 12, the final probabilistic calculation for complexity leveraging can be examined in (16), for instance. As a reminder, an integral overview that puts together all above solutions, and more, is given in the paragraph **Proof Strategy** (on page 21) before the proof of Theorem 12.

2 Preliminaries

We write $[n]$ to denote the set $\{1, 2, \ldots, n\}$ for an integer n. For any $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers with addition and multiplication modulo q. For a prime q and an integer N, we denote by $GL_N(\mathbb{Z}_q)$ the general linear group of of degree N over $\mathbb{Z}_q$. We write vectors as row-vectors, unless stated otherwise. For a vector $\mathbf{x}$ of dimension n, the notation $\mathbf{x}[i]$ indicates the i-th coordinate of $\mathbf{x}$, for $i \in [n]$, and we write $\mathbf{1}_{\mathbf{x}} \in \{0,1\}^n$ to denote the indicator vector of $\mathbf{x}$. For two vectors $\mathbf{x}, \mathbf{y}$ of the same length, we write the Hadamard product $\mathbf{x} \circ \mathbf{y} := (\mathbf{x}[i]\mathbf{y}[i])_i$ to denote the component-wise product of $\mathbf{x}$ and $\mathbf{y}$. We will follow the implicit notation in [24] 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".

Hardness Assumptions. We need some the **Decisional Diffie-Hellman** (DDH) assumption in a cyclic group $\mathbb{G}$ of prime order q. In a cyclic group $\mathbb{G}$ of prime order q, the (DDH) assumption in $\mathbb{G}$ assumes that no ppt adversary can distinguish the distributions $\{([\![1]\!], [\![a]\!], [\![b]\!], [\![ab]\!])\}$ and $\{([\![1]\!], [\![a]\!], [\![b]\!], [\![c]\!])\}$ for $a, b, c \xleftarrow{\$} \mathbb{Z}_q$. 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 $\mathbb{G}_2$.

Dual Pairing Vector Spaces. Formal definitions can be found in Appendix A.2. Details of basis changes are recalled in the appendix A.5. We use 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. Let us fix $N \in \mathbb{N}$ and consider $\mathbb{G}_1^N$ having N copies of $\mathbb{G}_1$. Any $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1 \in \mathbb{G}_1^N$ is identified as the vector $(x_1, \ldots, x_N) \in \mathbb{Z}_q^N$. The $\mathbf{0}$-vector is $\mathbf{0} = [\![(0, \ldots, 0)]\!]_1$. The addition of two vectors, and $\mathbb{Z}_q$-scalar multiplication, in $\mathbb{G}_1^N$ are defined by coordinate-wise addition. Viewing $\mathbb{Z}_q^N$ as a vector space of dimension N over $\mathbb{Z}_q$ with the notions of bases, we can obtain naturally a similar notion of bases for $\mathbb{G}_1^N$. More specifically, any invertible matrix $B \in GL_N(\mathbb{Z}_q)$ identifies a basis $\mathbf{B}$ of $\mathbb{G}_1^N$, whose i-th row $\mathbf{b}_i$ is $[\![B^{(i)}]\!]_1$, where $B^{(i)}$ is the i-th row of B. Naturally we can extend basis changes in $GL_N(\mathbb{Z}_q)$ to changes of bases of $\mathbb{G}_1^N$ by the fact that $\mathbb{G}_1$ is cyclic. Treating $\mathbb{G}_2^N$ similarly, we can furthermore define a product of two vectors $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1 \in \mathbb{G}_1^N, \mathbf{y} = [\![(y_1, \ldots, y_N)]\!]_2 \in \mathbb{G}_2^N$ by $\mathbf{x} \times \mathbf{y} := \prod_{i=1}^N \mathbf{e}(\mathbf{x}[i], \mathbf{y}[i]) = [\![\langle (x_1, \ldots, x_N), (y_1, \ldots, y_N) \rangle]\!]_\mathsf{t}$. Given a basis $\mathbf{B} = (\mathbf{b}_i)_{i \in [N]}$ of $\mathbb{G}_1^N$, we define $\mathbf{B}^*$ to be a basis of $\mathbb{G}_2^N$ by first defining $B' := (B^{-1})^\top$ and the i-th row $\mathbf{b}_i^*$ of $\mathbf{B}^*$ is $[\![B'^{(i)}]\!]_2$. It holds that $B(B')^\top = I_N$ the identity matrix and $\mathbf{b}_i \times \mathbf{b}_j^* = [\![\delta_{i,j}]\!]_\mathsf{t}$ for every $i, j \in [N]$, where $\delta_{i,j} = 1$ if and only if $i = j$. We call the pair $(\mathbf{B}, \mathbf{B}^*)$ a *pair of dual orthogonal bases* of $(\mathbb{G}_1^N, \mathbb{G}_2^N)$. If $\mathbf{B}$ is constructed by a random invertible matrix $B \xleftarrow{\$} GL_N(\mathbb{Z}_q)$, we call the resulting $(\mathbf{B}, \mathbf{B}^*)$ a pair of random dual bases. A DPVS is a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q, N)$ with dual orthogonal bases.

Access Structure and Linear Secret Sharing Schemes. We recall the definitions of access structures and linear secret sharing schemes in Appendix A.3. In short, an access structure $\mathbb{A} \subseteq 2^{\mathsf{Att}} \setminus \{\varnothing\}$ over an attribute space Att is a family of sets S of attributes. A secret sharing scheme for an access structure $\mathbb{A}$ over the attributes $\mathsf{Att} = \{\mathsf{att}_1, \mathsf{att}_2, \ldots, \mathsf{att}_m\}$ allows sharing a secret s among the m attributes att_j for $1 \leq j \leq m$, such that: (1) Any authorized set S in $\mathbb{A}$

can be used to reconstruct s from the shares of its elements; (2) Given any unauthorized set and its shares, the secret s is statistically identical to a uniform random value. A linear secret sharing scheme (LSSS) is a way to linearly share a secret. More specifically, let K be a field, $d, f \in \mathbb{N}$, and Att be a finite universe of attributes. A *Linear Secret Sharing Scheme* LSSS over K for an access structure $\mathbb{A}$ over Att is specified by a share-generating matrix $\mathbf{A} \in K^{d \times f}$ such that for any $I \subset [d]$, there exists a vector $\mathbf{c} \in K^d$ with support I and $\mathbf{c} \cdot \mathbf{A} = (1, 0, \ldots, 0)$ if and only if $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$. Finally, let $y \in \mathbb{Z}_q$ where q is prime and for the sake of simplicity, let $\mathsf{Att} \subset \mathbb{Z}_q$ be a set of attributes. Let $\mathbb{A}$ be a monotone access structure over Att realizable by an LSSS over $\mathbb{Z}_q$. A *random labeling* procedure $\Lambda_y(\mathbb{A})$ is a secret sharing of y using LSSS:

$$\Lambda_y(\mathbb{A}) := (y, v_2, v_3, \ldots, v_f) \cdot \mathbf{A}^\top \in \mathbb{Z}_q^d \tag{3}$$

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

The Masking Lemma with Repetitions. We state a technical lemma that is employed throughout our proofs. This is a generalized version of [31, Lemma 4], where the masks can be introduced even when *repetitions* of $\mathbf{c}$-vectors over j and root are allowed. A detailed proof can be found in Appendix B.

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 $\mathsf{S} \subseteq \mathsf{Att}$ be a set of attributes. Let $(\mathbf{H}, \mathbf{H}^*)$ and $(\mathbf{F}, \mathbf{F}^*)$ be two random dual bases of $(\mathbb{G}_1^2, \mathbb{G}_2^2)$ and $(\mathbb{G}_1^8, \mathbb{G}_2^8)$, respectively. The vectors $(\mathbf{h}_1, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ are public, while all other vectors are secret. Suppose we have two random labelings $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ for $a_0, a'_0 \xleftarrow{\$} \mathbb{Z}_q$. Let J denote the maximum number of* repetitions *at each $j \in \mathsf{S}$ for $\mathbf{c}_j$ or for $\mathbf{c}_{\mathsf{root}}$. Then, under the SXDH assumption in $(\mathbb{G}_1, \mathbb{G}_2)$, the following two distributions are computationally indistinguishable:*

$$\left\{ \begin{array}{ll} (x^{(\mathsf{rep})}), y & \\ \mathbf{c}^{(\mathsf{rep})}_{j \in \mathsf{S}} & = (\sigma_j^{(\mathsf{rep})}(1, -j),\ \psi^{(\mathsf{rep})},\ 0^5)_{\mathbf{F}} \\ \mathbf{k}^*_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} & = (\pi_j \cdot (j, 1),\ a_j z,\ 0^5)_{\mathbf{F}^*} \\ \mathbf{c}^{(\mathsf{rep})}_{\mathsf{root}} & = (\psi^{(\mathsf{rep})},\ 0)_{\mathbf{H}} \\ \mathbf{k}^*_{\mathsf{root}} & = (a_0 z,\ 0)_{\mathbf{H}^*} \end{array} \right\} ; \left\{ \begin{array}{ll} (x^{(\mathsf{rep})}), y & \\ \mathbf{c}^{(\mathsf{rep})}_{j \in \mathsf{S}} & = (\sigma_j^{(\mathsf{rep})}(1, -j),\ \psi^{(\mathsf{rep})},\ 0^2,\ \boxed{\tau z_j x^{(\mathsf{rep})}},\ 0^2)_{\mathbf{F}} \\ \mathbf{k}^*_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} & = (\pi_j (j, 1),\ a_j z,\ 0^2,\ \boxed{a'_j y / z_j},\ 0^2)_{\mathbf{F}^*} \\ \mathbf{c}^{(\mathsf{rep})}_{\mathsf{root}} & = (\psi^{(\mathsf{rep})},\ \boxed{\tau x^{(\mathsf{rep})}})_{\mathbf{H}} \\ \mathbf{k}^*_{\mathsf{root}} & = (a_0 z,\ \boxed{a'_0 y})_{\mathbf{H}^*} \end{array} \right\}$$

for any $x^{(\mathsf{rep})}, y \in \mathbb{Z}_q$, where $\mathsf{rep} \in [J]$, and $z_j, \sigma_j, \pi_j, \psi, \tau, z, r'_0 \xleftarrow{\$} \mathbb{Z}_q$.

Below is the main function class for inner products with access control that is studied in this paper.

Definition 2 (Inner Products with LSSS). *We consider the functionality $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$ and $\mathcal{F}^{\mathsf{IP}}_{\mathsf{subvec}}$ that contains $F_{\mathbf{y}_1,\ldots,\mathbf{y}_n} : \prod_{i \in [n]} (\mathbb{Z}_q^{N_i}) \to \mathbb{Z}_q$ defined as $F_{\mathbf{y}_1,\ldots,\mathbf{y}_n}(\mathbf{x}_1, \ldots, \mathbf{x}_n) := \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$, which receives as inputs and parameters where for all i, $\max(\|\mathbf{x}_i\|_\infty, \|\mathbf{y}_i\|_\infty) < B$, with $B = \mathsf{poly}(\lambda) \in \mathbb{N}$ being a polynomial. The access control is given by $\mathsf{Rel} : \mathsf{LSSS} \times \left(\prod_{i=1}^n 2^{\mathsf{Att}}\right) \to \{0, 1\}$ as $\mathsf{Rel}(\mathbb{A}, (\mathsf{S}_i)_i) = \prod_i \mathbb{A}(\mathsf{S}_i)$. The class LSSS contains Linear Secret Sharing Schemes over Att, and 2^{Att} denotes the superset of an attribute space $\mathsf{Att} \subseteq \mathbb{Z}_q$.*

3 Multi-Client Functional Encryption with Public Inputs

In this section we refine the definition of *multi-client functional encryption* in which at the time of encryption, each client can specify their own *public* data, while the function class contains functions that evaluate *both* the combined private and public data of clients. In Section 3.2 we prove that this general notion covers the original MCFE notion *with* and *without* fine-grained access control, and even more, *e.g.* the notion of *public-attributes* ABE. Interestingly, the syntax of previous formal definitions of FE, either in single-client [16] or multi-client [25, 18], allows no public data and let *public-attributes* ABE escape their scope, or has to include an artificial function in the class. More specifically, we discuss in Theorem 8 how our formal definition of MCFE with public inputs can be related to other existing primitives.

3.1 Definitions

Definition 3 (Functions with public inputs). *Let $\lambda, n \in \mathbb{N}$ and let $\mathcal{D}_{\lambda,i}$ and $\mathcal{R}_\lambda$ be domains and ranges indexed by λ in some ensembles $\{\mathcal{D}_{\lambda,i}\}_\lambda$ where $i \in [n]$, $\{\mathcal{R}_\lambda\}_\lambda$, respectively. A* function class $\mathcal{F} = \{F_{\lambda,n}\}_{\lambda,n}$ with public inputs $(\mathcal{Z}_{\lambda,i})_{i\in[n]}$, *where $\mathcal{Z}_{\lambda,i} := \{0,1\}^{poly(\lambda)}$, is defined to contain $F_{\lambda,n} : \prod_{i=1}^{n} (\mathcal{D}_{\lambda,i} \times \mathcal{Z}_{\lambda,i}) \to \mathcal{R}_\lambda$.*

In the following the index n is a function in λ and we omit it for clarity.

Definition 4 (Multi-client functional encryption with public inputs). *A* multi-client functional encryption (MCFE) scheme with public inputs, *for the class $\mathcal{F}$ with public inputs $(\mathcal{Z}_{\lambda,i})_{i\in[n]}$ where $\mathcal{Z}_{\lambda,i} := \mathsf{Tag} \times \widetilde{\mathcal{Z}}_{\lambda,i}$ for some set $\mathsf{Tag} = \{0,1\}^{poly(\lambda)}$, consists of four algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$*:*

$\mathsf{Setup}(1^\lambda, 1^n)$**:** *Given as inputs 1^λ for a security parameter λ, and a number of clients n, output a master secret key msk and n encryption keys $(\mathsf{ek}_i)_{i\in[n]}$.*

$\mathsf{Extract}(\mathsf{msk}, F_\lambda)$**:** *Given a function description $F_\lambda : \prod_{i=1}^{n} (\mathcal{D}_{\lambda,i} \times \mathcal{Z}_{\lambda,i}) \to \mathcal{R}_\lambda$ in $\mathcal{F}$, and the master secret key msk, output a decryption key dk_{F_λ}.*

$\mathsf{Enc}(\mathsf{ek}_i, x_i, z_i)$**:** *Given as inputs* public *data $z_i = (\mathsf{tag}, \tilde{z}_i) \in \mathcal{Z}_{\lambda,i}$ that contains some tag, an encryption key ek_i, a message $x_i \in \mathcal{D}_{\lambda,i}$, output a ciphertext $(\mathsf{ct}_{\mathsf{tag},i}, z_i)$. For a specific client i, the sets $\mathcal{D}_{\lambda,i}$ and $\mathcal{Z}_{\lambda,i}$ are indexed by λ in some ensembles $\{\mathcal{D}_{\lambda,i}\}_\lambda, \{\mathcal{Z}_{\lambda,i}\}_\lambda$.*

$\mathsf{Dec}(\mathsf{dk}_{F_\lambda}, \mathbf{c})$**:** *Given the decryption key dk_{F_λ} and a vector of ciphertexts $\mathbf{c} := (\mathsf{ct}_{\mathsf{tag},i}, z_i)_i$ of length n, output an element in $\mathcal{R}_\lambda$.*

Our syntax can be seen as a particular case of the general primitive *Multi-Party Functional Encryption* (MPFE) [8] in which we consider the particular case of multi-client while the key generation stays centralized. The main difference is in terms of security where ours is less restrictive (see Definition 5), which is sufficient for establishing connection to other primitives as we will see in Section 3.2. Regarding the concrete class calculating *inner products with access control*, we will revisit the connection from MIFE to MCFE in Section 4.

Correctness. For sufficiently large $\lambda \in \mathbb{N}$, for all $(\mathsf{msk}, (\mathsf{ek}_i)_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda)$, all functions $F_{\lambda,n} : \prod_i (\mathcal{D}_{\lambda,i} \times \mathcal{Z}_{\lambda,i}) \to \mathcal{R}_\lambda$ and $\mathsf{dk}_{F_{\lambda,n}} \leftarrow \mathsf{Extract}(\mathsf{msk}, F_{\lambda,n})$, for all $\mathsf{tag} \in \mathsf{Tag}$ and $(z_i)_{i=1}^n \in \mathcal{Z}_{\lambda,1} \times \cdots \times \mathcal{Z}_{\lambda,n}$, for all $(x_i)_{i\in[n]} \in \mathcal{D}_{\lambda,1} \times \cdots \times \mathcal{D}_{\lambda,n}$, if $F_\lambda((x_i, z_i)_i) \neq \perp$ and $z_i = (\mathsf{tag}, \tilde{z}_i) \in \mathcal{Z}_i$ for all i, the following holds with overwhelming probability, over the random coins of the algorithms:

$$\mathsf{Dec}\left(\mathsf{dk}_{F_\lambda}, (\mathsf{Enc}(\mathsf{ek}_i, x_i, z_i))_{i\in[n]}\right) = F_{\lambda,n}((x_i, z_i)_i) \ .$$

Security. First of all we define *admissible* adversaries $\mathcal{A}$ against an MCFE $\mathcal{E}$. We use the recent formulation of admissibility in [32].

Definition 5 (Admissible adversaries with public inputs). *Let $\mathcal{A}$ be a* ppt *adversary and let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be* an MCFE scheme with public inputs *for the function class $\mathcal{F}$ with public inputs $\mathcal{Z}_{\lambda,i} := \mathsf{Tag} \times \widetilde{\mathcal{Z}}_{\lambda,i}$. In the security game given in Figure 1 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 the following condition holds:*

> *There exist $\mathsf{tag} \in \mathsf{Tag}$, a function $F \in \mathcal{F}$ is queried to* **Extract**, *challenges $(x_i^{(0)}, x_i^{(1)}, (\mathsf{tag}, \tilde{z}_i^{(chal)}))_{i\in[n]}$ is queried to* **LoR**, *with public inputs $\tilde{z}_i^{(chal)} \in \widetilde{\mathcal{Z}}_{\lambda,i}$, and there exist vectors $(\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \mathbf{v}^{(chal)})$ so that $\forall\, i \in \mathcal{H} : \mathbf{t}^{(b)}[i] = x_i^{(b)}$ and $\mathbf{v}^{(chal)}[i] = \tilde{z}_{\mathsf{ekey}}^{(chal)}[i]$ satisfying*
>
> $$F((\mathbf{t}^{(0)}[i], (\mathsf{tag}, \mathbf{v}[i]))_{i\in[n]}) \neq F((\mathbf{t}^{(1)}[i], (\mathsf{tag}, \mathbf{v}[i]))_{i\in[n]}) \ . \tag{4}$$

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

Discussion on admissibility. We develop below some discussion on the admissibility notion in Definition 5:

- *(Repetitions)* In comparison to the original security of MCFE in [18], an adversary is still *admissible* if they query multiple times to the challenge oracle for a fixed (i, tag), whereas an admissible adversary *as per* [18] is allowed to query at most once for each (i, tag). This aspect of *repetitions* in the admissibility was first studied in [19] and later generalized in [20]. It is important that when repetitions are allowed for ciphertexts, the security model of MCFE automatically encompasses that of MIFE by replacing tags with a constant value, as confirmed in recent works [12]. Lastly, in our notion of MCFE with public inputs, we can also consider restricted repetitions only on the private parts $(x_i^{(0)}, x_i^{(1)})$ (see the weaker notion rep-priv in the following) and not on the public parts $z_i^{(chall)}$ to the challenge oracle. This form of restricted repetitions gives a weaker notion of security, but it still covers the security of classical MIFE *without* public inputs, as studied in [25, 13, 5, 21, 3, 41, 6, 9].
- *(Weaker constraints)* Regarding the corrupted $i \in \mathcal{C}$ in general, the admissibility check is done in **Finalise** at the end of the security experiment, and Definition 5 *per se* allows the adversary to query the challenge oracle **LoR**, whether the corruption is static or not, on

$$i, x_i^{(0)}, x_i^{(1)}, (\mathsf{tag}^*, \tilde{z}_i^{(chal)})$$

  where $x_i^{(0)} \neq x_i^{(1)}$. The adversary stays admissible as long as the condition (4) is not satisfied, *i.e.* the foregoing $x_i^{(0)} \neq x_i^{(1)}$ of corrupted $i \in \mathcal{C}$ does not make F differ with respect to the challenge bit $b \xleftarrow{\$} \{0, 1\}$. The original security of MCFE in [18] does not allow attacks where there exists $i \in \mathcal{C}$ such that $x_i^{(0)} \neq x_i^{(1)}$. By allowing a such query, we apparently allow more attacks than the original security model of MCFE in [18]. The work [32] examines the legitimacy of this condition in the plain (Decentralized) MCFE (DMCFE) setting and proposes a stronger security model that does allow $x_i^{(0)} \neq x_i^{(1)}$ of corrupted $i \in \mathcal{C}$ (thus considers more attacks admissible).
- *(Corrupted ciphertexts)* In terms of usage of the corrupted ek_i, for the admissible conditions 4 we do *not* put any quantifier on the ciphertexts that can be crafted by the adversary using a corrutped ek_i for $i \in \mathcal{C}$. Because when decrypting jointly a such ciphertext $\bar{\mathsf{ct}}_i \leftarrow \mathsf{Enc}(\mathsf{ek}_i, \bar{x}_i, \bar{z}_i)$ with other challenge ciphertext components (up to repetitions) vis-à-vis a function F, the evaluation will provide

$$F((\mathbf{t}^{(b)}[j], (\mathsf{tag}, \mathbf{v}[j]))_{j \neq i}, (\bar{x}_i, \bar{z}_i), (\mathbf{t}^{(b)}[j'], (\mathsf{tag}, \mathbf{v}[j']))_{j' \neq i})$$

  that always has the same i-th argument and cannot change the output of F. The same reasoning applies when the adversary crafts themselves multiple corrupted ciphertexts.
- *(Checking admissibility)* The admissiblity in Definition 5 for general function class may not be efficiently decidable. As we will see later in Section 4.2, within the scope of this paper, the class of functions is restricted to computing inner products with access control by LSSS, and the admissibility can be decided efficiently using concrete conditions 1 and 2 prior to the proof of Theorem 12.

In Theorem 8 we discuss how an MCFE that is provably secure under the admissibility in Definition 5 will imply a provably secure MIFE, and more. For the concrete class of computing *inner products with access control* which is the main subject of Section 4, we refer to Remark 14.

Definition 6 (IND-security with repetitions for MCFE with public inputs). *An MCFE scheme with public inputs* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the function class* $\mathcal{F}$ *with public inputs is* IND-secure *if for all* ppt *adversaries* $\mathcal{A}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*, the following probability is negligible*

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

$\mathbf{Initialise}(1^\lambda)$ $\boxed{\mathbf{Initialise}(1^\lambda, (x_i^{(0)}, x_i^{(1)})_{i\in[n]})}$

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

$\mathbf{Enc}(i, x_i, (\mathsf{tag}, \tilde{z}_i))$

Return $\mathsf{Enc}(\mathsf{ek}_i, x_i, (\mathsf{tag}, \tilde{z}_i))$

$\mathbf{Finalise}(b')$

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

$\mathbf{LoR}(i, x_i^{(0)}, x_i^{(1)}, (\mathsf{tag}^*, \tilde{z}_i^{(chal)}))$ $\boxed{\mathbf{LoR}(i, (\mathsf{tag}^*, \tilde{z}_i^{(chal)}))}$

$\mathsf{Enc}(\mathsf{ek}_i, x_i^{(b)}, (\mathsf{tag}^*, \tilde{z}_i^{(chal)})) \rightarrow \mathsf{ct}^{(b)}_{\mathsf{tag}^*,i}$
Return $\mathsf{ct}^{(b)}_{\mathsf{tag}^*,i}$

$\mathbf{Corrupt}(i)$

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

$\mathbf{Extract}(F)$

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

Fig. 1: The security game $\mathsf{Expr}^{\mathsf{mc\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$, $\boxed{\mathsf{Expr}^{\mathsf{mc\text{-}sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)}$ for Definition 6

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

In a more relaxed notion, the scheme $\mathcal{E}$ is *selectively IND-secure* with the security game $\mathsf{Expr}^{\mathsf{mc\text{-}sel\text{-}ind\text{-}cpa}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$, where the challenges are chosen before the setup.

Weaker notions. We can relax the admissibility notion from Definition 5, with more exclusions, to obtain weaker security notions considered in literature. They are simpler to achieve, and some generic conversions allow to lift from a weaker to a stronger scheme. One can consider a weaker notion of security for MCFE in which either all or none of honest components in the challenge are queried., the MCFE scheme said to be secure against *complete* queries only and add the following exclusion to the admissibility: *"There exist a tag* tag *and* $i, j \in \mathcal{H}$ *such that* $i \neq j$*, there exists a query* $(i, x_i^{(0)}, x_i^{(1)}, (\mathsf{tag}, *))$ *to* **LoR** *but there exist no query* $(j, x_j^{(0)}, x_j^{(1)}, (\mathsf{tag}, *))$ *to* **LoR***"*; The flag pos in the name of the experiment denotes this weaker notion in admissibility. We also define a notion of security where only one challenge tag tag^* is allowed, with the following exclusion to the admissibility: *"There exist two tags* $\mathsf{tag} \neq \mathsf{tag}'$ *and queries* $(*, *, *, (\mathsf{tag}, *))$, $(*, *, *, (\mathsf{tag}', *))$ *to* **LoR***"* ; The scheme $\mathcal{E}$ is *one-time IND-secure*, with the flag 1chal in the name of the experiment. Finally, if we allow only repetitions on the *private* parts $(x_i^{(0)}, x_i^{(1)})$ and not on the public parts $z_i^{(chal)}$ to **LoR** (or x_i and not on z_i to **Enc**), we denote the corresponding experiment with this weaker notion with the flag rep-priv, with the additional exclusion: *"There exist a tag* tag*, an index* i *and two public values* $z \neq z'$*, with queries* $(i, *, *, (\mathsf{tag}, z))$ *to* **LoR** *or* $(i, *, (\mathsf{tag}, z))$ *to* **Enc***, and* $(i, *, *, (\mathsf{tag}, z'))$ *to* **LoR** *or* $(i, *, (\mathsf{tag}, z'))$ *to* **Enc***"*.

All these flags pos, 1chal, rep-priv can be combined and added to the experiments presented in Figure 1.

Lemma 7 allows us to concentrate on the notion of one-time IND-security for our construction. The proof is a standard hybrid argument, thanks to the **Enc**-oracle access (in the case of secret-key encryption), in addition to **LoR**.

Lemma 7. *Let* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *for the function class* $\mathcal{F}$ *be an MCFE scheme with public inputs. If* $\mathcal{E}$ *is one-time IND-secure, then* $\mathcal{E}$ *is IND-secure.*

3.2 Implications between Notions: MCFE, MIFE, and more

Since its introduction in [25], a long line of works [13, 5, 21, 3, 41, 6, 9] considers MIFE having only *one* encryptor who can use a master secret key to encrypt independent components of a message. Our definition of MCFE from Definition 4 can capture this widely studied (one-encryptor) notion of MIFE, *with* and *without* access control, and in the latter case with *public* attributes. Generally, Theorem 8 demonstrates that given a secure MCFE *as per* Definition 6 for a strong enough

function class with public inputs, we can obtain secure instantiations of standard existing MIFE/MCFE notions in the secret-key setting as well as (single-client) FE/KP-ABE notions in the public-key setting. Relevant notions are recalled in Appendix A.4.

Theorem 8. *Let $\mathcal{F}$ be a function class with public inputs $(\mathcal{Z}_{\lambda,i})_{i\in[n]}$ where $\mathcal{Z}_{\lambda,i} := \mathsf{Tag} \times \widetilde{\mathcal{Z}}_{\lambda,i}$ for some tag space $\mathsf{Tag} = \{0,1\}^{poly(\lambda)}$. The elements of $\mathcal{F}$ are $F_{\lambda,n} : \prod_{i=1}^{n} (\mathcal{D}_{\lambda,i} \times \mathcal{Z}_{\lambda,i}) \to \mathcal{R}_\lambda$. Suppose that $\mathcal{F}$ contains the identity function $F^{\mathsf{id}}_{\lambda,n}$ where for all $(x_i, z_i)_i$, $F^{\mathsf{id}}_{\lambda,n}((x_i, z_i)_i) = (x_i, z_i)_i$. We suppose further that $\mathcal{F}_{\lambda,n}$ can encode a policy class Pol whose attributes are contained in $\mathsf{Att} \subseteq \widetilde{\mathcal{Z}}_{\lambda,i}$ for all $i \in [n]$. We have the following commutative diagram:*

$$\begin{array}{ccc}
\mathit{MCFE}^{\mathsf{xxx\text{-}rep\text{-}priv}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}] & \xrightarrow{\mathsf{rep\text{-}priv}} & \mathit{MIFE}^{\mathsf{xxx}}[\mathcal{F}] \\
\mathsf{adm} \big\downarrow \textit{(Def. 5)} & & \\
\mathit{FE}^{\mathsf{xxx}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}] & \xrightarrow[\mathsf{Att}\subseteq\widetilde{\mathcal{Z}}_{\lambda,i}]{\textit{pub. input}} & \mathit{KP\text{-}ABE}^{\mathsf{xxx}}_{\mathsf{pub}}[\mathsf{Pol}, \mathsf{Att}]
\end{array}$$

where

- *Each arrow "$\to$" preserves the IND-CPA security level* $\mathsf{xxx} \in \{\mathsf{sel}, \mathsf{adp}\} \times \{\mathsf{dyn}, \mathsf{stat}\}$ *of (challenge-selective, challenge-adaptive) combining with (dynamic, static) corruption security respectively. The label of the arrow indicates the necessary property for it to hold, detailed in the proof.*
- $\mathit{MIFE}^{\mathsf{xxx}}[\mathcal{F}]$ *denotes an MIFE following Definition 23, that can be adapted to capture MIFE for calculations in $\mathcal{F}$* without *access control as defined in [13, 5, 21, 3, 41, 6, 9].*
- $\mathit{FE}^{\mathsf{xxx}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}]$ *following Definition 21, that can be adapted to capture FE* with *access control as in [4, 31, 11], or* without *access control [16].*
- $\mathit{KP\text{-}ABE}^{\mathsf{xxx}}_{\mathsf{pub}}[\mathsf{Pol}, \mathsf{Att}]$ *denotes a KP-ABE for the policy class* Pol *with* public *attributes. The notion follows Definition 19.*

Proof (Of Implication MCFE to (single-client, public-key) FE.). We perform the reduction going from MCFE to (single-client, public-key) FE below. For completeness, other reductions are given in formal details in Appendix C. Let $\mathsf{MCFE}^{\mathsf{xxx}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}]$ be a secure MCFE following Definition 6. We denote by $(\mathsf{Setup}^{\mathsf{mc}}, \mathsf{Extract}^{\mathsf{mc}}, \mathsf{Enc}^{\mathsf{mc}}, \mathsf{Dec}^{\mathsf{mc}})$ the algorithms of the MCFE.

The function class is $\mathcal{F}$ containing $F_\lambda : \mathcal{D}_\lambda \times \mathcal{Z}_\lambda \to \mathcal{R}_\lambda$. Following Definition 20, the obtained FE is defined by algorithms:

$\mathsf{Setup}^{\mathsf{pk}}(1^\lambda)$**:** Run $\mathsf{Setup}^{\mathsf{mc}}(1^\lambda, 1^1) \to (\mathsf{msk}^{\mathsf{mc}}, \mathsf{ek}^{\mathsf{mc}})$. Output $\mathsf{msk} := \mathsf{msk}^{\mathsf{mc}}, \mathsf{pk} := (\mathsf{ek}^{\mathsf{mc}})$.
$\mathsf{Extract}^{\mathsf{pk}}(\mathsf{msk}, F_\lambda)$**:** Run $\mathsf{Extract}^{\mathsf{mc}}(\mathsf{msk}^{\mathsf{mc}}, F_\lambda) \to \mathsf{dk}_{F_\lambda}$ and output dk_{F_λ}.
$\mathsf{Enc}^{\mathsf{pk}}(\mathsf{pk}, x, z)$**:** Parse $\mathsf{pk} := (\mathsf{ek}^{\mathsf{mc}})$ and $z := (\epsilon, \tilde{z})$ as there is no tag in single client and public key FE. Sample $\mathsf{tag} \xleftarrow{\$} \mathsf{Tag}$ and run $\mathsf{Enc}^{\mathsf{mc}}(\mathsf{ek}^{\mathsf{mc}}, x, (\mathsf{tag}, \tilde{z})) \to \mathsf{ct}$. Finally output ct.
$\mathsf{Dec}^{\mathsf{pk}}(\mathsf{dk}_{F_\lambda}, \mathsf{ct})$**:** Run and output $\mathsf{Dec}^{\mathsf{mc}}(\mathsf{dk}_{F_\lambda}, \mathsf{ct})$.

Correctness follows from the correctness of the MCFE. If the function class captures access control, then the FE is for the same class having access control as well. In terms of security, let $\mathcal{A}$ be an adversary against the FE *as per* Definition 21. We construct an adversary $\mathcal{B}$ breaking $\mathsf{MCFE}^{\mathsf{xxx\text{-}rep\text{-}priv}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}]$, with *static* corruptions, using $\mathcal{A}$. The adversary $\mathcal{B}$ simulates the FE game by (i) first querying its MCFE challenger on $(1^\lambda, 1)$ to obtain the public parameters pp (if any) then *queries* **Corrupt**(1), gets ek, and forwards $\mathsf{pk} := \mathsf{ek}$ together with pp to $\mathcal{A}$. We note that the corrupted client is known from the beginning; (ii) simulating the FE's challenge queries by forwarding the challenge queries (*i.e.* sample $\mathsf{tag} \xleftarrow{\$} \mathsf{Tag}$ and define the challenge to be $(1, x^{(0)}, x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$) to its MCFE challenger given $\left(\left(x^{(0)}, (\epsilon, \tilde{z}^{(chal)})\right), \left(x^{(1)}, (\epsilon, \tilde{z}^{(chal)})\right)\right)$ by $\mathcal{A}$; (iii) the key extraction queries are forwarded to the MCFE challenger in a straightforward manner. If the FE adversary $\mathcal{A}$ is admissible, *i.e.* $x^{(0)} \neq x^{(1)}$ but $F(x^{(0)}, (\epsilon, \tilde{z}^{(chal)})) = F(x^{(1)}, (\epsilon, \tilde{z}^{(chal)}))$

for all F queried to **Extract**, then the challenge query $(1, x^{(0)}, x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$ is on a pair of inputs $(x^{(0)}, (\mathsf{tag}, \tilde{z}^{(chal)})) \neq (x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$ conforming to the admissibility. This implies that $\mathcal{B}$ is also admissible following Definition 5. Moreover, the fact that every encryption query is defined on a *freshly sampled* tag implies that there is no repetitions for any pair $(1, \mathsf{tag})$ registered to the MCFE challenger. This allows us to allow encrypting different public inputs even though the MCFE is for *private inputs repetitions* only. Therefore, if $\mathcal{A}$ wins the FE game, then $\mathcal{B}$ wins the MCFE game. □

Remark 9. **(From secret key to public key)** We emphasize that the crucial point allowing us to go from the *secret key* setting of MCFE to the *public key* setting of FE is the *admissibility* in Definition 5 under corruption. More specifically, Definition 5 allows the reduction to forward the challenge queries of its (public key) FE to the MCFE challenger, for the only client as $n = 1$,

$$(x^{(0)}, (\mathsf{tag}, \tilde{z}^{(chal)})) \neq (x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$$

as long as $F(x^{(0)}, (\epsilon, \tilde{z}^{(chal)})) = F(x^{(1)}, (\epsilon, \tilde{z}^{(chal)}))$ for all F queried to **Extract**. The only *secret encryption key* ek is corrupted up front and known to the FE adversary as a *public key* pk. Comparing to existing admissibility notions in [18], which excludes attacks where there exists $i \in \mathcal{C}$ such that $x_i^{(0)} \neq x_i^{(1)}$, the only queries that the reduction can forward are the *trivial* one from the FE adversary where $x^{(0)} = x^{(1)}$. Hence, existing admissibility notions in [18] and subsequent works are not sufficient to capture the reduction from MCFE to FE with meaningful CPA-security. Furthermore KP-ABE is made possible (without attribute-hiding) thanks to the public inputs.

Remark 10. **(Concrete instantiations)** Another key observation of Theorem 8 is that starting from any provably MCFE, we obtain an MIFE for the same function class by fixing one public tag for all ciphertexts. The security of the resulted MIFE comes from the fact that the security of the underlying MCFE allows *repetitions* at each position i, under the fixed tag, thanks to the admissibility in Definition 5. In this paper, our final construction for MCFE with access-control (see Corollary 13) satisfies this security with repetitions along with other favorable properties to be lifted to an MIFE with access-control. We consider the function class $\mathcal{F}_{\mathsf{subvec}}^{\mathsf{IP},B} \times \mathsf{LSSS}$ as defined in definition 2. Applying Theorem 8 to our MCFE in Section 4.2 gives concrete instantiations of the corresponding primitives.

4 MCFE for Inner Products with Access Control: Encrypting Vectors with Security against Repetitions

First of all, we specialize the general notion of MCFE with public inputs so as to define and give the model of security for *multi-client functional encryption with fine-grained access control* in Section 4.1. Our main goal is to improve the MCFE construction in [31, Section 5], which supports only encrypting scalars and does not tolerate *repetitions* of challenge ciphertexts. Section 4.2 gives an extension to encrypt subvectors, in a security model where the admissibility allows *repetitions* at positions under a challenge tag. Towards Corollary 13, we remove all *one-challenge* and *complete* challenge queries, and the resulted MCFE can be made MIFE by fixing a public tag. This clarifies the conversion from MCFE to MIFE in [31, Remark 16]. A subtlety is that the fixed public tag is processed by hashing, leading to a MIFE that inherits all security properties of the MCFE but without tags and without corruption. Hence, putting forward the fact that our MCFE does *not* allow repetitions on the attributes per client but only repetitions their private inputs, the obtained MIFE is secure only against *repetitions on private inputs*, *i.e.* potentially repetitive private $\mathbf{x}_i$ and no repetitions on the attributes S_i of each i. We discuss further our construction and revisit the MIFE regime for comparison with [4, 31] in Remark 14.

4.1 Definitions

We specialize the notion of MCFE with public inputs in Definition 4 to define the notion of multi-client functional encryption with fine-grained access control, *key-policy* and with *public attributes*.

Specialized function class with access control. Let $\lambda \in \mathbb{N}$ be a security parameter and we denote by n the number of clients in the system, which is fixed at set up time. We describe the function class $\mathcal{F} \times \mathsf{AC\text{-}K}$ for the multi-client functional encryption with fine-grained access control below:

- The public attributes of each client i come from $\mathcal{Z}_{\lambda,i} := \mathsf{Tag} \times \mathsf{AC\text{-}Ct}_i$ for some set $\mathsf{AC\text{-}Ct}_i$ and a tag space $\mathsf{Tag} = \{0,1\}^{poly(\lambda)}$.
- The access control is defined via a relation $\mathsf{Rel} : \mathsf{AC\text{-}K} \times \mathsf{AC\text{-}Ct}_1 \times \cdots \times \mathsf{AC\text{-}Ct}_n \to \{0,1\}$, for some set $\mathsf{AC\text{-}K}$.
- The function class $\mathcal{F} \times \mathsf{AC\text{-}K}$ contains $(F_\lambda, \mathsf{ac\text{-}k})$ having public inputs $(\mathcal{Z}_{\lambda,i})_{i\in[n]}$.

A plaintext for client i consists of $x_i \in \mathcal{D}_{\lambda,i}$, where $\mathcal{D}_{\lambda,i}$ denotes the domain from which each client i gets their inputs. The corresponding ciphertext can be decrypted to $F_\lambda(x)$ using the functional key $\mathsf{sk}_{F_\lambda,\mathsf{ac\text{-}k}}$ for $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$ if and only if $\mathsf{Rel}(\mathsf{ac\text{-}k}, (\mathsf{ac\text{-}ct}_i)_i) = 1$. Given the above specialization, the syntax of MCFE with access control can be derived from the general syntax of MCFE with public inputs in Definition 4. For the sake of analysis of our scheme later on, we give below only the *correctness* and *security* definitions for the specialized function class $\mathcal{F} \times \mathsf{AC\text{-}K}$.

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

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

where $F_\lambda : \mathcal{D}_{\lambda,1} \times \cdots \times \mathcal{D}_{\lambda,n} \to \mathcal{R}_\lambda$ and the probability is taken over the coins of algorithm.

Security. The security game is depicted in Figure 1, where the functionality class is $\mathcal{F} \times \mathsf{AC\text{-}K}$, the set of public data for each client i is $\mathcal{Z}_{\lambda,i} := \mathsf{Tag} \times \mathsf{AC\text{-}Ct}_i$. We recall that our general admissibility in Definition 11 allows an adversary to query *multiple* times to the challenge oracle for a fixed (i, tag). In particular, we consider also attacks where multiple $\mathbf{x}_i^{(\mathsf{rep})}$ are queried for the same (i, tag) to the oracle **LoR**, namely with *repetitions* at position i under the challenge tag tag. The formal definition, which is concretely interpreted for the class $\mathcal{F} \times \mathsf{AC\text{-}K}$ based on the general Definition 11 of MCFE with public inputs, is given below.

Definition 11 (Admissible adversaries with fine-grained access control). *Let $\mathcal{A}$ be a* ppt *adversary and let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an* MCFE scheme with fine-grained access control *for the functionality class $\mathcal{F} \times \mathsf{AC\text{-}K}$. In the security game given in Figure 1 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 exist $\mathsf{tag} \in \mathsf{Tag}$, a function $(F, \mathsf{ac\text{-}k}) \in \mathcal{Q}$ is queried to* **Extract**, *two challenges $(x_i^{(0)}, x_i^{(1)}, (\mathsf{tag}, \mathsf{ac\text{-}ct}_i))_{i\in[n]}$ are queried to* **LoR**, *with public inputs $\mathsf{ac\text{-}ct}_i \in \mathsf{AC\text{-}Ct}_{\lambda,i}$, a pair $(\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \mathbf{v}^{(chal)})$ so that for $b \in \{0,1\}$, $\forall\, i \in \mathcal{H} : \mathbf{t}^{(b)}[i] = x_i^{(b)}$ and $\mathbf{v}^{(chal)}[i] = \mathsf{ac\text{-}ct}_i$, and*
>
> - *The policy passes[a]: $\mathsf{Rel}(\mathsf{ac\text{-}k}, \mathbf{v}^{(chal)}) = 1$.*
> - *The function evaluation differs:*
>
> $$F\left(\mathbf{t}^{(0)}\right) \neq F\left(\mathbf{t}^{(1)}\right) \quad . \tag{5}$$

[a] This is up to attributes replacement in the corrupted slots $i \in \mathcal{C}$, therefore we only required $\mathbf{v}^{(chal)}$ to coincide with only with the *honest* attributes $(\mathsf{ac\text{-}ct}_i)_{i\in\mathcal{H}}$ and leave free the *corrupted* part.

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

We recall the weaker notion considering only *complete* queries, while facing repetitions, for this concrete $\mathcal{F} \times \mathsf{AC\text{-}K}$.

Weaker notions. We can relax Definition 11 to obtain weaker notions, in a similar manner which we use to relax Definition 5. The *selective*, *private-input only repetitions*, *complete*, and *one-time* security relaxations are straightforward.

4.2 Extension to Sub-vectors

In this section we present an MCFE scheme with fine-graine access control whose i-th ciphertext can encrypt *subvectors* of length N_i. In Remark 14 we discuss how to turn our final MCFE for inner products with access control, into an MIFE in the standard model, for computing inner products without access control. The bilinear group is $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, \mathbf{e}, q)$. The function class of interests is $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$ as defined in definition 2. A high-level overview of the construction is given in paragraph 7.

Construction. The details are given below:

$\mathsf{Setup}(1^\lambda)$**:** Choose $n+1$ pairs of dual orthogonal bases $(\mathbf{H}_i, \mathbf{H}^*_i, \mathbf{B}_i, \mathbf{B}^*_i)$ for $i \in [n]$ and $(\mathbf{F}, \mathbf{F}^*, \mathbf{G}, \mathbf{G}^*)$,$(\mathbf{H}_i, \mathbf{H}^*_i)$ is a pair of dual bases for $(\mathbb{G}_1^{2N+4}, \mathbb{G}_2^{2N+4})$, $(\mathbf{B}_i, \mathbf{B}^*_i)$ is a pair of dual bases for $(\mathbb{G}_1^{N+4}, \mathbb{G}_2^{N+4})$, $(\mathbf{F}, \mathbf{F}^*)$ is a pair of dual bases for $(\mathbb{G}_1^{2N+6}, \mathbb{G}_2^{2N+6})$, $(\mathbf{G}, \mathbf{G}^*)$ is a pair of dual bases for $(\mathbb{G}_1^{2N+6}, \mathbb{G}_2^{2N+6})$ [4]. Sample $\mu \xleftarrow{\$} \mathbb{Z}_q^*, \mathbf{S}, \mathbf{U}, \xleftarrow{\$} \prod_{i=1}^n (\mathbb{Z}_q^*)^N$ and write $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_n)$, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$. Perform an n-out-of-n secret sharing on 1, that is, choose $p_i \in \mathbb{Z}_q$ such that $1 = p_1 + \cdots + p_n$. Then, for each $i \in [n]$, sample N random values $\theta_{i,k} \xleftarrow{\$} \mathbb{Z}_q$. Output the master secret key and the encryption keys as

$$\begin{cases} \mathsf{msk} := \Big(\mathbf{S},\ \mathbf{U},\ (\theta_{i,k})_{i\in[N],k\in[N]}, (\mathbf{b}^*_{i,k})_{k\in[N+2]},\ \mathbf{f}^*_1,\ \mathbf{f}^*_2,\ \mathbf{f}^*_3, \\ \qquad \mathbf{g}^*_1,\ \mathbf{g}^*_2,\ \mathbf{g}^*_3,\ (\mathbf{h}^*_{i,1}, \mathbf{h}^*_{i,2}, \mathbf{h}^*_{i,3}, (\mathbf{h}^*_{i,N+3+k})_{k=1}^N)_{i\in[n]}\Big) \\ \mathsf{ek}_i := \Big(\mathbf{s}_i,\ \mathbf{u}_i, (B_i^{(k)})_{k\in[N+2]}, \mathbf{b}_{i,N+3},\ \mathbf{f}_1,\ \mathbf{f}_2,\ \mathbf{f}_3, \\ \qquad \mathbf{g}_1,\ \mathbf{g}_2,\ \mathbf{g}_3,\ p_i \cdot H_i^{(1)},\ p_i \cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ (\theta_{i,k}\mathbf{h}_{i,N+3+k})_{k=1}^N\Big) \end{cases}$$

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

$\mathsf{Extract}(\mathsf{msk}, (\mathbf{y}_i)_{i\in[n]} \in \prod_{i=1}^n \mathbb{Z}_q^N, \mathsf{ac\text{-}k} := \mathbb{A})$**:** Let $\mathbb{A}$ be an LSSS-realizable monotone access structure over a set of attributes $\mathsf{Att} \subseteq \mathbb{Z}_q$. First, sample $a_{i,0} \xleftarrow{\$} \mathbb{Z}_q$ and run the labeling algorithm $\Lambda_{a_{i,0}}(\mathbb{A})$ (see Definition 8) to obtain the labels $(a_{i,j})_j$ where j runs over the attributes in Att. In the end, it holds that $a_{i,0} = \sum_{j\in A} c_{i,j} \cdot a_{i,j}$ where j runs over some authorized set $A_i \in \mathbb{A}$ and $\mathbf{c}_i = (c_{i,j})_j$ is the reconstruction vector from LSSS w.r.t A_i. We denote by $\mathsf{List\text{-}Att}(\mathbb{A})$ the list of attributes appearing in $\mathbb{A}$, with possible repetitions. For each $i \in [n]$, each $k \in [N]$, sample $d_{\mathbb{A},i,k} \xleftarrow{\$} \mathbb{Z}_q$ such that $\sum_{i=1}^n \sum_{k=1}^N \theta_{i,k} d_{\mathbb{A},i,k} = 0$. For each $i \in [n]$, compute

$$\mathbf{m}_i := \left(\mathbf{y}_i, \sum_{i=1}^n a_{i,0}, \mathsf{rnd}_i, 0, 0\right)_{\mathbf{B}^*_i}; \widetilde{\mathbf{m}}_{i,j} := (\tilde{\pi}_{i,j} \cdot (j,1),\ a_{i,j},\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{G}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$

$$\mathbf{k}_{i,j} := (\pi_{i,j} \cdot (j,1),\ a_{i,j} \cdot z,\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{F}^*} \text{ for } j \in \mathsf{List\text{-}Att}(\mathbb{A})$$

$$\mathbf{k}_{i,\mathsf{ipfe}} := \left(\sum_{i=1}^n \langle \mathbf{s}_i, \mathbf{y}_i\rangle, \sum_{i=1}^n \langle \mathbf{u}_i, \mathbf{y}_i\rangle,\ a_{i,0} \cdot z,\ 0^N,\ (d_{\mathbb{A},i,k})_{k=1}^N, \mathsf{rnd}_{i,\mathsf{ipfe}}\right)_{\mathbf{H}^*_i}$$

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

[4] We denote the basis changing matrices for $(\mathbf{F}, \mathbf{F}^*), (\mathbf{B}_i, \mathbf{B}^*_i), (\mathbf{H}_i, \mathbf{H}^*_i)$ as $(F, F' := (F^{-1})^\top), (B_i, B'_i := (B_i^{-1})^\top), (H_i, H'_i := (H_i^{-1})^\top)$ respectively (see the appendix A.5 for basis changes in DPVS).

$\mathsf{Enc}(\mathsf{ek}_i, \mathbf{x}_i \in \mathbb{Z}_q^N, z_i := (\mathsf{tag}, \mathsf{S}_i))$**:** Parse

$$\mathsf{ek}_i := \Big(\mathbf{s}_i,\ \mathbf{u}_i, (B_i^{(k)})_{k\in[N+2]}, \mathbf{b}_{i,N+3}, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, p_i \cdot H_i^{(1)},\ p_i \cdot H_i^{(2)},\ \mathbf{h}_{i,3},\ (\theta_{i,k}\mathbf{h}_{i,N+3+k})_{k=1}^N\Big)$$

and $\mathsf{S}_i \subseteq \mathsf{Att} \subseteq \mathbb{Z}_q$ as the set of attributes, compute $\mathsf{H}(\mathsf{tag}) \to (\llbracket\omega\rrbracket_1, \llbracket\omega'\rrbracket_1) \in \mathbb{G}_1^2$. Use $p_i H_i^{(1)}$ and $p_i H_i^{(2)}$ to compute

$$p_i H_i^{(1)} \cdot \llbracket\omega\rrbracket_1 + p_i H_i^{(2)} \cdot \llbracket\omega'\rrbracket_1 = p_i \cdot \Big(\omega H_i^{(1)} \cdot g_1 + \omega' H_i^{(2)} \cdot g_1\Big) = p_i \cdot (\omega \mathbf{h}_{i,1} + \omega' \mathbf{h}_{i,2})\ .$$

For each $j \in \mathsf{S}_i$, sample $\psi_i, \nu_i \xleftarrow{\$} \mathbb{Z}_q$ and compute

$$\begin{aligned}
\widetilde{\mathbf{t}}_{i,j} &= (\widetilde{\sigma}_{i,j} \cdot (1, -j),\ \nu_i,\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{G}} \\
\mathbf{c}_{i,j} &= \sigma_{i,j} \cdot \mathbf{f}_1 - j \cdot \sigma_{i,j} \cdot \mathbf{f}_2 + \psi_i \cdot \mathbf{f}_3 = (\sigma_{i,j} \cdot (1, -j),\ \psi_i,\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{F}}
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{t}_i &:= \sum_{k\in[N]} \Big(\llbracket\omega\rrbracket_1 \cdot \mathbf{s}_i[k] \cdot B_i^{(k)} + \llbracket\omega'\rrbracket_1 \cdot \mathbf{u}_i[k] \cdot B_i^{(k)} + \llbracket\mathbf{x}_i[k]\rrbracket_1\Big) + \nu_i \cdot \mathbf{b}_{i,N+1} + \rho_i \cdot \mathbf{b}_{i,N+3} \\
&= (\omega \cdot \mathbf{s}_i + \omega' \cdot \mathbf{u}_i + \mathbf{x}_i, \nu_i, 0, \rho_i)_{\mathbf{B}_i} \\
\mathbf{c}_{i,\mathsf{ipfe}} &:= p_i \cdot (\omega \cdot \mathbf{h}_{i,1} + \omega' \cdot \mathbf{h}_{i,2}) + \psi_i \cdot \mathbf{h}_{i,3} + \sum_{k=1}^N \theta_{i,k}\mathbf{h}_{i,N+3+k} = (\omega p_i,\ \omega' p_i,\ \psi_i,\ 0^N,\ (\theta_{i,k})_{k=1}^N,\ 0)_{\mathbf{H}_i}
\end{aligned}$$

and output $\mathsf{ct}_{\mathsf{tag},i} := \Big(\Big(\mathbf{c}_{i,j}, \widetilde{\mathbf{t}}_{i,j}\Big)_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}}\Big)$.

$\mathsf{Dec}(\mathsf{dk}_{\mathbb{A},\mathbf{y}}, \mathbf{c} := (\mathsf{ct}_{\mathsf{tag},i}), \mathsf{aux\text{-}d} := \mathsf{tag})$**:** Parse

$$\mathsf{ct}_{\mathsf{tag},i} = \Big(\Big(\mathbf{c}_{i,j}, \widetilde{\mathbf{t}}_{i,j}\Big)_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}}\Big) \quad \text{and} \quad \mathsf{dk}_{\mathbb{A},\mathbf{y}} := \Big((\mathbf{k}_{i,j}, \widetilde{\mathbf{m}}_{i,j})_{i,j}, (\mathbf{m}_i, \mathbf{k}_{i,\mathsf{ipfe}})_{i\in[n]}\Big)\ .$$

For each $i \in [n]$, if there exists $A_i \subseteq \mathsf{S}_i$ and $A_i \in \mathbb{A}$, then compute the reconstruction vector $(c_{i,j})_j$ of for A_i and perform Algorithm 2. Finally, compute the discrete logarithm and output the small value $\mathsf{out} \in [-nNB^2, nNB^2] \subsetneq \mathbb{Z}_q$ [5].

We now state the security theorem. For simplicity, this theorem proves a weaker notion following Definition 11. In the subsequent lemmas we will show how to remove most of the above constraints.

Theorem 12. *Let $\mathcal{E}$ be a MCFE scheme with fine-grained access control for the function class $\mathcal{F}_{\mathsf{subvec}}^{\mathsf{IP},B} \times \mathsf{LSSS}$, given in Section 4.2 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, in the random oracle model, $\mathcal{E}$ is* one-time statically IND-secure *against* complete *challenge queries with* private-inputs only repetitions *(as per Definition 11), under the* SXDH *in $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Concrete Interpretation of Admissibility. Before going into the proof, we present specific conditions for admissible attacks in the case of *one-challenge, complete, with repetitions on private inputs* with repect to Definition 5:

1. For all vectors $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)}, (\mathsf{tag}, \mathsf{S}_i))$ that is queried to **LoR**, for all $((\mathbf{y}_i)_{i\in[n]}, \mathbb{A}) \in \mathcal{Q}$, let $\mathcal{H}$ be the set of honest clients and $b \xleftarrow{\$} \{0,1\}$ be the challenge bit. Then fof any $j_i \in [J_i]$, if $\prod_i \mathbb{A}(\mathsf{S}_i) = 1$ then: $\sum_{i\in\mathcal{H}} \langle \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i\rangle = 0$. This implies $\langle \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i\rangle$ is constant for any $j_i \in [J_i]$. We recall that we are in the *private-inputs only repetitions* and therefore there are no repetitions over $(\mathsf{tag}, \mathsf{S}_i)$.
2. For all vectors $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)}, (\mathsf{tag}, \mathsf{S}_i))$ that is queried to **LoR**, for all $((\mathbf{y}_i)_{i\in[n]}, \mathbb{A}) \in \mathcal{Q}$. Let $\mathcal{C} := [n] \setminus \mathcal{H}$ be the set of corrupted clients. Then, for all $i \in \mathcal{C}$, all $j_i \in [J]$: $\langle \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i\rangle = 0$.

[5] we represent $\mathbb{Z}_q$ as the ring of integers with addition and multiplication modulo q, containing the representatives in the interval $(-q/2,\ q/2)$.

Input: $\mathsf{ct}_{\mathsf{tag},i} = \left(\left(\mathbf{c}_{i,j}, \widetilde{\mathbf{t}}_{i,j}\right)_j, \mathbf{t}_i, \mathbf{c}_{i,\mathsf{ipfe}}\right)$ and $\mathsf{dk}_{\mathbb{A},\mathbf{y}} := \left(\left(\mathbf{k}_{i,j}, \widetilde{\mathbf{m}}_{i,j}\right)_{i,j}, (\mathbf{m}_i, \mathbf{k}_{i,\mathsf{ipfe}})_{i\in[n]}\right)$, as well as the reconstruction vector $(c_{i,j})_j$ of the LSSS for a reconstruction set A_i for each i

1. For each j in the reconstruction set A, compute

$$\widetilde{\mathbf{t}}_{0,j} = \sum_i \widetilde{\mathbf{t}}_{i,j} = (\widetilde{\sigma}_{0,j} \cdot (1, -j),\ \sum_i \nu_i,\ 0^N,\ 0,\ 0^N,\ 0,\ 0)_{\mathbf{G}}$$

   where $\widetilde{\sigma}_{0,j} = \sum_i \widetilde{\sigma}_{i,j}$ being a uniformly random value as $\widetilde{\sigma}_{i,j} \xleftarrow{\$} \mathbb{Z}_q$.
2. For each i compute

$$X_i = \sum_{j\in A_i} \widetilde{\mathbf{t}}_{0,j} \times (c_{i,j} \cdot \widetilde{\mathbf{m}}_{i,j}) = \left[\!\!\left[(\sum_i \nu_i) \cdot \left(\sum_{j\in A_i} c_{i,j} \cdot a_{i,j} \right) \right]\!\!\right]_{\mathsf{t}} = \left[\!\!\left[(\sum_i \nu_i) \cdot a_{i,0} \right]\!\!\right]_{\mathsf{t}}$$

$$Y_i = \sum_{j\in A_i} \mathbf{c}_{i,j} \times (c_{i,j} \cdot \mathbf{k}_{i,j}) = [\![\psi_i \cdot a_{i,0} \cdot z]\!]_{\mathsf{t}}$$

   and in the end summing all X_i to obtain $\mathsf{mask} = \sum_i X_i = [\![(\sum_i \nu_i) \cdot (\sum_i a_{i,0})]\!]_{\mathsf{t}}$
3. Compute

$$W = \sum_i \mathbf{t}_i \times \mathbf{m}_i = \left[\!\!\left[\sum_i \left(\omega \cdot \langle \mathbf{s}_i, \mathbf{y}_i \rangle + \omega' \cdot \langle \mathbf{u}_i, \mathbf{y}_i \rangle + \langle \mathbf{x}_i, \mathbf{y}_i \rangle\right) + (\sum_i \nu_i) \cdot (\sum_i a_{i,0}) \right]\!\!\right]_{\mathsf{t}}$$

   as well as $Z = \sum_i (\mathbf{c}_{i,\mathsf{ipfe}} \times \mathbf{k}_{i,\mathsf{ipfe}} - Y_i) = [\![\omega \cdot \sum_i \langle \mathbf{s}_i, \mathbf{y}_i \rangle + \omega' \cdot \sum_i \langle \mathbf{u}_i, \mathbf{y}_i \rangle]\!]_{\mathsf{t}}$ thanks to $\sum_{i=1}^n \sum_{k=1}^N \theta_{i,k} d_{\mathbb{A},i,k} = 0$ and $\sum_i p_i = 1$.
4. Finally, compute $\mathsf{out} = W - Z - \mathsf{mask} = [\![\sum_i \langle \mathbf{x}_i, \mathbf{y}_i \rangle]\!]_{\mathsf{t}}$ and then a discrete log of out in base g_{t} to obtain $\sum_i \langle \mathbf{x}_i, \mathbf{y}_i \rangle$.

Fig. 2: The final computation of decryption for the MCFE in Section 4.2, whose *correctness* can be verified according to construction. We note that because of all the probabilistic vectorization duringe encryption (cf. **Solution to the First Obstacle** starting from page 7 in Section 1.2), the decryption algorithm is much more complex than the one in [31] and [32].

We recall that these conditions are for the *one-challenge, complete, with repetitions on private inputs* case and are checked in **Finalise** procedure at the end of the security experiment. Particularly, condition 2 is checked for all corrupted clients $i \in \mathcal{C}$ and all $j_i \in [J]$, given any queries that are made to the oracle **LoR** for $i \in \mathcal{C}$ by the adversary. This makes sense even in the case of *static corruption*, since we do *not* prohibit such queries even after the set $\mathcal{C}$ is fixed. Finally, condition 2 does *not* need to cover private inputs of corrupted $i \in \mathcal{C}$ that are not queried to the oracle **LoR** because there exists no challenge bit b in those self-crafted ciphertexts $\mathsf{ct}_{\mathsf{tag},i} \leftarrow \mathsf{Enc}(\mathsf{ek}_i, \mathbf{z}_i, (\mathsf{tag}, \mathsf{S}_i))$. Decrypting $\mathsf{ct}_{\mathsf{tag},i}$ jointly with others challenge ciphertexts $\mathsf{ct}^{(b)}_{\mathsf{tag},j\neq i}$ under some key $\mathsf{dk}_{\mathbb{A},(\mathbf{y}_i)_{i\in[n]}}$ always gives the same i-th component $\langle \mathbf{z}_i, \mathbf{y}_i \rangle$ regardless of b.

Proof Strategy. Before presenting the details of the proof, we give an overview of the strategy. The sequence of games is given in Figure 6, 7, and 8 in Appendix D. The high level objective of each step is described below:

G_0**:** We start from the first game G_0 which is the security experiment for *one-time statically IND-security* against *complete* challenge queries with *private-inputs only repetitions*. For simplicity, we add a constraint that the challenge tag tag is *not* queried to **Enc**. This incurs a multiplicative loss factor in advantage up to an inverse of polynomial in λ, where we can reduce to the normal $\mathsf{1chal}$ by guessing the challenge tag among the tags for encryption, and responding all of its **Enc** queries $(i, \mathbf{x}_i, (\mathsf{tag}, \mathsf{ac\text{-}ct}_i))$ by $\mathbf{LoR}(i, \mathbf{x}_i, \mathbf{x}_i, (\mathsf{tag}, \mathsf{ac\text{-}ct}_i))$.

$\mathsf{G}_0 \to \mathsf{G}_1$**:** To go to G_1, we perform a sequence of hybrids over the key queries, which are indexed by $\ell \in [K]$. The main goal is to introduce $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ for each client $i \in \mathcal{H}$ (known in advance by static corruption) and repetition $j_i \in [J_i]$ in one (block of) coordinates of the challenge components $\mathbf{c}_{i,j}^{(j_i)}$. The corresponding (block of) coordinates in the key component $\mathbf{k}_{i,j}^{(\ell)}$ will be modified accordingly to contain a random copy of $R \cdot \mathbf{y}_i$ for some random $R \xleftarrow{\$} \mathbb{Z}_q$. The details of the reductions are given in the full proof in appendix D, we highlight here the fact that the *correctness* is necessarily preserved thanks to the admissiblity. When the key allows decryption, summing up over all honest clients $i \in \mathcal{H}$ contains

$$R \cdot \sum_{i\in\mathcal{H}} \langle \Delta\mathbf{x}_i, \mathbf{y}_i \rangle = R \cdot \sum_{i\in\mathcal{H}} \langle \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i \rangle = 0 \ . \tag{6}$$

Condition 1 ensures first that $\mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ is constant for all $j_i \in [J_i]$[6] and the sum over index $i \in \mathcal{H}$ is well defined. Finally this sum leads to (6) which is 0 and does not intervene the correct decryption[7]. As a final remark, this step exploits the *probabilistic vectorization* and its randomness (see overview in section 1.2): first to apply Lemma 1 and then follow-ups with the *complexity leveraging* argument, while taking into accounts the private-input repetitions (**Solution to the First/Second Obstacle** in section 1.2, pages 7 and 10).

$\mathsf{G}_1 \to \mathsf{G}_2$**:** After the hybrids $\mathsf{G}_0 \to \mathsf{G}_1$, we proceeed to G_2 to rewrite the adversary's view of the challenge ciphertext component on the aggregation of the honest i for $\widetilde{\mathbf{t}}_{i,j}^{(j_i)}$: $\widetilde{\mathbf{t}}_{0,j}^{(j_i)} = \sum_{i \in \mathcal{H}} \widetilde{\mathbf{t}}_{i,j}^{(j_i)}$. Thanks to *static* corruption, the set $\mathcal{H}$ is known in advance and $\widetilde{\mathbf{t}}_{0,j}^{(j_i)}$ is well defined. This is a completely formal rewriting that conforms to the calculations in the decryption algorithm (Algorithm 2) and hence preserves *correctness*.

$\mathsf{G}_2 \to \mathsf{G}_3$**:** In the next step, we proceed to G_3 by applying the masking lemma (Lemma 1), over the each key $\left(\left(\mathbf{k}_{i,j}^{(\ell)}, \widetilde{\mathbf{m}}_{i,j}^{(\ell)}\right)_{i,j}, (\mathbf{m}_i^{(\ell)}, \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)})_{i \in [n]}\right)$ that is indexed by $\ell \in [K]$. This masking application introduces $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ for each client $i \in \mathcal{H}$ (known in advance by static corruption) and repetition $j_i \in [J_i]$ in one (block of) coordinates of the challenge components $\mathbf{t}_i^{(j_i)}$, while the corresponding (block of) coordinates in the key component $\mathbf{m}_i^{(\ell)}$ will be modified accordingly to contain $R \cdot \mathbf{y}_i$. We remark that this pair of masks are the same as what are introduce in the step $\mathsf{G}_0 \to \mathsf{G}_1$, which is feasible under Lemma 1, and are needed for later steps in the proof. The correctness is preserved thanks to a similar argument as in the previous step.

$\mathsf{G}_3 \to \mathsf{G}_4$**:** We move to the *complexity leveraging* argument. As already briefly introduced in paragraph **Solution to the Third Obstacle** of Section 1, the complexity leveraging argument is a technique that unfolds as follows:

1. We define an event E that happens with fixed probability and whose probability space depends on the data that can be *adaptively* chosen by the adversary. Then, condition on E we move to the *selective* version $\mathsf{G}_i^*, \mathsf{G}_{i+1}^*, \ldots, \mathsf{G}_{i+K}^*$.
2. Next, we want to prove the sequence of *perfect indistinguishability* involving

$$\{\mathsf{G}_i^* \mid E\} \equiv \{\mathsf{G}_{i+1}^* | E\} \equiv \cdots \equiv \{\mathsf{G}_{i+K}^* \mid E\} \tag{7}$$

   where E happens with fixed probability and is *independent* of the view of the adversary during the reductions $\{\mathsf{G}_{i+t}^* \mid E\} \equiv \{\mathsf{G}_{i+t+1}^* | E\}$ in the sequence, for all $t \in [K-1]$.
3. We go back to the orginal *adaptive* games, without resorting to event E, a probabilistic argument concludes that $\{\mathsf{G}_i\} \equiv \{\mathsf{G}_{i+1}\} \equiv \cdots \equiv \{\mathsf{G}_{i+K}\}$. The main idea is given any ppt adaptive adversary, we can construct a simulator of the adaptive games $\{\mathsf{G}_i, \mathsf{G}_{i+1}, \ldots, \mathsf{G}_{i+K}\}$ can *(i)* first guess the adaptively chosen data for event E, *(ii)* interact with its selective challenger, while *(iii)* using the afterwards selective challenger's responses to interact with the adaptive adversary, and *(iv)* in the end, only when E holds, forward the adaptive adversary's final result to the selective challenger.

In the reduction of step 3, the guess at *(i)* is done by the simulator and following the check at *(iv)*, it incurs the simulator's advantage against the selective games being equal to a fixed loss factor $\Pr[E]$ mutiplied to the advantage of the adaptive adversary. However, thanks to the perfect indistinguishability (7), between the selective games for all simulators the advantage is 0. Therefore, for the particular above simulator the advantage is also 0 and that implies the arbitrary adaptive adversary's advantage is 0. It remains constructing the selective games $\{\mathsf{G}_i^*, \mathsf{G}_{i+1}^*, \ldots, \mathsf{G}_{i+K}^*\}$ and proving the perfect indistinguishability (7). To this end, we exploit formal basis changes in DPVS (cf. examples 1, 2, 3 in appendix A.5). In a

[6] This term $\Delta\mathbf{x}_i = \mathbf{0}$ the vector of all 0 when $b = 1$ and can be non-zero when $b = 0$.

[7] There is a step in this transition we already use *complexity leveraging*, for a common explanation we refer to $\mathsf{G}_3 \to \mathsf{G}_4$ below.

simplified notation the (block of) coordinares in ciphertexts and keys are changed as follows:

$$\text{(Formal quotient)} \begin{cases} \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & = (\cdots, \boxed{r \cdot \mathbf{1}_{\Delta \mathbf{x}_i}}, \boxed{r' \cdot \mathbf{1}}, \cdots)_{\mathbf{H}_i}; \\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & = (\cdots, \boxed{R \cdot (\Delta \mathbf{x}_i \circ \mathbf{y}_i^{(\ell)})}, \boxed{(\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N}, \cdots)_{\mathbf{H}_i^*}; \end{cases}$$

$$\text{(Formal switch, } b \text{ to } 1) \equiv \begin{cases} \mathbf{t}_i^{(j_i)} & = (\omega \cdot \mathbf{s}_i + \omega' \cdot \mathbf{u}_i + \boxed{\mathbf{x}_i^{(1,j_i)}}, \cdots, \Delta \mathbf{x}_i, \cdots)_{\mathbf{B}_i} \\ \mathbf{m}_i^{(\ell)} & = (\mathbf{y}_i^{(\ell)}, \cdots, \boxed{R' \cdot \mathbf{y}_i^{(\ell)}}, \cdots)_{\mathbf{B}_i^*} \\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & = (\cdots, r \cdot \mathbf{1}_{\Delta \mathbf{x}_i}, \boxed{(r' + r)} \mathbf{1}, \cdots)_{\mathbf{H}_i} \\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & = (\cdots, \boxed{R' \cdot \left(\Delta \mathbf{x}_i \circ \mathbf{y}_i^{(\ell)}\right)}, \boxed{(\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N}, \cdots)_{\mathbf{H}_i^*} \end{cases}$$

$$\text{(Redo formal quotient)} \equiv \begin{cases} \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & = (\cdots, \boxed{\Delta \mathbf{x}_i}, \boxed{(\theta_{i,k})_{k=1}^N}, 0)_{\mathbf{H}_i} \\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & = (\cdots, \boxed{R' \cdot \mathbf{y}_i^{(\ell)}}, \boxed{(d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N}, \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)})_{\mathbf{H}_i^*} \end{cases}$$

where $R, R' \xleftarrow{\$} \mathbb{Z}_q$, $\Delta \mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ is constant for $i \in \mathcal{H}$ over repetitions. We refer to the definition of the event for guesses in Equation (17) of the main proof, which ensures that under those formal basis changes correctness is preserved necessarily and we obtain the desired effects on vectors.

$\mathsf{G}_4 \to \mathsf{G}_5$**:** The remaining step is to clean auxiliary coordinates we have modified in the previous steps.

The full proof that develops all details of the above steps is given in Appendix D. An application of technical lemmas to enhance security is given in Appendix D.1. We have the following Corollary in the end:

Corollary 13. *We consider 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 the functionality is* $\mathcal{F}_{\mathsf{subvec}}^{\mathsf{IP},B} \times \mathsf{LSSS}$. *Then, there exists a multi-client IPFE scheme with fine-grained access control via LSSS that is* statically IND-secure *in the ROM, against* multiple incomplete *challenge queries with* repetitions on private inputs*, under the* SXDH *assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

Remark 14. **(Towards MIFE for inner products)** Corollary 13 presents an MCFE for subvectors with fine-grained access control so that its security adapted to the case of subvectors (see Definition 6), with *multiple (with possible repetitions on private inputs)* , under a given challenge tag and against *incomplete* queries. We can obtain an MIFE for *inner products* in the standard model by fixing one tag for every ciphertext, *i.e.* the random oracle can be removed by publishing a random fixed value corresponding to H(tag) for encryption. The security of the resulted MIFE is implied from the security of our MCFE in Corollary 13 thanks to the fact that the adversary can make multiple challenge queries to **LoR** for each slot $i \in [n]$, following the admissiblity in Definition 11. In particular, security with possible repetitions on private inputs of the MCFE implies security of the obtained MIFE when repetitive private $\mathbf{x}_i$ are used for the same i. In particular, we obtain an MIFE for inner-products with adaptive security in the standard model, whose keys can be control by LSSS restraining no repetitions on attributes per client.

Allowing Repetitions on Attributes. As mentioned at the beginning of this section, the above Lemma 27 deals with *incomplete* challenge queries, but only with respect to the *private* input $\mathbf{x}_i$ of each client i. It cannot lift our restriction that we do *not* allow repetitions on the public attributes S_i. This explains why this constraint persists in our final MCFE from Corollary 13. The fact that AoNE cannot deal with repetitions on public attributes is also mentioned in recent works [11] and we leave it as potential extension to remove this constraint.

Acknowledgments

This work was supported in part by the French ANR Project ANR-19-CE39-0011 PRESTO and the France 2030 ANR Project ANR-22-PECY-003 SecureCompute.

References

1. Abdalla, M., Benhamouda, F., Gay, R.: From single-input to multi-client inner-product functional encryption. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 552–582. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_19
2. Abdalla, M., Benhamouda, F., Kohlweiss, M., Waldner, H.: Decentralizing inner-product functional encryption. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 128–157. Springer, Heidelberg (Apr 2019). https://doi.org/10.1007/978-3-030-17259-6_5
3. Abdalla, M., Catalano, D., Fiore, D., Gay, R., Ursu, B.: Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 597–627. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_20
4. Abdalla, M., Catalano, D., Gay, R., Ursu, B.: Inner-product functional encryption with fine-grained access control. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part III. LNCS, vol. 12493, pp. 467–497. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64840-4_16
5. Abdalla, M., Gay, R., Raykova, M., Wee, H.: Multi-input inner-product functional encryption from pairings. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 601–626. Springer, Heidelberg (Apr / May 2017). https://doi.org/10.1007/978-3-319-56620-7_21
6. Agrawal, S., Goyal, R., Tomida, J.: Multi-input quadratic functional encryption from pairings. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 208–238. Springer, Heidelberg, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84259-8_8
7. Agrawal, S., Goyal, R., Tomida, J.: Multi-party functional encryption. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part II. LNCS, vol. 13043, pp. 224–255. Springer, Heidelberg (Nov 2021). https://doi.org/10.1007/978-3-030-90453-1_8
8. Agrawal, S., Goyal, R., Tomida, J.: Multi-party functional encryption. In: Theory of Cryptography. Springer International Publishing (2021)
9. Agrawal, S., Goyal, R., Tomida, J.: Multi-input quadratic functional encryption: Stronger security, broader functionality. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. LNCS, vol. 13747, pp. 711–740. Springer, Heidelberg (Nov 2022). https://doi.org/10.1007/978-3-031-22318-1_25
10. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 333–362. Springer, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53015-3_12
11. Agrawal, S., Tomida, J., Yadav, A.: Attribute-based multi-input FE (and more) for attribute-weighted sums. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part IV. LNCS, vol. 14084, pp. 464–497. Springer, Heidelberg (Aug 2023). https://doi.org/10.1007/978-3-031-38551-3_15
12. Agrawal, S., Tomida, J., Yadav, A.: Attribute-based multi-input fe (and more) for attribute-weighted sums. In: Advances in Cryptology - IACR CRYPTO 2023. Springer-Verlag (2023), https://eprint.iacr.org/2023/1191
13. Ananth, P., Jain, A.: Indistinguishability obfuscation from compact functional encryption. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 308–326. Springer, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-47989-6_15
14. Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D. thesis, Technion - Israel Institute of Technology, Haifa, Israel (June 1996), https://www.cs.bgu.ac.il/~beimel/Papers/thesis.pdf
15. Benaloh, J.C., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 27–35. Springer, Heidelberg (Aug 1990). https://doi.org/10.1007/0-387-34799-2_3
16. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19571-6_16
17. Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) PAIRING 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (May 2013). https://doi.org/10.1007/978-3-642-36334-4_8
18. Chotard, J., Dufour Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Decentralized multi-client functional encryption for inner product. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 703–732. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03329-3_24
19. Chotard, J., Dufour Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with repetition for inner product. Cryptology ePrint Archive, Report 2018/1021 (2018), https://eprint.iacr.org/2018/1021
20. Chotard, J., Dufour-Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Dynamic decentralized functional encryption. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 747–775. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56784-2_25
21. Datta, P., Okamoto, T., Tomida, J.: Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part II. LNCS, vol. 10770, pp. 245–277. Springer, Heidelberg (Mar 2018). https://doi.org/10.1007/978-3-319-76581-5_9

22. Datta, P., Pal, T.: Decentralized multi-authority attribute-based inner-product FE: Large universe and unbounded. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 587–621. Springer, Heidelberg (May 2023). https://doi.org/10.1007/978-3-031-31368-4_21
23. Delerablée, C., Gouriou, L., Pointcheval, D.: Key-policy abe with delegation of rights. Cryptology ePrint Archive, Report 2021/867 (2021), https://ia.cr/2021/867
24. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_8
25. Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 578–602. Springer, Heidelberg (May 2014). https://doi.org/10.1007/978-3-642-55220-5_32
26. Gordon, S.D., Katz, J., Liu, F.H., Shi, E., Zhou, H.S.: Multi-input functional encryption. Cryptology ePrint Archive, Report 2013/774 (2013), https://eprint.iacr.org/2013/774
27. Lai, Q., Liu, F.H., Wang, Z.: New lattice two-stage sampling technique and its applications to functional encryption - stronger security and smaller ciphertexts. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 498–527. Springer, Heidelberg (Oct 2021). https://doi.org/10.1007/978-3-030-77870-5_18
28. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2_27
29. Libert, B., Titiu, R.: Multi-client functional encryption for linear functions in the standard model from LWE. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 520–551. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_18
30. Nguyen, D.: Dynamic decentralized functional encryptions from pairings in the standard model. Cryptology ePrint Archive, Paper 2024/580 (2024), https://eprint.iacr.org/2024/580
31. Nguyen, K., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with fine-grained access control. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part I. LNCS, vol. 13791, pp. 95–125. Springer, Heidelberg (Dec 2022). https://doi.org/10.1007/978-3-031-22963-3_4
32. Nguyen, K., Phan, D.H., Pointcheval, D.: Optimal security notion for decentralized multi-client functional encryption. In: Tibouchi, M., Wang, X. (eds.) ACNS 23, Part II. LNCS, vol. 13906, pp. 336–365. Springer, Heidelberg (Jun 2023). https://doi.org/10.1007/978-3-031-33491-7_13
33. Nguyen, K., Pointcheval, D., Schädlich, R.: Dynamic decentralized functional encryption with strong security. Cryptology ePrint Archive, Paper 2022/1532 (2022), https://eprint.iacr.org/2022/1532
34. Nguyen, K., Pointcheval, D., Schädlich, R.: Decentralized multi-client functional encryption with strong security. IACR Communications in Cryptology **1**(2) (2024). https://doi.org/10.62056/andkp2fgx
35. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_11
36. Okamoto, T., Takashima, K.: Adaptively attribute-hiding (hierarchical) inner product encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 591–608. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_35
37. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (Dec 2012). https://doi.org/10.1007/978-3-642-34961-4_22
38. Pal, T., Dutta, R.: Attribute-based access control for inner product functional encryption from LWE. In: Longa, P., Ràfols, C. (eds.) LATINCRYPT 2021. LNCS, vol. 12912, pp. 127–148. Springer, Heidelberg (Oct 2021). https://doi.org/10.1007/978-3-030-88238-9_7
39. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176
40. Shi, E., Vanjani, N.: Multi-client inner product encryption: Function-hiding instantiations without random oracles. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 622–651. Springer, Heidelberg (May 2023). https://doi.org/10.1007/978-3-031-31368-4_22
41. Tomida, J.: Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 459–488. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_16
42. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-03356-8_36

A Additional Definitions

A.1 Hardness Assumptions

We state the assumptions needed for our constructions.

Definition 15. *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 16. *In the bilinear setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$, the* **Symmetric eXternal Diffie-Hellman** (SXDH) *assumption makes the* DDH *assumption in both $\mathbb{G}_1$ and $\mathbb{G}_2$.*

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

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

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

A.2 Dual Pairing Vector Spaces

Our constructions rely on the *Dual Pairing Vector Spaces* (DPVS) framework in prime-order bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$ and $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ are all written additively. The DPVS technique dates back to the seminal work by Okamoto-Takashima [35, 36, 37] aiming at adaptive security for ABE as well as IBE, together with the *dual system methodology* introduced by Waters [42]. In [28], the setting for dual systems is composite-order bilinear groups. Continuing on this line of works, Chen *et al.* [17] used prime-order bilinear groups under the SXDH assumption. Let us fix $N \in \mathbb{N}$ and consider $\mathbb{G}_1^N$ having N copies of $\mathbb{G}_1$. Any $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1 \in \mathbb{G}_1^N$ is identified as the vector $(x_1, \ldots, x_N) \in \mathbb{Z}_q^N$. There is no ambiguity because $\mathbb{G}_1$ is a cyclic group of order q prime. The $\mathbf{0}$-vector is $\mathbf{0} = [\![(0, \ldots, 0)]\!]_1$. The addition of two vectors in $\mathbb{G}_1^N$ is defined by coordinate-wise addition. The scalar multiplication of a vector is defined by $t \cdot \mathbf{x} := [\![t \cdot (x_1, \ldots, x_N)]\!]_1$, where $t \in \mathbb{Z}_q$ and $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1$. The additive inverse of $\mathbf{x} \in \mathbb{G}_1^N$ is defined to be $-\mathbf{x} := [\![(-x_1, \ldots, -x_N)]\!]_1$. Viewing $\mathbb{Z}_q^N$ as a vector space of dimension N over $\mathbb{Z}_q$ with the notions of bases, we can obtain naturally a similar notion of bases for $\mathbb{G}_1^N$. More specifically, any invertible matrix $B \in GL_N(\mathbb{Z}_q)$ identifies a basis $\mathbf{B}$ of $\mathbb{G}_1^N$, whose i-th row $\mathbf{b}_i$ is $[\![B^{(i)}]\!]_1$, where $B^{(i)}$ is the i-th row of B. The canonical basis $\mathbf{A}$ of $\mathbb{G}_1^N$ consists of $\mathbf{a}_1 := [\![(1, 0 \ldots, 0)]\!]_1, \mathbf{a}_2 := [\![(0, 1, 0 \ldots, 0)]\!]_1, \ldots, \mathbf{a}_N := [\![(0, \ldots, 0, 1)]\!]_1$. It is straightforward that we can write $\mathbf{B} = B \cdot \mathbf{A}$ for any basis $\mathbf{B}$ of $\mathbb{G}_1^N$ corresponding to an invertible matrix $B \in GL_N(\mathbb{Z}_q)$. We write $\mathbf{x} = (x_1, \ldots, x_N)_{\mathbf{B}}$ to indicate the representation of $\mathbf{x}$ in the basis $\mathbf{B}$, i.e. $\mathbf{x} = \sum_{i=1}^N x_i \cdot \mathbf{b}_i$. By convention the writing $\mathbf{x} = (x_1, \ldots, x_N)$ concerns the canonical basis $\mathbf{A}$.

Treating $\mathbb{G}_2^N$ similarly, we can furthermore define a product of $\mathbf{x} = [\![(x_1, \ldots, x_N)]\!]_1 \in \mathbb{G}_1^N, \mathbf{y} = [\![(y_1, \ldots, y_N)]\!]_2 \in \mathbb{G}_2^N$ by $\mathbf{x} \times \mathbf{y} := \prod_{i=1}^N \mathbf{e}(\mathbf{x}[i], \mathbf{y}[i]) = [\![\langle (x_1, \ldots, x_N), (y_1, \ldots, y_N) \rangle]\!]_t$. Given a basis $\mathbf{B} = (\mathbf{b}_i)_{i \in [N]}$ of $\mathbb{G}_1^N$, we define $\mathbf{B}^*$ to be a basis of $\mathbb{G}_2^N$ by first defining $B' := (B^{-1})^\top$ and the i-th row $\mathbf{b}_i^*$ of $\mathbf{B}^*$ is $[\![B'^{(i)}]\!]_2$. It holds that $B \cdot (B')^\top = I_N$ the identity matrix and $\mathbf{b}_i \times \mathbf{b}_j^* = [\![\delta_{i,j}]\!]_t$ for every $i, j \in [N]$, where $\delta_{i,j} = 1$ if and only if $i = j$. We call the pair $(\mathbf{B}, \mathbf{B}^*)$ a *pair of dual orthogonal bases* of $(\mathbb{G}_1^N, \mathbb{G}_2^N)$. If $\mathbf{B}$ is constructed by a random invertible matrix $B \xleftarrow{\$} GL_N(\mathbb{Z}_q)$, we call the resulting $(\mathbf{B}, \mathbf{B}^*)$ a pair of random dual bases. A DPVS is a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q, N)$ with dual orthogonal bases. In this work, we also use extensively *basis changes* over dual orthogonal bases of a DPVS to argue the steps of switching key as well as ciphertext vectors to semi-functional mode in our proofs. The details of such basis changes are recalled in the appendix A.5.

A.3 Access Structure and Linear Secret Sharing Schemes

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

Definition 18 (LSSS [14]). *Let K be a field, $d, f \in \mathbb{N}$, and Att be a finite universe of attributes. A* Linear Secret Sharing Scheme LSSS *over K for an access structure $\mathbb{A}$ over Att is specified by a share-generating matrix $\mathbf{A} \in K^{d \times f}$ such that for any $I \subset [d]$, there exists a vector $\mathbf{c} \in K^d$ with support I and $\mathbf{c} \cdot \mathbf{A} = (1, 0, \ldots, 0)$ if and only if $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$.*

In order to share s using an LSSS over K, one first picks uniformly random values $v_2, v_3, \ldots, v_f \xleftarrow{\$} K$ and the share for an attribute att_i is the i-th coordinate $\mathbf{s}[i]$ of the share vector $\mathbf{s} := (s, v_2, v_3, \ldots, v_f) \cdot \mathbf{A}^\top$. Then, only an authorized set $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$ for some $I \subseteq [d]$ can recover $\mathbf{c}$ to reconstruct s from the shares by: $\mathbf{c} \cdot \mathbf{s}^\top = \mathbf{c} \cdot (\mathbf{A} \cdot (s, v_2, v_3, \ldots, v_f)^\top) = s$. Some canonical examples of LSSS include Shamir's secret sharing scheme for any f-out-of-d threshold gate [39] or Benaloh and Leichter's scheme for any monotone formula [15]. An access structure $\mathbb{A}$ is said to be LSSS-*realizable* if there exists a linear secret sharing scheme implementing $\mathbb{A}$.

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

$$\Lambda_y(\mathbb{A}) := (y, v_2, v_3, \ldots, v_f) \cdot \mathbf{A}^\top \in \mathbb{Z}_q^d \tag{8}$$

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

A.4 More Cryptographic Primitives

We recall necessary cryptographic primitives used in this work.

Key-policy Attribute-Based Encryption (KP-ABE). A *key-policy attribute-based encryption* scheme is defined by a tuple of algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. The Setup algorithm takes as input a security parameter 1^λ and outputs a public key pk and a master secret key msk. The KeyGen algorithm takes as input a master secret key msk, a policy $\mathbb{A}$, and outputs a secret key $\mathsf{sk}_{\mathbb{A}}$. The Enc algorithm takes as input a public key pk, a message m in some message space $\mathcal{M}$, and a set of attributes S, and outputs a ciphertext ct_{S}. The Dec algorithm takes as input a secret key $\mathsf{sk}_{\mathbb{A}}$ and a ciphertext ct_{S}, and outputs a message m. A KP-ABE is *correct* if for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathbb{A} \in \mathsf{Pol}$, all $\mathsf{S} \subseteq \mathsf{Att}$, all $m \in \mathcal{M}$, and all $\mathsf{sk}_{\mathbb{A}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \mathbb{A})$, if Pol accepts S, it holds that $\mathsf{Dec}(\mathsf{sk}_{\mathbb{A}}, \mathsf{Enc}(\mathsf{pk}, m, \mathsf{S})) = m$.

The *security* of a KP-ABE is defined below.

Definition 19. *A KP-ABE scheme $\mathcal{E}$ with respect to a class of policies Pol having attribute space Att is* CPA-secure *if for every ppt adversary $\mathcal{A}$, the following probability is negligible in λ:*

$$\mathsf{Adv}^{\mathsf{kpabe}}_{\mathsf{Pol},\mathsf{Att},\mathcal{A}}(1^\lambda) := \left| \Pr[\mathsf{Expr}^{\mathsf{kpabe}}_{\mathsf{Pol},\mathsf{Att},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \right|$$

where the experiment $\mathsf{Expr}^{\mathsf{kpabe}}_{\mathsf{Pol},\mathsf{Att},\mathcal{A}}(1^\lambda)$ is defined as follows:

1. *The challenger runs* $\mathsf{Setup}(1^\lambda)$ *to obtain* $(\mathsf{pk}, \mathsf{msk})$ *and outputs* pk *to* $\mathcal{A}$*. In the following the adversary* $\mathcal{A}$ *can make queries adaptively in any order before Finalize.*
2. *(Key queries) The adversary* $\mathcal{A}$ *adaptively outputs a policy* $\mathbb{A}$*. The challenger runs* $\mathsf{sk}_{\mathbb{A}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \mathbb{A})$ *and returns* $\mathsf{sk}_{\mathbb{A}}$ *to* $\mathcal{A}$*.*
3. *(Challenge) The adversary* $\mathcal{A}$ *outputs a pair of messages* (m_0, m_1) *and a set of attributes* S^**. The challenger chooses a bit* $b \in \{0, 1\}$ *and runs* $\mathsf{ct}_{\mathsf{S}^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b, \mathsf{S}^*)$*.*
4. *(Finalize) The adversary* $\mathcal{A}$ *outputs a guess* $\hat{b}$*. If there exists a policy* $\mathbb{A}$ *such that* S^* *satisfies* $\mathbb{A}$*, then the expriment outputs* 0*. Otherwise, the experiment outputs* $\hat{b} \stackrel{?}{=} b$*.*

We can define similar weaker notions of *selective* challenge message and/or *selective* challenges attributes.

Functional Encryption (FE). Below is a recall of the syntax and security of (public key) single client FE.

Definition 20. *A* functional encryption *scheme for a class* $\mathcal{F}$ *is defined by a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$*. The* Setup *algorithm takes as input a security parameter* 1^λ *and outputs a public key* pk *and a master secret key* msk*. The* $\mathsf{Extract}$ *algorithm takes as input a master secret key* msk *and a function description* $F_\lambda : \mathcal{M}_\lambda \to \mathcal{R}_\lambda$*, and outputs a secret key* sk_F*. The* Enc *algorithm takes as input a public key* pk*, a message* m *in some message space* $\mathcal{M}$*, outputs a ciphertext* ct*. The* Dec *algorithm takes as input a secret key* sk_F *and a ciphertext* ct*, and outputs an element in* $\mathcal{R}$*. An FE for a class* $\mathcal{F}$ *is* correct *if for all* $\lambda \in \mathbb{N}$*, all* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*, all* $F_\lambda \in \mathcal{F}$*, all* $m \in \mathcal{M}$*, and all* $\mathsf{sk}_F \leftarrow \mathsf{Keygen}(F_\lambda, \mathsf{msk})$*, it holds that* $\mathsf{Dec}(\mathsf{sk}_F, \mathsf{Enc}(\mathsf{pk}, m)) = F_\lambda(m)$*.*

The *security* of an FE scheme is defined below.

Definition 21. *A FE scheme* $\mathcal{E}$ *with respect to a class of functions* $\mathcal{F}$ *is* CPA-secure *if for every* ppt *adversary* $\mathcal{A}$*, the following probability is negligible in* λ*:*

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

where the experiment $\mathsf{Expr}^{\mathsf{fe}}_{\mathcal{E},\mathcal{F},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

1. *The challenger runs* $\mathsf{Setup}(1^\lambda)$ *to obtain* $(\mathsf{pk}, \mathsf{msk})$ *and outputs* pk *to* $\mathcal{A}$*. In the following the adversary* $\mathcal{A}$ *can make queries adaptively in any order before Finalize.*
2. *(Key queries) The adversary* $\mathcal{A}$ *adaptively outputs a function description* F_λ*. The challenger runs* $\mathsf{sk}_F \leftarrow \mathsf{Extract}(F_\lambda, \mathsf{msk})$ *and returns* sk_F *to* $\mathcal{A}$*.*
3. *(Challenge) The adversary* $\mathcal{A}$ *outputs a pair of messages* (m_0, m_1)*. The challenger chooses a bit* $b \in \{0, 1\}$ *and runs* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$*.*
4. *(Finalize) The adversary* $\mathcal{A}$ *outputs a guess* $\hat{b}$*. If there exists a function description* F_λ *such that* $F(m_0) \neq F(m_1)$*, then the expriment outputs* 0*. Otherwise, the experiment outputs* $\hat{b} \stackrel{?}{=} b$*.*

We can define similar weaker notions of *selective* challenge message and/or *selective* functional decryptionkey queries. The notion of FE with *access control* can be captured by considering the class $\mathcal{F}$ that does not only include the calulating function F_λ, but also the access control policies $\mathbb{A}$ given any member $(F_\lambda, \mathbb{A})$ in $\mathcal{F}$ (see Section 4.1 for a formal treatment in the case of MCFE). The *correctness* is adapted that the decryption key $\mathsf{sk}_{F,\mathbb{A}}$ can only decrypt the ciphertexts ct to $F(m)$ if the access control policy $\mathbb{A}$ accepts the attributes S of the ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m, \mathsf{S})$. The notion of *security* is defined similarly as Definition 21, except that the syntax is adapted to the FE with access control.

Multi-Input Functional Encryption (MIFE). We recall in the following the syntax and security of multi-input functional encryption, following [25].

Definition 22. *A* multi-input functional encryption *scheme is defined by a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$*. The* Setup *algorithm takes as input a security parameter* 1^λ *and a number of slots* n*, and outputs a public parameter* pp*, a master secret key* msk*, and* n *encryption keys* ek_i*. The* $\mathsf{Extract}$ *algorithm takes as input a function description* $F_\lambda : \prod_{i=1}^n \mathcal{D}_{\lambda,i} \to \mathcal{R}_\lambda$ *and the master secret key* msk*, and outputs a decryption key* dk_F*. The* Enc *algorithm takes as input an encryption key* ek_i *and a message* m_i *in some message space* $\mathcal{D}_{\lambda,i}$*, and outputs a ciphertext* ct_i*. The* Dec *algorithm takes as input a decryption key* dk_F *and a vector of ciphertexts* ct_i *of length* n*, and outputs an element in* $\mathcal{R}_\lambda$ *or* $\perp$*. An MIFE for a class* $\mathcal{F}$ *is* correct *if for all* $\lambda \in \mathbb{N}$*, all* $(\mathsf{pp}, \mathsf{msk}, (\mathsf{ek}_i)_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$*, all* $F_\lambda \in \mathcal{F}$*, all* $m_i \in \mathcal{D}_{\lambda,i}$*, and all* $\mathsf{dk}_F \leftarrow \mathsf{Extract}(F_\lambda, \mathsf{msk})$*, it holds that* $\mathsf{Dec}(\mathsf{dk}_{F_\lambda}, (\mathsf{Enc}(\mathsf{ek}_i, m_i))_{i\in[n]}) = F_\lambda(m_i)_{i\in[n]}$*.*

The *security* of an MIFE is defined below.

Definition 23. *An MIFE scheme* $\mathcal{E}$ *with respect to a class of functions* $\mathcal{F}$ *is* secure *if for every* ppt *adversary* $\mathcal{A}$*, the following probability is negligible in* λ*:*

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

where the experiment $\mathsf{Expr}^{\mathsf{mife}}_{\mathcal{F},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

1. *The challenger runs* $\mathsf{Setup}(1^\lambda, 1^n)$ *to obtain* $(\mathsf{pp}, \mathsf{msk}, (\mathsf{ek}_i)_{i\in[n]})$ *and outputs* pp *to* $\mathcal{A}$*. In the following the adversary* $\mathcal{A}$ *can make queries adaptively in any order before Finalize.*
2. *(Corruption) In the works of [4, 9], the adversary against the MIFE is futhermore allowed to corrupt* ek_i *for some* $i \in [n]$*. This notion of security for MIFE with corruption allows one more oracle for the adversary to corrupt* ek_i *for any slot* $i \in [n]$ *of their choices.*
3. *(Key queries) The adversary* $\mathcal{A}$ *adaptively outputs a function description* F_λ*. The challenger runs* $\mathsf{dk}_F \leftarrow \mathsf{Extract}(F_\lambda, \mathsf{msk})$ *and returns* dk_F *to* $\mathcal{A}$*.*
4. *(Challenge) The adversary* $\mathcal{A}$ *outputs a query* $(i, m_i^{(0)}, m_i^{(1)})$ *for some* $i \in [n]$*. The challenger chooses a bit* $b \in \{0,1\}$ *and encrypts* $m_i^{(b)}$ *to obtain* $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{ek}_i, m_i^{(b)})$*. The ciphertext* ct_i *is returned to* $\mathcal{A}$*.*
5. *(Encryption) The adversary* $\mathcal{A}$ *outputs a query* (i, m_i) *for some* $i \in [n]$*. The challenger encrypts* m_i *to obtain* $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{ek}_i, m_i)$*. The ciphertext* ct_i *is returned to* $\mathcal{A}$*.*
6. *(Finalize) The adversary* $\mathcal{A}$ *outputs a guess* $\hat{b}$*. If the following condition is satisfied, the experiment outputs* $\hat{b} \stackrel{?}{=} b$*: let* $I \subset [n]$ *be the set of corrupted indices, for* $b \in \{0,1\}$ *we define* $\mathbf{X}^{(b)} := \{x_{1,j}^{(b)}, \ldots, {}_{n,j}\}_{j=1}^q$ *to be the queried challenges*
   (a) *The pair* $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}$ *satisfies that for all* F *queried by* $\mathcal{A}$*, all* $I' = \{i_1, \ldots, i_t\} \subseteq I \cup \varnothing$*, all* $\{x'_{i_1}, \ldots, x'_{i_t}\}$*, all* $j_1, \ldots, j_{n-t} \in [q]$ *we have*

$$\begin{aligned} &F\left(\mathsf{order}\left(x^{(0)}_{i_1,j_1}, \ldots, x^{(0)}_{i_{n-t},j_{n-t}}, x'_{i_1}, \ldots, x'_{i_t}\right)\right) \\ &\quad = F\left(\mathsf{order}\left(x^{(1)}_{i_1,j_1}, \ldots, x^{(1)}_{i_{n-t},j_{n-t}}, x'_{i_1}, \ldots, x'_{i_t}\right)\right) \end{aligned}$$

   (b) *The set* $\{F\}$ *queried by* $\mathcal{A}$ *satisfies that for all* $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}$ *challenges, all* $I' = \{i_1, \ldots, i_t\} \subseteq I \cup \varnothing$*, all* $\{x'_{i_1}, \ldots, x'_{i_t}\}$*, all* $j_1, \ldots, j_{n-t} \in [q]$ *we have*

$$\begin{aligned} &F\left(\mathsf{order}\left(x^{(0)}_{i_1,j_1}, \ldots, x^{(0)}_{i_{n-t},j_{n-t}}, x'_{i_1}, \ldots, x'_{i_t}\right)\right) \\ &\quad = F\left(\mathsf{order}\left(x^{(1)}_{i_1,j_1}, \ldots, x^{(1)}_{i_{n-t},j_{n-t}}, x'_{i_1}, \ldots, x'_{i_t}\right)\right) \end{aligned}$$

   such that the ℓ*-input receives its correspond value by the permutation* $\mathsf{order}(\cdot)$*. Otherwise, the experiment outputs* 0*.*

We can define similar weaker notions of *selective* challenge message and/or *selective* functional decryption key queries. The notion of MIFE with *access control* can be done in the same manner as we do for FE with access control in the previous paragraph. The *correctness* is adapted that the decryption key $\mathsf{sk}_{F,\mathbb{A}}$ can only decrypt the ciphertexts $(\mathsf{ct}_i)_i$ to $F((m_i)_i)$ if the access control policy $\mathbb{A}$ accepts the attributes S_i of the ciphertext $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i, \mathsf{S}_i)$ for all slots $i \in [n]$.

A.5 Dual Pairing Vector Spaces - Basis Changes

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 $\mathbf{B} := B \cdot \mathbf{U}$, which means:

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

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

$$(x_1, \ldots, x_N)_{\mathbf{B}} = ((x_1, \ldots, x_N) \cdot B)_{\mathbf{U}} \ ; \quad (y_1, \ldots, y_N)_{\mathbf{U}} = \left((y_1, \ldots, y_N) \cdot B^{-1}\right)_{\mathbf{B}} \ . \tag{9}$$

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

$$(x_1, \ldots, x_N)_{\mathbf{B}^*} = \left((x_1, \ldots, x_N) \cdot B'\right)_{\mathbf{U}^*} \ ; \quad (y_1, \ldots, y_N)_{\mathbf{U}^*} = \left((y_1, \ldots, y_N) \cdot B^\top\right)_{\mathbf{B}^*} \ . \tag{10}$$

It 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 are taken from the identity matrix I_N.

The basis changes are particularly useful in our security proofs. Intuitively these changes constitute a transition from a hybrid G having vectors expressed in $(\mathbf{U}, \mathbf{U}^*)$ to the next hybrid $\mathsf{G}_{\mathsf{next}}$ having vectors expressed in $(\mathbf{B}, \mathbf{B}^*)$. We focus on two types of basis changes, which are elaborated below. For simplicity, we consider dimension $N = 2$:

Formal Basis Changes: We change $(\mathbf{U}, \mathbf{U}^*)$ into $(\mathbf{B}, \mathbf{B}^*)$ using

$$\begin{aligned} B &:= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}_{1,2} & B' &:= \left(B^{-1}\right)^\top = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}_{1,2} \\ \mathbf{B} &= B \cdot \mathbf{U} & \mathbf{B}^* &= B' \cdot \mathbf{U}^* \ . \end{aligned}$$

We use this type in situations such as: in G we have vectors *all* of the form $(x_1, 0)_{\mathbf{U}}, (y_1, 0)_{\mathbf{U}^*}$, and we want to go to $\mathsf{G}_{\mathsf{next}}$ having vectors *all* of the form $(x_1, 0)_{\mathbf{B}}, (y_1, \boxed{y_1})_{\mathbf{B}^*}$. The simulator writes *all* vectors $(x_1, 0)_{\mathbf{U}}, (y_1, 0)_{\mathbf{U}^*}$ in $(\mathbf{U}, \mathbf{U}^*)$ and under this basis change they are written into

$$(x_1,\ 0)_{\mathbf{U}} = (x_1 - 0,\ 0)_{\mathbf{B}} = (x_1,\ 0)_{\mathbf{B}}; \quad (y_1,\ 0)_{\mathbf{U}^*} = (y_1,\ 0 + y_1)_{\mathbf{B}^*} = (y_1,\ y_1)_{\mathbf{B}^*}$$

following the calculations in (9) and (10). The products between two dual vectors are invariant, *all* vectors are formally written from $(\mathbf{U}, \mathbf{U}^*)$ (corresponding to G) to $(\mathbf{B}, \mathbf{B}^*)$ (corresponding to $\mathsf{G}_{\mathsf{next}}$), the adversary's view over the vectors is thus identical from G to $\mathsf{G}_{\mathsf{next}}$. In particular, this is a kind of *information-theoretic property* of DPVS by basis changing that we exploit to have identical hybrids' hop in the security proof. We list some formal basis changes that are extensively used in this work:

1. *(Duplication)* This is the above example, vectors $\mathbf{b}_2, \mathbf{b}_1^*$ are secret:

$$B := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}_{1,2} \qquad B' := \left(B^{-1}\right)^\top = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}_{1,2}$$
$$\mathbf{B} = B \cdot \mathbf{U} \qquad \mathbf{B}^* = B' \cdot \mathbf{U}^* \ .$$

and $\{(x_1, 0)_{\mathbf{U}}, (y_1, 0)_{\mathbf{U}^*}\} \equiv \left\{(x_1, 0)_{\mathbf{B}}, (y_1, \boxed{y_1})_{\mathbf{B}^*}\right\}$.

2. *(Quotient, by randomness $r \xleftarrow{\$} \mathbb{Z}_q^*$)* The matrices, vector $\mathbf{b}_1$ is secret, are:

$$B := \begin{bmatrix} r & 0 \\ 0 & 1 \end{bmatrix}_{1,2} \qquad B' := \left(B^{-1}\right)^\top = \begin{bmatrix} 1/r & 0 \\ 0 & 1 \end{bmatrix}_{1,2}$$
$$\mathbf{B} = B \cdot \mathbf{U} \qquad \mathbf{B}^* = B' \cdot \mathbf{U}^* \ .$$

and $\{(x_1, 0)_{\mathbf{U}}, (y_1, 0)_{\mathbf{U}^*}\} \equiv \left\{(\boxed{x_1 \cdot r}, 0)_{\mathbf{B}}, (\boxed{y_1 \cdot 1/r}, 0)_{\mathbf{B}^*}\right\}$.

3. *(Formal Switch)* this is the same as *(Duplication)*, but the starting coordinates are not 0:

$$B := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}_{1,2} \qquad B' := \left(B^{-1}\right)^\top = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}_{1,2}$$
$$\mathbf{B} = B \cdot \mathbf{U} \qquad \mathbf{B}^* = B' \cdot \mathbf{U}^* \ .$$

and $\{(x_1, x_2)_{\mathbf{U}}, (y_1, y_2)_{\mathbf{U}^*}\} \equiv \left\{(\boxed{x_1 - x_2}, x_2)_{\mathbf{B}}, (y_1, \boxed{y_2 + y_1})_{\mathbf{B}^*}\right\}$.

<u>*Computational Basis Change:*</u> Given an instance of a computational problem, *e.g.* $[\![(a, b, c)]\!]_1$ of DDH in $\mathbb{G}_1$ where $c - ab = 0$ or $\delta \xleftarrow{\$} \mathbb{Z}_q$, we change $(\mathbf{U}, \mathbf{U}^*)$ into $(\mathbf{B}, \mathbf{B}^*)$ using

$$B := \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix}_{1,2} \qquad B' := \left(B^{-1}\right)^\top = \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}_{1,2}$$
$$\mathbf{B} = B \cdot \mathbf{U} \qquad \mathbf{B}^* = B' \cdot \mathbf{U}^* \ .$$

One situation where this type of basis change can be useful is: in G we have *some* target vectors of the form $(0, \mathsf{rnd})_{\mathbf{U}}$, where $\mathsf{rnd} \xleftarrow{\$} \mathbb{Z}_q$ is a random scalar, together with other $(z_1, z_2)_{\mathbf{U}}$, and *all* the dual is of the form $(0,\ y_2)_{\mathbf{U}^*}$. We want to go to $\mathsf{G}_{\mathsf{next}}$ having $(\boxed{\widetilde{\mathsf{rnd}}},\ \mathsf{rnd})_{\mathbf{B}}$ masked by some randomness $\widetilde{\mathsf{rnd}} \xleftarrow{\$} \mathbb{Z}_q$, while keeping $(0,\ y_2)_{\mathbf{B}^*}$. Because $[\![a]\!]_1$ is given, the simulator can simulate vectors $(z_1, z_2)_{\mathbf{U}}$ directly in $\mathbf{B}$ using $[\![a]\!]_1$ as well as the known coordinates z_1, z_2. The basis change will be employed for the simulation of target vectors:

$$(c,\ b)_{\mathbf{U}} + (0, \mathsf{rnd})_{\mathbf{B}} = (c - a \cdot b,\ \mathsf{rnd} + b)_{\mathbf{B}};$$
$$(0,\ y_2)_{\mathbf{U}^*} = (0,\ y_2 + a \cdot 0)_{\mathbf{B}^*} = (0,\ y_2)_{\mathbf{B}^*}$$

where *all* vectors in $\mathbf{B}^*$ must be written first in $\mathbf{U}^*$, since we do not have $[\![a]\!]_2$, to see how the basis change affects them. Using the basis change we simulate those target vectors by $(c - a \cdot b,\ \mathsf{rnd} + b)_{\mathbf{B}}$ with rnd implicitly being updated to $\mathsf{rnd} + b$, the uninterested $(z_1, z_2)_{\mathbf{B}}$ are simulated correctly in $\mathbf{B}$, meanwhile the dual vectors $(0,\ y_2)_{\mathbf{B}^*}$ stays the same. Depending on the DDH instance, if $c - ab = 0$ the target vectors are in fact $(0,\ \mathsf{rnd})_{\mathbf{B}}$ and we are simulating G, else $c - ab = \delta \xleftarrow{\$} \mathbb{Z}_q$ the target vectors are simulated for $\mathsf{G}_{\mathsf{next}}$ and $\widetilde{\mathsf{rnd}} := \delta$. Hence, under the hardness of DDH in $\mathbb{G}_1$, a computationally bounded adversary cannot distinguish its views in the hybrids' hop from G to $\mathsf{G}_{\mathsf{next}}$. Under the SXDH assumption in the DPVS setting, we list some computational basis changes that are extensively used in this work:

1. *(Subspace)* Given the DDH instance $[\![(a, b, c)]\!]$ in the group w.r.t $\mathbf{B}$, this is the above example, the matrices, vectors $\mathbf{b}_2, \mathbf{b}_1^*$ are secret, are:

$$B := \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix}_{1,2} \qquad B' := \left(B^{-1}\right)^\top = \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}_{1,2}$$
$$\mathbf{B} = B \cdot \mathbf{U} \qquad \mathbf{B}^* = B' \cdot \mathbf{U}^* \ .$$

and $\{(z_1, z_2)_{\mathbf{B}}, (0, \mathsf{rnd})_{\mathbf{U}}, (0,\ y_2)_{\mathbf{B}^*}\} \approx_c \left\{(z_1, z_2)_{\mathbf{B}}, (\boxed{\widetilde{\mathsf{rnd}}},\ \mathsf{rnd})_{\mathbf{B}}, (0,\ y_2)_{\mathbf{B}^*}\right\}$.

2. *(Swap)* Given the DDH instance $[\![(a, b, c)]\!]$ in the group w.r.t $\mathbf{B}$, the matrices, vectors $\mathbf{b}_3, \mathbf{b}_1^*, \mathbf{b}_2^*$ are secret, are:

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

$$\mathbf{B} = B \cdot \mathbf{U} \qquad \mathbf{B}^* = B' \cdot \mathbf{U}^* \ .$$

and $\{(z_1, z_2, z_3)_{\mathbf{B}}, (x, 0, y)_{\mathbf{U}}, (r, r, r')_{\mathbf{U}^*}\} \approx_c \left\{(z_1, z_2, z_3)_{\mathbf{B}}, (\boxed{0, x}, y)_{\mathbf{B}}, (r, r, r')_{\mathbf{B}^*}\right\}$.

We remark that the basis changes will modify basis vectors and for the indistinguishability to hold, perfectly in *formal* change and computationally in *computational* changes, all impacted basis vectors must not be revealed to the adversary.

Additional Notations. Any $\mathbf{x} = [\![(m_1, \ldots, m_N)]\!]_1 \in \mathbb{G}_1^N$ is identified as $(m_1, \ldots, m_N) \in \mathbb{Z}_q^N$. 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 $t \cdot \mathbf{x} := [\![t \cdot (m_1, \ldots, m_N)]\!]_1$, where $t \in \mathbb{Z}_q$ and $\mathbf{x} = [\![(m_1, \ldots, m_N)]\!]_1$. The additive inverse of $\mathbf{x} \in \mathbb{G}_1^N$ is defined to be $-\mathbf{x} := [\![(-m_1, \ldots, -m_N)]\!]_1$. The canonical basis $\mathbf{A}$ of $\mathbb{G}_1^N$ consists of $\mathbf{a}_1 := [\![(1, 0 \ldots, 0)]\!]_1, \mathbf{a}_2 := [\![(0, 1, 0 \ldots, 0)]\!]_1, \ldots, \mathbf{a}_N := [\![(0, \ldots, 0, 1)]\!]_1$. By convention the writing $\mathbf{x} = (m_1, \ldots, m_N)$ concerns the canonical basis $\mathbf{A}$.

B Deferred Proofs - Proof of Lemma 1

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* $\mathsf{S} \subseteq \mathsf{Att}$ *be a set of attributes. Let $(\mathbf{H}, \mathbf{H}^*)$ and $(\mathbf{F}, \mathbf{F}^*)$ be two random dual bases of $(\mathbb{G}_1^2, \mathbb{G}_2^2)$ and $(\mathbb{G}_1^8, \mathbb{G}_2^8)$, respectively. The vectors $(\mathbf{h}_1, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ are public, while all other vectors are secret. Suppose we have two random labelings $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ for $a_0, a'_0 \xleftarrow{\$} \mathbb{Z}_q$. Let J denote the maximum number of* repetitions *at each $j \in \mathsf{S}$ for $\mathbf{c}_j$ or for $\mathbf{c}_{\mathsf{root}}$. Then, under the* SXDH *assumption in $(\mathbb{G}_1, \mathbb{G}_2)$, the following two distributions are computationally indistinguishable:*

$$\left\{\begin{array}{ll} (x^{(\mathsf{rep})}), y & \\ \mathbf{c}^{(\mathsf{rep})}_{j \in \mathsf{S}} & = (\sigma_j^{(\mathsf{rep})}(1, -j),\ \psi^{(\mathsf{rep})},\ 0^5)_{\mathbf{F}} \\ \mathbf{k}^*_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} & = (\pi_j \cdot (j, 1),\ a_j z,\ 0^5)_{\mathbf{F}^*} \\ \mathbf{c}^{(\mathsf{rep})}_{\mathsf{root}} & = (\psi^{(\mathsf{rep})},\ 0)_{\mathbf{H}} \\ \mathbf{k}^*_{\mathsf{root}} & = (a_0 z,\ 0)_{\mathbf{H}^*} \end{array}\right\} ; \left\{\begin{array}{ll} (x^{(\mathsf{rep})}), y & \\ \mathbf{c}^{(\mathsf{rep})}_{j \in \mathsf{S}} & = (\sigma_j^{(\mathsf{rep})}(1, -j),\ \psi^{(\mathsf{rep})},\ 0^2,\ \boxed{\tau z_j x^{(\mathsf{rep})}},\ 0^2)_{\mathbf{F}} \\ \mathbf{k}^*_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} & = (\pi_j(j, 1),\ a_j z,\ 0^2,\ \boxed{a'_j y / z_j},\ 0^2)_{\mathbf{F}^*} \\ \mathbf{c}^{(\mathsf{rep})}_{\mathsf{root}} & = (\psi^{(\mathsf{rep})},\ \boxed{\tau x^{(\mathsf{rep})}})_{\mathbf{H}} \\ \mathbf{k}^*_{\mathsf{root}} & = (a_0 z,\ \boxed{a'_0 y})_{\mathbf{H}^*} \end{array}\right\}$$

for any $x^{(\mathsf{rep})}, y \in \mathbb{Z}_q$, where $\mathsf{rep} \in [J]$, and $z_j, \sigma_j, \pi_j, \psi, \tau, z, r'_0 \xleftarrow{\$} \mathbb{Z}_q$.

Proof (Of Lemma 1). The proof is done through a sequence of games, starting from G_0 where the adversary receives D_1 and ending in G_4 where the adversary receives D_2. The games are depicted in Figure 3.

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

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

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

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

Game G_0 :

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

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

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

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

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

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

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

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

Game $\mathsf{G}_3 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q, a_0' \xleftarrow{\$} \mathbb{Z}_q,\ (a_j')_{j \in \mathcal{J}} \leftarrow \Lambda_{a_0'}(\mathbb{A})$

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

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

Game $\mathsf{G}_4 : \tau, z_j \xleftarrow{\$} \mathbb{Z}_q, a_0' \xleftarrow{\$} \mathbb{Z}_q,\ (a_j')_{j \in \mathcal{J}} \leftarrow \Lambda_{a_0'}(\mathbb{A})$

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

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

Fig. 3: 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 S for the $\mathbf{c}$-vectors.

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

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

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:

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

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

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

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 $|\mathsf{Adv}(\mathsf{G}_1) - \mathsf{Adv}(\mathsf{G}_0)| \leq \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{DDH}}(1^\lambda)$, where $\mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{DDH}}(1^\lambda)$ 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 $\mathbf{c}_j^{(\mathsf{rep})}$:

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

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 $(\mathbf{F}, \mathbf{F}^*)$ will be changed as follows:

$$\begin{aligned}
F &:= \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & -a \\ 0 & 0 & 1 \end{bmatrix}_{1,2,6} & F' &:= \left(F^{-1}\right)^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & a & 1 \end{bmatrix}_{1,2,6} \\
\mathbf{F} &= F \cdot \mathbf{W}; & \mathbf{F}^* &= F' \cdot \mathbf{W}^*
\end{aligned}$$

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 $j \in \mathsf{S}$, the simulator can sample $\alpha_j, \beta_j \xleftarrow{\$} \mathbb{Z}_q$, compute (in the exponent) $b_j = \alpha_j \cdot b + \beta_j$ and $c_j = \alpha_j \cdot c + \beta_j \cdot a$. We use the random self-reducibility of DDH, then virtually set

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

where $z_j = \alpha_j(1+j)\zeta/\tau$. The repetition-related randomness $\sigma_j^{(\mathsf{rep})} := b_j \cdot x^{(\mathsf{rep})}$ is under affect of $x^{(\mathsf{rep})}$ as expected. If $\zeta = 0$ then $\mathbf{c}_j$ is computed as in G_1, else we are in the current game. Consequently, the difference in advantages of an adversary against G_0 and G_1 is bounded by

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

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

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

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 $(\mathbf{F}, \mathbf{F}^*), (\mathbf{H}, \mathbf{H}^*)$ will be changed by matrices:

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

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

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

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

More generally, if we treat a *vector* $\mathbf{x}^{(\mathsf{rep})}$ that is stored in multiple coordinates of the $\mathbf{c}$-vectors, the above basis change can be generalized so that the repetition-related randomness is instead updated to $\psi^{(\mathsf{rep})} + a\tau \cdot \sum_k \mathbf{x}^{(\mathsf{rep})}[k]$, individually by each coordinate of $\mathbf{x}^{(\mathsf{rep})}$. Let $(d_j')_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ be a random labeling obtained from $\Lambda_1(\mathbb{A})$, i.e. we perform a secret sharing of 1 using the LSSS realizing $\mathbb{A}$. We simulate the vectors

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

When $\rho = 0$ we are in the previous game, where $\psi^{(\mathsf{rep})} + a\tau \cdot x^{(\mathsf{rep})}$ is used instead of $\psi^{(\mathsf{rep})}$ 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 $1/z_j$:

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

This transition is discussed separately in Lemma 24, which show the indistinguishability.

The proof is concluded. □

Lemma 24. *Assuming the* SXDH *assumption for* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, the difference between advantages* $|\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3)|$ *in the proof of Lemma 1 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. The goal is to randomized, for each $j \in \mathsf{List\text{-}Att}(\mathbb{A})$, the label a'_j into a'_j/z_j that is *i.i.d uniformly random* among j, not being a set of shares from labeling of a'_0 anymore. More precisely, the game $\mathsf{G}_{3.m}$ is indexed by $m \in \{0, \ldots, P\}$, where P is the number of attributes in $\mathsf{List\text{-}Att}(\mathbb{A})$ and :

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

This leads to $\mathsf{G}_{3.0} = \mathsf{G}_3$ and $\mathsf{G}_{3.P} = \mathsf{G}_4$. The current form of other vectors, where S is the set of attributes associated to the $\mathbf{c}$-vectors, is:

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

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 [23], we can "factor out" the current attribute-related parts of a'_j in $\mathbf{k}$-vectors, then manipulate the remaining appropriate random linear factor for obtaining the desired new labels (multiplicatively). This requires some rewriting. For two

labelings $\tilde{\mathbf{a}} := (\tilde{a}_0, (\tilde{a}_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}) \leftarrow \Lambda_{\tilde{a}_0}(\mathbb{A})$ and $(a''_0, (a''_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}) \leftarrow \Lambda_{a''_0}(\mathbb{A})$, together with uniformly random scalars $\rho, \delta \xleftarrow{\$} \mathbb{Z}_q^*$ we rewrite the vectors as follows

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

and thus we have

$$\begin{aligned}
a'_0 &= \rho \cdot a''_0; & a_0 &= \tilde{a}_0 + \delta \cdot a''_0 \\
a'_j &= \rho \cdot a''_j; & a_j &= \tilde{a}_j + \delta \cdot a''_j \\
\pi_j &= \Pi_j + a''_j \cdot \tilde{\pi}_j \ . & &
\end{aligned} \tag{11}$$

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 (11), 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 4. 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:

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

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

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

The basis change is done following these matrices:

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

and the simulator can set

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Fig. 4: Games for Lemma 24. 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 (11) 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 5.

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:

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

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

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

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

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

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

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

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

The other vector components stay as in the previous game. More generally, if we treat a *vector* $\mathbf{x}^{(\mathsf{rep})}$ instead of scalars, the above basis change can be adapted with more coordinates in the $\mathbf{c}$-vectors and $\mathbf{h}^*$-vectors. 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 $\tau x z_j / z_m$:

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

We apply Lemma 25 to consider the transition from $\mathsf{G}_{3.m-1.2}$ to $\mathsf{G}_{3.m-1.3}$. We do a sequence of hybrids indexed by $m' \in \mathsf{List\text{-}Att}(\mathbb{A}) \setminus \{m\}$. The coordinates affected are $(1, 2, 5, 7, 8)$ of $(\mathbf{F}, \mathbf{F}^*)$. We note that during each application of the lemma for an index m', only the vectors $\mathbf{c}_{m'}$ and $\mathbf{k}^*_m$ are taken into account and affected by the basis changes (w.r.t the gray boxes). The main reason that we have to do index by index, for $m' \in \mathsf{List\text{-}Att}(\mathbb{A}) \setminus \{m\}$, to change $\mathbf{c}_{m'}$ is the fact that we use formal basis changes to randomize the $(7, 8)$ coordinates, which in turn provide randomness to change the 5-th coordinate of $\mathbf{c}_{m'}$. Indeed, if we change more than 2 vectors $\mathbf{c}_{m'}$ at the same time, there will be more than 2 linear relations in a linear system binding the $(7, 8)$ coordinates. The solution of this system uses the fact that $m' - m \neq 0$ and $1/(m' - m)$ is well-defined, see the arithmetics (12). The more relations it has, the more restrictive it becomes and in the end our formal basis change cannot be well-defined, *i.e.* we cannot obtain an invertible matrix. The setting with *repetitions* also put more constraints on

the formal basis change, see (13) that is needed to be satisfied for the formal basis change to be well-defined. Thus, we can only deal with 1 vector $\mathbf{c}_{m'}$, where $m' \in \mathsf{List\text{-}Att}(\mathbb{A}) \setminus \{m\}$. For other vectors, the concerning coordinates can be written directly in the target bases because they are all 0. We proceed by a sequence of games depicted in Figure 5. The changes that make the transitions between games are highlighted in gray.

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:

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

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

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

$$F := \begin{bmatrix} \alpha/\rho & 0 \\ 1/z_m & 1 \end{bmatrix}_{5,6} \qquad F' := \left(F^{-1}\right)^\top = \begin{bmatrix} \rho/\alpha & -\rho/(z_m\alpha) \\ 0 & 1 \end{bmatrix}_{5,6}$$

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

Note that this basis change will affect only the $\mathbf{h}$-vector of attribute $m \in \mathsf{List\text{-}Att}(\mathbb{A})$, because by construction the other components have coordinate 0 for $\mathbf{f}_5^*$ and have the same writing before and after the basis change. Moreover, the basis change can be applied before the simulator sees the vectors along with $\mathbb{A}$ and S, by first sampling a value $z \xleftarrow{\$} \mathbb{Z}_q$ and use z in the basis change. Afterwards, when all attributes are declared, z would be the mask at the attribute m corresponding to the current hybrid. Last but not least, we target specifically the $\mathbf{h}$-vector of attribute m and the matrix is well-defined without relating to repetitions. We have

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

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

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

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

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

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

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

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

$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ if $m \neq j < m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ 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	$)_{\mathbf{F}^*}$ if $j < m$
$\mathbf{h}_m^*$	$($	$\tilde{\pi}_m \cdot (m, 1)$	$\delta \cdot z$	0	ρy	0	0	0	$)_{\mathbf{F}^*}$ if $j = m$
$\mathbf{h}_j^*$	$($	$\tilde{\pi}_j \cdot (j, 1)$	$\delta \cdot z$	ρy	0	0	0	0	$)_{\mathbf{F}^*}$ if $j > m$

Game $\mathsf{G}_{3.m-1.2.m'-1.1} : z_j \xleftarrow{\$} \mathbb{Z}_q$ (Application Lemma 25 - first game hop)

$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ if $m \neq j < m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ 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	$)_{\mathbf{F}^*}$ if $j < m$
$\mathbf{h}_m^*$	$($	$\tilde{\pi}_m \cdot (m, 1)$	$\delta \cdot z$	0	ρy	0	$j\theta_j$	θ_j	$)_{\mathbf{F}^*}$ if $j = m$
$\mathbf{h}_j^*$	$($	$\tilde{\pi}_j \cdot (j, 1)$	$\delta \cdot z$	ρy	0	0	0	0	$)_{\mathbf{F}^*}$ if $j > m$

Game $\mathsf{G}_{3.m-1.2.m'-1.2} : z_j \xleftarrow{\$} \mathbb{Z}_q$ (Application Lemma 25 - second game hop, masking with $\mu_j^{(\mathsf{rep})}$)

$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ if $m \neq j < m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	$\mu_j^{(\mathsf{rep})}$	$-j\mu_j^{(\mathsf{rep})}$	$)_{\mathbf{F}}$ if $m \neq j = m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ 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	$)_{\mathbf{F}^*}$ if $j < m$
$\mathbf{h}_m^*$	$($	$\tilde{\pi}_m \cdot (m, 1)$	$\delta \cdot z$	0	ρy	0	$m\theta_m$	θ_m	$)_{\mathbf{F}^*}$ if $j = m$
$\mathbf{h}_j^*$	$($	$\tilde{\pi}_j \cdot (j, 1)$	$\delta \cdot z$	ρy	0	0	0	0	$)_{\mathbf{F}^*}$ if $j > m$

Game $\mathsf{G}_{3.m-1.2.m'-1.3} : z_j \xleftarrow{\$} \mathbb{Z}_q$ (Application Lemma 25 2nd to 3rd game - *formal* basis change, generalizing example 2 of *randomized quotient* technique, w.r.t conditions (13))

$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ if $m \neq j < m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	$\mu_1^{(\mathsf{rep})}$	$\mu_2^{(\mathsf{rep})}$	$)_{\mathbf{F}}$ if $m \neq j = m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ 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	$)_{\mathbf{F}^*}$ if $j < m$
$\mathbf{h}_m^*$	$($	$\tilde{\pi}_m \cdot (m, 1)$	$\delta \cdot z$	0	ρy	0	θ_1	θ_2	$)_{\mathbf{F}^*}$ if $j = m$
$\mathbf{h}_j^*$	$($	$\tilde{\pi}_j \cdot (j, 1)$	$\delta \cdot z$	ρy	0	0	0	0	$)_{\mathbf{F}^*}$ if $j > m$

Game $\mathsf{G}_{3.m-1.2.m'-1.4} : z_j \xleftarrow{\$} \mathbb{Z}_q$ (Application Lemma 25 - use the previously randomized coordinates $\mu_1^{(\mathsf{rep})}, \mu_2^{(\mathsf{rep})}$)

$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ if $m \neq j < m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	$\mu_1^{(\mathsf{rep})}$	$\mu_2^{(\mathsf{rep})}$	$)_{\mathbf{F}}$ if $m \neq j = m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ 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	$)_{\mathbf{F}^*}$ if $j < m$
$\mathbf{h}_m^*$	$($	$\tilde{\pi}_m \cdot (m, 1)$	$\delta \cdot z$	0	ρy	0	θ_1	θ_2	$)_{\mathbf{F}^*}$ if $j = m$
$\mathbf{h}_j^*$	$($	$\tilde{\pi}_j \cdot (j, 1)$	$\delta \cdot z$	ρy	0	0	0	0	$)_{\mathbf{F}^*}$ if $j > m$

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

$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})}$	$\tau x^{(\mathsf{rep})} z_j / z_m$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ if $m \neq j \leq m'$
$\mathbf{c}_j^{(\mathsf{rep})}$	$($	$\sigma_j^{(\mathsf{rep})} \cdot (1, -j)$	$\psi^{(\mathsf{rep})}$	τx	$\tau x^{(\mathsf{rep})}$	$\tau z_j x^{(\mathsf{rep})}$	0	0	$)_{\mathbf{F}}$ 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	$)_{\mathbf{F}^*}$ if $j < m$
$\mathbf{h}_m^*$	$($	$\tilde{\pi}_m \cdot (m, 1)$	$\delta \cdot z$	0	ρy	0	0	0	$)_{\mathbf{F}^*}$ if $j = m$
$\mathbf{h}_j^*$	$($	$\tilde{\pi}_j \cdot (j, 1)$	$\delta \cdot z$	ρy	0	0	0	0	$)_{\mathbf{F}^*}$ if $j > m$

Fig. 5: The hybrids to go from $\mathsf{G}_{3.m-1.2.m'-1.0}$ to $\mathsf{G}_{3.m-1.2.m'-1.5=3.m-1.2.m'}$ is coming from Lemma 25, on coordinates $(1, 2, 5, 7, 8)$, while taking $\mathbf{h}_m^*$ as the $\mathbf{k}$-vector and changing $\mathbf{c}_{m'}$ where $m' \neq m$ in the application of the lemma. 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 (11) for the rewriting of $\mathbf{k}_j^*$ into $\mathbf{h}_j^*$.

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

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

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 proof is concluded. □

Lemma 25. *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 $\beta, \alpha^{(\mathsf{rep})}, \gamma^{(\mathsf{rep})} \in \mathbb{Z}_q$ are chosen constants, where* rep *are index for repetitions of the* $\mathbf{c}$*-vectors. Then, under the* SXDH *assumption, the following two distributions are computationally indistinguishable, where* $\mathbf{c}$*-vectors are* repetitive *over the same j with independent randomness:*

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

and

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

where $\sigma^{(\mathsf{rep})}, \pi \xleftarrow{\$} \mathbb{Z}_q$ are unknown and random, and $\sigma^{(\mathsf{rep})}$ are independent *among different* rep.

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

The masks $\tau^{(\mathsf{rep})}$ are depending on the repetitions index rep, for which we use the random self-reducibility of the DDH assumption. 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 $c - ab$ is either 0 or uniformly random, we use the following basis changing matrices (F, F'):

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

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

The vector $\mathbf{c}^{(\mathsf{rep})}$ can be simulated as follows. First, we randomize independently $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ into $([\![a]\!]_1, [\![b^{(\mathsf{rep})}]\!]_1, [\![c^{(\mathsf{rep})}]\!]_1)$ so that $c^{(\mathsf{rep})} - ab^{(\mathsf{rep})}$ is uniformly random or 0 following $c - ab$. We then compute

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

simulating $\sigma^{(\mathsf{rep})} := b^{(\mathsf{rep})}$, while $\tau^{(\mathsf{rep})} := c^{(\mathsf{rep})} - ab^{(\mathsf{rep})}$. 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 $\pi + a\rho$. The important point is that our basis changing matrix depends only on a, that is not randomized in the randomly self-reduced DDH instances and thus independent from rep.

If $c^{(\mathsf{rep})} - ab^{(\mathsf{rep})} = 0$ 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:

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

where $\theta_1, \theta_2 \xleftarrow{\$} \mathbb{Z}_q$ are chosen uniformly at random.
We consider the basis changing matrices (F, F'):

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

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:

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

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

$$\begin{cases} \tau^{(\mathsf{rep})}(z_4 + j z_3) = \mu_1^{(\mathsf{rep})} \\ -\tau^{(\mathsf{rep})}(z_2 + j z_1) = \mu_2^{(\mathsf{rep})} \\ \rho(m z_1 + z_2) = \theta_1 \\ \rho(m z_3 + z_4) = \theta_2 \end{cases} \Leftrightarrow \begin{cases} z_4 + j z_3 = \mu_1^{(\mathsf{rep})}/\tau^{(\mathsf{rep})} \\ m z_3 + z_4 = \theta_2/\rho \\ z_2 + j z_1 = -\mu_2^{(\mathsf{rep})}/\tau^{(\mathsf{rep})} \\ m z_1 + z_2 = \theta_1/\rho \end{cases}$$
$$\Leftrightarrow \begin{cases} (j - m) z_3 = \mu_1^{(\mathsf{rep})}/\tau^{(\mathsf{rep})} - \theta_2/\rho \\ m z_3 + z_4 = \theta_2/\rho \\ (j - m) z_1 = -\mu_2^{(\mathsf{rep})}/\tau^{(\mathsf{rep})} - \theta_1/\rho \\ m z_1 + z_2 = \theta_1/\rho \end{cases} . \tag{12}$$

The system has a solution if and only if $j \neq m$, which is already our hypothesis. We note that since $\mu_1^{(\mathsf{rep})}, \mu_2^{(\mathsf{rep})}, \theta_1, \theta_2$ are uniformly random chosen values and fixed to determine (z_1, z_2, z_3, z_4), we can always perform normalization using $\mu_1^{(\mathsf{rep})}, \mu_2^{(\mathsf{rep})}, \theta_1, \theta_2$ to ensure $z_1 z_4 - z_2 z_3 = 1$ for the basis change. Moreover, it is important that in the current setting of *repetitions*, $(\mu_1^{(\mathsf{rep})}, \mu_2^{(\mathsf{rep})}, \theta_1, \theta_2)$ are chosen such that

$$\frac{\mu_1^{(\mathsf{rep})}}{\tau^{(\mathsf{rep})}} = \mathsf{const}_1 \text{ and } \frac{\mu_2^{(\mathsf{rep})}}{\tau^{(\mathsf{rep})}} = \mathsf{const}_2 \tag{13}$$

are constants $\mathsf{const}_1, \mathsf{const}_2 \in \mathbb{Z}_q$ over different rep. Otherwise the basis change matrix is not well defined because its entries (z_1, z_2, z_3, z_4) expresed by $(\mu_1, \mu_2, \theta_1, \theta_2)$ depend on rep. In other words, at the time of defining the basis change matrix, $\mathsf{const}_1, \mathsf{const}_2$ are fixed and independent of rep, then $(\mu_1^{(\mathsf{rep})}, \mu_2^{(\mathsf{rep})}, \tau^{(\mathsf{rep})})$ are chosen during simulation of $\mathbf{c}^{(\mathsf{rep})}$ following (13). 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 $\gamma^{(\mathsf{rep})}$ in $\mathbf{c}$ to another constant $\alpha^{(\mathsf{rep})}$:

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

Let $(\mathbf{W}, \mathbf{W}^*)$ be the canonical bases of $\mathbb{G}_1^5$ and $\mathbb{G}_2^5$. The security loss of this game hop depends on the maximum number of repetitions over $\mathbf{c}$-vectors that the adversary can query. Given a DSDH instance $([\![a]\!]_1, [\![b]\!]_1, [\![c]\!]_1)$ where $c - ab$ is either 1 or the constant 0, we use the following basis changing matrices (F, F'):

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

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

$$\begin{aligned}\mathbf{c}^{(\mathsf{rep})} &= (\sigma^{(\mathsf{rep})} \cdot (1, -j),\ \gamma^{(\mathsf{rep})},\ \mu_1^{(\mathsf{rep})},\ \mu_2^{(\mathsf{rep})})_{\mathbf{F}} \\ &\quad + (0,\ 0,\ c \cdot (\gamma^{(\mathsf{rep})} - \alpha^{(\mathsf{rep})}),\ b \cdot (\alpha^{(\mathsf{rep})} - \gamma^{(\mathsf{rep})}),\ 0)_{\mathbf{W}} \\ &= (\sigma^{(\mathsf{rep})} \cdot (1, -j),\ \gamma^{(\mathsf{rep})} + (c - ab) \cdot (\alpha^{(\mathsf{rep})} - \gamma^{(\mathsf{rep})}),\ b \cdot (\alpha^{(\mathsf{rep})} - \gamma^{(\mathsf{rep})}),\ \mu_2^{(\mathsf{rep})})_{\mathbf{F}}\ .\end{aligned}$$

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

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

and θ_1 is updated to $\theta_1 + a\beta$. It follows form previous game that the last two coordinates of the $\mathbf{k}$-vectors do not depend on repetitions of $\mathbf{c}$, which is impossible anyway, and th basis changing uses only a from the DDH instance. If $c - ab = 0$ we are in the previous game, otherwise we are in the current game. The difference in advantages is $|\mathsf{Adv}(\mathsf{G}_4) - \mathsf{Adv}(\mathsf{G}_3)| \leq 2 \cdot \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{DDH}}(1^\lambda)$.

Game G_5: In this game we clean the masks $\mu_1^{(\mathsf{rep})}, \mu_2^{(\mathsf{rep})}, \theta_1, \theta_2$ by doing the reverse transition from G_3 back to G_0.

The proof of the lemma is concluded. □

C Deferred Proofs - Proof of Theorem 8

Proof (Of Theorem 8). Let $\mathsf{MCFE}^{\mathsf{xxx}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i \in [n]}]$ be a secure MCFE following Definition 6. We denote by $(\mathsf{Setup}^{\mathsf{mc}}, \mathsf{Extract}^{\mathsf{mc}}, \mathsf{Enc}^{\mathsf{mc}}, \mathsf{Dec}^{\mathsf{mc}})$ the algorithms of the MCFE.

From MCFE to MIFE. Following Definition 22, we consider the notion of MIFE having only *one* encryptor who can use a master secret key to encrypt independent components of a message. The function class is $\mathcal{F}$ containing $F_{\lambda,n} : \prod_{i=1}^n (\mathcal{D}_{\lambda,i} \times \mathcal{Z}_{\lambda,i}) \to \mathcal{R}_\lambda$. There is *no* public inputs as we are concentrating on the classic MIFE *as per* [25]. The obtained MIFE is defined by the algorithms:

$\mathsf{Setup}^{\mathsf{mi}}(1^\lambda, 1^n)$**:** Run $\mathsf{Setup}^{\mathsf{mc}}(1^\lambda, 1^n) \to (\mathsf{msk}^{\mathsf{mc}}, (\mathsf{ek}_i^{\mathsf{mc}})_i)$. Sample a tag $\mathsf{tag} \xleftarrow{\$} \mathsf{Tag}$ and output $\mathsf{msk} := \mathsf{msk}^{\mathsf{mc}}, (\mathsf{ek}_i := (\mathsf{ek}_i^{\mathsf{mc}}, \mathsf{tag}))_i$.

$\mathsf{Extract}^{\mathsf{mi}}(\mathsf{msk}, F_\lambda)$**:** Run $\mathsf{Extract}^{\mathsf{mc}}(\mathsf{msk}^{\mathsf{mc}}, F_\lambda) \to \mathsf{dk}_{F_\lambda}$ and output dk_{F_λ}.

$\mathsf{Enc}^{\mathsf{mi}}(\mathsf{ek}_i, x_i)$**:** Parse $\mathsf{ek}_i := (\mathsf{ek}_i^{\mathsf{mc}}, \mathsf{tag})$. Run $\mathsf{Enc}^{\mathsf{mc}}(\mathsf{ek}_i^{\mathsf{mc}}, x_i, (\mathsf{tag}, \epsilon)) \to \mathsf{ct}_i$ as there is no public inputs in classical MIFE, then output ct_i.

$\mathsf{Dec}^{\mathsf{mi}}(\mathsf{dk}_{F_\lambda}, (\mathsf{ct}_i)_i)$**:** Run and output $\mathsf{Dec}^{\mathsf{mc}}(\mathsf{dk}_{F_\lambda}, (\mathsf{ct}_i)_i)$.

Correctness follows from the correctness of the MCFE. In terms of security, let $\mathcal{A}$ be an adversary against the MIFE *as per* Definition 23. We construct an adversary $\mathcal{B}$ breaking $\mathsf{MCFE}^{\text{xxx-rep-priv}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}]$ using $\mathcal{A}$.

The adversary $\mathcal{B}$ simulates the MIFE game by (i) first querying its MCFE challenger on $(1^\lambda, 1^n)$ to obtain the public parameters (if any) then forwards to $\mathcal{A}$; (ii) simulating the MIFE's encryption/challenge queries by fixing a tag tag for all encryption (respectively challenge) ciphertexts and forwarding the encryption (that is, $(i, x_i, (\mathsf{tag}, \epsilon))$) (respectively challenge (that is, $(i, x_i^{(0)}, x_i^{(1)}, (\mathsf{tag}, \epsilon))$)) queries given (i, x_i) or $(i, x_i^{(0)}, x_i^{(1)})$ by $\mathcal{A}$ against the MIFE; (iii) the key-extraction queries are forwarded to the MCFE challenger in a straightforward manner. In the end $\mathcal{B}$ outputs the same as $\mathcal{A}$. If $\mathcal{A}$ wins the MIFE game, then $\mathcal{B}$ wins the MCFE game. We remark that when $\mathcal{A}$ makes *repetitions* over the encryption queries (*i.e.* same i but different messages), the forwarded queries to the MCFE challenger are *repetitions* over the *private* inputs as well, while the public inputs stay (tag, ϵ) for both **Enc** and **LoR**). In particular if $\mathcal{A}$ is admissible following Definition 23, all queries by $\mathcal{B}$ to its challenger are admissible *as per* Definition 5, in the *private-only repetitions*, because the conditions of MIFE security imposes more restricting conditions, due to the fact that there are more possibilities to combine ciphertexts[8].

From MCFE to (single-client, public-key) FE. The function class is $\mathcal{F}$ containing $F_\lambda : \mathcal{D}_\lambda \times \mathcal{Z}_\lambda \to \mathcal{R}_\lambda$. Following Definition 20, the obtained FE is defined by algorithms:

$\mathsf{Setup}^{\mathsf{pk}}(1^\lambda)$**:** Run $\mathsf{Setup}^{\mathsf{mc}}(1^\lambda, 1^1) \to (\mathsf{msk}^{\mathsf{mc}}, \mathsf{ek}^{\mathsf{mc}})$. Output $\mathsf{msk} := \mathsf{msk}^{\mathsf{mc}}, \mathsf{pk} := (\mathsf{ek}^{\mathsf{mc}})$.
$\mathsf{Extract}^{\mathsf{pk}}(\mathsf{msk}, F_\lambda)$**:** Run $\mathsf{Extract}^{\mathsf{mc}}(\mathsf{msk}^{\mathsf{mc}}, F_\lambda) \to \mathsf{dk}_{F_\lambda}$ and output dk_{F_λ}.
$\mathsf{Enc}^{\mathsf{pk}}(\mathsf{pk}, x, z)$**:** Parse $\mathsf{pk} := (\mathsf{ek}^{\mathsf{mc}})$ and $z := (\epsilon, \tilde{z})$ as there is no tag in single client and public key FE. Sample $\mathsf{tag} \xleftarrow{\$} \mathsf{Tag}$ and run $\mathsf{Enc}^{\mathsf{mc}}(\mathsf{ek}^{\mathsf{mc}}, x, (\mathsf{tag}, \tilde{z})) \to \mathsf{ct}$. Finally output ct.
$\mathsf{Dec}^{\mathsf{pk}}(\mathsf{dk}_{F_\lambda}, \mathsf{ct})$**:** Run and output $\mathsf{Dec}^{\mathsf{mc}}(\mathsf{dk}_{F_\lambda}, \mathsf{ct})$.

Correctness follows from the correctness of the MCFE. If the function class captures access control, then the FE is for the same class having access control as well. In terms of security, let $\mathcal{A}$ be an adversary against the FE *as per* Definition 21. We construct an adversary $\mathcal{B}$ breaking $\mathsf{MCFE}^{\text{xxx-rep-priv}}[\mathcal{F}, (\mathcal{Z}_{\lambda,i})_{i\in[n]}]$, with *static* corruptions, using $\mathcal{A}$. The adversary $\mathcal{B}$ simulates the FE game by (i) first querying its MCFE challenger on $(1^\lambda, 1)$ to obtain the public parameters pp (if any) then *queries* **Corrupt**(1), gets ek, and forwards $\mathsf{pk} := \mathsf{ek}$ together with pp to $\mathcal{A}$. We note that the corrupted client is known from the beginning; (ii) simulating the FE's challenge queries by forwarding the challenge queries (*i.e.* sample $\mathsf{tag} \xleftarrow{\$} \mathsf{Tag}$ and define the challenge to be $(1, x^{(0)}, x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$) to its MCFE challenger given $\left(\left(x^{(0)}, (\epsilon, \tilde{z}^{(chal)})\right), \left(x^{(1)}, (\epsilon, \tilde{z}^{(chal)})\right)\right)$ by $\mathcal{A}$; (iii) the key extraction queries are forwarded to the MCFE challenger in a straightforward manner. If the FE adversary $\mathcal{A}$ is admissible, *i.e.* $x^{(0)} \neq x^{(1)}$ but $F(x^{(0)}, (\epsilon, \tilde{z}^{(chal)})) = F(x^{(1)}, (\epsilon, \tilde{z}^{(chal)}))$ for all F queried to **Extract**, then the challenge query $(1, x^{(0)}, x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$ is on a pair of inputs $(x^{(0)}, (\mathsf{tag}, \tilde{z}^{(chal)})) \neq (x^{(1)}, (\mathsf{tag}, \tilde{z}^{(chal)}))$ conforming to the admissibility. This implies that $\mathcal{B}$ is also admissible following Definition 5. Moreover, the fact that every encryption query is defined on a *freshly sampled* tag implies that there is no repetitions for any pair $(1, \mathsf{tag})$ registered to the MCFE challenger. This allows us to allow encrypting different public inputs even though the MCFE is for *private inputs repetitions* only. Therefore, if $\mathcal{A}$ wins the FE game, then $\mathcal{B}$ wins the MCFE game.

Implication to KP-ABE. The implication to KP-ABE follows from the (single-client, public-key) FE case for $\mathcal{F}$ containing $F_\lambda : \mathcal{D}_\lambda \times \mathcal{Z}_\lambda \to \mathcal{R}_\lambda$. Moreover, the identity function is in $\mathcal{F}$ and allows the *all-or-nothing* decryption of KP-ABE, without any evaluation on the plaintext. In particular, thanks to the hypothesis that the function class $\mathcal{F}$ can encode a policy class Pol. The

[8] In the one-encryptor setting there is no corruption oracle in the MIFE game, *e.g.* see the original in [25].

attribute space Att is contained in $\mathcal{Z}_\lambda$, as public inputs, or in the secret inputs $\mathcal{D}_\lambda$ in case of *attribute-hiding*. A reduction from FE to KP-ABE can be obtained with ease. □

Remark 26. **(On the encoding of attributes in the proof of Theorem 8)** In the reduction to KP-ABE, implicitly the security comes from what is the hypothesis of the MCFE: even though the MCFE is set up for one slot, each time an encryption is created, a fresh tag is sampled therefore *not* leading to a fully repetitive on *both* private and public inputs. This thus allows encrypting on *different* attribute sets while there is no full repetitions for any pair $(1, \mathsf{tag})$ registered to the MCFE challenger. Finally, an adversary breaking the KP-ABE allows breaking the MCFE.

D Deferred Proofs - Proof of Theorem 12

Theorem 12. *Let $\mathcal{E}$ be a MCFE scheme with fine-grained access control for the function class $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$, given in Section 4.2 in a bilinear group setting $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, \mathbf{e}, q)$. Then, in the random oracle model, $\mathcal{E}$ is* one-time statically IND-secure *against* complete *challenge queries with* private-inputs only repetitions *(as per Definition 11), under the* SXDH *in $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Proof (Of Theorem 12). The sequence of games can be found in Figure 6, 7, and 8. 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 $\boxed{\text{boxed}}$. 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, where the private-input repetitions at each position $i \in [n]$ are indexed by $\mathsf{rep} \in [J_i]$ where J_i is the maximum repetitions queried for position i. We note that for different i, the bound J_i can be different. The challenge ciphertext encrypts subvectors $\mathbf{x}_i^{(b,\mathsf{rep})} \in \mathbb{Z}_q^N$. For simplicity, we add a constraint that the challenge tag tag is *not* queried to **Enc**. This incurs a multiplicative loss factor in advantage up to an inverse of polynomial in λ, where we can reduce to the normal $\mathsf{1chal}$ by guessing the challenge tag among the tags for encryption, and responding all of its **Enc** queries $(i, \mathbf{x}_i, (\mathsf{tag}, \mathsf{ac\text{-}ct}_i))$ by $\mathbf{LoR}(i, \mathbf{x}_i, \mathbf{x}_i, (\mathsf{tag}, \mathsf{ac\text{-}ct}_i))$.

Game G_1: We perform a sequence of hybrids over the key queries $(\mathbf{y}_i^{(\ell)})_i$ for $\ell \in [K]$. We denote $\mathsf{G}_{0.\ell}$ the hybrid where all the $\le (\ell - 1)$-th key is programmed

$$\begin{array}{rllll|l|c|c|cl}
\mathbf{LoR}\ \mathbf{t}_i^{(j_i)} & (& \omega \cdot \mathbf{s}_i + \omega' \cdot \mathbf{u}_i + \mathbf{x}_i^{(b,j_i)} & & & \nu_i^{(j_i)} & \mathbf{0} & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\le \ell-1)} & (& \mathbf{y}_i^{(\le \ell-1)} & & & \sum_{i=1}^n a_{i,0}^{(\le \ell-1)} & \mathbf{0} & \mathsf{rnd}_i^{(\le \ell-1)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|c|c|c|c|cl}
\mathbf{LoR}\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta\mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 &)_{\mathbf{H}_i} \\
\mathbf{k}_{i,\mathsf{ipfe}}^{(\le \ell-1)} & (\sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\le \ell-1)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\le \ell-1)} \rangle & a_{i,0}^{(\le \ell-1)} z^{(\le \ell-1)} & \boxed{R \cdot \mathbf{y}_i^{(\le \ell-1)}} & (d_{\mathbb{A},i,k}^{(\le \ell-1)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\le \ell-1)} &)_{\mathbf{H}_i^*}
\end{array}$$

while other ciphertext components from **Enc** are kept in normal form. It holds that $\mathsf{G}_0 = \mathsf{G}_{0.0}$. For $\ell \in [K]$, the transition from $\boxed{\mathsf{G}_{0.\ell-1}}$ to $\boxed{\mathsf{G}_{0.\ell}}$ is as follows: $\boxed{\mathsf{G}_{0.\ell.0}}$ is the same as $\boxed{\mathsf{G}_{0.\ell-1}}$. $\boxed{\mathsf{G}_{0.\ell.1}}$ is the same as $\boxed{\mathsf{G}_{0.\ell.0}}$ except that we apply Lemma 1 to introduce a set of masks in the ciphertexts : $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$. The proof of Lemma 1 can be found in Appendix B. We remark that $\Delta\mathbf{x}_i$ is a vector of differences of the challenge ciphertexts at position i, being constants at each i over all repetitions j_i, under the admissibility. Moreover, the *strong admissibility* also ensures that:

$$\begin{cases} \sum_{i \in \mathcal{H}} \langle \Delta\mathbf{x}_i, \mathbf{y}_i^{(\ell)} \rangle = 0 \\ \langle \Delta\mathbf{x}_i, \mathbf{y}_i^{(\ell)} \rangle = 0 \ \forall\, i \in \mathcal{C} \end{cases}$$

Game G_0 : $\mathsf{H}(\mathsf{tag}) \to ([\![\omega_{\mathsf{tag}}]\!]_1, [\![\omega'_{\mathsf{tag}}]\!]_1)$, $\mathsf{H}(\mathsf{tag}') \to ([\![\chi_{\mathsf{tag}'}]\!]_1, [\![\chi'_{\mathsf{tag}'}]\!]_1)$, (for **Enc** queries H on tag' are noted by χ and χ') ; $\ell \in [K]$ indexes key queries
$a_{i,0}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q$, $(a_{i,j}^{(\ell)})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_{i,0}^{(\ell)}}(\mathbb{A})$, $\sum_{i=1}^{n} \sum_{k=1}^{N} d_{\mathbb{A},i,k}^{(\ell)} \theta_{i,k} = 0$

$$
\begin{array}{rlcc|c|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,j}^{(j_i)} & (& \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \psi_i^{(j_i)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{F}} \\
\textbf{LoR}\ \widetilde{\mathbf{t}}_{i,j}^{(j_i)} & (& \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \nu_i^{(j_i)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{G}} \\
\textbf{Enc}\ \mathbf{c}_{i,j}^{(j_i)} & (& \sigma'_{i,j} \cdot (1,-j) & \overline{\psi}_i^{(j_i)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{k}_{i,j}^{(\ell)} & (& \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} \cdot z^{(\ell)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{F}^*} \\
\widetilde{\mathbf{m}}_{i,j}^{(\ell)} & (& \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{G}^*}
\end{array}
$$

$$
\begin{array}{rlcc|c|c|c|c}
\textbf{LoR}\ \mathbf{t}_i^{(j_i)} & (& \omega \cdot \mathbf{s}_i + \omega' \cdot \mathbf{u}_i + \mathbf{x}_i^{(b,j_i)} & \nu_i^{(j_i)} & \mathbf{0} & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\textbf{Enc}\ \mathbf{t}_i^{(j_i)} & (& \chi \cdot \mathbf{s}_i + \chi' \cdot \mathbf{u}_i + \overline{\mathbf{x}}_i^{(j_i)} & \nu_i^{(j_i)} & \mathbf{0} & 0 & \overline{\rho}_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (& \mathbf{y}_i^{(\ell)} & \sum_{i=1}^{n} a_{i,0}^{(\ell)} & \mathbf{0} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}
$$

$$
\begin{array}{rlcc|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \mathbf{0} & (\theta_{i,k})_{k=1}^{N} & 0 &)_{\mathbf{H}_i} \\
\textbf{Enc}\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& p_i \chi_{\mathsf{tag}'} & p_i \chi'_{\mathsf{tag}'} & \overline{\psi}_i^{(j_i)} & \mathbf{0} & (\theta_{i,k})_{k=1}^{N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i} \\
\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a_{i,0}^{(\ell)} z^{(\ell)} & \mathbf{0} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^{N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}
$$

Game G_1 : $z_j \xleftarrow{\$} \mathbb{Z}_q^*$, $\Delta \mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ (Masking Application - Lemma 1, hybrids over each key query $(\mathbf{y}_i^{(\ell)})_i$, using the DPVS basis changes from Appendix A.5, *i.e.* formal ones (1, 2, 3) and computational ones (1, 2))

$\boxed{\mathsf{G}_{0.\ell.1}}$ where $\ell \in [K]$ and K is the maximum number of key queries. We are in the setting of *private-input only* repetitions $a'_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'_{i,j})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_{i,0}}(\mathbb{A})$

$$
\begin{array}{rlcc|c|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,j}^{(j_i)} & (& \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \psi_i^{(j_i)} & \mathbf{0} & 0 & \boxed{z_j \cdot \Delta \mathbf{x}_i} & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{k}_{i,j}^{(\ell)} & (& \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} \cdot z^{(\ell)} & \mathbf{0} & 0 & \boxed{(a'_{i,j}/z_j) \cdot \mathbf{y}_i^{(\ell)}} & 0 & 0 &)_{\mathbf{F}^*}
\end{array}
$$

$$
\begin{array}{rlcc|c|c|c|c}
\textbf{LoR}\ \mathbf{t}_i^{(j_i)} & (& \omega \cdot \mathbf{s}_i + \omega' \cdot \mathbf{u}_i + \mathbf{x}_i^{(b,j_i)} & \nu_i^{(j_i)} & \mathbf{0} & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (& \mathbf{y}_i^{(\ell)} & \sum_{i=1}^{n} a_{i,0}^{(\ell)} & \mathbf{0} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}
$$

$$
\begin{array}{rlcc|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \boxed{\Delta \mathbf{x}_i} & (\theta_{i,k})_{k=1}^{N} & 0 &)_{\mathbf{H}_i} \\
i \in \mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a_{i,0}^{(\ell)} z^{(\ell)} & \boxed{a'_{i,0} \cdot \mathbf{y}_i^{(\ell)}} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^{N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}
$$

$\boxed{\mathsf{G}_{0.\ell.2}}$: $R \xleftarrow{\$} \mathbb{Z}_q$, $\Delta \mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$(Randomization, the honest $\mathcal{H}$ and corrupted $\mathcal{C}$ are known due to *static* corruption, use *formal basis changes*)

$$
\begin{array}{rlcc|c|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,j}^{(j_i)} & (& \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \psi_i^{(j_i)} & \mathbf{0} & 0 & \boxed{z_j \cdot \Delta \mathbf{x}_i} & 0 & 0 &)_{\mathbf{F}} \\
\mathbf{k}_{i,j}^{(\ell)} & (& \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} \cdot z^{(\ell)} & \mathbf{0} & 0 & \boxed{(a'_{i,j}/z_j) \cdot \mathbf{y}_i^{(\ell)}} & 0 & 0 &)_{\mathbf{F}^*}
\end{array}
$$

$$
\begin{array}{rlcc|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta \mathbf{x}_i & (\theta_{i,k})_{k=1}^{N} & 0 &)_{\mathbf{H}_i} \\
i \in \mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a_{i,0}^{(\ell)} z^{(\ell)} & \boxed{(a'_{i,0} + R) \cdot \mathbf{y}_i^{(\ell)}} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^{N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}
$$

$\boxed{\mathsf{G}_{0.\ell.3}}$: $R \xleftarrow{\$} \mathbb{Z}_q$, $\Delta \mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$(Reverse Masking Application - Lemma 1, only mask $R \cdot \mathbf{y}_i^{(\ell)}$ remains in $\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$ for $i \in \mathcal{H}$)

$$
\begin{array}{rlcc|c|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,j}^{(j_i)} & (& \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \psi_i^{(j_i)} & \mathbf{0} & 0 & \boxed{\mathbf{0}} & 0 & 0 &)_{\mathbf{F}} \\
\textbf{LoR}\ \widetilde{\mathbf{t}}_{i,j}^{(j_i)} & (& \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \nu_i^{(j_i)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{G}} \\
\mathbf{k}_{i,j}^{(\ell)} & (& \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} \cdot z^{(\ell)} & \mathbf{0} & 0 & \boxed{\mathbf{0}} & 0 & 0 &)_{\mathbf{F}^*} \\
\widetilde{\mathbf{m}}_{i,j}^{(\ell)} & (& \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{G}^*}
\end{array}
$$

$$
\begin{array}{rlcc|c|c|c|c|c}
\textbf{LoR}\ \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta \mathbf{x}_i & (\theta_{i,k})_{k=1}^{N} & 0 &)_{\mathbf{H}_i} \\
i \in \mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a_{i,0}^{(\ell)} z^{(\ell)} & \boxed{R \cdot \mathbf{y}_i^{(\ell)}} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^{N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}
$$

$\mathsf{G}_1 := \mathsf{G}_{0.K.3}$, where $\ell \in [K]$ and K is the maximum number of key queries.

Fig. 6: Games $\mathsf{G}_0, \mathsf{G}_1$ for Theorem 12.

corresponding to any inner product function of $(\mathbf{y}_i)_i$ (together with an LSSS). The ℓ-th key components are programmed to also accomodate newly independent values: $a'_{i,0} \xleftarrow{\$} \mathbb{Z}_q$, $(a'_{i,j})_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_{i,0}}(\mathbb{A})$, $z_j \xleftarrow{\$} \mathbb{Z}_q^*$. We emphasize that the random values introduced in key components are randomized secret shares $(a'_{i,j}/z_j) \cdot \mathbf{y}_i$ in which $a'_{i,j}$ are shares of $a'_{i,0}$ by the attributes in $\mathsf{List\text{-}Att}(\mathbb{A})$. Thus, over all honest $i \in \mathcal{H}$, due to the simulated vectors at decryption will cancel the masks, while for corrupted $i \in \mathcal{C}$, the masks are already 0 after

Game G_2 : $R \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ (Rewriting game's description, summing of $\widetilde{\mathbf{t}}_{i,j}^{(j_i)}$ over $i \in \mathcal{H}$ known statically, not affecting correctness)

$$\begin{array}{rl|l|c|c|c|c|cl}
\textbf{LoR } \mathbf{c}_{i,j}^{(j_i)} & (\ \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \psi_i^{(j_i)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{F}} \\
\textbf{LoR } \widetilde{\mathbf{t}}_{0,j}^{(j_i)} = \sum_{i\in\mathcal{H}} \widetilde{\mathbf{t}}_{i,j}^{(j_i)} & (\ \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \sum_{i\in\mathcal{H}} \nu_i^{(j_i)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{G}} \\
\mathbf{k}_{i,j}^{(\ell)} & (\ \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} \cdot z^{(\ell)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{F}^*} \\
\widetilde{\mathbf{m}}_{i,j}^{(\ell)} & (\ \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 &)_{\mathbf{G}^*}
\end{array}$$

$$\begin{array}{rl|l|c|c|c|l}
\textbf{LoR } \mathbf{t}_i^{(j_i)} & (\ \omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \mathbf{x}_i^{(b,j_i)} & \nu_i^{(j_i)} & \mathbf{0} & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (\ \mathbf{y}_i^{(\ell)} & \sum_{i=1}^n a_{i,0}^{(\ell)} & \mathbf{0} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|l|l|c|c|c|l}
\textbf{LoR } \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta\mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 &)_{\mathbf{H}_i} \\
i\in\mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \sum_i\langle\mathbf{s}_i,\mathbf{y}_i^{(\ell)}\rangle & \sum_i\langle\mathbf{u}_i,\mathbf{y}_i^{(\ell)}\rangle & a_{i,0}^{(\ell)} z^{(\ell)} & R\cdot\mathbf{y}_i^{(\ell)} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

Game G_3 : $R \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ (Masking Application - Lemma 1, hybrids over each key query $(\mathbf{y}_i^{(\ell)})_i$, similar to $\mathsf{G}_1 \to \mathsf{G}_2$)

$$\begin{array}{rl|l|c|c|c|c|cl}
\textbf{LoR } \widetilde{\mathbf{t}}_{0,j}^{(j_i)} = \sum_{i\in\mathcal{H}} \widetilde{\mathbf{t}}_{i,j}^{(j_i)} & (\ \sigma_{i,j}^{(j_i)} \cdot (1,-j) & \sum_{i\in\mathcal{H}} \nu_i^{(j_i)} & \mathbf{0} & 0 & \boxed{\mathbf{0}} & 0 & 0 &)_{\mathbf{G}} \\
\widetilde{\mathbf{m}}_{i,j}^{(\ell)} & (\ \pi_{i,j}^{(\ell)} \cdot (j,1) & a_{i,j}^{(\ell)} & \mathbf{0} & 0 & \boxed{\mathbf{0}} & 0 & 0 &)_{\mathbf{G}^*}
\end{array}$$

$$\begin{array}{rl|l|c|c|c|l}
\textbf{LoR } \mathbf{t}_i^{(j_i)} & (\ \omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \mathbf{x}_i^{(b,j_i)} & \nu_i^{(j_i)} & \boxed{\Delta\mathbf{x}_i} & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (\ \mathbf{y}_i^{(\ell)} & \sum_{i=1}^n a_{i,0}^{(\ell)} & \boxed{R\cdot\mathbf{y}_i^{(\ell)}} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|l|l|c|c|c|l}
\textbf{LoR } \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta\mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 &)_{\mathbf{H}_i} \\
i\in\mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \sum_i\langle\mathbf{s}_i,\mathbf{y}_i^{(\ell)}\rangle & \sum_i\langle\mathbf{u}_i,\mathbf{y}_i^{(\ell)}\rangle & a_{i,0}^{(\ell)} z^{(\ell)} & R\cdot\mathbf{y}_i^{(\ell)} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

Fig. 7: Games $\mathsf{G}_2, \mathsf{G}_3$ for Theorem 12.

Game G_4 : $R, R' \xleftarrow{\$} \mathbb{Z}_q$, $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}$ (Switching $\mathbf{x}_i^{(b,j_i)}$ to $\mathbf{x}_i^{(1,j_i)}$, using *complexity leveraging*, the invariant coordinates are grouped as "$\cdots$")

$\boxed{\mathsf{G}_{3.1}}$ (Formal Quotient, using $\Delta\mathbf{x}_i$ is constant for $i \in \mathcal{H}$ over repetitions, Hadamard product is denoted "$\circ$", see example 2 on DPVS basis changes)

$$\begin{array}{rl|l|c|c|c|l}
\textbf{LoR } \mathbf{t}_i^{(j_i)} & (\ \omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \mathbf{x}_i^{(b,j_i)} & \nu_i^{(j_i)} & \Delta\mathbf{x}_i & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (\ \mathbf{y}_i^{(\ell)} & \sum_{i=1}^n a_{i,0}^{(\ell)} & R\cdot\mathbf{y}_i^{(\ell)} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|c|c|c|l}
\textbf{LoR } \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ \cdots & \boxed{r\mathbf{1}_{\Delta\mathbf{x}_i}} & \boxed{r'\mathbf{1}} & 0 &)_{\mathbf{H}_i} \\
i\in\mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \cdots & \boxed{R\cdot(\Delta\mathbf{x}_i \circ \mathbf{y}_i^{(\ell)})} & \boxed{(\theta_{i,k}\cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

$\boxed{\mathsf{G}_{3.2}}$ (Switching, updating secret shares of 0, Hadamard product is denoted "$\circ$", see example 3 on DPVS basis changes)

$$\begin{array}{rl|l|c|c|c|l}
\textbf{LoR } \mathbf{t}_i^{(j_i)} & (\ \omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \boxed{\mathbf{x}_i^{(1,j_i)}} & \nu_i^{(j_i)} & \Delta\mathbf{x}_i & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (\ \mathbf{y}_i^{(\ell)} & \sum_{i=1}^n a_{i,0}^{(\ell)} & \boxed{R'\cdot\mathbf{y}_i^{(\ell)}} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|c|c|c|l}
\textbf{LoR } \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ \cdots & r\mathbf{1}_{\Delta\mathbf{x}_i} & (r'-r)\mathbf{1} & 0 &)_{\mathbf{H}_i} \\
i\in\mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \cdots & \boxed{R'\cdot\left(\Delta\mathbf{x}_i \circ \mathbf{y}_i^{(\ell)}\right)} & \boxed{(\theta_{i,k}\cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

$\boxed{\mathsf{G}_4 := \mathsf{G}_{3.3}}$ (Formal Quotient, using $\Delta\mathbf{x}_i$ is constant for $i \in \mathcal{H}$ over repetitions, see example 2 on DPVS basis changes)

$$\begin{array}{rl|l|c|c|c|l}
\textbf{LoR } \mathbf{t}_i^{(j_i)} & (\ \omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \mathbf{x}_i^{(1,j_i)} & \nu_i^{(j_i)} & \Delta\mathbf{x}_i & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (\ \mathbf{y}_i & \sum_{i=1}^n a_{i,0}^{(\ell)} & R'\cdot\mathbf{y}_i^{(\ell)} & \mathsf{rnd}_i^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|l|l|c|c|c|l}
\textbf{LoR } \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \boxed{\Delta\mathbf{x}_i} & \boxed{\theta_{i,k}} & 0 &)_{\mathbf{H}_i} \\
i\in\mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \sum_i\langle\mathbf{s}_i,\mathbf{y}_i^{(\ell)}\rangle & \sum_i\langle\mathbf{u}_i,\mathbf{y}_i^{(\ell)}\rangle & a_{i,0}^{(\ell)} z & \boxed{R'\cdot\mathbf{y}_i^{(\ell)}} & \boxed{(d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N} & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

Game G_5 : (Cleaning)

$$\begin{array}{rl|l|c|c|c|l}
\textbf{LoR } \mathbf{t}_i^{(j_i)} & (\ \omega\cdot\mathbf{s}_i + \omega'\cdot\mathbf{u}_i + \mathbf{x}_i^{(1,j_i)} & \nu_i^{(j_i)} & \boxed{\mathbf{0}} & 0 & \rho_i^{(j_i)} &)_{\mathbf{B}_i} \\
\mathbf{m}_i^{(\ell)} & (\ \mathbf{y}_i^{(\ell)} & \sum_{i=1}^n a_{i,0}^{(\ell)} & \boxed{\mathbf{0}} & \mathsf{rnd}^{(\ell)} & 0 &)_{\mathbf{B}_i^*}
\end{array}$$

$$\begin{array}{rl|l|l|c|c|c|l}
\textbf{LoR } \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \boxed{\mathbf{0}} & \boxed{\theta'_{i,k}} & 0 &)_{\mathbf{H}_i} \\
i\in\mathcal{H}\ \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \sum_i\langle\mathbf{s}_i,\mathbf{y}_i^{(\ell)}\rangle & \sum_i\langle\mathbf{u}_i,\mathbf{y}_i^{(\ell)}\rangle & a_{i,0}^{(\ell)} z^{(\ell)} & \boxed{\mathbf{0}} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

Fig. 8: Games $\mathsf{G}_4, \mathsf{G}_5$ for Theorem 12.

performing the product between the i-th ciphertext component and the i-th key component

$$\langle z_j \Delta\mathbf{x}_i, a'_{i,j}/z_j \mathbf{y}_i^{(\ell)} \rangle = \langle \Delta\mathbf{x}_i, a'_{i,j}\mathbf{y}_i^{(\ell)} \rangle = a'_{i,j} \langle \Delta\mathbf{x}_i, \mathbf{y}_i^{(\ell)} \rangle = 0\ .$$

Moreover, because we are dealing with *vectors* $\Delta\mathbf{x}_i, \mathbf{y}_i^{(\ell)}$, the Lemma 1 is applied by components, which is possible due to the appropriate dimension of $\mathbf{c}$-components and $\mathbf{k}$-components, as well as the proof of the Lemma 1 itself (see Appendix B).

$\boxed{\mathsf{G}_{0.\ell.2}}$ We randomize the values $a'_{i,0}$ in the key components by adding a independent fixed random mask R. First of all, we remark that for $i \in \mathcal{H}$ where $\mathbb{A}(\mathsf{S}_i) = 0$, where $\mathbb{A}$ is the LSSS associated to the ℓ-th key query, the change is even perfectly indistinguishable. This is because of the facts that

- the randomized shares $a'_{i,j}/z_j$ are uniformly random and independent thanks to z_j,
- even with *repetitions* at a position i, for a challenge tag tag, the shares z_j are independent for different repetitions given the *private-only* repetitions.
- more importantly, when $\mathbb{A}(\mathsf{S}_i) = 0$, it holds that z_j *never* appears in any of the ciphertexts returned to the adversary
- as a consequence, the shares $a'_{i,j}/z_j$ is information theoretically hidden and making $a'_{i,0}$ information theoretically hidden for the adversary.

In the end, in this case for $i \in \mathcal{H}$ where $\mathbb{A}(\mathsf{S}_i) = 0$, what we do is just rewriting an information theoretically hidden value $a'_{i,0}$ to another information theoretically hidden value $a'_{i,0} + R$, and this change goes perfectly indistinguishable. However, there can be the case where some $i \in \mathcal{H}$ it holds $\mathbb{A}(\mathsf{S}_i) = 1$. This case can be treated by formal basis changes together with a *complexity leveraging* argument. We detail below, the details of calculation for DPVS basis changes are recalled and can be revised in appendix A.5.

The main idea is to consider the *selective* version $\mathsf{G}^*_{0.\ell.1.t}$ for $t \in \{1, 2, 3, 4\}$, where the values $(\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]}$ are guessed in advance. We then use formal argument for the transitions $\mathsf{G}^*_{0.\ell.1.1} \to \mathsf{G}^*_{0.\ell.1.4}$ to obtain for $j \in [3]$,

$$\Pr[\mathsf{G}^*_{0.\ell.j} = 1] = \Pr[\mathsf{G}^*_{0.\ell.j+1} = 1] \ . \tag{14}$$

In the end, we use a *complexity leveraging* argument to conclude that thanks to (14), we have $\Pr[\mathsf{G}_{0.\ell.1} = \mathsf{G}_{0.\ell.1.1} = 1] = \Pr[\mathsf{G}_{0.\ell.2} = \mathsf{G}_{0.\ell.1.4} = 1]$.

For the sequence $\mathsf{G}_{0.\ell.1.1} \to \mathsf{G}_{0.\ell.1.4}$, we make a guess for the values $(\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]}$, choose $R \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, random secret sharings $(\theta_{i,k} \cdot d^{(\ell)}_{\mathbb{A},i,k})_{k=1}^N$ of 0 where $\theta_{i,k} \neq 0$. We define the event E that the guess is correct on $(\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]}$ and for all $k \in [N]$

$$\theta_{i,k} \cdot d^{(\ell)}_{\mathbb{A},i,k} = -R \cdot \Delta\mathbf{x}_i[k]\mathbf{y}_i^{(\ell)}[k] \ . \tag{15}$$

We describe the *selective* games below, starting from $\mathsf{G}^*_{0.\ell.1} = \mathsf{G}^*_{0.\ell.1.1}$, where event E is assumed true:

Game $\mathsf{G}^*_{0.\ell.1} = \mathsf{G}^*_{0.\ell.1.1}$: The vectors have form:

$$\begin{array}{lllc|c|c|c|c|cl}
\textbf{LoR} & \mathbf{c}^{(j_i)}_{i,\mathsf{ipfe}} & (& p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta\mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 &)_{\mathbf{H}_i} \\
i \in \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a^{(\ell)}_{i,0} z^{(\ell)} & a'_{i,0} \cdot \mathbf{y}_i^{(\ell)} & (d^{(\ell)}_{\mathbb{A},i,k})_{k=1}^N & \mathsf{rnd}^{(\ell)}_{i,\mathsf{ipfe}} &)_{\mathbf{H}_i^*}
\end{array}$$

Game $\mathsf{G}^*_{0.\ell.1.2}$: We perform a formal basis change to the key components, for $i \in \mathcal{H}$ to change $(\mathbf{H}_i, \mathbf{H}_i^*)$ following matrices: for $r, r' \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$,

$$H_i[row, col] = \begin{cases} 1 & \text{if } row = col \leq 3 \\ \frac{r}{\Delta\mathbf{x}_i[z]} & \text{if } \exists z \in [N] \text{ s.t. } row = col = 3 + z \ \wedge \ \Delta\mathbf{x}_i[z] \neq 0 \\ 1 & \text{if } \exists z \in [N] \text{ s.t. } row = col = 3 + z \ \wedge \ \Delta\mathbf{x}_i[z] = 0; H'_i := \left(H_i^{-1}\right)^\top \\ \frac{r'}{\theta_{i,z}} & \text{if } \exists z \in [N] \text{ s.t. } row = col = N + 3 + z \\ 0 & \text{otherwise} \end{cases} .$$

We remark that the matrix does not have to check non-zeroness of $\theta_{i,z}$, as it is guaranteed by the event E. The vectors have form: we denote the Hadamard product by "$\circ$", and $\mathbf{1}_{\Delta\mathbf{x}_i}$ is the vector of 1's at the positions where $\Delta\mathbf{x}_i$ is non-zero

$$\begin{array}{lllllll} \textbf{LoR} & \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& \cdots & r \cdot \boxed{\mathbf{1}_{\Delta \mathbf{x}_i}} & r' \cdot \boxed{\mathbf{1}} & 0 \quad)_{\mathbf{H}_i} \\ i \in \mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \cdots & a'_{i,0} \cdot \left(\Delta \mathbf{x}_i \circ \mathbf{y}_i^{(\ell)}\right) & (\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} \quad)_{\mathbf{H}_i^*} \end{array}$$

Game $\mathsf{G}^*_{0.\ell.1.3}$: We perform a formal basis change to the key components, for $i \in \mathcal{H}$ to change $(\mathbf{H}_i, \mathbf{H}_i^*)$ following matrices: for $r, r' \xleftarrow{\$} \mathbb{Z}_q^*$, (for ease of presenting basis changes we write the transposed matrix $H_i^\top$)

$$H_i^\top[row, col] = \begin{cases} 1 & \text{if } row = col \notin \{4+N, \ldots, 3+2N\} \\ 1 & \text{if } row = col \in \{4+N, \ldots, 3+2N\} \ \wedge \ \Delta \mathbf{x}_i[row - N - 3] \neq 0 \\ \frac{r'}{r'+r} & \text{if } row = col \in \{4+N, \ldots, 3+2N\} \ \wedge \ \Delta \mathbf{x}_i[row - N - 3] = 0; \\ -1 & \text{if } \exists z \in [N] \text{ s.t. } row = 3+N+z \ \wedge \ col = 3+z \\ 0 & \text{otherwise} \end{cases} \quad H_i' := \left(H_i^{-1}\right)^\top .$$

We note that on the diagonal $\tilde{z} := row = col \in \{4+N, 3+2N\} \ \wedge \ \Delta \mathbf{x}_i[\tilde{z} - N - 3] = 0$, because coordinate $\mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)}[\tilde{z} - N] = 0$ as $\Delta \mathbf{x}_i[\tilde{z} - N - 3] = 0$, the moving by $H_i^\top[3 + N + \tilde{z} - N - 3, 3 + \tilde{z} - N - 3]$ has no effect on $\mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)}[\tilde{z}]$. Thus $H_i^{-1}[row, col]$ needs to multiply a factor $(r' + r)/r'$[9] to the coordinate $\mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)}[\tilde{z}]$ to make sure that after the basis change it becomes $r' + r$. Dually the coordinate $\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}[\tilde{z}] = \theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)}$ stays correctly thanks to the relation (15) and we pay attention that $\Delta \mathbf{x}_i[\tilde{z} - N - 3] = 0$. The vectors have form: we denote the Hadamard product by "$\circ$"

$$\begin{array}{lllllll} \textbf{LoR} & \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& \cdots & r \cdot \mathbf{1}_{\Delta \mathbf{x}_i} & \boxed{(r + r') \cdot \mathbf{1}} & 0 \quad)_{\mathbf{H}_i} \\ i \in \mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \cdots & \boxed{(a'_{i,0} + R) \cdot (\Delta \mathbf{x}_i \circ \mathbf{y}_i^{(\ell)})} & (\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} \quad)_{\mathbf{H}_i^*} \end{array}$$

using the hypothesis that event E happens along with the relation (15) specifically. Consequently, we just update one secret share of 0 by another. The randomness r' is updated to $r' + r$, indentically distributed.

Game* $\mathsf{G}^*_{0.\ell.1.4}$: We undo the formal basis changes $\mathsf{G}^*_{0.\ell.1.1} \to \mathsf{G}^*_{0.\ell.1.2}$, where the division by $1/r, 1/(r + r')$ can be done with overwhelming probability since $r, r' \xleftarrow{\$} \mathbb{Z}_q^*$ at the beginning of the game to define the matrices. This gives

$$\begin{array}{lllllll} \textbf{LoR} & \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (\ p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta \mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 \quad)_{\mathbf{H}_i} \\ i \in \mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (\ \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a_{i,0}^{(\ell)} z^{(\ell)} & \boxed{(a'_{i,0} + R) \cdot \mathbf{y}_i^{(\ell)}} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} \quad)_{\mathbf{H}_i^*} \end{array}$$

The above games demonstrate relation (14). We now employ the complexity leveraging argument. Let us fix $j \in \{1, 2, 3\}$. For $u \in \{0.\ell.1.j, 0.\ell.1.j + 1\}$ let $\mathsf{Adv}_u(\mathcal{A}) := |\Pr[\mathsf{G}_u(\mathcal{A}) = 1] - 1/2|$ denote the advantage of a ppt adversary $\mathcal{A}$ in game G_u. We build a ppt adversary $\mathcal{B}^*$ playing against G_u^* such that its advantage $\mathsf{Adv}_u^*(\mathcal{B}^*) := |\Pr[\mathsf{G}_u^*(\mathcal{B}^*) = 1] - 1/2|$ equals $\gamma \cdot \mathsf{Adv}_u(\mathcal{A})$ for $u \in \{t, t + 1\}$, for some constant γ.

The adversary $\mathcal{B}^*$ first guesses the values $(\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]}$, choose $R \xleftarrow{\$} \mathbb{Z}_q^*$, random secret sharings $(\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N$ of 0. Then $\mathcal{B}^*$ defines the event E that:

> the guess is correct on $(\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]}$ and for all $k \in [N]$, $\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)} = -R \cdot \Delta \mathbf{x}_i[k] \mathbf{y}_i^{(\ell)}[k]$.

When $\mathcal{B}^*$ guesses successfully and E happens, then the simulation of $\mathcal{A}$'s view in G_t is perfect. Otherwise, $\mathcal{B}^*$ aborts the simulation and outputs a random bit b'. Since E happens with

[9] Therefore the corresponding position on the diagonal of $H_i^\top[\tilde{z}, \tilde{z}] = \frac{r'}{r'+r}$.

some fixed probability γ and is independent from the view of $\mathcal{A}$, we have[10]:

$$\begin{aligned}
\mathsf{Adv}_u^*(\mathcal{B}^*) &= \left|\Pr[\mathsf{G}_u^*(\mathcal{B}^*) = 1] - \frac{1}{2}\right| \\
&= \left|\Pr[E] \cdot \Pr[\mathsf{G}_u^*(\mathcal{B}^*) = 1 \mid E] + \frac{\Pr[\neg E]}{2} - \frac{1}{2}\right| \\
&= \left|\gamma \cdot \Pr[\mathsf{G}_u^*(\mathcal{B}^*) = 1 \mid E] + \frac{1-\gamma-1}{2}\right| \\
&\overset{(*)}{=} \gamma \cdot \left|\Pr[\mathsf{G}_u(\mathcal{A}) = 1] - \frac{1}{2}\right| = \gamma \cdot \mathsf{Adv}_u(\mathcal{A})
\end{aligned} \tag{16}$$

where $(*)$ comes from the fact that conditioned on E, $\mathcal{B}$ simulates perfectly G_u for $\mathcal{A}$, therefore $\Pr[\mathsf{G}_u(\mathcal{A}) = 1 \mid E] = \Pr[\mathsf{G}_u^*(\mathcal{B}^*) = 1 \mid E]$, then we apply the independence between E and $\mathsf{G}_u(\mathcal{A}) = 1$. Together with relation (14), this concludes that $\Pr[\mathsf{G}_{0.\ell.1.j} = 1] = \Pr[\mathsf{G}_{0.\ell.1.j+1} = 1]$ for any fixed $j \in \{1, 2, 3\}$, in particular $\Pr[\mathsf{G}_{0.\ell.1} = \mathsf{G}_{0.\ell.1.1} = 1] = \Pr[\mathsf{G}_{0.\ell.2} = \mathsf{G}_{0.\ell.1.4} = 1]$. Union bounds on $\mathbb{A}(\mathsf{S}_i) = 0$ (perfect indistinguishability by information-theoretic argument on z_j and $a'_{i,j}/z_j$) and $\mathbb{A}(\mathsf{S}_i) = 1$ (perfect indistinguishability by complexity leveraging) give the conclusion that the game hop is perfectly indistinguishable.

$\boxed{\mathsf{G}_{0.\ell.3}}$ Reverse Masking Application - Lemma 1, so that only mask $R\mathbf{y}_i$ remains for $i \in \mathcal{H}$, in $\mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)}$. Once again, the mask R will be canceled by the admissibility condition:

$$\sum_{i \in \mathcal{H}} \langle \mathbf{x}_i^{(b,j_i)} - \mathbf{x}_i^{(1,j_i)}, \mathbf{y}_i^{(\ell)} \rangle = 0 \ .$$

We arrive at G_1 after $\mathsf{G}_{0.K.3}$.

Game G_2: We rewrite the game's description to program the vectors $\widetilde{\mathbf{t}}_{0,j}^{(j_i)} = \sum_{i \in \mathcal{H}} \widetilde{\mathbf{t}}_{i,j}^{(j_i)}$. The goal is to consider $\widetilde{\mathbf{t}}_{0,j}^{(j_i)}$ in the subsequent games, *i.e.* we look at the vectors $\widetilde{\mathbf{t}}_{0,j}^{(j_i)}$ instead of the given $\widetilde{\mathbf{t}}_{i,j}^{(j_i)}$ returned to the adversary. The rewriting is totally formal as it follows exactly what is described in Figure 2.

Game G_3: We apply similarly Lemma 1 as in $\mathsf{G}_1 \to \mathsf{G}_2$, by a sequence of hybrids over the ℓ-th functional key, one after another. We remark that the random factor $R \xleftarrow{\$} \mathbb{Z}_q$ is the same as that one introduced in $\mathsf{G}_1 \to \mathsf{G}_2$, this simplifies one guess during the complexity leveraging argument. The formal basis changes resembles those in $\mathsf{G}_1 \to \mathsf{G}_2$ and in the end, the game hop is perfectly indistinguishable.

Game G_4: We use a *complexity leveraging* argument, that depends only on *formal basis changes*. The goal is to switch from $\mathbf{x}_i^{(b,j_i)}$ to $\mathbf{x}_i^{(1,j_i)}$ for $i \in \mathcal{H}$. The details of the *selective* underlying games are given in Figure 8. First of all, we make a guess for the values $(\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]}$, choose $R \xleftarrow{\$} \mathbb{Z}_q^*$, random secret sharings $(\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N$ of 0 where $\theta_{i,k} \neq 0$. We define the event F that

$$\begin{aligned}
&\text{the guess is correct on } (\mathbf{x}_i^{(1,j_i)}, \mathbf{x}_i^{(0,j_i)}, \mathbf{y}_i^{(\ell)})_{i \in [n]}^{j_i \in [J_i]} \text{ and for all } k \in [N] \\
&\qquad \theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)} = -\Delta \mathbf{x}_i[k] \mathbf{y}_i^{(\ell)}[k] \ ,
\end{aligned} \tag{17}$$

so as to make sure $\theta_{i,k} \cdot d_{\mathbb{A},i,k}^{(\ell)}$ is a secret sharing of 0 conditioned on F. We give the matrices' definitions as follows to demonstrate how the calculation is performed:

Game $\mathsf{G}_{3.1}^* = \mathsf{G}_3^*$: The vectors have form:

$$\begin{array}{lcc|c|c|c|c|c|c}
\mathbf{LoR} & \mathbf{c}_{i,\mathsf{ipfe}}^{(j_i)} & (& p_i \omega_{\mathsf{tag}} & p_i \omega'_{\mathsf{tag}} & \psi_i^{(j_i)} & \Delta \mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 &)_{\mathbf{H}_i} \\
i \in \mathcal{H} & \mathbf{k}_{i,\mathsf{ipfe}}^{(\ell)} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}_i^{(\ell)} \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}_i^{(\ell)} \rangle & a_{i,0}^{(\ell)} z^{(\ell)} & R \cdot \mathbf{y}_i^{(\ell)} & (d_{\mathbb{A},i,k}^{(\ell)})_{k=1}^N & \mathsf{rnd}_{i,\mathsf{ipfe}}^{(\ell)} &)_{\mathbf{H}_i^*}
\end{array}$$

[10] This calculation (16) to relate $\mathsf{Adv}_u^*(\mathcal{B}^*)$ to $\mathsf{Adv}_u(\mathcal{A})$ is the core of our complexity levaraging argument, being built upon the previous information-theoretic game transtions and the probability of event E.

Game $\mathsf{G}^*_{3.2}$: We perform a formal basis change to the key components, for $i \in \mathcal{H}$ to change $(\mathbf{H}_i, \mathbf{H}^*_i)$ following matrices: for $r, r' \xleftarrow{\$} \mathbb{Z}^*_q$,

$$H_i[row, col] = \begin{cases} 1 & \text{if } row = col \leq 3 \\ \frac{r}{\Delta \mathbf{x}_i[z]} & \text{if } \exists z \in [N] \text{ s.t. } row = col = 3 + z \ \wedge \ \Delta \mathbf{x}_i[z] \neq 0 \\ 1 & \text{if } \exists z \in [N] \text{ s.t. } row = col = 3 + z \ \wedge \ \Delta \mathbf{x}_i[z] = 0 \\ \frac{r'}{\theta_{i,z}} & \text{if } \exists z \in [N] \text{ s.t. } row = col = N + 3 + z \\ 1 & \text{if } \exists \tilde{j} \in [J], z \in [N] \text{ s.t. } row = col = N + 3 + z \\ 0 & \text{otherwise} \end{cases} ; H'_i := \left(H_i^{-1}\right)^\top .$$

We remark that the matrix does not have to check non-zeroness of $\theta_{i,z}$, as it is guaranteed by the event F. The vectors have form: we denote the Hadamard product by "$\circ$", and $\mathbf{1}_{\Delta \mathbf{x}_i}$ is the vector of 1's at the positions where $\Delta \mathbf{x}_i$ is non-zero

$$\begin{array}{lcc|c|c|c|c} \textbf{LoR} & \mathbf{c}^{(j_i)}_{i,\mathsf{ipfe}} & (\ \cdots & r \cdot \boxed{\mathbf{1}_{\Delta \mathbf{x}_i}} & r' \cdot \boxed{\mathbf{1}} & 0 &)_{\mathbf{H}_i} \\ i \in \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (\ \cdots & \boxed{R \cdot \left(\Delta \mathbf{x}_i \circ \mathbf{y}^{(\ell)}_i\right)} & \boxed{(\theta_{i,k} \cdot d^{(\ell)}_{\mathbb{A},i,k})_{k=1}^N} & \mathsf{rnd}^{(\ell)}_{i,\mathsf{ipfe}} &)_{\mathbf{H}^*_i} \end{array}$$

Game $\mathsf{G}^*_{3.3}$: We perform a formal basis change to the key components, for $i \in \mathcal{H}$ to change $(\mathbf{B}_i, \mathbf{B}^*_i)$, $(\mathbf{H}_i, \mathbf{H}^*_i)$ following matrices: for $r, r' \xleftarrow{\$} \mathbb{Z}^*_q$, (for ease of presenting basis changes we write the transposed matrix $H_i^\top$ and B_i^{-1})

$$H_i^\top[row, col] = \begin{cases} 1 & \text{if } row = col \notin \{4 + n, \ldots, 3 + 2N\} \\ 1 & \text{if } row = col \in \{4 + n, \ldots, 3 + 2N\} \ \wedge \ \Delta \mathbf{x}_i[row - N - 3] \neq 0 \\ \frac{r'}{r' + r} & \text{if } row = col \in \{4 + n, \ldots, 3 + 2N\} \ \wedge \ \Delta \mathbf{x}_i[row - N - 3] = 0 \\ -1 & \text{if } \exists z \in [N] \text{ s.t.} \\ & \quad row = 3 + z \ \wedge \ col = 3 + N + z \\ 0 & \text{otherwise} \end{cases} ; \ H'_i := \left(H_i^{-1}\right)^\top$$

$$B_i^{-1}[row, col] = \begin{cases} 1 & \text{if } row = col \\ -1 & \text{if } \exists z \in [N] \text{ s.t.} \\ & \quad row = 1 + N + z \ \wedge \ col = z \\ 0 & \text{otherwise} \end{cases} ; \ B'_i := \left(B_i^{-1}\right)^\top .$$

Following the matrices

- The formal changes of $(\mathbf{B}_i, \mathbf{B}^*_i)$ switch $\mathbf{x}^{(b,j_i)}_i$ to $\mathbf{x}^{(1,j_i)}_i$ for $i \in \mathcal{H}$, where for $z \in [N]$ under B_i^{-1}, the coordinate $\mathbf{t}^{(j_i)}_i[z]$ is updated to

  $$\mathbf{t}^{(j_i)}_i[z] - \Delta \mathbf{x}_i[z] = \omega \cdot \mathbf{s}_i[z] + \omega' \cdot \mathbf{u}_i[z] + \mathbf{x}^{(b,j_i)}_i[z] + \mathbf{x}^{(1,j_i)}_i[z] - \mathbf{x}^{(b,j_i)}_i[z] = \omega \cdot \mathbf{s}_i[z] + \omega' \cdot \mathbf{u}_i[z] + \mathbf{x}^{(1,j_i)}_i[z] .$$

  While dually in $\mathbf{m}^{(\ell)}_i[1 + N + z]$ the matrix $B_i^\top$ introduces $R' \mathbf{y}^{(\ell)} := (R + 1) \cdot \mathbf{y}^{(\ell)}_i$ staying regroupable with the corresponding $\Delta \mathbf{x}_i$ in $\mathbf{t}^{(j_i)}_i[1 + N + z]$.
- The changes of $(\mathbf{H}_i, \mathbf{H}^*_i)$ are also to correct R to R' in the key components, thanks to (17) of the games that we recall under this selective sequence, so that the decryption's correctness is preserved. We note that the diagonal of $H_i^\top$ also takes care of the case where $\Delta \mathbf{x}_i[z] = 0$ for $z \in [N]$, in the same manner as we have done for $\mathsf{G}^*_{0.\ell.1.2} \to \mathsf{G}^*_{0.\ell.1.3}$ previously.

The vectors have form: we denote the Hadamard product by "$\circ$"

$$\begin{array}{lcc|c|c|c|c|c} \textbf{LoR} & \mathbf{t}^{(j_i)}_i & (\ \omega \cdot \mathbf{s}_i + \omega' \cdot \mathbf{u}_i + \boxed{\mathbf{x}^{(1,j_i)}_i} & \nu^{(j_i)}_i & \Delta \mathbf{x}_i & 0 & \rho^{(j_i)}_i &)_{\mathbf{B}_i} \\ & \mathbf{m}^{(\ell)}_i & (\ \mathbf{y}^{(\ell)}_i & \sum_{i=1}^n a^{(\ell)}_{i,0} & \boxed{R' \cdot \mathbf{y}^{(\ell)}_i} & \mathsf{rnd}^{(\ell)}_i & 0 &)_{\mathbf{B}^*_i} \end{array}$$

$$\begin{array}{lcc|c|c|c|c} \textbf{LoR} & \mathbf{c}^{(j_i)}_{i,\mathsf{ipfe}} & (\ \cdots & r \cdot \mathbf{1}_{\Delta \mathbf{x}_i} & \boxed{(r' + r) \cdot \mathbf{1}} & 0 &)_{\mathbf{H}_i} \\ i \in \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (\ \cdots & \boxed{R' \cdot \left(\Delta \mathbf{x}_i \circ \mathbf{y}^{(\ell)}_i\right)} & \boxed{(\theta_{i,k} \cdot d^{(\ell)}_{\mathbb{A},i,k})_{k=1}^N} & \mathsf{rnd}^{(\ell)}_{i,\mathsf{ipfe}} &)_{\mathbf{H}^*_i} \end{array}$$

using the hypothesis that event F happens along with the relation (17) specifically. Consequently, we just update one secret share of 0 by another. The randomness r' is updated to $r' + r$, indentically distributed.

Game* $\mathsf{G}^*_{3.4}$**:** We undo the formal basis changes $\mathsf{G}^*_{3.1} \to \mathsf{G}^*_{3.2}$ and obtain

$$\begin{array}{llll|c|c|c|c|cl}
\textbf{LoR} & \mathbf{c}^{(j_i)}_{i,\mathsf{ipfe}} & (& p_i\omega_{\mathsf{tag}} & p_i\omega'_{\mathsf{tag}} & \psi^{(j_i)}_i & \Delta\mathbf{x}_i & (\theta_{i,k})_{k=1}^N & 0 &)_{\mathbf{H}_i} \\
i \in \mathcal{H} & \mathbf{k}^{(\ell)}_{i,\mathsf{ipfe}} & (& \sum_i \langle \mathbf{s}_i, \mathbf{y}^{(\ell)}_i \rangle & \sum_i \langle \mathbf{u}_i, \mathbf{y}^{(\ell)}_i \rangle & a^{(\ell)}_{i,0} z^{(\ell)} & \boxed{R' \cdot \mathbf{y}^{(\ell)}_i} & (d^{(\ell)}_{\mathbb{A},i,k})_{k=1}^N & \mathsf{rnd}^{(\ell)}_{i,\mathsf{ipfe}} &)_{\mathbf{H}^*_i}
\end{array}$$

Game G_5**:** We clean the masks so that the adversary's view is independent of the challenge b.

The bit b does not appear in the responses to the adversary anymore, completing the proof. □

D.1 Upgrading Security from Theorem 12

Next, we can apply a layer of All-or-Nothing Encapsulation (AoNE) so as to remove the tradeoff with respect to *incomplete* challenge ciphertexts (*i.e.* remove *pos*-condition in Definition 11, under the private-input repetitions). That is, the adversary can now omit some honest slot $i \in \mathcal{H}$ with respect to $(\mathsf{tag}, \mathsf{S}_i)$ for different S_i, up to repetition $(\mathbf{x}_i^{(0,j_i)}, \mathbf{x}_i^{(1,j_i)})$. More specifically, we apply the generic transformation from [33, Lemma 16], to treat the case of MCFE with access control as a special case in the above lemma so as to remove *pos*-condition, given the private-input repetitions. The formal statement is stated below.

Lemma 27 (Incomplete Security with Private-Only Repetitions). *Assume there exist (1) a one-challenge* MCFE *scheme* $\mathcal{E}^{\mathsf{pos}}$ *for the function class* $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$ *that is secure against* complete *queries,* i.e. *satisfying* pos-*security and (2) an* AoNE *scheme* $\mathcal{E}^{\mathsf{aone}}$ *whose message space contains the ciphertext space of* $\mathcal{E}^{\mathsf{pos}}$*. Then there exists a one-challenge* MCFE *scheme* $\mathcal{E}$ *for the same function class* $\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS}$ *that is even secure against* incomplete *queries. More precisely, for any* ppt *adversary* $\mathcal{A}$*, there exist* ppt *algorithms* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{rep\text{-}priv\text{-}1chal\text{-}xxx}}_{\mathcal{E},\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS},\mathcal{A}}(1^\lambda) \leq 12 \cdot \left(\mathsf{Adv}^{\mathsf{rep\text{-}priv\text{-}pos\text{-}1chal\text{-}xxx}}_{\mathcal{E}^{\mathsf{pos}},\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS},\mathcal{B}_1}(1^\lambda) + \mathsf{Adv}^{\mathsf{rep\text{-}priv\text{-}1chal\text{-}xxx}}_{\mathcal{E}^{aone},\mathcal{F}^{\mathsf{IP},B}_{\mathsf{subvec}} \times \mathsf{LSSS},\mathcal{B}_2}(1^\lambda) \right)$$

where $\mathsf{xxx} \subseteq \{\mathsf{stat}, \mathsf{sel}\}$.

We refer to the proof of the more general lemma in [33, Lemma 16], with repetitions on the private inputs $\mathbf{x}_i$. Finally, by combining with Lemma 7 to allow multiple challenge tags, where the only restriction remains: solely for *private* inputs, and not *public* attributes per client, will repetitions be allowed.