Multi-Client Attribute-Based Unbounded Inner Product Functional Encryption, and More

Subhranil Dutta[†], Aikaterini Mitrokotsa[†], Tapas Pal[⋆], Jenit Tomy[†]

[†]University of St. Gallen, St. Gallen, Switzerland
{subhranil.dutta, katerina.mitrokotsa, jenit.tomy}@unisg.ch
[⋆]Karlsruhe Institute of Technology, KASTEL SRL, Karlsruhe, Germany
tapas.pal@kit.edu

March 5, 2025

Abstract

This paper presents the concept of a multi-client functional encryption (MC-FE) scheme for *attribute-based inner product functions* (AB-IP), initially proposed by Abdalla et al. [ASIACRYPT'20], in an *unbounded* setting. In such a setting, the setup is independent of vector length constraints, allowing secret keys to support functions of arbitrary lengths, and clients can dynamically choose vector lengths during encryption. The functionality outputs the sum of inner products if vector lengths and indices meet a specific relation, and all clients' attributes satisfy the key's policy. We propose the following constructions based on the matrix decisional Diffie-Hellman assumption in a natural permissive setting of unboundedness:

- the *first* multi-client attribute-based unbounded IPFE (MC-AB-UIPFE) scheme secure in the standard model, overcoming previous limitations where clients could only encrypt fixed-length data;
- the *first* multi-input AB-UIPFE (MI-AB-UIPFE) in the public key setting; improving upon prior bounded constructions under the same assumption;
- the *first* dynamic decentralized UIPFE (DD-UIPFE); enhancing the dynamism property of prior works.

Technically, we follow the blueprint of Agrawal et al. [CRYPTO'23] but begin with a new unbounded FE called *extended slotted unbounded* IPFE. We first construct a single-input AB-UIPFE in the standard model and then extend it to multi-input settings. In a nutshell, our work demonstrates the applicability of function-hiding security of IPFE in realizing variants of multi-input FE capable of encoding unbounded length vectors both at the time of key generation and encryption.

Keywords. Inner product functional encryption · Multi-client · Dynamic decentralized · Unbounded · Attribute-based

Contents

1 Introduction

Multi-Party Functional Encryption. Functional encryption (FE) [16, 44] is a powerful cryptographic primitive that enables the computation of a function on encrypted data, departing from the traditional *all-or-nothing* approach of public key encryption. Crucially, it reveals only the output of the specified function, without disclosing any additional information about the underlying data. FE supporting a specific function class $\mathcal{F}$, issues a secret key SK_f associated with a function $f \in \mathcal{F}$ using a master secret key, computes a ciphertext $\mathsf{CT}_{\mathbf{x}}$ associated with a message $\mathbf{x}$. By combining the SK_f with $\mathsf{CT}_{\mathbf{x}}$, the decryptor learns $f(\mathbf{x})$, but gains no further insight into the original message $\mathbf{x}$. FE has received significant attention in the literature, with numerous schemes being proposed to achieve a wide array of functionalities under diverse cryptographic assumptions [29, 30, 49, 3, 5, 50].

Initially, FE was defined for the single input setting, *i.e.,* assuming that there is a single encryptor of the data and a single key generator. However, in realistic applications, data often originates from multiple sources, and joint computations across this distributed data are frequently necessary—such as for performing aggregate statistical analyses over data owned by various parties. To capture these more complex and realistic scenarios, several extensions of FE have been proposed ranging from *multi-authority* FE [8, 23], *multi-input* [31, 9, 4, 12], *multi-client* FE [20, 1, 35, 12, 39] to *decentralized multi-client* FE [18, 2, 34], and *dynamic decentralized* FE [19, 12]. To unify and generalize the primitives that enable multi-user functionality in FE, Agrawal et al. [8] introduced the concept of *multi-party functional encryption* (MP-FE) that allows both distributed ciphertexts and distributed keys, specifying how these can be combined to facilitate function evaluation. In this work, we focus on the following types of MP-FEs:

- *Multi-Client FE (MC-FE)*: MC-FE [31, 18] considers a fixed number of parties or clients, say n, each with their own inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and allow computing joint functions on their data *i.e.,* $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. In more detail, party i encrypts its input $\mathbf{x}_i$ under a label/timestamp L_i to obtain CT_i, a key authority that holds the master secret MSK generates a functional key SK_f that enables the decryptor to compute $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ from the collection of ciphertexts $\{\mathsf{CT}_i\}_{i \in [n]}$ only if they share the same label, *i.e.*, $L_i = L$ for all $i \in [n]$.

- *Multi-Input FE (MI-FE)*: MI-FE [31] generalizes the concept of MC-FE in the sense that MI-FE sets no restriction on the way that ciphertexts can be combined and allows all possible combinations of ciphertexts during decryption. An MI-FE scheme in the public key setting with corruption is rather challenging to construct compared to an MC-FE for the same function class due to the absence of labels.

  Recently, attribute-based extensions of MC-FE and MI-FE (MC-AB-FE, MI-AB-FE) have been proposed in [5, 38, 12] which integrate an additional layer of access control on top of the functionality already offered by MC-FE and MI-FE. Here, the inputs are associated with clients' attributes and a policy is embedded into the secret key such that the decryption recovers the functional value only when the attributes of all the clients individually satisfy the policy of the secret key.

- *Dynamic Decentralized FE (DD-FE)*: DD-FE [19] is a decentralized variant of FE that enables the local and independent generation of *both* ciphertexts and keys, eliminating the need for a central authority. In a DD-FE, the clients can dynamically join the system without reliance on any central authority, allowing greater flexibility and autonomy. During the encryption or key generation process, users can specify a set of participants whose inputs can be combined during decryption to perform joint computations. Currently, DD-FEs are designed to support linear functions [19] and attribute-weighted sums computations [12].

A common limitation shared by all these MP-FEs is that the input length for each party, and consequently the size of the functions operating on those inputs, is fixed at the time of setup. This *boundedness* significantly restricts the scope of applications for current MP-FE schemes, despite their ability to support various interesting and useful classes of functions.

Unbounded Functional Encryption. Although there exist several FEs capable of supporting arbitrary circuits and Turing machines [29, 49, 14, 11, 33] they currently rely on impractical cryptographic primitives such as indistinguishability obfuscation or multi-linear maps. In contrast, FE for specific function classes such as linear and quadratic functions and their variants [3, 10] are built upon well-established, standard assumptions, making them more feasible for practical use. Unbounded FE (UFE) offers greater flexibility compared to its bounded counterparts, as it allows the generation of secret keys and ciphertexts for functions and messages of arbitrary lengths. This makes the setup of UFE independent of any predetermined bound on function or message lengths, a crucial and essential feature—particularly in the context of MP-FE—as it enables the parties to encrypt variable-length data during encryption or generate keys for functions of any size. Moreover, unlike (bounded) FE, where the sizes of all ciphertexts and keys depend on the maximum bound set for the data/function length during setup, UFE produces input-specific sizes for ciphertexts and keys. This is a highly desirable property in multi-party settings, as it allows parties to allocate storage sizes tailored to their specific input requirements.

The concept of *unboundedness* in single-input FE was concurrently studied by Tomida and Takashima [47] and Dufour Sans et al. [26] for linear functions called unbounded inner product functional encryption (UIPFE). A UIPFE generates a secret key $\mathsf{SK}_{\mathbf{y}}$ for a vector $\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}}$ and computes a ciphertext $\mathsf{CT}_{\mathbf{x}}$ for a vector $\mathbf{x} \in \mathbb{Z}^{\ell}$. We adopt the *permissive* unboundedness property for decryption [26], referred to as "ct-dominant" in [47], which is considered most practical for real-world applications. For two vectors $\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}}$ an input to key generation and $\mathbf{x} = (x_i)_{i \in [\ell]}$ an input to the encryption, we say the permissive unboundedness condition holds for decryption if $I_{\mathbf{y}} \subseteq [\ell]$ and the inner product is defined as $\sum_{i \in I_{\mathbf{y}}} x_i y_i$. In contrast, *strict* unboundedness [26] requires $I_{\mathbf{y}} = [\ell]$ for decryption. Since the permissive case of unboundedness is more natural, many subsequent works [25, 24, 46] built UFE for variants of linear and quadratic functions from standard assumptions. The permissive setting, particularly in multi-input scenarios, better aligns with the flexible nature of FE, offering finer control over unbounded encrypted vectors.

Recently, Datta and Pal [23] developed UFE for attribute-based linear functions in the multi-authority setting with distributed secret keys. However, to the best of our knowledge, UFE schemes have yet to be explored in multi-client or dynamic decentralized settings. In this work, we initiate the study of MC-AB-UFE, MI-AB-UFE and DD-UFE for specific function classes, thereby enriching the landscape of MP-FEs and addressing more practical applications.

1.1 Our Results

In this work, we enrich the domain of multi-input functional encryption schemes for attribute-based linear functions by introducing unbounded input vector lengths. Specifically, we formalize the concepts of multi-client attribute-based unbounded IPFE (MC-AB-UIPFE), multi-input attribute-based unbounded IPFE (MI-AB-UIPFE), and dynamic decentralized unbounded IPFE (DD-UIPFE), where the *unboundedness* is naturally defined for practical applications. We also present constructions of these primitives in a selective corruption model, achieving indistinguishability-based security (IND-security) relying on the matrix DDH (MDDH) assumption. The attribute-based access control is considered in the key-policy setting, where secret keys are generated for access structures $\mathbb{A}$ realizable by LSSS [32] and ciphertexts are computed under a set of attributes S. We proceed by defining functionalities and their features.

Multi-Client AB-UIPFE. We design an MC-AB-UIPFE where the number of clients n is fixed in the setup, and each client is given an encryption key which is independent of the lengths of vectors. The secret keys are generated by the authority for the tuple $(\mathbb{A}, (\mathbf{y}_k = (y_{k,i})_{i \in I_{\mathbf{y}_k}})_{k \in [n]})$, the ciphertexts are computed by the clients corresponding to their attributes S_k, label L_k and chosen vectors $\mathbf{x}_k \in \mathbb{Z}_p^{\ell_k}$, and the decryption recovers:

$$f((\mathbb{A}, (\mathbf{y}_k, I_{\mathbf{y}_k})_{k \in [n]}), (\mathsf{S}_k, L_k, \mathbf{x}_k, \ell_k)_{k \in [n]})$$
$$= \begin{cases} \sum_{k \in [n]} \sum_{i \in I_{\mathbf{y}_k}} x_{k,i} y_{k,i} & \text{if } (I_{\mathbf{y}_k} \subseteq [\ell_k]) \wedge (\mathbb{A}(\mathsf{S}_k) = 1) \wedge (L_k = L),\ \forall k \in [n] \\ \perp & \text{otherwise} \end{cases}$$

Work	Scheme	Parties	Input length	Function length	Access Control	Label	Corr.	Assumption
[5]	MI-AB-IPFE	n	bnd	bnd	MSP	N/A	×	DDH
[18]	MC-IPFE	n	bnd	bnd	N/A	✓	✓	DDH
	DMC-IPFE	n	bnd	bnd	N/A	✓	✓	SXDH
[1]	MC-IPFE	n	bnd	bnd	N/A	✓	✓	DDH, LWE, DCR
[19]	DD-IPFE	unbd	bnd	bnd	N/A	✓	✓	DDH
[38]	MC-AB-IPFE	n	bnd	bnd	LSSS	OT	✓	SXDH
[9]	MI-QFE	n	bnd	bnd	N/A	N/A	✓	MDDH
[12]	MC-FE for AWS	n	unbd	bnd	N/A	✓	✓	MDDH
	MI-AB-FE for AWS	n	unbd	bnd	ABP	N/A	✓	MDDH
	DD-FE for AWS	unbd	unbd	bnd	N/A	✓	✓	MDDH
[45]	MC-IPFE	n	bnd	bnd	N/A	✓	✓	MDDH
[40]	DMC-IPFE	n	bnd	bnd	N/A	OT	✓	SXDH
This work	MC-AB-UIPFE	n	unbd	unbd	LSSS	OT	✓	MDDH
	MI-AB-UIPFE	n	unbd	unbd	LSSS	N/A	✓	MDDH
	DD-UIPFE	unbd	unbd	unbd	N/A	✓	✓	MDDH

Table 1: Comparison among multi-party FE schemes. Here, DMC means decentralized multi-client; bnd, unbd mean bounded, unbounded; Assum is a shorthand for assumption; SXDH, LWE, DCR stand for symmetric external Diffie-Hellman, learning with errors, decision composite residuosity; MSP means monotone span program; Label refers to the capability of labelling functionality that restricts decryption such that it is allowed only when all labels are equal. OT means each label can be used once per input; Corr is a shorthand for Corruption.

To the best of our knowledge, this is the first unbounded FE in a multi-input setting where both the size of functions and messages remain unrestricted during setup. Previously, Nguyen, Phan, and Pointcheval [38] built an MC-AB-IPFE with bounded vectors, under the same security model and assumption. More recently, Agrawal, Tomida and Yadav [12] developed an MC-FE for AWS functionality with unbounded slots relying on the same matrix DDH assumption. However, while their scheme allows encrypting unbounded-length messages, the size of each slot, and thus the functions operating on them, remains bounded. Our approach differs since both the function and message vectors in our MC-AB-UIPFE are unbounded. Furthermore, their MC-FE is not attribute-based, whereas we construct MC-FE for an unbounded attribute-based inner product functionality. A detailed comparison is provided in Section 2.

Along the way, we design the first single-input AB-UIPFE where $n = 1$ (and $L_k = \epsilon$) under the same assumption, proven secure in the standard model. Our single-input AB-UIPFE is a subclass of attribute-based unbounded quadratic FE recently built by Tomida [46] where the access control is provided by arithmetic branching programs (ABP). However, our construction is simpler and more direct than [46], since we only deal with linear function and LSSS access policies on top of it. Additionally, Tomida's scheme relies on the random oracle model (ROM), whereas ours operates in the standard model. Previously, Datta and Pal [23] constructed a multi-authority AB-UIPFE which essentially implies an AB-UIPFE, but their unboundedness follows a strict model, while ours adopts the more flexible and natural permissive setting.

Multi-Input AB-UIPFE. We construct the first MI-AB-UIPFE in the public key setting by extending our MC-AB-UIPFE. Note that an MI-FE scheme in the public key setting with corruption is much more challenging to construct than an MC-FE scheme due to the absence of labels during decryption. An adversary can decrypt any combination of ciphertexts in the multi-input setting whereas, in the multi-client setting, decryption is guaranteed only when all the ciphertexts are computed under the same label or timestamp. In

our MI-AB-UIPFE, the decryption reveals:

$$f((\mathbb{A}, (\mathbf{y}_k, I_{\mathbf{y}_k})_{k\in[n]}), (\mathsf{S}_k, \mathbf{x}_k, \ell_k)_{k\in[n]}) = \begin{cases} \sum_{k\in[n]} \sum_{i\in I_{\mathbf{y}_k}} x_{k,i} y_{k,i} & \text{if } (I_{\mathbf{y}_k} \subseteq [\ell_k]) \wedge (\mathbb{A}(\mathsf{S}_k) = 1),\ \forall k \in [n] \\ \perp & \text{otherwise} \end{cases}$$

As mentioned above, our AB-UIPFE functionality differs from the attribute-based AWS functionality (with unbounded slots) [12]. More importantly, the MI-FE scheme of [12] cannot capture the permissive unboundedness even if it supports encrypting unbounded length vectors at each input.

Dynamic Decentralized UIPFE. In the literature, dynamic decentralized FE has been constructed only for the bounded class of FE schemes such as inner products [19], AWS [12]. In a DD-FE, there is no authority, clients can dynamically join the system, selecting a set of users $\mathcal{U}_k$ during encryption or key generation whose inputs can be combined during decryption. However, despite this dynamic feature, all clients must agree on a fixed input length, which limits flexibility. We argue that a truly dynamic system should allow clients to choose their own input sizes during key generation and encryption. Our notion of dynamic decentralized unbounded FE (DD-UFE) extends the dynamic nature of conventional DD-FE by removing this limitation. We construct DD-UFE for linear functions (DD-UIPFE) where the secret keys are generated for the tuple $(\mathbf{y}_k = (y_{k,i})_{i\in I_{\mathbf{y}_k}})_{k\in\mathcal{U}_{k,\mathsf{key}}}$, the ciphertexts are computed for the tuple $(L_k, \mathbf{x}_k \in \mathbb{Z}_p^{\ell_k}, \mathcal{U}_{k,\mathsf{msg}})$, and decryption outputs:

$$f(((\mathbf{y}_k, I_{\mathbf{y}_k})_{k\in\mathcal{U}_{k,\mathsf{key}}}), (L_k, \mathbf{x}_k, \ell_k)_{k\in\mathcal{U}_{k,\mathsf{msg}}}) = \begin{cases} \sum_{k\in\mathcal{U}} \sum_{i\in I_{\mathbf{y}_k}} x_{k,i} y_{k,i} & \text{if } (\mathcal{U} = \mathcal{U}_{k,\mathsf{key}} = \mathcal{U}_{k,\mathsf{msg}}) \wedge ((I_{\mathbf{y}_k} \subseteq [\ell_k]) \wedge (L_k = L)\ \forall k \in \mathcal{U}) \\ \perp & \text{otherwise} \end{cases}$$

We build our DD-UIPFE using the blueprint of previous works [19, 13].

Our primary technical contributions involve the design of an extended slotted unbounded IPFE scheme, which utilizes the *extended* functionality to integrate an attribute-based access control layer while leveraging the *unbounded slotted* feature to facilitate unboundedness and multi-input extensions. A comparative analysis with existing works is presented in Table 1, and further technical details can be found in Section 2.

Applications of multi-input (attribute-based) UIPFE. Consider a scenario where a research institute aims to optimize disease diagnosis by utilizing data from multiple medical centers. The *k-th* center contributes patient data in the form of an unbounded-length input vector $\mathbf{x}_k$ representing various medical measurements, e.g., blood pressure, body temperature, red blood cell count, collected from patients treated at that particular center. Since the number of patients at each center may vary over time, the length of the input vector $\mathbf{x}_k$ is unbounded, reflecting real-world unpredictability. Additionally, each medical center is associated with a weight vector $\mathbf{y}_k$, which could represent the importance of certain measurements or the confidence level assigned to the data, depending on the center's practices.

Suppose there are n such medical centers. In that case, the goal is to compute the sum defined as $\sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k \rangle = \sum_{k\in[n], i\in I_{\mathbf{y}_k}} x_{k,i} y_{k,i}$, where both $\mathbf{x}_k, \mathbf{y}_k$ are of unbounded length, allowing for dynamic, varying-length data inputs for each center. For instance, if $\mathbf{x}_k$ includes blood pressure measurements in odd indices and body temperature readings in even indices, researchers can perform targeted computations such as the average blood pressure by selecting the relevant elements of the vectors $\mathbf{y}_k$. Let's assume three medical centers contribute data as follows: the first center treats three patients, the second four, and the third five. Their input vectors are $\mathbf{x}_1 = (x_{11}, x_{12}, x_{13}), \mathbf{x}_2 = (x_{21}, x_{22}, x_{23}, x_{24})$, and $\mathbf{x}_3 = (x_{31}, x_{32}, x_{33}, x_{34}, x_{35})$, respectively. If the researcher aims to compute the average of blood pressure measurements (stored in the odd indices), they would require a secret key for the function $\big(\mathbf{y}_1 = (y_{11}, y_{13}), \mathbf{y}_2 = (y_{21}, y_{23}), \mathbf{y}_3 = (y_{31}, y_{33}, y_{35})\big)$. This would allow the desired permissive inner product computation $\sum_k \sum_{i=\text{odd}} x_{ki} y_{ki}$. Moreover, if the number of medical centers is not fixed, *i.e.*, n is also unbounded then DD-UIPFE can be employed in

such a scenario. This example illustrates how MC-UIPFE or DD-UIPFE enables a robust computation of aggregation statistics and aligns with realistic applications.

Additionally, for more fine-grained access control, each input $\mathbf{x}_k$ could be associated with an attribute S_k, and data from a medical center would only be used if it satisfies a specific policy, such as $\mathbb{A}(\mathsf{S}_k) = 1$. For example, the computation might focus on blood pressure measurements for patients treated in the *k-th* medical center situated in location/state X_k, where the policy $\mathbb{A}(X_k) = 1$ filters the relevant data. Our MI-AB-UIPFE or MC-AB-UIPFE allow computing such *dynamic* aggregates on private data, something beyond the scope of existing (bounded input) MI(MC)-FEs. More generally, our MP-UFEs support a wide range of real-world applications, from healthcare to other domains such as financial data aggregation or electricity consumption analysis. This flexibility makes it ideal for environments where data inputs and policies are diverse and continuously evolving.

2 Technical Overview

Recap: MC-AB-IPFE of [38]. We begin with a concise overview of the multi-client FE of [38], referred to as NPP. They provided a construction of MC-AB-IPFE in the key-policy setting using *dual pairing vector spaces* (DPVS), a rich mathematical framework introduced by Okamoto and Takashima [43]. At the core of the MC-AB-IPFE of NPP, there is a single client version: the setup algorithm defines a bound n on the vector sizes and generates master keys accordingly. The access control part is enforced via the LSSS [15] policies. During key generation, the authority embeds an access structure $\mathbb{A}$ and a vector $\mathbf{y} \in \mathbb{Z}_p^n$ into the key. The encryption process encodes a vector $\mathbf{x} \in \mathbb{Z}_p^n$ under a set of attribute S to compute a ciphertext. If attributes in S satisfy $\mathbb{A}$, decryption reveals the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$, otherwise nothing is learned about $\mathbf{x}$.

In the multi-client setting, it is assumed that the number of clients is equal to the length of the vectors. More specifically, the setup algorithm parses the master secret key of the single client scheme into n pieces to create clients' encryption keys EK_i. It connects the keys using an n-out-of-n secret sharing of a specific component of the underlying bases of DPVS. The key generation works as before. Each client holds a single entry of the vector $\mathbf{x}$ and encrypts it using EK_i under the same attribute set S and a label L. When all the n ciphertexts are computed with the same label L, the secret key holder can decrypt them together to $\langle \mathbf{x}, \mathbf{y} \rangle$, given that the attributes in S satisfy the access structure. They achieve adaptive indistinguishability-based security under the SXDH assumption in the ROM. Although NPP demonstrates a blueprint of how to integrate an access control mechanism to a DPVS-style IPFE scheme and upgrade the single-input version into a multi-client one, there are a few challenges to face while supporting unbounded length vectors.

Challenges in NPP. Let us now perceive the high-level obstacles that one must overcome in NPP to support unbounded vector lengths:

- The fact that the clients can encrypt only a single entry of the vector is against the property of encrypting an arbitrary length vector in each encryption. A trivial way out is to run the same encryption algorithm for all the entries of the arbitrary length vector. However, it is clear that such an approach would rather fail because an adversary can easily combine ciphertext components of different vectors to create a valid ciphertext for an unwanted vector.
- The procedure of connecting the clients' encryption keys using an n-out-of-n secret sharing strategy would not work in our setting since the number of clients must not be the same as the lengths of vectors encrypted by the clients in our setting. Looking ahead, in fact, the number of clients is also not pre-decided in our DD-UIPFE scheme.

2.1 Integrating Unboundedness

Recall that in an MC-AB-UIPFE, n clients can choose arbitrary lengths of vectors during encryption. Let us assume that the client k selects a vector $\mathbf{x}_k = (x_{k,i})_{i \in [\ell_k]}$ along with an attribute set S_k and computes a ciphertext CT_k. The secret key SK is generated by the authority for a function $\mathbf{y} = (\mathbf{y}_k)_{k \in [n]}$ with associated

index sets $\{I_k\}_{k\in[n]}$ and an access structure $\mathbb{A}$. Given that S_k satisfies $\mathbb{A}$ and $I_k \subseteq [\ell_k]$ for all $k \in [n]$, the decryption recovers the sum of inner product values $\sum_{k\in[n]}\langle \mathbf{x}_k, \mathbf{y}_k\rangle$ if all the ciphertexts are computed under the same label.

Even without access control, constructing an MC-UIPFE remains challenging. A natural approach might involve adapting the transformation by Abdalla et al. [1], which extends a single input IPFE to a multi-client IPFE. One might attempt to replace the underlying single input IPFE using an existing UIPFE [47] to achieve an MC-UIPFE. However, this approach fails since the encryption algorithm requires the vector lengths to be known in advance. Specifically, each vector $\mathbf{x}_k$ is embedded into a larger vector $\widetilde{\mathbf{x}}_k$ of length $n\ell$ (where ℓ is the vector length) and masked using $\mathbf{t}_k$, a (n-out-of-n) secret share of $\mathbf{0}_{n\ell}$, before applying IPFE encryption. Therefore, this transformation cannot accommodate dynamic vector lengths, a key feature of MC-UIPFE. Furthermore, achieving *permissiveness*—where decryption succeeds only if $I_k \subseteq [\ell_k]$ for each $k \in [n]$—is unclear in this setting. Incorporating access control further complicates the process. Previously, non-generic construction of MC-IPFE [18] generates clients' encryption keys depending on the vector length fixed at the time of setup.

Comparison with MC-FE of [12]. A recent work by Agrawal et al. [12], henceforth ATY, builds an MC-FE scheme for AWS with unbounded slots (FE-AWS) from pairings, originally introduced by Abdalla et al. [7]. FE-AWS generalizes IPFE by allowing an encryptor to encode $\{\mathbf{x}_j, \mathbf{z}_j\}_{j\in[N]}$ where N is unbounded, $\mathbf{x}_j$ and $\mathbf{z}_j$s are called pubic and private attributes respectively, the key is generated for a function f which is usually an ABP, and decryption recovers $\sum_{j\in[N]}\langle f(\mathbf{x}_j), \mathbf{z}_j\rangle$. In an MC-FE-AWS, each client encrypts an *unbounded-slot* input $\{\mathbf{x}_{k,j}, \mathbf{z}_{k,j}\}_{j\in[N_k]}$, where N_k is unbounded, and decryption recovers $\sum_k \sum_{j\in[N_k]}\langle f_k(\mathbf{x}_{k,j}), \mathbf{z}_{k,j}\rangle$. The term "unbounded-slot" in FE-AWS is quite *different* from our definition of *unboundedness* in MC-AB-UIPFE. While FE-AWS allows the encryptor to choose an unbounded number of vectors, both vector sizes and function classes are fixed during setup, meaning the encryption key depends on these sizes. In contrast, MC-AB-UIPFE allows unbounded message and function vectors, with successful decryption requiring a permissive relation between index sets. Although MC-AB-UIPFE might seem reducible to MC-FE-AWS by encoding unbounded vectors into AWS slots, verifying the permissive relation between index sets complicates this approach, making it inapplicable for constructing MC-AB-UIPFE or DD-UIPFE directly.

Our Approach. Instead of integrating the unboundedness property to existing MC-IPFE or MC-AB-IPFE, we investigate whether it is possible to upgrade available single input UIPFE [26, 47, 24] or AB-UIPFE [23] into the multi-client setting. Along this direction, we use the blueprint of NPP to construct a single-input FE first, and then upgrade it to the multi-client setting. To build a suitable single-input FE (without access control) that can later be enriched with an attribute-based access control extension, we follow the idea of ATY that builds an extended FE-AWS equipped with an additional inner product. Furthermore, for integrating the unboundedness feature into this framework, we observe that existing works [47, 24] used function-hiding security of the underlying IPFE to realize the permissive case of unboundedness which we are aiming for. The function-hiding security of IPFE has been independently exploited for achieving the permissive case of unboundedness [47, 24] and multi-client realization of a certain class of FEs [12]. The former works utilize the function-hiding security to carry out an index-encoding methodology for realizing the unboundedness property whereas the latter uses it for converting a single-input scheme into a multi-input scheme without relying on the ciphertext homomorphism property desired in [20]. In this work, we develop a methodology that demonstrates how function-hiding security of IPFE can be compelled to obtain both of these properties together for the function class of AB-IP.

Constructing UIPFE of [47] using Slotted IPFE. Our starting point is the UIPFE construction of Tomida and Takashima [47] which we call TT. Although UIPFE of TT is a direct construction based on pairing, we can view it as a generic construction based on slotted IPFE (sIPFE) [36, 24] which is a hybrid of a public key IPFE and a secret key function-hiding IPFE. A vector $\mathbf{x} \in \mathbb{Z}_p^n$ in sIPFE is divided into parts $(\mathbf{x}_{\mathsf{pub}}, \mathbf{x}_{\mathsf{priv}}) \in \mathbb{Z}_p^{n_{\mathsf{pub}}} \times \mathbb{Z}_p^{n_{\mathsf{priv}}}$ such that $n = n_{\mathsf{pub}} + n_{\mathsf{priv}}$. While one can encrypt the public part $\mathbf{x}_{\mathsf{pub}}$ using

the public key through a slotted encryption algorithm, encrypting the whole vector $\mathbf{x} = (\mathbf{x}_{\mathsf{pub}}, \mathbf{x}_{\mathsf{priv}})$ requires the knowledge of the master secret key, just like a secret key IPFE, which is done by a normal encryption algorithm. The ciphertexts obtained by the slotted and normal modes are indistinguishable when $\mathbf{x}_{\mathsf{priv}}$ is set to $\mathbf{0}_{n_{\mathsf{priv}}}$. Hence, we can only hide the private part of the function vector $\mathbf{y}_{\mathsf{priv}}$ while $\mathbf{y}_{\mathsf{pub}}$ remains public in the secret key generated for $\mathbf{y} = (\mathbf{y}_{\mathsf{pub}}, \mathbf{y}_{\mathsf{priv}})$.

The sIPFE of [36] is built using asymmetric prime-order pairing groups with the pairing operation $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We denote $[\![a]\!]_i$ by an element g_i^a in the group $\mathbb{G}_i$ and $[\![(a_1, \ldots, a_n)]\!]_i$ by a vector of group elements $(g_i^{a_1}, \ldots, g_i^{a_n})$ for $i \in \{1, 2, T\}$. Let $\mathsf{sIPFE} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ be a sIPFE scheme supporting vectors of length $4 = n_{\mathsf{pub}}$. We ignore the private slots, which can be added later as those are required only for security analysis.

Construction 1 (UIPFE from sIPFE) The setup of UIPFE generates $(\mathsf{iMPK}, \mathsf{iMSK})$ using iSetup. A secret key for a vector $\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}}$, and a ciphertext for a vector $\mathbf{x} = (x_i)_{i \in [\ell]}$ are computed as

$$\begin{aligned} \mathsf{SK} : &\quad \{\ \mathsf{iSK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(i\sigma_i, \sigma_i, y_i, r_i)]\!]_2)\ \}_{i \in I_{\mathbf{y}}} \\ \mathsf{CT} : &\quad \{\ \mathsf{iCT}_i \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![(\pi_i, -i\pi_i, x_i, \alpha)]\!]_1)\ \}_{i \in [\ell]} \end{aligned}$$

where $r_i \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}}} r_i = 0$. Note that σ_i, π_i are chosen uniformly at random from $\mathbb{Z}_p$ to encode the indices such that the inner product between the encodings vanishes only when they have matching indices. This index encoding methodology was first introduced by Okamoto and Takashima [42] in the context of achieving unbounded inner product encryption. If the unboundedness is permissive, i.e. $I_{\mathbf{y}} \subseteq [\ell]$, then decryption works by computing $\sum_{i \in I_{\mathbf{y}}} \mathsf{iDec}(\mathsf{iSK}_i, \mathsf{iCT}_i) = [\![\sum_{i \in I_{\mathbf{y}}} x_i y_i]\!]_T$ as $\sum_{i \in I_{\mathbf{y}}} r_i = 0$.

We use the DDH assumption and the function-hiding security of sIPFE. Since we consider selective security, the length of the challenge vector is known in advance. Let us denote $\mathsf{sk}(y_i) = (i\sigma_i, \sigma_i, y_i, r_i)$ and $\mathsf{ct}(x_i) = (\pi_i, -i\pi_i, x_i, \alpha)$. In the following, we will add private slots, indicated by dashed underline, to $\mathsf{sk}(y_i)$ and $\mathsf{ct}(x_i)$ when needed for proving the security discussed in three steps:

1. *duplicating secret shares.* The secret shares $\{r_i\}_{i \in I_{\mathbf{y}}}$ are copied to a private slot of $\mathsf{sk}(y_i)$, i.e.,

$$\mathsf{sk}(y_i) \leftarrow (i\sigma_i, \sigma_i, y_i, r_i, \underline{\widetilde{r}_i}), \quad \mathsf{ct}(x_i^{(0)}) \leftarrow (\pi_i, -i\pi_i, x_i^{(0)}, \alpha, \underline{\widetilde{\alpha}})$$

   with $\sum_{i \in I_{\mathbf{y}}} \widetilde{r}_i = 0$. This hybrid is indistinguishable from the original game by the DDH assumption and the function-hiding security of sIPFE.

2. *handling non-permissive keys.* We call the secret keys with $I_{\mathbf{y}} \not\subset [\ell]$ as non-permissive keys. For such keys, the duplicated secret shares $\{\widetilde{r}_j\}_{j \in I_{\mathbf{y}} \setminus [\ell]}$ are chosen uniformly at random. We observe that the inner product between the index encoding parts, i.e. $\langle (\pi_i, -i\pi_i), (j\sigma_j, \sigma_j) \rangle$ is non-zero as $i \neq j$. This produces an extra entropy, sufficient to change the secret shares to random values using the function-hiding security of sIPFE.

3. *statistical shift.* In the final step, the secret keys and the ciphertext are changed to a special form as:

$$\mathsf{sk}(y_i) \leftarrow (i\sigma_i, \sigma_i, y_i, r_i, \underline{\widetilde{r}_i - \xi_i y_i}),$$
$$\mathsf{ct}(x_i^{(0)}) \leftarrow (\pi_i, -i\pi_i, x_i^{(0)} + \xi_i \widetilde{\alpha}, \alpha, \underline{\widetilde{\alpha}})$$

   where ξ_i is sampled uniformly at random from $\mathbb{Z}_p$. The indistinguishability follows from the function-hiding security of sIPFE since the inner product between $\mathsf{sk}(y_i)$ and $\mathsf{ct}(x_i^{(0)})$ remains unchanged. Now, we apply a statistical transformation through ξ_i by shifting it as $\xi_i \leftarrow \xi_i + (x_i^{(1)} - x_i^{(0)})/\widetilde{\alpha}$. It does not change the distribution of the secret shares $\{\widetilde{r}_i - \xi_i y_i\}$ due to the admissibility condition that $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$ for all secret keys with $I_{\mathbf{y}} \subseteq [\ell]$. On the other hand, it changes $\mathsf{ct}(x_i^{(0)})$ to $\mathsf{ct}(x_i^{(1)})$.

Adding Access Control to UIPFE. As the next step towards our goal, we aim to integrate attribute-based access control into the UIPFE described above. To achieve this, we leverage the technique of NPP, which encodes access control within the DPVS structure using an LSSS [15]. An LSSS allows us to secret share a random element $a_0 \leftarrow \mathbb{Z}_p$ depending on an access structure $\mathbb{A}$ over an attribute space Att into several shares $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ such that only an authorized set $\{\mathsf{att}_i\}_{i \in \mathsf{S}} \subseteq \mathsf{Att}$ can recover $\mathbf{c} = (c_j)_j$ to reconstruct $a_0 = \sum_{j \in \mathsf{S}} c_j a_j$. We generically construct an AB-UIPFE by combining an sIPFE, for realizing the access control part, and a UIPFE, for the inner product computation.

Construction 2 (Candidate AB-UIPFE) Let us consider an $\mathsf{sIPFE} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ and a $\mathsf{UIPFE} = (\mathsf{uSetup}, \mathsf{uKeyGen}, \mathsf{uEnc}, \mathsf{uDec})$. The setup of AB-UIPFE samples $(\mathsf{iMPK}, \mathsf{iMSK})$, $(\mathsf{uMPK}, \mathsf{uMSK})$ by running iSetup, uSetup respectively. It sets $\mathsf{MPK} = (\mathsf{iMPK}, \mathsf{uMPK})$, $\mathsf{MSK} = (\mathsf{iMSK}, \mathsf{uMSK})$. A secret key corresponding to $(\mathbf{y} = (y_i)_{i \in I_\mathbf{y}}, \mathbb{A})$ and a ciphertext for $(\mathbf{x} \in \mathbb{Z}_p^\ell, \mathsf{S})$ are computed as follows:

$$\mathsf{SK}: \quad \left\{ \begin{array}{ll} \mathsf{iSK}_j & \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(j\sigma_j, \sigma_j, a_j \cdot z)]\!]_2) \\ \mathsf{uSK} & \leftarrow \mathsf{uKeyGen}(\mathsf{uMSK}, [\![(\mathbf{y}, a_0 \cdot z)]\!]_2) \end{array} \right\}_{j \in \mathsf{List\text{-}Att}(\mathbb{A})},$$

$$\mathsf{CT}: \quad \left\{ \begin{array}{ll} \mathsf{iCT}_j & \leftarrow \mathsf{iEnc}(\mathsf{iMPK}, [\![(\pi_j, -j\pi_j, \psi)]\!]_1) \\ \mathsf{uCT} & \leftarrow \mathsf{uEnc}(\mathsf{uMPK}, [\![(\mathbf{x}, \psi)]\!]_1) \end{array} \right\}_{j \in \mathsf{S}},$$

where $\sigma_j, a_0, z, \pi_j, \psi \leftarrow \mathbb{Z}_p$. It is easy to observe that the correctness works if the attributes associated with S satisfy the access structure $\mathbb{A}$ and $I_\mathbf{y} \subseteq [m]$. More specifically, the decryption first reconstructs $[\![a_0 z\psi]\!]_T = \prod_{j \in \mathsf{S}} c_j \cdot \mathsf{iDec}(\mathsf{iSK}_j, \mathsf{iCT}_j)$ and computes $[\![\sum_{i \in I_\mathbf{y}} x_i y_i + a_0 z\psi]\!]_T = \mathsf{uDec}(\mathsf{uSK}, \mathsf{uCT})$. Then, it extracts the inner product value $[\![\sum_{i \in I_\mathbf{y}} x_i y_i]\!]_T$. Unfortunately, the scheme is not secure. In an AB-UIPFE, the adversary is allowed to query secret keys for $(\mathbf{y} = (y_i)_{i \in I_\mathbf{y}}, \mathbb{A})$ such that either $I_\mathbf{y} \not\subseteq [m]$ (*non-permissive*) or $\mathbb{A}$ is not satisfied by the attributes associated with S (*non-accepting*). The permissive case of unboundedness can be handled by the underlying UIPFE. However, to prevent the adversary from extracting any information about the message vector using the permissive but non-accepting keys, we have to implement a masking strategy similar to [41,42,38]. In more detail, the masking term is created by first copying the secret value a_0 and the shares $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ into additional slots of the secret keys (similar to *duplicating secret shares* step) and then randomizing the shares for the case of non-accepting keys (similar to the *handling non-permissive keys* step). Although slotted IPFE could allow adding some additional slots for the purpose, the UIPFE does *not* endorse modifying slots of the vectors embedded in the keys, since it is not function-hiding. Therefore, the above construction fails to provide a secure AB-UIPFE.

Extended Functionality and Function-hiding Security. We now show how to extend the UIPFE functionality to execute the masking strategy in the above scheme. Looking ahead, we also need to devise a way to link multiple instances of the UIPFE into a single scheme for building MC-AB-UIPFE. For that, we use an *extended functionality* mechanism where the actual functionality is extended to integrate an additional randomization strategy into the system. In more detail, we need an augmented primitive that supports encrypting unbounded length vectors and, concurrently, possesses the ability to attach secret random values to the computation when required during the security analysis. Consequently, the extended functionality must have enough space for realizing a normal UIPFE whilst it must have an extended possibly bounded space working like a function-hiding IPFE. At this juncture, we define the notion of *extended unbounded* slotted IPFE (esUIPFE) which precisely enables the required extended functionality. More specifically, each vector $\mathbf{x}$ is partitioned into two parts $(\mathbf{x}_\mathsf{pub}, \mathbf{x}_\mathsf{priv})$ as in a normal slotted IPFE except that the public slot $\mathbf{x}_\mathsf{pub}$ is further partitioned into two parts $(\mathbf{x}_\mathsf{upub}, \mathbf{x}_\mathsf{bpub})$ where $\mathbf{x}_\mathsf{upub}$ is unbounded with an associated index set $I_{\mathbf{x}_\mathsf{upub}}$ and $\mathbf{x}_\mathsf{bpub}$ is bounded. A secret key is generated for a vector of the form $\mathbf{y} = (\mathbf{y}_\mathsf{upub}, \mathbf{y}_\mathsf{bpub}, \mathbf{y}_\mathsf{priv}) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^{n_1} \times \mathbb{Z}_p^{n_2}$ with an associated index set $I_{\mathbf{y}_\mathsf{upub}}$, a ciphertext is computed by encrypting a vector $\mathbf{x} = (\mathbf{x}_\mathsf{upub}, \mathbf{x}_\mathsf{bpub}, \mathbf{x}_\mathsf{priv}) \in \mathbb{Z}_p^\ell \times \mathbb{Z}_p^{n_1} \times \mathbb{Z}_p^{n_2}$ in the normal mode. The decryption reveals $\sum_{i \in I_{\mathbf{y}_\mathsf{upub}}} x_{\mathsf{upub},i} y_{\mathsf{upub},i} + \langle (\mathbf{x}_\mathsf{bpub}, \mathbf{x}_\mathsf{priv}), (\mathbf{y}_\mathsf{bpub}, \mathbf{y}_\mathsf{priv}) \rangle$ if $I_{\mathbf{y}_\mathsf{upub}} \subseteq [\ell]$. In the slotted mode of encryption, $\mathbf{x}_\mathsf{priv}$ is ignored, similar to a usual slotted IPFE and the function-hiding security holds only in the private slot.

We note that similar techniques of extending functionality have been used in previous works [7,22,46,12] in different contexts either for achieving unbounded slot AWS functionality from a single input one [7,

22], building (single-input) unbounded quadratic FE using the function-hiding security of IPFE [46] or for converting the multi-input IPFE of [21] to a multi-client FE-AWS [12]. We emphasize that Tomida [46] has also considered a notion named unbounded slotted IPFE, however, in his application, the public slot is *completely* unbounded (and is *not* extended by a bounded slot) as the motivation was to support *unbounded quadratic* computation, not to connect multiple threads of UIPFE. In this work, we demonstrate another application of the *extended functionality* for designing a multi-client unbounded FE.

We build an esUIPFE generically using a sIPFE. The construction follows the idea of utilizing sIPFE to build UIPFE as in Construction 1. Here, we use the public slots of sIPFE for extending the UIPFE functionality with an additional inner product computation.

Construction 3 (esUIPFE) The setup of esUIPFE generates $(\mathsf{iMPK}, \mathsf{iMSK})$ using iSetup. A secret key for a vector $\mathbf{y} = (\mathbf{y}_{\mathsf{upub}}, \mathbf{y}_{\mathsf{bpub}}, \mathbf{y}_{\mathsf{priv}}) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^{n_1} \times \mathbb{Z}_p^{n_2}$ with an associated index set $I_{\mathbf{y}_{\mathsf{upub}}}$, and a ciphertext for a vector $\mathbf{x} = (\mathbf{x}_{\mathsf{upub}}, \mathbf{x}_{\mathsf{bpub}}, \mathbf{x}_{\mathsf{priv}}) \in \mathbb{Z}_p^{\ell} \times \mathbb{Z}_p^{n_1} \times \mathbb{Z}_p^{n_2}$ are computed as follows:

$$\mathsf{SK}: \quad \{\ \mathsf{iSK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(i\sigma_i, \sigma_i, y_{\mathsf{upub},i}, r_i, \mathbf{y}_{\mathsf{bpub},i}, \underline{\mathbf{y}_{\mathsf{priv},i}})]\!]_2)\ \}_{i \in I_{\mathbf{y}_{\mathsf{upub}}}}$$

$$\mathsf{CT}_{\mathrm{slot}}: \quad \{\ \mathsf{iCT}_i \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![(\pi_i, -i\pi_i, x_{\mathsf{upub},i}, \alpha, \mathbf{x}_{\mathsf{bpub}})]\!]_1)\ \}_{i \in [\ell]}$$

$$\mathsf{CT}_{\mathrm{norm}}: \quad \{\ \mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMSK}, [\![(\pi_i, -i\pi_i, x_{\mathsf{upub},i}, \alpha, \mathbf{x}_{\mathsf{bpub}}, \underline{\mathbf{x}_{\mathsf{priv}}})]\!]_1)\ \}_{i \in [\ell]}$$

where $\mathbf{y}_{\mathsf{bpub}} = \sum_{i \in I_{\mathbf{y}_{\mathsf{upub}}}} \mathbf{y}_{\mathsf{bpub},i}$, $\mathbf{y}_{\mathsf{priv}} = \sum_{i \in I_{\mathbf{y}_{\mathsf{upub}}}} \mathbf{y}_{\mathsf{priv},i}$ and $\mathsf{CT}_{\mathrm{slot}}, \mathsf{CT}_{\mathrm{norm}}$ refer to the ciphertexts computed in slotted and normal modes respectively. The selective IND-security of our esUIPFE can be argued similarly to the UIPFE based on the DDH assumption. Upon replacing the UIPFE with esUIPFE in the candidate Construction 2 of AB-UIPFE, we essentially get a slotted version of AB-UIPFE (AB-sUIPFE). The normal mode of ciphertext can be ignored in the case of single-client AB-UIPFE which is a public-key primitive whereas, looking ahead, the normal mode becomes useful while upgrading it into a multi-client FE. We now describe our AB-sUIPFE that trivially captures AB-UIPFE supporting the same class of policies.

Construction 4 (AB-sUIPFE from esUIPFE) Let us consider an $\mathsf{sIPFE} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ and an $\mathsf{esUIPFE} = (\mathsf{eSetup}, \mathsf{eKeyGen}, \mathsf{eEnc}, \mathsf{eSlotEnc}, \mathsf{eDec})$ with $n_1 = 1$. The setup of AB-sUIPFE samples $(\mathsf{iMPK}, \mathsf{iMSK})$, $(\mathsf{eMPK}, \mathsf{eMSK})$ by running iSetup, eSetup respectively. It sets $\mathsf{MPK} = (\mathsf{iMPK}, \mathsf{eMPK})$, $\mathsf{MSK} = (\mathsf{iMSK}, \mathsf{eMSK})$. A secret key corresponding to $(\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}}, \mathbf{y}_{\mathsf{priv}}, \mathbb{A})$ and a ciphertext for $(\mathbf{x} \in \mathbb{Z}_p^{\ell}, \mathbf{x}_{\mathsf{priv}}, \mathsf{S})$ are computed as follows:

$$\mathsf{SK}: \quad \begin{Bmatrix} \mathsf{iSK}_j & \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(j\sigma_j, \sigma_j, a_j \cdot z)]\!]_2)\ \}_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}\ , \\ \mathsf{eSK} & \leftarrow \mathsf{eKeyGen}(\mathsf{eMSK}, [\![(\mathbf{y}, a_0 \cdot z, \underline{\mathbf{y}_{\mathsf{priv}}})]\!]_2) \end{Bmatrix}$$

$$\mathsf{CT}_{\mathrm{slot}}: \quad \begin{Bmatrix} \mathsf{iCT}_j & \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![(\pi_j, -j\pi_j, \psi)]\!]_1)\ \}_{j \in \mathsf{S}}\ , \\ \mathsf{eCT} & \leftarrow \mathsf{eSlotEnc}(\mathsf{eMPK}, [\![(\mathbf{x}, \psi)]\!]_1) \end{Bmatrix}$$

$$\mathsf{CT}_{\mathrm{norm}}: \quad \begin{Bmatrix} \mathsf{iCT}_j & \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![(\pi_j, -j\pi_j, \psi)]\!]_1)\ \}_{j \in \mathsf{S}}\ , \\ \mathsf{eCT} & \leftarrow \mathsf{eEnc}(\mathsf{eMSK}, [\![(\mathbf{x}, \psi, \underline{\mathbf{x}_{\mathsf{priv}}})]\!]_1) \end{Bmatrix}$$

Next, we only analyze the security of $\mathsf{CT}_{\mathrm{slot}}$ and refer to Sec. 5.2 for a formal and complete proof of security. The first hybrid switches to $\mathsf{eSlotEnc}$ from eEnc for encrypting the challenge message for activating the private slots. As before, let us denote $\mathsf{sk}_j(\mathbb{A}) = (j\sigma_j, \sigma_j, a_j \cdot z), \mathsf{sk}(\mathbf{y}) = (\mathbf{y}, a_0 \cdot z)$ and $\mathsf{ct}_j(\mathsf{S}) = (\pi_j, -j\pi_j, \psi), \mathsf{ct}(\mathbf{x}^{(0)}) = (\mathbf{x}^{(0)}, \psi)$. The proof proceeds with the following steps.

1. *shifting secret shares to private slots.* The secret shares $\{a_j\}_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ are shifted to a *private* slot of the vectors. The modified vectors will be:

$$\mathsf{sk}_j(\mathbb{A}) \leftarrow (j\sigma_j, \sigma_j, a_j \cdot z, \underline{a_j \cdot z}), \qquad \mathsf{sk}(\mathbf{y}) \leftarrow (\mathbf{y}, a_0 \cdot z, \underline{a_0 \cdot z})\ ,$$

$$\mathsf{ct}_j(\mathsf{S}) \leftarrow (\pi_j, -j\pi_j, 0, \underline{\psi}), \qquad \mathsf{ct}(\mathbf{x}^{(0)}) \leftarrow (\mathbf{x}^{(0)}, 0, \underline{\psi})\ .$$

The indistinguishability is guaranteed by the function-hiding security of sIPFE and $\mathsf{esUIPFE}$.

2. *adding masking shares.* We call the secret keys for which the associated policy is not satisfied by the attributes of the challenge ciphertext, *i.e.,* $\mathbb{A}(\mathsf{S}) = 0$, *non-accepting* keys. Due to the presence of private slots in both the IPFEs, we can apply a masking strategy, adapted from [41,42,38], to disable the decryption ability of the non-accepting keys in the next step, even when the keys satisfy the permissive case of the unboundedness. In particular, we change the vectors to

$$\mathsf{sk}_j(\mathbb{A}) \leftarrow (j\sigma_j, \sigma_j, a_j \cdot z, \underline{\ a_j \cdot z,\ a'_j\delta \cdot z/v\ }), \qquad \mathsf{sk}(\mathbf{y}) \leftarrow (\mathbf{y}, a_0 \cdot z, \underline{\ a_0 \cdot z,\ a'_0\delta \cdot z\ }),$$
$$\mathsf{ct}_j(\mathsf{S}) \leftarrow (\pi_j, -j\pi_j, 0, \underline{\ \psi,\ \psi' \cdot v\ }), \qquad \mathsf{ct}(\mathbf{x}^{(0)}) \leftarrow (\mathbf{x}^{(0)}, 0, \underline{\ \psi,\ \psi'\ })$$

where $\delta = \langle \mathbf{x}^{(0)} - \mathbf{x}^{(1)}, \mathbf{y} \rangle$.

3. *handling non-accepting keys.* In this step, the masking secret value a'_0 is replaced with a uniformly chosen value r_0 (uncorrelated with a'_j) only for the keys non-accepting keys. Note that, here we use the advantage of proving security against a selective adversary that is restricted to sending all the key queries before getting the challenge ciphertext. This step is information-theoretic because of the fact that the reconstruction of the secret value of LSSS is possible *only if* the attributes satisfy the access structure. Let us write $\mathsf{sk}(\mathbf{y}) = (\mathbf{y}, a_0 \cdot z, \underline{\ a_0 \cdot z,\ \widetilde{a}\delta \cdot z\ })$ where $\widetilde{a}$ is equal to a'_0 for accepting keys and r_0 for non-accepting keys.

4. *statistical shift.* In the final step, a statistical shifting is performed to the random element r_0 as $r'_0 \leftarrow r_0 + 1/z\psi'$ for all non-accepting keys. It allows us to change $\mathsf{ct}(\mathbf{x}^{(0)})$ into an encoding of $\mathbf{x}^{(1)}$ as $\mathsf{ct}(\mathbf{x}^{(1)}) \leftarrow (\mathbf{x}^{(1)}, 0, \underline{\ \psi,\ \psi'\ })$. The indistinguishability follows from the function-hiding security of esUIPFE since

$$\begin{aligned}\langle \mathbf{x}^{(1)}, \mathbf{y} \rangle + \psi a_0 \cdot z + r'_0\delta\psi' \cdot z &= \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle + \psi a_0 \cdot z + r_0\delta\psi' \cdot z + \delta \\ &= \langle \mathbf{x}^{(0)}, \mathbf{y} \rangle + \psi a_0 \cdot z + r_0\delta\psi' \cdot z\end{aligned}$$

holds for all non-accepting keys. On the other hand, for the accepting keys the value of $\widetilde{a}$ remains a'_0 and hence $\langle \mathsf{ct}(\mathbf{x}^{(0)}), \mathsf{sk}(\mathbf{y}) \rangle = \langle \mathsf{ct}(\mathbf{x}^{(1)}), \mathsf{sk}(\mathbf{y}) \rangle$ holds for such keys due to the admissibility condition.

Adding Multiple Clients into the System. As we achieved a single-input AB-UIPFE, the next step would be to add multiple clients into the system by connecting several encryption algorithms run by the clients. Recall that in MC-AB-UIPFE, the setup samples a master secret key MSK and encryption keys $\{\mathsf{EK}_k\}_{k\in[n]}$ for the n clients, each client computes a ciphertext CT_k by encrypting a message $(\mathbf{x}_k \in \mathbb{Z}_p^{\ell_k}, \mathsf{S}_k)$ under a label L, the authority generates a key SK corresponding to a function $\mathbf{y} = (\mathbf{y}_k)_{k\in[n]}$ with an associated index set $I_{\mathbf{y}_k}$ and an access structure $\mathbb{A}$. Now, decrypting all clients' ciphertexts together using the key SK, the following conditions must hold:

(*i*) The index sets $\{I_{\mathbf{y}_k}\}_{k\in[n]}$ must satisfy the client-wise permissiveness condition, i.e. $I_{\mathbf{y}_k} \subseteq [\ell_k]$ for all $k \in [n]$.

(*ii*) All clients' attributes $\{\mathsf{S}_k\}_{k\in[n]}$ must satisfy the access structure $\mathbb{A}$, i.e. $\mathbb{A}(\mathsf{S}_k) = 1$ for all $k \in [n]$.

(*iii*) All the ciphertexts CT_k must be encrypted under the same label L.

To construct MC-AB-UIPFE, we follow the template of ATY which uses an extended FE to connect multiple ciphertexts of the FE for AWS functionality. Although their MC-FE achieves stronger security allowing multiple use of labels, it does not provide any access control. In contrast, we develop an MC-FE scheme for the AB-UIP functionality that provides access control and supports unbounded data, function sizes, but it achieves a weaker one-time label security [17,38,1,40]. Concretely, the extended FE of ATY is replaced with our AB-sUIPFE. We describe our MC-AB-UIPFE as follows.

Construction 5 (MC-AB-UIPFE from AB-sUIPFE) Let us consider an $\mathsf{AB\text{-}sUIPFE} = (\mathsf{aSetup}, \mathsf{aKeyGen}, \mathsf{aSlotEnc}, \mathsf{aDec})$ and a pseudorandom function $\mathsf{PRF} : \{0,1\}^* \to \mathbb{Z}_p^m$ with key space $\mathcal{K}$. The setup samples $(\mathsf{aMPK}_k, \mathsf{eMSK}_k)$ by running aSetup and $\mathsf{seed}_{k,k'} \leftarrow \mathcal{K}$ such that $\mathsf{seed}_{k,k'} = \mathsf{seed}_{k',k}$ for $k, k' \in [n]$. It sets $\mathsf{EK}_k = (\mathsf{aMSK}_k, \{\mathsf{seed}_{k,k'}\}_{k \neq k'})$. A secret key corresponding to a function $(\{\mathbf{y}_k\}_{k\in[n]}, I_{\mathbf{y}_k}, \mathbb{A})$ and a ciphertext for $(\mathbf{x}_k \in \mathbb{Z}_p^{\ell_k}, L, \mathsf{S}_k)$ are computed as:

$$\mathsf{SK}: \quad \{ \quad \mathsf{aSK}_k \quad \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\ \mathbf{y}_k,\ \underline{\boldsymbol{\alpha}, 0}\ )]\!]_2, \mathbb{A})\ \}_{k\in[n]}$$

$$\mathsf{CT}_k: \quad \mathsf{aCT}_k \quad \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\ \mathbf{x}_k,\ \underline{\mathbf{s}_k, 0}\ )]\!]_1, \mathsf{S}_k)$$

where $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_p$ and $\mathbf{s}_k = \sum_{k'\neq k}(-1)^{k'<k}\mathsf{PRF}^{\mathsf{seed}_{k',k}}(L)$. If $I_{\mathbf{y}_k} \subseteq [\ell_k]$ and $\mathbb{A}(\mathsf{S}_k) = 1$ for all $k \in [n]$ then $\mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k)$ returns $[\![v_k]\!]_T = [\![\sum_{i\in I_{\mathbf{y}_k}} x_{k,i}y_{k,i} + \langle \boldsymbol{\alpha}, \mathbf{s}_k\rangle]\!]_T$ and, finally we recover $\prod_{k\in[n]}[\![v_k]\!]_T = [\![\sum_{k\in[n], i\in I_{\mathbf{y}_k}} x_{k,i}y_{k,i}]\!]_T$ since by definition $\sum_{k\in[n]}\langle \boldsymbol{\alpha}, \mathbf{s}_k\rangle = 0$.

We observe that the clients' ciphertexts are connected by a n-out-of-n secret sharing of 0 obtained by PRF keys $\mathsf{seed}_{k,k'}$. We must put the shares into private slots as it would enable us to modify the structure of the shares for the honest clients while keeping their distribution intact using the function-hiding security of $\mathsf{AB\text{-}sUIPFE}$. Let us write $\mathsf{sk}(\mathbf{y}_k) = (\mathbf{y}_k,\ \underline{\boldsymbol{\alpha}, 0})$ and $\mathsf{ct}(\mathbf{x}_k) = (\mathbf{x}_k,\ \underline{\mathbf{s}_k, 0})$ and, for simplicity of this overview, assume that all the clients are honest. Then, in the original game, the adversary gets secret keys and ciphertexts corresponding to the vectors $\mathsf{sk}(\mathbf{y}_k), \mathsf{ct}(\mathbf{x}_k^{(0)})$. In the next hybrid, depending on the security of the employed PRF, we change $\mathbf{s}_k$ to a uniformly random $\widetilde{\mathbf{s}}_k$ such that $\sum_{k\in[n]}\widetilde{\mathbf{s}}_k = 0$. Then, we change the vectors to $\mathsf{sk}(\mathbf{y}_k) = (\mathbf{y}_k,\ \underline{\boldsymbol{\alpha}, \langle \boldsymbol{\alpha}, \widetilde{\mathbf{s}}_k\rangle + \delta_k})$ and $\mathsf{ct}(\mathbf{x}_k^{(1)}) = (\mathbf{x}_k^{(1)},\ \underline{\mathbf{0}, 1})$ where $\delta_k = \langle \mathbf{x}_k^{(0)} - \mathbf{x}_k^{(1)}, \mathbf{y}_k\rangle$. The indistinguishability follows from function-hiding security of AB-sUIPFE. Next, we replace $\langle \boldsymbol{\alpha}, \widetilde{\mathbf{s}}_k\rangle$ with a uniformly random value r_k using the MDDH assumption over $\mathbb{G}_2$. This implies that r_k can absorb δ_k and, hence, we can go back to the original vectors $\mathsf{sk}(\mathbf{y}_k) = (\mathbf{y}_k,\ \underline{\boldsymbol{\alpha}, 0})$ and $\mathsf{ct}(\mathbf{x}_k^{(1)}) = (\mathbf{x}_k^{(1)},\ \underline{\mathbf{s}_k, 0})$. This concludes the proof.

From Multi-Client to Multi-Input. We now show how to convert our MC-AB-UIPFE into an MI-AB-UIPFE in the *public-key* setting with corruption. At first glance, it seems that if we fix the label of MC-AB-UIPFE to a unique value, say $H(L) = [\![1]\!]_1$ for all L, then it gives us an MI-AB-UIPFE. Unfortunately, this is not the case since in MI-AB-UIPFE the adversary can use a secret key to decrypt any combination of ciphertexts from different slots. Let us consider a toy example of a two-input MI-AB-UIPFE, where we have two ciphertexts $\mathsf{CT}_1^1, \mathsf{CT}_1^2$ at the first slot encrypting $(\mathsf{S}_1^1, \mathbf{x}_1^1)$ and $(\mathsf{S}_1^2, \mathbf{x}_1^2)$ respectively, a single ciphertext CT_2 at the second slot encrypting $(\mathsf{S}_2, \mathbf{x}_2)$. Now, in our MC-AB-UIPFE, given a secret key SK for an access structure $\mathbb{A}$ such that $\mathbb{A}(\mathsf{S}_1^1) = \mathbb{A}(\mathsf{S}_1^2) = 1, \mathbb{A}(\mathsf{S}_2) = 0$ and a vector $(\mathbf{y}_1, \mathbf{y}_2)$, the adversary can recover $[\![\sum_{i\in I_{\mathbf{y}_1}} x_{1,i}^1 y_{1,i} + s_1]\!]_T$ and $[\![\sum_{i\in I_{\mathbf{y}_1}} x_{1,i}^2 y_{1,i} + s_1]\!]_T$, and eventually $[\![\sum_{i\in I_{\mathbf{y}_1}} (x_{1,i}^1 - x_{1,i}^2) y_{1,i}]\!]_T$. This leakage is not permitted in the multi-input FE with the standard security notion [6, 21, 12]. However, such a leakage is inevitable if we would have $\mathbb{A}(\mathsf{S}_2) = 0$. Since in that case, the adversary can recover the same value by decrypting $(\mathsf{CT}_1^1, \mathsf{CT}_2)$ and $(\mathsf{CT}_1^2, \mathsf{CT}_2)$ with the same secret key SK and then subtracting the results. Therefore, the leakage occurs if the adversary is allowed to query *only illegitimate* secret keys, which cannot decrypt any combination of ciphertexts. We say a secret key $(\mathbb{A}, (\mathbf{y}_1, \ldots, \mathbf{y}_n))$ is *illegitimate* if the adversary does not have a ciphertext for S_i at slot i such that $\mathbb{A}(\mathsf{S}_i) = 1$. In other words, the MI-AB-UIPFE obtained from MC-AB-UIPFE is secure against only legitimate keys which can decrypt any combination of ciphertexts that the adversary has. To achieve security against any keys, it is necessary to restrict the adversary in getting the partial values $[\![\sum_{i\in I_{\mathbf{y}_k}} x_{k,i}y_{k,i} + s_k]\!]_T$, for each $k \in [n]$, *only* when it has a legitimate key. For this, we use the blueprint of [12] where they additionally utilize a ciphertext-policy ABE along with an n-out-of-n secret sharing to convert their MI-AB-FE with security against legitimate keys to an MI-AB-FE secure against any keys. In our setting, we employ the CP-ABE of [37] capturing the predicates realizable by (monotone) span programs based on the MDDH assumption. The core idea of the transformation is that each client will receive an additional master secret key of the CP-ABE. At the time of key generation, the

secret key of MI-AB-UIPFE (secure against the legitimate keys) is first secret-shared using an n-out-of-n secret sharing and then each share is encrypted using the CP-ABE under the same access structure. The secret key consists of all these CP-ABE ciphertexts. The ciphertext of each client additionally contains a secret key of the CP-ABE computed for their attribute sets. Therefore, reconstruction of the secret key of the underlying MI-AB-UIPFE (secure against the legitimate keys) from the n shares requires that the access structure of the key must be satisfied by all the attribute sets present in the ciphertext combination.

The above transformation only achieves an MI-AB-UIPFE in the secret key setting, i.e. security without corruption. As also mentioned in [12], the limitation arises from the fact that there exist access structures that never evaluate to 1 (say, non-accepting access structure) and for the transformation to work in the corruption model the underlying CP-ABE must satisfy the property that the adversary should not be able to decrypt a ciphertext computed for such non-accepting access structure even if it gains access to the master secret key. Such a CP-ABE is very hard to construct from standard assumptions, since it implies witness encryption verifiable by monotone boolean formulae. To circumvent this issue, we use a wildcard attribute $\mathsf{S}^\star$ that satisfies all access structures realizable by LSSS. The existence of such a wildcard attribute set will provide no additional information with the leakage of the master secret keys of the CP-ABE which correspond to the corrupted slots, since the adversary can always decrypt any ciphertexts of those slots. Moreover, an MI-ABE with corruption generally implies witness encryption [28], which is also the case in our work. However, we bypass this implication by adding the wildcard attribute set. The use of wildcards in our setting is motivated by the previous works [28,12]. It is easy to see that our MC-AB-UIPFE supports such a wildcard by simply setting $\psi = 0$ (see Construction 2) when computing a ciphertext for $\mathsf{S}^\star$.

Dynamic Decentralized UIPFE. We now present the technical details for obtaining a DD-UIPFE. Our approach builds upon the framework established in [9] and integrates our esUIPFE to facilitate the *dynamic* joining of parties into the system, allowing them to encrypt arbitrary-length vectors. At a very high level, each party joins the system by sampling a PRF seed seed_k. Subsequently, they dynamically sample $(\mathsf{iMPK}_k, \mathsf{iMSK}_k)$ and $(\mathsf{euMPK}_k, \mathsf{euMSK}_k)$ using the PRF with a user set $\mathcal{U}$ as input. During key generation, each party computes a vector $[\![\boldsymbol{\alpha}]\!]_2 = H(\{\mathbf{y}_k\}_k, \mathcal{U})$ using a hash function, encoding it into $\mathsf{sk}(\mathbf{y}_k) = (\mathbf{y}_k, \underline{\boldsymbol{\alpha}})$. In parallel, each party employs another PRF with input $(\mathcal{U}, L)$ to compute vectors $\mathbf{s}_k$ such that $\sum_{k\in\mathcal{U}} \mathbf{s}_k = \mathbf{0}$, encoding these into $\mathsf{ct}(\mathbf{x}_k) = (\mathbf{x}_k, \underline{\mathbf{s}_k})$. The seed for this PRF is derived using a non-interactive key exchange protocol. The unbounded nature of the underlying vectors is preserved through the esUIPFE. For further details and a concrete description of our DD-UIPFE, we refer the reader to Section 7.

A roadmap of our constructions is illustrated in Figure 1.

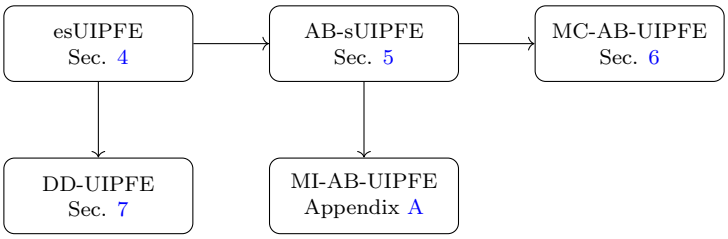

Figure 1: Roadmap of our constructions. Note that AB-sUIPFE captures AB-UIPFE, and MC(MI)-AB-UIPFE captures MC(MI)-UIPFE.

3 Preliminaries

Notations. For some prime p, $\mathbb{Z}_p$ denotes a finite field of order p, and for $n \in \mathbb{N}$, the set $\mathsf{GL}_n(\mathbb{Z}_p)$ denotes all $n \times n$ invertible matrices with entries from $\mathbb{Z}_p$. We indicate the process of random sampling of an element a from the finite set S by $a \leftarrow S$. We use $L(S)$ to denote the set of finite lists of elements from S, and $[n]$ to

denote the set $\{1,\ldots,n\}$. A bold uppercase letter represents a matrix, e.g., $\mathbf{A}$, while a bold lowercase letter indicates a vector, e.g., $\mathbf{x}$. The index set of the vector $\mathbf{a}$ is denoted by $I_{\mathbf{a}}$. For example, if $\mathbf{a} = (a_1, a_3, a_8)$, we write $I_{\mathbf{x}} = \{1, 3, 8\}$. The concatenation of vectors is denoted by $\mathbf{a}_1||\mathbf{a}_2||\ldots||\mathbf{a}_n$. The length of a vector $\mathbf{a}$ is denoted by $|\mathbf{a}|$. For any two vectors $\mathbf{a} = (a_i)_{i\in I_{\mathbf{a}}}$ and $\mathbf{b} = (b_i)_{i\in I_{\mathbf{b}}}$ with the respective index sets $I_{\mathbf{a}}$ and $I_{\mathbf{b}}$, a permissive relation $\mathcal{R}$ is defined as follows: $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}$ if and only if $I_{\mathbf{b}} \subseteq I_{\mathbf{a}}$. The inner product $\langle \mathbf{a}, \mathbf{b}\rangle_p$ in permissive case is defined as $\sum_{i\in I_{\mathbf{b}}} a_i b_i$. If both the vectors are in same length m, then $\langle \mathbf{a}, \mathbf{b}\rangle$ represents the normal inner product as $\sum_{i\in[m]} a_i b_i$. Consider g_ι as a generator of the cyclic group $\mathbb{G}_\iota$. If $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ is an n-tuple vector, then $[\![\mathbf{a}]\!]_\iota = (g_\iota^{a_1}, g_\iota^{a_2}, \ldots, g_\iota^{a_n})$. For $c, u \in \mathbb{Z}_p$, we represent $c[\![u]\!]_\iota = g_\iota^{cu}$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathsf{GL}_n(\mathbb{Z}_p)$, we define $[\![\mathbf{A}]\!]_\iota = g_\iota^{\mathbf{A}}$, where exponentiation is carried out component-wise, and $\mathbf{a}_i$ represents the i-th row vector of $\mathbf{A}$. A function $\mathsf{negl} : \mathbb{N} \to [0, 1]$ is said to be *negligible* if, for every $c \in \mathbb{N}$, there exists a $\lambda_c \in \mathbb{N}$ such that $\mathsf{negl}(\lambda) \leq \frac{1}{\lambda^c}$ for all $\lambda > \lambda_c$. Consider two distributions A and B. Then, $A \approx_s B$ denotes that the two distributions are statistically indistinguishable, while $A \approx_c B$ represents computational indistinguishability. If $A \equiv B$, the two distributions are identically distributed. We discuss the remaining preliminaries in the following.

Definition 1 (Pairing Groups) A bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ consists of a prime p, two multiplicative source groups $\mathbb{G}_1, \mathbb{G}_2$ and a target group $\mathbb{G}_T$ with the order $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$ where g_1, g_2 are the generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. We consider a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies the following:

- *bilinearity:* $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2,\ a, b \in \mathbb{Z}_p$ and
- *non-degeneracy:* $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.

A bilinear group generator $\mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ takes the security parameter λ and outputs a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ with a λ-bit prime integer p.

Assumption 1 (Decisional Diffie-Hellman) Let $\mathbb{G}$ be a cyclic group of prime order p. We define the distribution $(D, [\![h_{\mathfrak{b}}]\!])$ over $\mathbb{G}$ as

$$D = (\mathbb{G}, [\![1]\!], [\![f]\!], [\![g]\!]) \text{ for } f, g \leftarrow \mathbb{Z}_p; \quad h_{\mathfrak{b}} = \begin{cases} fg & \text{if } \mathfrak{b} = 0 \\ h \leftarrow \mathbb{Z}_p & \text{if } \mathfrak{b} = 1. \end{cases}$$

We say that the Decisional Diffie-Hellman (DDH) assumption holds in $\mathbb{G}$ if for all PPT adversaries $\mathcal{A}$, there exists a *negligible* function $\mathsf{negl}(\cdot)$ satisfying the following:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda) = \Big|\Pr[\mathcal{A}(D, [\![h_0]\!]) = 1] - \Pr[\mathcal{A}(D, [\![h_1]\!]) = 1]\Big| \leq \mathsf{negl}(\lambda).$$

Assumption 2 (Matrix Decisional Diffie-Hellman [27]) Let $\ell, k \in \mathbb{N}$ such that $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution over the matrices in $\mathbb{Z}_p^{\ell\times k}$ if it outputs a full-rank matrix with overwhelming probability. Without loss of generality, we assume the first k rows of the matrix $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}$ form an invertible matrix. We define the distributions $(D_i, [\![\mathbf{k}_{\mathfrak{b}}]\!]_i)$ for $i \in \{1, 2\}$ over groups $\mathbb{G}_1, \mathbb{G}_2$ in bilinear group G as

$$D_i = (\mathsf{G}, [\![\mathbf{A}]\!]_i) \text{ for } \mathbf{m} \leftarrow \mathbb{Z}_p^k; \quad \mathbf{k}_{\mathfrak{b}} = \begin{cases} \mathbf{Am} & \text{if } \mathfrak{b} = 0 \\ \mathbf{k} \leftarrow \mathbb{Z}_p^\ell & \text{if } \mathfrak{b} = 1. \end{cases}$$

We say that the Matrix Decisional Diffie-Hellman (MDDH) assumption over $\mathbb{G}_i$ for $i \in \{1, 2\}$ holds if for all PPT adversaries $\mathcal{A}$, there exists a *negligible* function $\mathsf{negl}(\cdot)$ satisfying the following:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MDDH}}(\lambda) = \Big|\Pr[\mathcal{A}(D_i, [\![\mathbf{k}_0]\!]_i) = 1] - \Pr[\mathcal{A}(D_i, [\![\mathbf{k}_1]\!]_i) = 1]\Big| \leq \mathsf{negl}(\lambda).$$

Definition 2 (Access Structure [38]) Let $\mathsf{Att} = \{\mathsf{att}_1, \ldots, \mathsf{att}_n\}$ be a finite set of attributes. An access structure over Att is a collection $\mathbb{A}$ of non-empty subsets of $\{\mathsf{Att}\}$, i.e., $\mathbb{A} \subseteq 2^{\{\mathsf{Att}\}} \setminus \{\phi\}$. A set contained in $\mathbb{A}$

is called an authorized, otherwise it is called unauthorized. An access structure $\mathbb{A}$ is *monotone* if $\mathsf{S}_1 \subseteq \mathsf{S}_2 \subseteq \mathbb{A}$ and $\mathsf{S}_1 \in \mathbb{A}$ implies $\mathsf{S}_2 \in \mathbb{A}$. 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. Note that, $\mathsf{List\text{-}Att}(\mathbb{A})$ is the list of attributes appearing in the access structure $\mathbb{A}$.

In this paper, we represent the access policies realizable by *linear secret sharing schemes* (LSSS) which we define below.

Definition 3 (Linear Secret Sharing Scheme [38]) 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 $\mathbf{c} \cdot \mathbf{A} = (1, 0, ..., 0)$ if and only if $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$.

To share a secret s, pick uniformly random values $v_2, \ldots, v_d \leftarrow K$ and generate a vector of n shares as $\mathbf{s} := (s, v_2, \ldots, v_d) \cdot \mathbf{A}^\top$ such that the share for attribute att_i is the i-th coordinate s_i of $\mathbf{s}$. Only an authorized set $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$ can recover $\mathbf{c}$ to reconstruct s by computing $\mathbf{c} \cdot \mathbf{s}^\top = \mathbf{c} \cdot (\mathbf{A} \cdot (s, v_2, \ldots, v_d)^\top) = s$. For any unauthorized set, reconstructing the secret will result in a uniformly random value.

Definition 4 (Pseudorandom Function) A pseudorandom function (PRF) family $\mathcal{F} = \{\mathsf{PRF}^{\mathsf{seed}}(\cdot)\}_{\mathsf{seed} \in \mathcal{K}_{\mathsf{prf}}}$ with a keyspace $\mathcal{K}_{\mathsf{prf}}$, domain $\mathcal{X}$ and codomain $\mathcal{Y}$ is a function family that consists of functions $\mathsf{PRF}^{\mathsf{seed}} : \mathcal{X} \to \mathcal{Y}$. Let Rand be the set of random functions with the same domain $\mathcal{X}$ and codomain $\mathcal{Y}$. Then for all PPT adversaries $\mathcal{A}$, there exists a *negligible* function $\mathsf{negl}(\cdot)$ satisfying the following:

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}^{\mathsf{PRF}^{\mathsf{seed}}(\cdot)}(\lambda) = 1] - \Pr[\mathcal{A}^{\mathsf{Rand}(\cdot)}(\lambda) = 1]| \leq \mathsf{negl}(\lambda)$$

with $\mathsf{seed} \leftarrow \mathcal{K}_{\mathsf{prf}}$ and $\mathsf{Rand}(\cdot) \leftarrow \mathsf{Rand}$.

Definition 5 (Non-Interactive Key Exchange [19]) A non-interactive key exchange (NIKE) scheme $\Pi_{\mathsf{nike}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyShared})$ for shared key space $\mathcal{K}_s$ consists of the following algorithms:

$\mathsf{Setup}(1^\lambda) \to \mathsf{PP}$: The setup algorithm takes as input the security parameter λ and outputs the public parameters PP.

$\mathsf{KeyGen}(\mathsf{PP}) \to (\mathsf{PK}, \mathsf{SK})$: The key generation algorithm takes as input PP and outputs party's public key PK and the corresponding secret key SK.

$\mathsf{KeyShared}(\mathsf{PK}, \mathsf{SK}) \to K$: The key shared algorithm takes as input a party's PK, SK and deterministically outputs a shared key $K \in \mathcal{K}_s$.

Correctness: For all $\lambda \in \mathbb{N}$, we require

$$\Pr\left[K_{i,j} = K_{j,i} : \begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{KeyGen}(\mathsf{PP}), (\mathsf{PK}_j, \mathsf{SK}_j) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ K_{i,j} \leftarrow \mathsf{KeyShared}(\mathsf{PK}_i, \mathsf{SK}_j), K_{j,i} \leftarrow \mathsf{KeyShared}(\mathsf{PK}_j, \mathsf{SK}_i) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 6 (Security of NIKE) The $\Pi_{\mathsf{nike}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyShared})$ scheme is secure if for all PPT adversaries $\mathcal{A}$, there exists a *negligible* function $\mathsf{negl}(\cdot)$ satisfying

$$\mathsf{Adv}^{\mathsf{nike}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{nike}}_{\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{nike}}_{\mathcal{A}}(\lambda, 1) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{nike}}_{\mathcal{A}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{nike}}_{\mathcal{A}}(\lambda,\beta):}$

1: $\mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda)$.
2: $Q \leftarrow \phi$.
3: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{NHonest}},\mathcal{O}_{\mathsf{Cor}},\mathcal{O}_{\mathsf{Reveal}},\mathcal{O}_{\mathsf{Test},\beta}}(\mathsf{PP})$.
4: output β'.

$\underline{\mathcal{O}_{\mathsf{NHonest}}():}$

1: $(\mathsf{PK},\mathsf{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$.
2: $Q = Q \cup \{(\mathsf{PK},\mathsf{SK})\}$.
3: output PK.

$\underline{\mathcal{O}_{\mathsf{Corr}}(\mathsf{PK}):}$

1: if $\exists(\mathsf{PK},\mathsf{SK}) \in Q$,
$Q = Q \setminus \{(\mathsf{PK},\mathsf{SK})\}$.
output SK.

$\underline{\mathcal{O}_{\mathsf{Reveal}}(\mathsf{PK},\mathsf{PK}'):}$

1: if $(\mathsf{PK},\mathsf{SK}) \in Q$,
output $\mathsf{KeyShared}(\mathsf{PK}',\mathsf{SK})$.
2: else if $(\mathsf{PK}',\mathsf{SK}') \in Q$,
output $\mathsf{KeyShared}(\mathsf{PK},\mathsf{SK}')$.

$\underline{\mathcal{O}_{\mathsf{Test},\beta}(\mathsf{PK},\mathsf{PK}'):}$

1: if $(\mathsf{PK},\mathsf{SK}) \notin Q$ or $(\mathsf{PK}',\mathsf{SK}') \notin Q$,
abort.
2: if $\beta = 0$,
output $\mathsf{KeyShared}(\mathsf{PK},\mathsf{SK}')$.
3: if $\beta = 1$,
output $K \leftarrow \mathcal{K}_s$.

Definition 7 (All-or-nothing Encryption [19]) An all-or-nothing encryption (AoNE) scheme $\Pi_{\mathsf{aone}} = (\mathsf{Setup},\mathsf{LocalSetup},\mathsf{Enc},\mathsf{Dec})$ is defined over the message space $\mathcal{M} = \{0,1\}^\ell \times 2^{\mathcal{ID}} \times \mathcal{L}$ with $\ell \in \mathbb{N}$, key space $\mathcal{K} = \phi$, identity space $\mathcal{ID}$ and label space $\mathcal{L}$. Note that, AoNE is a class of dynamic decentralized functional encryption (DDFE) scheme. The scheme consists of the following algorithms:

$\mathsf{GlobalSetup}(1^\lambda) \to \mathsf{PP}$: The global setup algorithm takes as input security parameter λ and output public parameter PP.

$\mathsf{LocalSetup}(\mathsf{PP}) \to (\mathsf{PK}_k,\mathsf{MSK}_k)$: The local setup algorithm takes as input PP and output the party's public key PK_k and secret key MSK_k. The following two algorithms implicitly take PK_k.

$\mathsf{Enc}(\mathsf{MSK}_k,(x,\mathcal{U},L)) \to \mathsf{CT}_k$: The encryption algorithm takes as input party's MSK_k, a message x, a user set $\mathcal{U}$, a label L and outputs the corresponding ciphertext CT_k.

$\mathsf{Dec}(\{\mathsf{CT}_k\}_{i\in\mathcal{U}_{\mathsf{Msg}}}) \to \zeta \ \vee \perp$. This algorithm takes as inputs $\{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}}$ where $\mathcal{U}_{\mathsf{Msg}} \subseteq \mathcal{ID}$ is any set of users. It outputs either ζ or $\perp$ indicating failure.

Correctness: For all $\lambda \in \mathbb{N}, x \in \{0,1\}^\ell, \mathcal{U}_{\mathsf{Msg}} \in 2^{\mathcal{ID}}$ and $L_k \in \mathcal{L}$, we require

$$\Pr\left[\zeta = f(\epsilon,\{k,(x_k,\mathcal{U}_k,L_k)\}_{k\in\mathcal{U}_{\mathsf{Msg}}}) : \begin{array}{l}\mathsf{PP} \leftarrow \mathsf{GlobalSetup}(1^\lambda)\\ (\mathsf{PK}_k,\mathsf{SK}_k) \leftarrow \mathsf{LocalSetup}(\mathsf{PP})\\ \mathsf{CT}_k \leftarrow \mathsf{Enc}(\mathsf{MSK}_k,(x_k,\mathcal{U}_k,L_k))\\ \zeta \leftarrow \mathsf{Dec}(\{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}})\end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

where the function f is defined as follows:

$$f(\epsilon,\{k,(x_k,\mathcal{U}_k,L_k)\}_{k\in\mathcal{U}_{\mathsf{Msg}}}) = \begin{cases}(x_k)_{k\in\mathcal{U}_{\mathsf{Msg}}} & \text{if } (\star) \text{ holds}\\ \perp & \text{otherwise.}\end{cases}$$

The conditions in $(\star)$ define as follows:
- for all $k \in \mathcal{U}_{\mathsf{Msg}}$, $\mathcal{U}_{\mathsf{Msg}} = \mathcal{U}_k$.
- for all $k_1,k_2 \in \mathcal{U}_{\mathsf{Msg}}$, $L_{k_1} = L_{k_2}$.

Note that, the KeyGen algorithm is not required and Dec works without the secret key components. The security definition is the same as Definition 19 except that no queries to $\mathcal{O}_{\mathsf{KG}}(\cdot)$ are provided to the adversary.

Definition 8 (Security of AoNE) The $\Pi_{\mathsf{aone}} = (\mathsf{GlobalSetup}, \mathsf{LocalSetup}, \mathsf{Enc}, \mathsf{Dec})$ is said to be xx-yy-*indistinguishability* (xx-yy-IND) secure for $\mathsf{xx} \in \{\mathsf{sel}, \mathsf{adp}\}$, $\mathsf{yy} \in \{\mathsf{sym}, \mathsf{asym}\}$ if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a *negligible* function negl such that the following holds

$$\mathsf{Adv}^{\mathsf{aone}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{aone}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{aone}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, 1) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{aone}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{aone}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, \beta):}$

1: $\mathsf{PP} \leftarrow \mathsf{GlobalSetup}(1^\lambda)$.
2: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{HonGen}}(\cdot), \mathcal{O}_{\mathsf{Corr}}(\cdot), \mathcal{O}_{\mathsf{E}}(\cdot), \mathcal{O}_{\mathsf{LoR},\beta}(\cdot)}(\mathsf{PP})$.
3: Output β' if condition $(*)$ is satisfied.

$\underline{\mathcal{O}_{\mathsf{Corr}}(k):}$
output MSK_k.

$\underline{\mathcal{O}_{\mathsf{HonGen}}(k):}$
output $\mathsf{LocalSetup}(\mathsf{PP})$.

$\underline{\mathcal{O}_{\mathsf{E}}(k, (x_k, \mathcal{U}_k, L_k)):}$
output $\mathsf{Enc}(\mathsf{MSK}_k, (x_k, \mathcal{U}_k, L_k))$.

$\underline{\mathcal{O}_{\mathsf{LoR},\beta}(k, (x^0_k, x^1_k, \mathcal{U}_k, L_k)):}$
output $\mathsf{Enc}(\mathsf{MSK}_k, (x^\beta_k, \mathcal{U}_k, L_k))$.

Let $\mathcal{CS}, \mathcal{HS}$ be the sets of all inputs $k \in \mathcal{ID}$ for which the adversary makes queries to the oracles $\mathcal{O}_{\mathsf{HonGen}}(\cdot)$ and $\mathcal{O}_{\mathsf{Corr}}(\cdot)$ respectively. The condition $(*)$ is that if there exist a subset of identities $\mathcal{U}_{\mathsf{Msg}} \subseteq \mathcal{HS}$, then it should satisfy all the following conditions

- $f(\epsilon, \{k, (x^0_k, \mathcal{U}_k, L_k)\}_{k \in \mathcal{U}_{\mathsf{Msg}}}) = f(\epsilon, \{k, (x^1_k, \mathcal{U}_k, L_k)\}_{k \in \mathcal{U}_{\mathsf{Msg}}})$.
- for all $k \in \mathcal{U}_{\mathsf{Msg}}$, $[\mathcal{O}_{\mathsf{LoR},\beta}(k, (x^0_k, x^1_k, \mathcal{U}_k, L_k))$ is queried or $\mathcal{O}_{\mathsf{E}}(k, (x_k, \mathcal{U}_k, L_k))$ is queried with $x^0_k = x^1_k = x_k]$ and $[x^0_k = x^1_k = x_k$ for $k \in \mathcal{CS}]$.

– For $\mathsf{xx} = \mathsf{sel}$: the adversary first generates the $\mathcal{CS}$ set in one shot, then all queries to $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ or $\mathcal{O}_{\mathsf{E}}(\cdot)$ oracles should be made.
– For $\mathsf{yy} = \mathsf{sym}$: for $i \in \mathcal{CS}$, the queries $\mathcal{O}_{\mathsf{LoR},\beta}(k, (x^0_k, x^1_k, \mathcal{U}_k, L_k))$ must satisfy $x^0_k = x^1_k$.

Definition 9 (Slotted Inner-Product Functional Encryption [36]) A slotted inner-product functional encryption (sIPFE) scheme $\Pi_{\mathsf{sip}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$ is defined over a slot specification $\mathcal{S}' = \mathcal{S}'_{\mathsf{pub}} \times \mathcal{S}'_{\mathsf{priv}}$, where $\mathcal{S}'_{\mathsf{pub}} = \mathbb{Z}^{n_1}_p$ represents the public slot of size n_1, and $\mathcal{S}'_{\mathsf{priv}} = \mathbb{Z}^{n_2}_p$ represents the private slot of size n_2. Let G be a bilinear group containing the groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of prime order p. The scheme consists of the following five algorithms:

$\mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}) \to (\mathsf{MSK}, \mathsf{MPK})$: The setup algorithm takes as input security parameter λ and outputs the master public key MPK and the master secret key MSK.

$\mathsf{KeyGen}(\mathsf{MSK}, [\![\mathbf{y}]\!]_2) \to \mathsf{SK}$: The key generation algorithm takes as input MSK, a slot vector $\mathbf{y} \in \mathcal{S}'$ in the exponent of the group $\mathbb{G}_2$ and outputs the secret key SK.

$\mathsf{Enc}(\mathsf{MSK}, [\![\mathbf{x}]\!]_1) \to \mathsf{CT}$: The encryption algorithm takes as input MSK, the slot vector $\mathbf{x} \in \mathcal{S}'$ in the exponent of the group $\mathbb{G}_1$ and outputs the ciphertext CT.

$\mathsf{SlotEnc}(\mathsf{MPK}, [\![\mathbf{x}]\!]_1) \to \mathsf{CT}$: The slotted encryption algorithm takes as input MPK, the public slot vector $\mathbf{x} \in \mathcal{S}'_{\mathsf{pub}}$ in the exponent of the group $\mathbb{G}_1$ and outputs the ciphertext CT.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{CT}) \to [\![d]\!]_T \ \vee \perp$: The deception algorithm takes as input SK, CT and outputs an element $[\![d]\!]_T \in \mathbb{G}_T$.

Correctness: For all $\lambda \in \mathbb{N}$, and $\mathbf{x}, \mathbf{y} \in \mathcal{S}'$, we require

$$\Pr\left[[\![d]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle]\!]_T \ : \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}) \\ \mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, [\![\mathbf{y}]\!]_2) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, [\![\mathbf{x}]\!]_1) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Slot-mode correctness: For all $\mathbf{x} \in \mathcal{S}'_{\mathsf{pub}}$, the following distributions are required to be identical:

$$\left\{(\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) : (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}), \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{0}^{n_2})]\!]_1)\right\},$$
$$\left\{(\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) : (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}), \mathsf{CT} \leftarrow \mathsf{SlotEnc}(\mathsf{MPK}, [\![\mathbf{x}]\!]_1)\right\}$$

Definition 10 (Security of sIPFE) A sIPFE scheme $\Pi_{\mathsf{sip}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$ is said to be xx-*function-hiding-indistinguishability* (xx-FH-IND) secure for xx $\in \{\mathsf{sel}, \mathsf{adp}\}$ if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that the following holds

$$\mathsf{Adv}^{\mathsf{sip}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda) = \left|\Pr\left[\mathsf{Expt}^{\mathsf{sip}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{sip}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{sip}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{sip}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, \beta) :}$
1: $(n_1, n_2) \leftarrow \mathcal{A}(1^\lambda)$.
2: $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2})$.
3: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KG},\beta}(\cdot,\cdot), \mathcal{O}_{\mathsf{E},\beta}(\cdot)}(\mathsf{MPK})$.
4: output β'.

$\underline{\mathcal{O}_{\mathsf{KG},\beta}(\mathbf{y}^{(0)}_\ell, \mathbf{y}^{(1)}_\ell) :}$
output $\mathsf{KeyGen}(\mathsf{MSK}, [\![\mathbf{y}^{(\beta)}_\ell]\!]_2)$.

$\underline{\mathcal{O}_{\mathsf{E},\beta}(\mathbf{x}^{(0)}_\kappa, \mathbf{x}^{(1)}_\kappa) :}$
output $\mathsf{Enc}(\mathsf{MSK}, [\![\mathbf{x}^{(\beta)}_\kappa]\!]_1)$.

Here, $(\mathbf{y}^{(0)}_\ell, \mathbf{y}^{(1)}_\ell)$ denotes the ℓ-th secret key query and $(\mathbf{x}^{(0)}_\kappa, \mathbf{x}^{(1)}_\kappa)$ denotes the κ-th encryption query. Let Q_k, Q_c be the numbers of queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$, $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ oracles respectively and $\mathbf{y}^{(\beta)}_\ell = (\mathbf{y}^{(\beta)}_{\ell,\mathsf{pub}}, \mathbf{y}^{(\beta)}_{\ell,\mathsf{priv}})$ with $\mathbf{y}^{(\beta)}_{\ell,\mathsf{pub}} \in \mathcal{S}'_{\mathsf{pub}}$ and $\mathbf{y}^{(\beta)}_{\ell,\mathsf{priv}} \in \mathcal{S}'_{\mathsf{priv}}$ for $\beta \in \{0, 1\}$. Then, the following conditions must hold:

$$[\![\langle \mathbf{x}^{(0)}_\kappa, \mathbf{y}^{(0)}_\ell \rangle]\!]_T = [\![\langle \mathbf{x}^{(1)}_\kappa, \mathbf{y}^{(1)}_\ell \rangle]\!]_T \text{ for all } \ell \in [Q_k], \kappa \in [Q_c] \text{ and } \mathbf{y}^{(0)}_{\ell,\mathsf{pub}} = \mathbf{y}^{(1)}_{\ell,\mathsf{pub}} \text{ for all } \ell \in [Q_k].$$

– For xx = sel: Queries to $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ must be made before any queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$.
– For xx = adp: Queries to $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$, $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$ can be made in any order.

4 Extended Slotted UIPFE

In this section, we define the *extended slotted unbounded IPFE* (esUIPFE) with slot-specification $\mathcal{S} = \mathcal{S}_{\mathsf{pub}} \times \mathcal{S}_{\mathsf{priv}}$ where $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^* \times \mathbb{Z}_p^{n_1}$ and $\mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{n_2}$ represent the elements in the public and private slots respectively. Let $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ be a pairing group (see Definition 1) of prime order p.

Definition 11 An esUIPFE scheme $\Pi_{\mathsf{esi}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$, defined over the slot specification $\mathcal{S} = \mathcal{S}_{\mathsf{pub}} \times \mathcal{S}_{\mathsf{priv}}$, consists of the following five algorithms:

$\mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}) \rightarrow (\mathsf{MPK}, \mathsf{MSK})$: The setup algorithm takes as input the security parameter λ, the length n_1 of the bounded part of $\mathcal{S}_{\mathsf{pub}}$, and the length n_2 of the $\mathcal{S}_{\mathsf{priv}}$ part. It outputs the master public key MPK and the master secret key MSK.

$\mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}, \mathbf{r}, \mathbf{y}_{\mathsf{priv}})]\!]_2, I_{\mathbf{y}}) \rightarrow \mathsf{SK}$: The key generation algorithm takes as input MSK, the slot vector $(\mathbf{y}, \mathbf{r}, \mathbf{y}_{\mathsf{priv}}) \in \mathcal{S}$ in the exponent of the group $\mathbb{G}_2$ with an associated index set $I_{\mathbf{y}}$ of $\mathbf{y}$. It outputs a secret key SK.

$\mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{z}, \mathbf{x}_{\mathsf{priv}})]\!]_1) \rightarrow \mathsf{CT}$: The encryption algorithm takes as input MSK and the slot vector $(\mathbf{x}, \mathbf{z}, \mathbf{x}_{\mathsf{priv}}) \in \mathcal{S}$ in the exponent of the group $\mathbb{G}_1$ where $\mathbf{x} \in \mathbb{Z}_p^m$ (say) is an arbitrary length vector. It outputs a ciphertext CT.

$\mathsf{SlotEnc}(\mathsf{MPK}, [\![(\mathbf{x}, \mathbf{z})]\!]_1) \rightarrow \mathsf{CT}$: The slotted encryption algorithm takes as input MPK and the public-slot vector $(\mathbf{x}, \mathbf{z}) \in \mathcal{S}_{\mathsf{pub}}$ in the exponent of the group $\mathbb{G}_1$, where $\mathbf{x} \in \mathbb{Z}_p^m$ (of arbitrary length). It outputs a ciphertext CT.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{CT}) \to [\![d]\!]_T \ \vee \perp$: The decryption algorithm takes as input SK and CT. It either outputs an element $[\![d]\!]_T \in \mathbb{G}_T$ or a special symbol $\perp$ indicating failure.

Correctness: For all $\lambda \in \mathbb{N}, (\mathbf{x}, \mathbf{z}, \mathbf{x}_{\mathsf{priv}}), (\mathbf{y}, \mathbf{r}, \mathbf{y}_{\mathsf{priv}}) \in \mathcal{S}$ such that $\mathbf{x} \in \mathbb{Z}_p^m, \mathbf{y} \in \mathbb{Z}_p^{|I_{\mathbf{y}}|}$, $\mathbf{z}, \mathbf{r} \in \mathbb{Z}_p^{n_1}$ and $\mathbf{x}_{\mathsf{priv}}, \mathbf{y}_{\mathsf{priv}} \in \mathbb{Z}_p^{n_2}$ with $\mathcal{R}(\mathbf{x}, \mathbf{y}) = 1$, we require

$$\Pr\left[[\![d]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle_p + \langle \mathbf{z}, \mathbf{r} \rangle + \langle \mathbf{x}_{\mathsf{priv}}, \mathbf{y}_{\mathsf{priv}} \rangle]\!]_T \ : \ \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}) \\ \mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}, \mathbf{r}, \mathbf{y}_{\mathsf{priv}})]\!]_2, I_{\mathbf{y}}) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{z}, \mathbf{x}_{\mathsf{priv}})]\!]_1) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Slot-mode correctness: For all $(\mathbf{x}, \mathbf{z}) \in \mathcal{S}_{\mathsf{pub}}$, the following distributions must be identical:

$$\left\{ (\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) : (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}), \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{z}, \mathbf{0}^{n_2})]\!]_1) \right\},$$
$$\left\{ (\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) : (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2}), \mathsf{CT} \leftarrow \mathsf{SlotEnc}(\mathsf{MPK}, [\![(\mathbf{x}, \mathbf{z})]\!]_1) \right\}.$$

Definition 12 (Security of esUIPFE) The $\Pi_{\mathsf{esi}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$ scheme is said to be xx-function-hiding-indistinguishability (xx-FH-IND)-based secure for $\mathsf{xx} \in \{\mathsf{sel}, \mathsf{adp}\}$ if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds:

$$\mathsf{Adv}^{\mathsf{esi}}_{\mathcal{A},\mathsf{xx\text{-}FH\text{-}IND}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{xx\text{-}FH\text{-}IND}}(\lambda, 0) = 1 \right] - \Pr\left[\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{xx\text{-}FH\text{-}IND}}(\lambda, 1) = 1 \right] \right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{xx\text{-}FH\text{-}IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{xx\text{-}IND}}(\lambda, \beta)}$
1: $(n_1, n_2) \leftarrow \mathcal{A}(1^\lambda)$.
2: $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2})$.
3: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KG},\beta}(\cdot,\cdot), \mathcal{O}_{\mathsf{E},\beta}(\cdot)}(\mathsf{MPK})$.
4: output β'.

$\underline{\mathcal{O}_{\mathsf{KG},\beta}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}) :}$
1: output $\mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(\beta)}_{\ell,\mathsf{priv}})]\!]_2, I_{\mathbf{y}})$.

$\underline{\mathcal{O}_{\mathsf{E},\beta}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta \in \{0,1\}}) :}$
1: output $\mathsf{Enc}(\mathsf{MSK}[\![(\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{\mathsf{priv}})]\!]_1)$.

Here, $(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell})$ denotes the ℓ-th secret key query and $\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{z}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta \in \{0,1\}}$ denotes the κ-th encryption query where it must hold that: $|\mathbf{x}^{(0)}_\kappa| = |\mathbf{x}^{(1)}_\kappa| = m_\kappa$ (say). Let Q_k, Q_c be the numbers of queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$, $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ oracles respectively. Then, for all $\ell \in [Q_k], \kappa \in [Q_c]$ with $\mathcal{R}(\mathbf{x}^{(0)}_\kappa, \mathbf{y}_\ell) = \mathcal{R}(\mathbf{x}^{(1)}_\kappa, \mathbf{y}_\ell) = 1$, it must hold that

$$[\![\langle \mathbf{x}^{(0)}_\kappa, \mathbf{y}_\ell \rangle_p + \langle \mathbf{z}^{(0)}_\kappa, \mathbf{r}_\ell \rangle + \langle \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}} \rangle]\!]_T = [\![\langle \mathbf{x}^{(1)}_\kappa, \mathbf{y}_\ell \rangle_p + \langle \mathbf{z}^{(1)}_\kappa, \mathbf{r}_\ell \rangle + \langle \mathbf{x}^{(1)}_{\kappa,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}} \rangle]\!]_T.$$

– For $\mathsf{xx} = \mathsf{sel}$: Queries to $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ must be made before any queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$.
– For $\mathsf{xx} = \mathsf{adp}$: Queries to $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$, $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$ can be made in any order.

4.1 Construction

Consider $\Pi_{\mathsf{sip}} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ be a bounded sIPFE with slot-specification $\mathcal{S}' = \mathcal{S}'_{\mathsf{pub}} \times \mathcal{S}'_{\mathsf{priv}}$ with $\mathcal{S}'_{\mathsf{pub}} = \mathbb{Z}_p^{n_1+4}$, $\mathcal{S}'_{\mathsf{priv}} = \mathbb{Z}_p^{n_2} \times \mathbb{Z}_p^2 \times \mathbb{Z}_p^{n_2}$. We present our esUIPFE scheme $\Pi_{\mathsf{esi}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$ with $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^* \times \mathbb{Z}_p^{n_1}$ and $\mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{n_2}$ below. We discuss the bounded sIPFE in Definition 9.

$\mathsf{Setup}(1^\lambda, 1^{n_1}, 1^{n_2})$: The setup algorithm takes as input the security parameter λ, the lengths n_1, n_2 and executes the following steps:

1. Generates $(\mathsf{iMPK}, \mathsf{iMSK}) \leftarrow \mathsf{iSetup}(1^\lambda, 1^{n_1+4}, 1^{2n_2+2})$.
2. Outputs the master public key $\mathsf{MPK} = \mathsf{iMPK}$ and the master secret key $\mathsf{MSK} = \mathsf{iMSK}$.

$\mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}, \mathbf{r}, \mathbf{y}_{\mathsf{priv}})]\!]_2, I_{\mathbf{y}})$: The key generation takes input MSK with a vector tuple $[\![(\mathbf{y}, \mathbf{r}, \mathbf{y}_{\mathsf{priv}})]\!]_2$ and does the following steps:

1. Defines the vector $\mathbf{k}_{i,\mathsf{fe}}$ as follows:

$$\mathbf{k}_{i,\mathsf{fe}} = \quad (\quad \pi_i(i,1), \quad y_i, \quad \mathbf{s}_i, \quad r_i, \quad \mathbf{y}_{i,\mathsf{priv}}, \quad 0, \quad 0, \quad \mathbf{0}^{n_2} \quad) \quad \forall i \in I_{\mathbf{y}}$$

   where $\pi_i, r_i \leftarrow \mathbb{Z}_p$, $\mathbf{s}_i \leftarrow \mathbb{Z}_p^{n_1}, \mathbf{y}_{i,\mathsf{priv}} \leftarrow \mathbb{Z}_p^{n_2}$ such that $\sum_{i \in I_{\mathbf{y}}} r_i = 0$, $\sum_{i \in I_{\mathbf{y}}} \mathbf{s}_i = \mathbf{r}$, $\sum_{i \in I_{\mathbf{y}}} \mathbf{y}_{i,\mathsf{priv}} = \mathbf{y}_{\mathsf{priv}}$.
2. Generates $\mathsf{iSK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\mathbf{k}_{i,\mathsf{fe}}]\!]_2)$.
3. Outputs the secret key $\mathsf{SK} = \{\mathsf{iSK}_i\}_{i \in I_{\mathbf{y}}}$.

$\mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{z}, \mathbf{x}_{\mathsf{priv}})]\!]_1)$: The encryption algorithm takes as input MSK, a vector tuple $[\![(\mathbf{x}, \mathbf{z}, \mathbf{x}_{\mathsf{priv}})]\!]_1$ and proceeds to do the following steps:

1. Defines the vector $\mathbf{c}_{i,\mathsf{fe}}$ as follows:

$$\mathbf{c}_{i,\mathsf{fe}} = \quad (\quad \sigma_i(1,-i), \quad x_i, \quad \mathbf{z}, \quad \alpha, \quad \mathbf{x}_{\mathsf{priv}}, \quad 0, \quad 0, \quad \mathbf{0}^{n_2} \ ) \ \ \forall i \in [m]$$

   where $\alpha, \sigma_i \leftarrow \mathbb{Z}_p$ for all $i \in [m]$.
2. Generates $\mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMSK}, [\![\mathbf{c}_{i,\mathsf{fe}}]\!]_1)$.
3. Outputs the ciphertext $\mathsf{CT} = \{\mathsf{iCT}_i\}_{i \in [m]}$.

$\mathsf{SlotEnc}(\mathsf{MPK}, [\![(\mathbf{x}, \mathbf{z})]\!]_1)$: The slot encryption algorithm takes as input MPK, a vector tuple $(\mathbf{x}, \mathbf{z})$ and performs the following steps:

1. Defines the vector $\mathbf{c}_{i,\mathsf{fe}}$ as follows:

$$\mathbf{c}_{i,\mathsf{fe}} = \quad (\quad \sigma_i(1,-i), \quad x_i, \quad \mathbf{z}, \quad \alpha \ ) \ \ \forall i \in [m]$$

   where $\alpha, \sigma_i \leftarrow \mathbb{Z}_p$ for all $i \in [m]$.
2. Generates $\mathsf{iCT}_i \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![\mathbf{c}_{i,\mathsf{fe}}]\!]_1)$.
3. Outputs the ciphertext $\mathsf{CT} = \{\mathsf{iCT}_i\}_{i \in [m]}$.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{CT})$: The decryption algorithm takes as input the secret key SK, the ciphertext CT and proceeds as follows:

1. If $I_{\mathbf{y}} \subseteq [m]$, i.e., $\mathcal{R}(\mathbf{x}, \mathbf{y}) = 1$, then it computes $[\![d]\!]_T \leftarrow \prod_{i \in I_{\mathbf{y}}} \mathsf{iDec}(\mathsf{iSK}_i, \mathsf{iCT}_i)$ and returns $[\![d]\!]_T$.
2. Otherwise, it returns $\perp$.

Correctness: From the correctness of Π_{sip} with $\mathcal{R}(\mathbf{x}, \mathbf{y}) = 1$, we have

$$\mathsf{iDec}(\mathsf{iSK}_i, \mathsf{iCT}_i) = [\![x_i y_i + \langle \mathbf{s}_i, \mathbf{z} \rangle + \langle \mathbf{x}_{\mathsf{priv}}, \mathbf{y}_{i,\mathsf{priv}} \rangle]\!]_T \text{ and } \prod_{j \in I_{\mathbf{y}}} \mathsf{iDec}(\mathsf{iSK}_i, \mathsf{iCT}_i) \quad = [\![\langle \mathbf{x}, \mathbf{y} \rangle_p + \langle \mathbf{r}, \mathbf{z} \rangle + \langle \mathbf{x}_{\mathsf{priv}}, \mathbf{y}_{\mathsf{priv}} \rangle]\!]_T \ .$$

Slot-mode correctness: From the slot-mode correctness of Π_{sip}, we have

$$\mathsf{iEnc}(\mathsf{iMSK}, [\![\mathbf{c}'_{i,\mathsf{fe}}]\!]_1) \equiv \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![\mathbf{c}_{i,\mathsf{fe}}]\!]_1)$$

where $\mathbf{c}'_{i,\mathsf{fe}} = (\mathbf{u} || \mathbf{0}^{2n_2+2})$ and $\mathbf{c}_{i,\mathsf{fe}} = \mathbf{u}$ such that $\mathbf{u} \in \mathbb{Z}_p^{4+n_1}$.

4.2 Security Analysis

In Theorem 6, we present the security analysis of our esUIPFE scheme, as described in Construction 4.1. We will use the following lemma from [24] (adapted in our setting) to handle the non-permissive keys in the security analysis.

Lemma 1 (Handling Non-permissive Keys [24]) *Let* $\Pi_{\mathsf{sip}} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ *be a bounded sIPFE scheme with slot-specification* $\mathcal{S}'_{pub} = \mathbb{Z}_p^{n_1+4}$, $\mathcal{S}'_{priv} = \mathbb{Z}_p^{n_2} \times \mathbb{Z}_p^2 \times \mathbb{Z}_p^{n_2}$. *For the polynomials* $t = t(\lambda)$, $n = n(\lambda)$, *with* $n > t$, *we define the following vectors*

$$\begin{aligned}
\mathbf{k}_j &= (\quad \pi_j(j,1),\ 0,\ \mathbf{0}^{n_1},\ 0,\ \mathbf{0}^{n_2},\ 0,\ r_j,\ \mathbf{0}^{n_2}\ ) \ , \\
\mathbf{k}_j^{\beta} &= (\quad \pi_j(j,1),\ 0,\ \mathbf{0}^{n_1},\ 0,\ \mathbf{0}^{n_2},\ 0,\ r_j + \beta\hat{r}_j,\ \mathbf{0}^{n_2}\ ) \quad \forall j \in [t+1, n] \ , \\
\mathbf{c}_{k'} &= (\ \sigma_{k'}(1,-k'),\ 0,\ \mathbf{0}^{n_1},\ 0,\ \mathbf{0}^{n_2},\ 0,\ \widetilde{\alpha}_k,\ \mathbf{0}^{n_2}\ ) \quad \forall k' \in [t] \ ,
\end{aligned}$$

where $\pi_j, \rho_{k'}, r_j, \hat{r}_j, \widetilde{\alpha}_{k'} \leftarrow \mathbb{Z}_p, \beta \leftarrow \{0,1\}$. *For any* $\mathsf{iMSK} \leftarrow \mathsf{iSetup}(1^\lambda, 1^{n_1}, 1^{n_2})$, *the distributions* $\{\{\mathsf{iSK}_{\mathbf{k}_j}\}_{j\in[n]}, \{\mathsf{iSK}_{\mathbf{k}_j^\beta}\}_{j\in[t+1,n]}, \{\mathsf{iCT}_{\mathbf{c}_{k'}}\}_{k'\in[t]}\}$ *for* $\beta \leftarrow \{0,1\}$ *are computationally indistinguishable where*

$$\begin{aligned}
\mathsf{iSK}_{\mathbf{k}_j} &= \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\mathbf{k}_j]\!]_2) \quad \forall j \in [n] \ , \\
\mathsf{iSK}_{\mathbf{k}_j^\beta} &= \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\mathbf{k}_j^\beta]\!]_2) \quad \forall j \in [t+1, n] \ , \\
\mathsf{iCT}_{\mathbf{c}_{k'}} &= \mathsf{iEnc}(\mathsf{iMSK}, [\![\mathbf{c}_{k'}]\!]_1) \quad \forall k' \in [t] \ .
\end{aligned}$$

Theorem 6 *Our* Π_{esi} *scheme achieves sel-FH-IND security as per Definition 12 if DDH assumption holds in the group* $\mathbb{G}_2$ *and* Π_{sip} *scheme is sel-FH-IND as per Defintion 10.*

Proof. We prove Theorem 6 through a sequence of hybrids. We describe the hybrids below. The values Q_c and Q_k represent the number of ciphertext and key generation queries, respectively. We briefly provide indistinguishability arguments of security hybrids in Fig. 2. We represent the slots using dashed boxes, which are updated in the subsequent hybrid steps. In the subsequent hybrids, we will only mention the updated slots.

Hybrid 0. Same as the experiment $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ where the adversary can query the following:

> **Encryption queries**: On receiving the queries $(\mathbf{x}_\kappa^{(0)}, \mathbf{z}_\kappa^{(0)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}), (\mathbf{x}_\kappa^{(1)}, \mathbf{z}_\kappa^{(1)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(1)})$ from the adversary $\mathcal{A}$, the challenger computes the vectors $\mathbf{c}_{\kappa,i,\mathsf{fe}}$ for all $\kappa \in [Q_c]$ as follows:
>
> $$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{z}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \mathbf{c}_{\kappa,i,\mathsf{fe}} = (\sigma_{\kappa,i}(1,-i), \boxed{x_{\kappa,i}^{(0)}}, \boxed{\mathbf{z}_\kappa^{(0)}}, \alpha_\kappa, \boxed{\mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}}, \boxed{\perp}, \boxed{\perp}, \boxed{\perp})$$
>
> where $\alpha_\kappa \leftarrow \mathbb{Z}_p$, for all $i \in [m]$ and $\sigma_{\kappa,i} \leftarrow \mathbb{Z}_p$.
>
> **Key Generation queries**: On receiving ℓ-th functional query $(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell})$, compute the vectors $\mathbf{k}_{\ell,i,\mathsf{fe}}$ for all $i \in I_{\mathbf{y}_\ell}$ as follows:
>
> $$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \mathbf{k}_{\ell,i,\mathsf{fe}} = (\pi_{\ell,i}(i,1), y_{\ell,i}, \mathbf{s}_{\ell,i}, \boxed{r_{\ell,i}}, \boxed{\mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}}, \boxed{0}, 0, \boxed{\mathbf{0}^{n_2}})$$
>
> where $\pi_{\ell,i}, r_{\ell,i} \leftarrow \mathbb{Z}_p$, $\mathbf{s}_{\ell,i} \leftarrow \mathbb{Z}_p^{n_1}, \mathbf{y}_{\ell,i,\mathsf{priv}} \leftarrow \mathbb{Z}_p^{n_2}$ such that $\sum_{i\in I_{\mathbf{y}}} r_{\ell,i} = 0, \sum_{i\in I_{\mathbf{y}_\ell}} \mathbf{s}_{\ell,i} = \mathbf{r}_\ell, \sum_{i\in I_{\mathbf{y}_\ell}} \mathbf{y}_{\ell,i,\mathsf{priv}} = \mathbf{y}_{\ell,\mathsf{priv}}$.

Hybrid 1. This hybrid is the same as Hybrid 0 except that the vectors $\mathbf{c}_{\kappa,i,\mathsf{fe}}$ for all $\kappa \in [Q_c]$ are modified as follows.

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{z}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{\kappa,i,\mathsf{fe}} : \quad (\ x_{\kappa,i}^{(0)},\ \mathbf{z}_\kappa^{(0)},\ \mathbf{x}_{\kappa,\ \mathsf{priv}}^{(0)},\ \boxed{0},\ \boxed{0},\ \boxed{\mathbf{0}}\ ) \ .$$

The indistinguishability follows from the slot mode correctness of the Π_{sip}.

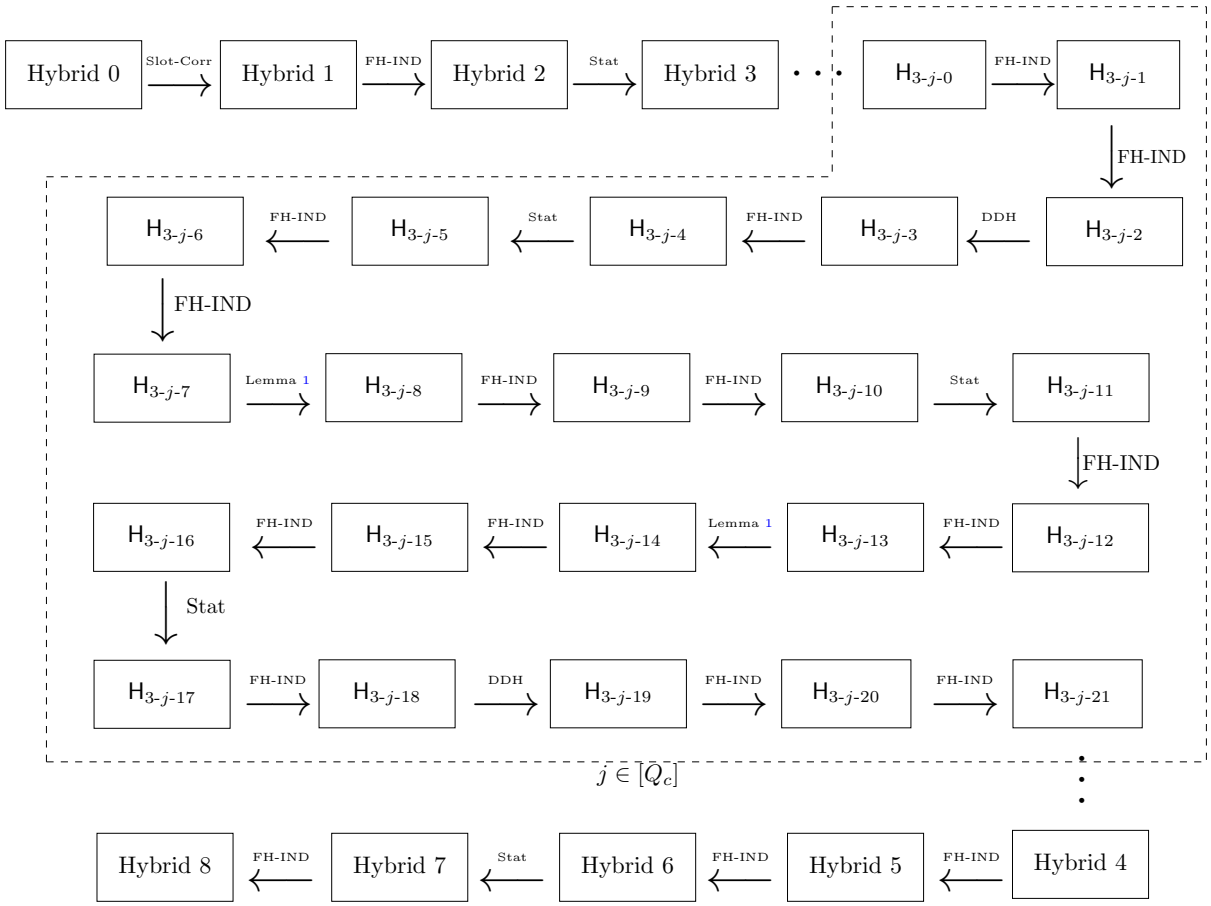

Figure 2: Outline of the security games for Theorem 6. Here, 'Stat' means statistically, 'Slot-Corr' is a shorthand for slot mode correctness of Π_{sip}, and 'FH-IND' is a shorthand for the function-hiding indistinguishability security of Π_{sip}.

Hybrid 2. We modify the vectors $\mathbf{k}_{\ell,i,\mathsf{fe}}$ for all $\ell \in [Q_k]$ as follows.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad \left(\; r_{\ell,i}, \quad \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, \quad \boxed{r_{\ell,i}}, \quad \boxed{\mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}} \; \right) .$$

Hybrid 3. In this hybrid, we set $r_{\ell,i} = \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}$ for all $\ell \in [Q_k]$ where $\widetilde{r}_{\ell,i}, \widetilde{r}'_{\ell,i}, \delta \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,i} = \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}'_{\ell,i} = 0$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad \left(\; \boxed{\widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, \boxed{\widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \; \right) .$$

Hybrid 4. This hybrid is the same as Hybrid 3 except that the ciphertext queries $\mathbf{c}_{\kappa,i,\mathsf{fe}}$ for all $\kappa \in [Q_c]$ are modified as below.

$$\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{z}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{\kappa,i,\mathsf{fe}} : \quad \left(\; \boxed{x^{(1)}_{\kappa,i}}, \boxed{\mathbf{z}^{(1)}_\kappa}, \boxed{\mathbf{0}}, 0, 0, \boxed{\mathbf{x}^{(1)}_{\kappa,\mathsf{priv}}} \; \right) .$$

Hybrid 5. In this hybrid, we modify the following vectors for all $\ell \in [Q_c]$ and $\kappa \in [Q_k]$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}: \quad \left(\widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \boxed{\mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}}, \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \boxed{\mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}} \right),$$
$$\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{z}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{\kappa,i,\mathsf{fe}}: \quad \left(x^{(1)}_{\kappa,i}, \mathbf{z}^{(1)}_\kappa, \boxed{\mathbf{x}^{(1)}_{\kappa,\mathsf{priv}}}, 0, 0, \boxed{\mathbf{0}} \right).$$

Hybrid 6. This hybrid is the same as Hybrid 5 except that all the $\mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}$ are replaced with $\mathbf{0}$ for all $\ell \in [Q_k]$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}: \quad \left(\widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}, \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \boxed{\mathbf{0}} \right).$$

Hybrid 7. We substitute $r_{\ell,i} = \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}$ for all $\ell \in [Q_k]$ in this hybrid.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}: \quad \left(\boxed{r_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}, \boxed{r_{\ell,i}}, \mathbf{0} \right).$$

Hybrid 8. In this last hybrid, we modify the vectors $\mathbf{k}_{\ell,i,\mathsf{fe}}$ for all $\ell \in [Q_k]$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}: \quad \left(r_{\ell,i}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}, \boxed{0}, \mathbf{0} \right).$$

This hybrid is the same as $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 1)$. Thanks to Lemma 2 to Lemma 9, we can conclude the proof of Theorem 6.

Lemma 2 *Hybrid 1 and Hybrid 2 are computationally indistinguishable if the underlying* Π_{sip} *scheme is sel-FH-IND secure.*

Proof. Consider a PPT adversary $\mathcal{A}$ that can distinguish between Hybrid 1 and Hybrid 2. We can use $\mathcal{A}$ to construct $\mathcal{B}$ that can break the sel-FH-IND security of the Π_{sip} scheme as follows. On receiving challenge messages $\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{z}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}$ from $\mathcal{A}$, and the challenger picks the bit $b = 0$, $\mathcal{B}$ computes the following ciphertext and secret key queries by forwarding them to the challenger as follows.

$$\mathsf{iCT}^{\text{Hybrid 1}}_{\kappa,i} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(\sigma_{\kappa,i}, -i\sigma_{\kappa,i}, x^{(0)}_{\kappa,i}, \mathbf{z}^{(0)}_{\kappa,i}, \alpha_\kappa, \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}, 0, 0, 0, \mathbf{0})]\!]_1) = \mathsf{iEnc}(\mathsf{iMSK}, [\![\widetilde{\mathbf{x}}^{(0)}_{\kappa,i}]\!]_1) \text{ and}$$
$$\mathsf{iSK}^{\text{Hybrid 1}}_{\ell,i} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(i\pi_i, \pi_{\ell,i}, y_{\ell,i}, \mathbf{s}_{\ell,i}, r_{\ell,i}\mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, 0, 0, 0, \mathbf{0}^{n_2})]\!]_2) = \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\widetilde{\mathbf{y}}^{(0)}_{\ell,i}]\!]_2).$$

When the challenger picks the bit $b = 1$, $\mathcal{B}$ computes the following vectors and forwards it to the challenger.

$$\mathsf{iCT}^{\text{Hybrid 2}}_{\kappa,i} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(\sigma_{\kappa,i}, -i\sigma_{\kappa,i}, x^{(0)}_{\kappa,i}, \mathbf{z}^{(0)}_{\kappa,i}, \alpha_\kappa, \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}, 0, 0, 0, \mathbf{0})]\!]_1) = \mathsf{iEnc}(\mathsf{iMSK}, [\![\widetilde{\mathbf{x}}^{(1)}_{\kappa,i}]\!]_1) \text{ and}$$
$$\mathsf{iSK}^{\text{Hybrid 2}}_{\ell,i} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(i\pi_i, \pi_{\ell,i}, y_{\ell,i}, \mathbf{s}_{\ell,i}, r_{\ell,i}\mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, r_{\ell,i}, 0, 0, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}]\!]_2) = \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\widetilde{\mathbf{y}}^{(1)}_{\ell,i}]\!]_2).$$

For all $\kappa \in [Q_c]$, $\ell \in [Q_k]$, we have

$$\begin{aligned}\mathsf{iDec}(\mathsf{iSK}^{\text{Hybrid 1}}_{\ell,i}, \mathsf{iCT}^{\text{Hybrid 1}}_{\kappa,i}) &= [\![y_{\ell,i}x^{(0)}_{\kappa,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{\kappa,i}\rangle + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}\rangle + r_{\ell,i}\alpha_\kappa]\!]_T \\ &= \mathsf{iDec}(\mathsf{iSK}^{\text{Hybrid 2}}_{\ell,i}, \mathsf{iCT}^{\text{Hybrid 2}}_{j,i}).\end{aligned}$$

$\mathcal{B}$ is an admissible adversary for the security of the Π_{sip} scheme. When $\mathcal{B}$ samples $b = 0$, the game is identical to Hybrid 1 and when $b = 1$, the game is identical to Hybrid 2. □

Lemma 3 *Hybrid 2 and Hybrid 3 are statistically indistinguishable.*

Proof. The distributions

$$\{r_{\ell,i} \leftarrow \mathbb{Z}_p : \sum_{i \in I_{\mathbf{y}_\ell}} r_{\ell,i} = 0\} \text{ and } \{\widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i} : \delta \leftarrow \mathbb{Z}_p, \sum_{i \in I_{\mathbf{y}_\ell}} (\widetilde{r}'_{\ell,i} + \delta \widetilde{r}'_{\ell,i}) = 0, \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,k} = \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}'_{\ell,i} = 0\}$$

are statistically close as $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i} = 0$ with $\{\widetilde{r}_{\ell,i}, \widetilde{r}'_{\ell,i}\}_{i \in I_{\mathbf{y}_\ell}}, \delta$ uniformly chosen over $\mathbb{Z}_p$ and satisfies $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,i} = \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}'_{\ell,i} = 0$. □

Lemma 4 *Hybrid 3 and Hybrid 4 are computationally indistinguishable if DDH assumption holds over the group $\mathbb{G}_2$ and Π_{sip} scheme is sel-FH-IND secure.*

Proof. We prove the lemma through a sequence of hybrids, namely $\mathsf{H}_{3,j}$ for every ciphertext query $j \in [Q_c]$. We define $\mathsf{H}_{3,0}$ and H_{3,Q_c} the same as Hybrid 3 and Hybrid 4, respectively. The hybrid $\mathsf{H}_{3,j}$ is the same as $\mathsf{H}_{3,j-1}$ with the following changes to the j-th ciphertext query $\mathbf{c}_{j,i,\mathsf{fe}}$.

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad \sigma_{j,i}(1,-i), \quad x_{j,i}^{(1)}, \quad \mathbf{z}_j^{(1)}, \quad \alpha_j, \quad \mathbf{0}, \quad 0, \quad 0 \quad \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .$$

Lemma 5 *Hybrid $\mathsf{H}_{3,j}$ and $\mathsf{H}_{3,j+1}$ are computationally indistinguishable if DDH assumption holds over the group $\mathbb{G}_2$ and Π_{sip} scheme is sel-FH-IND secure.*

Proof. We prove the lemma by introducing several sub-hybrids described as follows.

Hybrid $\mathsf{H}_{3,j,0}$: This hybrid is the same as $\mathsf{H}_{3,j-1}$. We represent the slots using dashed boxes, which are updated in the subsequent hybrid steps. In the subsequent hybrids, we will only mention the updated slots.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}} = \quad (\quad \pi_{\ell,i}(i,1), \quad y_{\ell,i}, \quad \mathbf{s}_{\ell,i}, \quad \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i}, \quad \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \quad \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i}, \quad 0, \quad \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}} = \quad (\quad \sigma_{j,i}(1,-i), \quad x_{j,i}^{(0)}, \quad \mathbf{z}_j^{(0)}, \quad \alpha_j, \quad \mathbf{x}_{j,\mathsf{priv}}^{(0)}, \quad 0, \quad 0 \quad \mathbf{0} \quad) \ .
\end{aligned}$$

Hybrid $\mathsf{H}_{3,j,1}$: In this hybrid, we modify the j-th ciphertext query $\mathbf{c}_{j,i,\mathsf{fe}}$ as follows.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i}, 0, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\quad \{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(0)}, \mathbf{z}_j^{(0)}, \ 0 \ , \mathbf{x}_{j,\mathsf{priv}}^{(0)}, \ \alpha_j \ , 0, \mathbf{0} \quad) \ .
\end{aligned}$$

We can show the indistinguishability between the hybrids through a reduction to the Π_{sip} security as in Lemma 2. We know for all $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,0}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,0}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{j,\mathsf{priv}}^{(0)} \rangle + \widetilde{r}_{\ell,i} \alpha_j + \delta \widetilde{r}'_{\ell,i} \alpha_j]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,1}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,1}}).
\end{aligned}$$

Thus, we have the same inner product values in both hybrids.

Hybrid $\mathsf{H}_{3,j,2}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,1}$ except the following changes to j-th ciphertext and all key generation queries $\ell \in [Q_k]$.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \ \alpha_j \widetilde{r}_{\ell,i} \ , \ \alpha_j \delta \widetilde{r}'_{\ell,i} \ , \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(0)}, \mathbf{z}_j^{(0)}, 0, \mathbf{x}_{j,\mathsf{priv}}^{(0)}, \ 1 \ , \ 1 \ , \mathbf{0} \quad) \ .
\end{aligned}$$

We know for all $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,1}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,1}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{j,\mathsf{priv}}^{(0)} \rangle + \widetilde{r}_{\ell,i}\alpha_j + \delta\widetilde{r}'_{\ell,i}\alpha_j]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,2}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,2}}).
\end{aligned}$$

In case of $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,1}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,1}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + \widetilde{r}_{\ell,i}\alpha_\iota + \delta\widetilde{r}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,2}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,2}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,1}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,1}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + \widetilde{r}_{\ell,i}\alpha_\iota + \delta\widetilde{r}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,2}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,2}}).
\end{aligned}$$

We have the same inner product values in both hybrids, which can be shown computationally indistinguishable, similar to the proof of Lemma 2.

Hybrid $\mathsf{H}_{3,j,3}$: In this hybrid, we replace $\alpha_j\delta$ with $c \leftarrow \mathbb{Z}_p$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \alpha_j\widetilde{r}_{\ell,i}, \quad \widetilde{cr}'_{\ell,i} \quad , \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ .$$

Claim 1 Hybrid $\mathsf{H}_{3,j,2}$ and Hybrid $\mathsf{H}_{3,j,3}$ are computationally indistinguishable if DDH assumption holds over the group $\mathbb{G}_2$.

Proof. Given an adversary $\mathcal{A}$ that can distinguish between the hybrids, we construct an adversary $\mathcal{B}$ that breaks the DDH assumption.

Let $\mathcal{B}$ receives a DDH instances $(\mathbb{G}_2, [\![f]\!]_2, [\![g]\!]_2, [\![h_{\mathfrak{b}}]\!]_2)$ where

$$h_{\mathfrak{b}} = \begin{cases} fg & \text{if } \mathfrak{b} = 0 \\ h \leftarrow \mathbb{Z}_p & \text{if } \mathfrak{b} = 1. \end{cases}$$

For the secret key vector $\mathbf{k}_{\ell,i,\mathsf{fe}}$ of Hybrid $\mathsf{H}_{3,j,2}$, $\mathcal{B}$ implicitly sets $\alpha_j = f$, $\delta = g$ and simulates the secret key component using the oracle $\mathcal{O}_{\mathsf{KG},\beta}$ as follows:

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \pi_{\ell,i}(i,1), \quad y_{\ell,i}, \quad \mathbf{s}_{\ell,i}, \quad \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \quad \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \quad f\widetilde{r}_{\ell,i}, \quad h_{\mathfrak{b}}\widetilde{r}'_{\ell,i}, \quad \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \ ) \ .$$

We know that

$$(\mathbb{G}_2, [\![f]\!]_2, [\![g]\!]_2, [\![fg]\!]_2) \approx_c (\mathbb{G}_2, [\![f]\!]_2, [\![g]\!]_2, [\![h]\!]_2)$$

by the DDH assumption. If $\mathfrak{b} = 0, h_{\mathfrak{b}} = fg$, then the adversarial view is the same as $\mathsf{H}_{3,j,2}$. When $\mathfrak{b} = 1$, $h_{\mathfrak{b}}$ is uniformly chosen from the group $\mathbb{G}_2$ and hence the adversarial view is similar to $\mathsf{H}_{3,j,3}$. Therefore, we have $\mathsf{H}_{3,j,2} \approx_c \mathsf{H}_{3,j,3}$ by the DDH assumption. □

Hybrid $\mathsf{H}_{3,j,4}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,3}$ except the following changes.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}: \quad (\ \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, \boxed{\widetilde{r}_{\ell,i}}, \boxed{\widetilde{r}'_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \ ) \ ,$$
$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}}: \quad (\ x^{(0)}_{j,i}, \mathbf{z}^{(0)}_j, 0, \mathbf{x}^{(0)}_{j,\mathsf{priv}}, \boxed{\alpha_j}, \boxed{\mathrm{c}}, \mathbf{0} \ ) \ .$$

Note that for all key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,3}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,3}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \widetilde{r}_{\ell,i}\alpha_j + \widetilde{r}'_{\ell,i} c]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,4}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,4}}_{j,i}).
\end{aligned}$$

In case of $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,3}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,3}}_{\iota,i}) \\
&= [\![y_{\ell,i} x^{(1)}_{\iota,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(1)}_{\iota,i} \rangle + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{\iota,\mathsf{priv}} \rangle + \widetilde{r}_{\ell,i}\alpha_\iota + \delta\widetilde{r}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,4}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,4}}_{\iota,i}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,3}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,3}}_{\iota,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{\iota,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{\iota,i} \rangle + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{\iota,\mathsf{priv}} \rangle + \widetilde{r}_{\ell,i}\alpha_\iota + \delta\widetilde{r}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,4}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,4}}_{\iota,i}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying Π_{sip}.

Hybrid $\mathsf{H}_{3,j,5}$: We modify the vector $\mathbf{c}_{j,i,\mathsf{fe}}$ in this hybrid as follows.

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}}: \quad (\ x^{(0)}_{j,i}, \quad \mathbf{z}^{(0)}_j, \quad 0, \quad \mathbf{x}^{(0)}_{j,\mathsf{priv}}, \quad \alpha_j, \quad \boxed{\widetilde{\alpha}_j + \delta\alpha_j}, \quad \mathbf{0} \ ) \ ,$$

where $\widetilde{\alpha}_j, \delta \leftarrow \mathbb{Z}_p$. The hybrids $\mathsf{H}_{3,j,4}$ and $\mathsf{H}_{3,j,5}$ are statistically indistinguishable as the distributions of $\{c : c \leftarrow \mathbb{Z}_p\}$ and $\{\widetilde{\alpha}_j + \delta\alpha_j : \widetilde{\alpha}_j, \delta \leftarrow \mathbb{Z}_p\}$ are statistically close.

Hybrid $\mathsf{H}_{3,j,6}$: We modify the queries as below.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}: \quad (\ \widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, \boxed{\widetilde{r}_{\ell,i} + \delta\widetilde{r}'_{\ell,i}}, \widetilde{r}'_{\ell,i}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \ ) \ ,$$
$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}}: \quad (\ x^{(0)}_{j,i}, \mathbf{z}^{(0)}_j, 0\mathbf{x}^{(0)}_{j,\mathsf{priv}}, \alpha_j, \boxed{\widetilde{\alpha}_j}, \mathbf{0} \ ) \ .$$

Note that for all key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,5}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,5}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \widetilde{r}_{\ell,i}\alpha_j + \delta\widetilde{r}'_{\ell,i}\alpha_j + \widetilde{r}'_{\ell,i}\widetilde{\alpha_j}]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,6}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,6}}_{j,i}).
\end{aligned}$$

In case of $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,5}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,5}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + \widetilde{r}_{\ell,i}\alpha_\iota + \delta \widetilde{r}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,6}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,6}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,5}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,5}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + \widetilde{r}_{\ell,i}\alpha_\iota + \delta \widetilde{r}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,6}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,6}}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying $\mathsf{\Pi_{sip}}$.

Hybrid $\mathsf{H}_{3,j,7}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,6}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad r_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \boxed{0}, \widetilde{r}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)}) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(0)}, \mathbf{z}_j^{(0)}, \boxed{\alpha_j}, \mathbf{x}_{j,\mathsf{priv}}^{(0)}, \boxed{0}, \widetilde{\alpha}_j, \mathbf{0} \quad) \ .
\end{aligned}$$

For the j-th ciphertext query and key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,6}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,6}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{j,\mathsf{priv}}^{(0)} \rangle + \widetilde{r}_{\ell,i}\alpha_j + \delta \widetilde{r}'_{\ell,i}\alpha_j + \widetilde{r}'_{\ell,i}\widetilde{\alpha_j}]\!]_T \\
&= [\![y_{\ell,i} x_{j,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(0)} \rangle + r_{\ell,i}\alpha_j + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{j,\mathsf{priv}}^{(0)} \rangle + \widetilde{r}'_{\ell,i}\widetilde{\alpha_j}]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,7}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,7}}).
\end{aligned}$$

In case of $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,6}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,6}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,7}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,7}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,6}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,6}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,7}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,7}}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying $\mathsf{\Pi_{sip}}$.

Hybrid $\mathsf{H}_{3,j,8}$: We modify the key vectors for the cases where $I_{\mathbf{y}_\ell} \nsubseteq [m]$ as follows. We denote it as case (II) where $\overline{r}'_{\ell,i} \leftarrow \mathbb{Z}_p$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II}) : \quad (\quad r_{\ell,i}, \quad \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \quad 0, \quad \boxed{\overline{r}'_{\ell,i}}, \quad \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ .$$

We use the following lemma to prove the indistinguishability between the hybrid Hybrid $\mathsf{H}_{3,j,7}$ and Hybrid $\mathsf{H}_{3,j,8}$,

Claim 2 Hybrid $\mathsf{H}_{3,j,7}$ and Hybrid $\mathsf{H}_{3,j,8}$ are computationally indistinguishable if Lemma 1 holds over groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

Proof. From Lemma 1 over the vectors $\mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I})$, $\mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II})$ and $\mathbf{c}_{j,i,\mathsf{fe}}$, $\mathsf{H}_{3,j,7}$ and $\mathsf{H}_{3,j,8}$ are computationally close. □

Hybrid $\mathsf{H}_{3,j,9}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,8}$ except the following changes. Note that, the $\mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I})$ and $\mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II})$ represent the secret keys corresponding to $I_{\mathbf{y}_\ell} \subseteq [m]$ and $I_{\mathbf{y}_\ell} \not\subseteq [m]$, respectively.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I}): \quad \left(\; r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \boxed{\widetilde{r}'_{\ell,i} - \xi_{j,i} y_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \; \right),$$

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II}): \quad \left(\; r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \boxed{\overline{r}'_{\ell,i} - \xi_{j,i} y_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \; \right),$$

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}}: \quad \left(\; \boxed{x^{(0)}_{j,i} + \xi_{j,i}\widetilde{\alpha}_j}, \mathbf{z}^{(0)}_j, \alpha_j, \mathbf{x}^{(0)}_{j,\mathsf{priv}}, 0, \widetilde{\alpha}_j, \mathbf{0} \; \right),$$

where $\xi_{j,i} \leftarrow \mathbb{Z}_p$. For $I_{\mathbf{y}_\ell} \subseteq [m]$, $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,8}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,8}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + r_{\ell,i}\alpha_j + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \widetilde{r}'_{\ell,i}\widetilde{\alpha_j}]\!]_T \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + y_{\ell,i}\xi_{j,i}\widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + r_{\ell,i}\alpha_j + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \widetilde{r}'_{\ell,i}\widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i}\widetilde{\alpha}_j]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{j,i})
\end{aligned}$$

and when $I_{\mathbf{y}_\ell} \not\subseteq [m]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,8}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,8}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + r_{\ell,i}\alpha_j + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \overline{r}'_{\ell,i}\widetilde{\alpha_j}]\!]_T \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + y_{\ell,i}\xi_{j,i}\widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + r_{\ell,i}\alpha_j + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \overline{r}'_{\ell,i}\widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i}\widetilde{\alpha}_j]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{j,i}).
\end{aligned}$$

For queries $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,8}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,8}}_{\iota,i}) \\
&= [\![y_{\ell,i} x^{(1)}_{\iota,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(1)}_{\iota,i} \rangle + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{\iota,\mathsf{priv}} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{\iota,i}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,8}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,8}}_{\iota,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{\iota,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{\iota,i} \rangle + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{\iota,\mathsf{priv}} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{\iota,i}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying Π_{sip} scheme.

Hybrid $\mathsf{H}_{3,j,10}$: In this hybrid, we modify the vectors $\mathbf{k}_{\ell,i,\mathsf{fe}}$ for all $\ell \in [Q_k]$ where $\boldsymbol{\eta}_j \leftarrow \mathbb{Z}_p^{n_1}$ and $\chi_{j,i} = \frac{\langle \mathbf{x}^{(0)}_{j,\mathsf{priv}}, \mathbf{y}^{(0)}_{j,i,\mathsf{priv}} \rangle - \langle \mathbf{x}^{(1)}_{j,\mathsf{priv}}, \mathbf{y}^{(1)}_{j,i,\mathsf{priv}} \rangle}{\widetilde{\alpha}_j}$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I}): \quad \big(\; r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \; \widetilde{r}'_{\ell,i} - \xi_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \chi_{j,i} \; , \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \; \big) \; ,$$

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II}): \quad \big(\; r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \; \overline{r}'_{\ell,i} - \xi_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \chi_{j,i} \; , \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \; \big) \; ,$$

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta \in \{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}} \quad : \quad \big(x^{(0)}_{j,i} + \xi_{j,i} \widetilde{\alpha}_j, \; \mathbf{z}^{(0)}_j + \boldsymbol{\eta}_j \widetilde{\alpha}_j \; , \alpha_j \; \mathbf{0}^{n_2}, 0, \widetilde{\alpha}_j, \; \mathbf{x}^{(1)}_{j,\mathsf{priv}} \; \big) \; .$$

In cases where $I_{\mathbf{y}_\ell} \subseteq [m]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + y_{\ell,i} \xi_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + r_{\ell,i} \alpha_j + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \widetilde{r}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j]\!]_T \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + y_{\ell,i} \xi_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + \langle \mathbf{s}_{\ell,i}, \boldsymbol{\eta}_j \widetilde{\alpha}_j \rangle + r_{\ell,i} \alpha_j + \widetilde{r}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j \\
&- \widetilde{\alpha}_j \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \chi_{j,i} \widetilde{\alpha}_j + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,10}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,10}}_{j,i}).
\end{aligned}$$

The non-permissive case follows the same. When $I_{\mathbf{y}_\ell} \not\subseteq [m]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + y_{\ell,i} \xi_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + r_{\ell,i} \alpha_j + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle + \overline{r}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j]\!]_T \\
&= [\![y_{\ell,i} x^{(0)}_{j,i} + y_{\ell,i} \xi_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{j,i} \rangle + \langle \mathbf{s}_{\ell,i}, \boldsymbol{\eta}_j \widetilde{\alpha}_j \rangle + r_{\ell,i} \alpha_j + \overline{r}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j \\
&- \widetilde{\alpha}_j \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \chi_{j,i} \widetilde{\alpha}_j + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,10}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,10}}_{j,i}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{\iota,i}) \\
&= [\![y_{\ell,i} x^{(1)}_{\iota,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(1)}_{\iota,i} \rangle + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{\iota,\mathsf{priv}} \rangle + r_{\ell,i} \alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,10}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,10}}_{\iota,i}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,9}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,9}}_{\iota,i}) \\
&= [\![y_{\ell,i} x^{(0)}_{\iota,i} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(0)}_{\iota,i} \rangle + \langle \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(0)}_{\iota,\mathsf{priv}} \rangle + r_{\ell,i} \alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,10}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,10}}_{\iota,i}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying Π_{sip} scheme.

Hybrid $\mathsf{H}_{3,j,11}$: This hybrid is the same as the previous hybrid except the following changes where $\widetilde{R}'_{\ell,i}, \overline{R}'_{\ell,i} \leftarrow \mathbb{Z}_p$, $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{R}'_{\ell,i} = 0$, and set $\xi_{j,i} = \xi'_{j,i} + \frac{x^{(1)}_{j,i} - x^{(0)}_{j,i}}{\widetilde{\alpha}_j}$ and $\boldsymbol{\eta}_j = \boldsymbol{\eta}'_j + \frac{\mathbf{z}^{(1)}_j - \mathbf{z}^{(0)}_j}{\widetilde{\alpha}_j}$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I}): \quad (\quad r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \boxed{\widetilde{R}'_{\ell,i} - \xi_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta'}_j, \mathbf{s}_{\ell,i} \rangle}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \quad) \ ,$$

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II}): \quad (r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \boxed{\overline{R}'_{\ell,i} - \xi_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta'}_j, \mathbf{s}_{\ell,i} \rangle} \ , \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \quad) \ ,$$

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}} \quad : \quad (\boxed{x^{(1)}_{j,i} + \xi'_{j,i}\widetilde{\alpha}_j} \ , \ \boxed{\mathbf{z}^{(1)}_j + \boldsymbol{\eta'}_j \widetilde{\alpha}_j} \ , \alpha_j, \mathbf{0}^{n_2}, 0, \widetilde{\alpha}_j, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \quad) \ .$$

We know that

$$x^{(0)}_{j,i} + \xi_{j,i}\widetilde{\alpha}_j = x^{(0)}_{j,i} + (\xi'_{j,i} + \frac{x^{(1)}_{j,i} - x^{(0)}_{j,i}}{\widetilde{\alpha}_j})\widetilde{\alpha}_j = x^{(1)}_{j,i} + \xi'_{j,i}\widetilde{\alpha}_j$$

$$\mathbf{z}^{(0)}_j + \boldsymbol{\eta}_j\widetilde{\alpha}_j = \mathbf{z}^{(0)}_j + (\boldsymbol{\eta}'_j + \frac{\mathbf{z}^{(1)}_j - \mathbf{z}^{(0)}_j}{\widetilde{\alpha}_j})\widetilde{\alpha}_j = \mathbf{z}^{(1)}_j + \boldsymbol{\eta'}_j\widetilde{\alpha}_j$$

$$\begin{aligned}
&\widetilde{r}'_{\ell,i} - \xi_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \chi_{j,i} \\
&= \widetilde{r}'_{\ell,i} - (\xi'_{j,i} + \frac{x^{(1)}_{j,i} - x^{(0)}_{j,i}}{\widetilde{\alpha}_j}) y_{\ell,i} - \langle \boldsymbol{\eta}'_j + \frac{\mathbf{z}^{(1)}_j - \mathbf{z}^{(0)}_j}{\widetilde{\alpha}_j}, \mathbf{s}_{\ell,i} \rangle + \frac{\langle \mathbf{x}^{(0)}_{j,\mathsf{priv}}, \mathbf{y}^{(0)}_{j,i,\mathsf{priv}} \rangle - \langle \mathbf{x}^{(1)}_{j,\mathsf{priv}}, \mathbf{y}^{(1)}_{j,i,\mathsf{priv}} \rangle}{\widetilde{\alpha}_j} \\
&= \widetilde{r}'_{\ell,i} - \xi'_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta'}_j, \mathbf{s}_{\ell,i} \rangle + \frac{(x^{(0)}_{j,i} - x^{(1)}_{j,i}) y_{\ell,i} + \langle \mathbf{z}^{(0)}_j - \mathbf{z}^{(1)}_j, \mathbf{s}_{\ell,i} \rangle + \langle \mathbf{x}^{(0)}_{j,\mathsf{priv}}, \mathbf{y}^{(0)}_{j,i,\mathsf{priv}} \rangle - \langle \mathbf{x}^{(1)}_{j,\mathsf{priv}}, \mathbf{y}^{(1)}_{j,i,\mathsf{priv}} \rangle}{\widetilde{\alpha}_j} \\
&= \widetilde{r}'_{\ell,i} - \xi'_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta'}_j, \mathbf{s}_{\ell,i} \rangle + \frac{\Delta_{\ell,j,i}}{\widetilde{\alpha}_j} \\
&\approx_s \widetilde{R}'_{\ell,i} - \xi'_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta'}_j, \mathbf{s}_{\ell,i} \rangle
\end{aligned}$$

where $\Delta_{\ell,j,i} = (x^{(0)}_{j,i} - x^{(1)}_{j,i}) y_{\ell,i} + \langle \mathbf{z}^{(0)}_j - \mathbf{z}^{(1)}_j, \mathbf{s}_{\ell,i} \rangle + \langle \mathbf{x}^{(0)}_{j,\mathsf{priv}}, \mathbf{y}^{(0)}_{j,i,\mathsf{priv}} \rangle - \langle \mathbf{x}^{(1)}_{j,\mathsf{priv}}, \mathbf{y}^{(1)}_{j,i,\mathsf{priv}} \rangle$. As $\sum_{i\in I_{\mathbf{y}_\ell}} \Delta_{\ell,j,i} = \langle \mathbf{x}^{(0)}_j - \mathbf{x}^{(1)}_j, \mathbf{y}_\ell \rangle_p + \sum_{i\in I_{\mathbf{y}_\ell}} \langle \mathbf{z}^{(0)}_j - \mathbf{z}^{(1)}_j, \mathbf{s}_{\ell,i} \rangle + \sum_{i\in I_{\mathbf{y}_\ell}} (\langle \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, \mathbf{x}^{(0)}_{j,\mathsf{priv}} \rangle - \langle \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}}, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \rangle) = 0$ from the security definition, we have $\Delta_{\ell,j,i}/\widetilde{\alpha}_j + \widetilde{r}'_{\ell,i}$ statistically close to $\widetilde{R}'_{\ell,i}$. Similarly,

$$\overline{r}'_{\ell,i} - \xi_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \chi_{j,i} \approx_s \widetilde{R}'_{\ell,i} - \xi'_{j,i} y_{\ell,i} - \langle \boldsymbol{\eta'}_j, \mathbf{s}_{\ell,i} \rangle.$$

This is a statistical modification.

Hybrid $\mathsf{H}_{3,j,12}$: In this hybrid, we modify the following vectors,

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I}): \quad (\quad r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \boxed{\widetilde{R}'_{\ell,i} - \xi_{j,i} y_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \quad) \ ,$$

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}): \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II}): \quad (\quad r_{\ell,i}, \mathbf{y}^{(0)}_{\ell,i,\mathsf{priv}}, 0, \boxed{\overline{R}'_{\ell,i} - \xi_{j,i} y_{\ell,i}}, \mathbf{y}^{(1)}_{\ell,i,\mathsf{priv}} \quad) \ ,$$

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_j, \mathbf{z}^{(\beta)}_j, \mathbf{x}^{(\beta)}_{j,\mathsf{priv}}\}_{\beta\in\{0,1\}}): \quad \mathbf{c}_{j,i,\mathsf{fe}} \quad : \quad (\quad x^{(1)}_{j,i} + \xi'_{j,i\widetilde{\alpha}_j}, \ \boxed{\mathbf{z}^{(1)}_j} \ , \alpha_j, \mathbf{0}^{n_2}, 0, \widetilde{\alpha}_j, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \quad) \ .$$

For $I_{\mathbf{y}_\ell} \subseteq [m]$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,12}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,12}}_{j,i}) \\
&= [\![y_{\ell,i} x^{(1)}_{j,i} + y_{\ell,i} \xi'_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(1)}_{j,i} \rangle + \langle \mathbf{s}_{\ell,i}, \boldsymbol{\eta}'_j \widetilde{\alpha}_j \rangle + r_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j \\
&- \widetilde{\alpha}_j \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \rangle]\!]_T \\
&= [\![y_{\ell,i} x^{(1)}_{j,i} + y_{\ell,i} \xi'_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}^{(1)}_{j,i} \rangle + r_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j + \langle \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \mathbf{x}^{(1)}_{j,\mathsf{priv}} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}^{\mathsf{H}_{3,j,13}}_{\ell,i}, \mathsf{iCT}^{\mathsf{H}_{3,j,13}}_{j,i}).
\end{aligned}$$

The non-permissive case follows the same. When $I_{\mathbf{y}_\ell} \not\subseteq [m]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + y_{\ell,i} \xi'_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + \langle \mathbf{s}_{\ell,i}, \boldsymbol{\eta}'_j \widetilde{\alpha}_j \rangle + r_{\ell,i} \alpha_j + \overline{R}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j \\
&- \widetilde{\alpha}_j \langle \boldsymbol{\eta}_j, \mathbf{s}_{\ell,i} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + y_{\ell,i} \xi'_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + r_{\ell,i} \alpha_j + \overline{R}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + r_{\ell,i} \alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + r_{\ell,i} \alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying Π_{sip} scheme.

Hybrid $\mathsf{H}_{3,j,13}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,12}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I}) : \quad (\quad r_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, 0, \boxed{\widetilde{R}'_{\ell,i}}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II}) : \quad (\quad r_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, 0, \boxed{\overline{R}'_{\ell,i}}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} \quad : \quad (\boxed{x_{j,i}^{(1)}}, \mathbf{z}_j^{(1)}, \alpha_j, \mathbf{0}^{n_2}, 0, \widetilde{\alpha}_j, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .
\end{aligned}$$

When $I_{\mathbf{y}_\ell} \subseteq [m]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + y_{\ell,i} \xi'_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + r_{\ell,i} \alpha_j + \widetilde{R}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + r_{\ell,i} \alpha_j + \widetilde{R}'_{\ell,i} \widetilde{\alpha_j} + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

The non-permissive case also follows similarly. For $I_{\mathbf{y}_\ell} \not\subseteq [m]$,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + y_{\ell,i} \xi'_{j,i} \widetilde{\alpha}_j + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + r_{\ell,i} \alpha_j + \overline{R}'_{\ell,i} \widetilde{\alpha_j} - \xi_{j,i} y_{\ell,i} \widetilde{\alpha}_j + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + r_{\ell,i} \alpha_j + \overline{R}'_{\ell,i} \widetilde{\alpha_j} + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,12}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,12}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,13}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,13}}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying Π_{sip} scheme.

Hybrid $\mathsf{H}_{3,j,14}$: Except for the following changes, this hybrid is the same as Hybrid $\mathsf{H}_{3,j,13}$.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad r_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, 0, \boxed{\widetilde{R}'_{\ell,i}}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, \alpha_j, \mathbf{0}^{n_2}, 0, \widetilde{\alpha}_j, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .
\end{aligned}$$

Claim 3 Hybrid $\mathsf{H}_{3,j,13}$ and Hybrid $\mathsf{H}_{3,j,14}$ are computationally indistinguishable if Lemma 1 holds over groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

Proof. From Lemma 1 over the vectors $\mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{I})$, $\mathbf{k}_{\ell,i,\mathsf{fe}}(\mathsf{II})$ and $\mathbf{c}_{j,i,\mathsf{fe}}$, $\mathsf{H}_{3,j,13}$ and $\mathsf{H}_{3,j,14}$ are computationally close. □

Hybrid $\mathsf{H}_{3,j,15}$: We set the value $r_{\ell,i} = \widetilde{R}_{\ell,i} + \delta\widetilde{R}'_{\ell,i}$ and modify the rest of the vectors as follows.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \boxed{\widetilde{R}_{\ell,i} + \delta\widetilde{R}'_{\ell,i}}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \boxed{\widetilde{R}_{\ell,i} + \delta\widetilde{R}'_{\ell,i}}, \widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, \boxed{0}, \mathbf{0}^{n_2}, \boxed{\alpha_j}, \widetilde{\alpha}_j, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .
\end{aligned}$$

For all $\ell \in [Q_k]$ have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,14}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,14}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + r_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\widetilde{\alpha_j} + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_j + \delta\widetilde{R}'_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\widetilde{\alpha_j} + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,15}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,15}}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,14}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,14}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta\widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,15}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,15}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,14}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,14}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + r_{\ell,i}\alpha_\iota]\!]_T \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta \widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,15}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,15}}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying Π_{sip} scheme.

Hybrid $\mathsf{H}_{3,j,16}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,15}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \widetilde{R}_{\ell,i} + \delta \widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \boxed{\widetilde{R}_{\ell,i}}, \widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, 0, \mathbf{0}^{n_2}, \alpha_j, \boxed{\widetilde{\alpha}_j + \delta\alpha_j}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .
\end{aligned}$$

For all key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,15}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,15}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\delta\alpha_j + \widetilde{R}'_{\ell,i}\widetilde{\alpha_j} + \langle \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,16}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,16}}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,15}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,15}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta \widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,16}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,16}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,15}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,15}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta \widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,16}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,16}}).
\end{aligned}$$

Thus, the hybrids are computationally indistinguishable by the security of the Π_{sip}.

Hybrid $\mathsf{H}_{3,j,17}$: We set $c = \widetilde{\alpha}_j + \delta\alpha_j$ as the distributions $\{c : c \leftarrow \mathbb{Z}_p\}$ and $\{\widetilde{\alpha}_j + \delta\alpha_j : \widetilde{\alpha_j}, \delta, \alpha_j \leftarrow \mathbb{Z}_p\}$ are statistically close. Thus, this modification is a statistical change.

$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, 0, \mathbf{0}^{n_2}, \alpha_j, \boxed{c}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .$$

Hybrid $\mathsf{H}_{3,j,18}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,17}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \widetilde{R}_{\ell,i} + \delta \widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \boxed{\alpha_j \widetilde{R}_{\ell,i}}, \boxed{c\widetilde{R}'_{\ell,i}}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, 0, \mathbf{0}^{n_2}, \boxed{1}, \boxed{1}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .
\end{aligned}$$

For all key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,17}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,17}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i} c + \langle \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,18}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,18}}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,17}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,17}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta\widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,18}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,18}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,17}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,17}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta\widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,18}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,18}}).
\end{aligned}$$

The indistinguishability follows from the security of the underlying scheme Π_{sip}.

Hybrid $\mathsf{H}_{3,j,19}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,18}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\quad \widetilde{R}_{\ell,i} + \delta\widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \alpha_j \widetilde{R}_{\ell,i}, \boxed{\delta\alpha_j \widetilde{R}'_{\ell,i}}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, 0, \mathbf{0}^{n_2}, 1, 1, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \quad) \ .
\end{aligned}$$

We can show that Hybrid $\mathsf{H}_{3,j,18}$ and Hybrid $\mathsf{H}_{3,j,19}$ are computationally indistinguishable through a DDH reduction similar to the proof of Claim 1.

Hybrid $\mathsf{H}_{3,j,20}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,19}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\widetilde{R}_{\ell,i} + \delta\widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \boxed{\widetilde{R}_{\ell,i} + \delta\widetilde{R}'_{\ell,i}}, \boxed{0}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)}) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\quad x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, 0, \mathbf{0}^{n_2}, \boxed{\alpha_j}, \boxed{0}, \mathbf{x}_{j,\mathsf{priv}}^{(1)}) \ .
\end{aligned}$$

For all key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,19}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,19}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\delta\alpha_j + \langle \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,20}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,20}}).
\end{aligned}$$

For $\iota \in [Q_c]$, $\iota < j$ and $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,19}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,19}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(1)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta\widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,20}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,20}}).
\end{aligned}$$

For all $\iota \in [Q_c]$, $\iota > j$ and $\ell \in [Q_k]$, we also have,

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,19}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,19}}) \\
&= [\![y_{\ell,i} x_{\iota,i}^{(0)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\iota,i}^{(0)} \rangle + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{x}_{\iota,\mathsf{priv}}^{(0)} \rangle + \widetilde{R}_{\ell,i}\alpha_\iota + \delta \widetilde{R}'_{\ell,i}\alpha_\iota]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,20}}, \mathsf{iCT}_{\iota,i}^{\mathsf{H}_{3,j,20}}).
\end{aligned}$$

The hybrids Hybrid $\mathsf{H}_{3,j,19}$ and Hybrid $\mathsf{H}_{3,j,20}$ are indistinguishable from the security of the underlying Π_{sip} scheme.

Hybrid $\mathsf{H}_{3,j,21}$: This hybrid is the same as Hybrid $\mathsf{H}_{3,j,20}$ except the following changes.

$$\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{r}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}) : \quad \mathbf{k}_{\ell,i,\mathsf{fe}} : \quad (\ \widetilde{R}_{\ell,i} + \delta \widetilde{R}'_{\ell,i}, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(0)}, \widetilde{R}_{\ell,i} + \delta \widetilde{R}'_{\ell,i}, 0, \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)} \ ) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_j^{(\beta)}, \mathbf{z}_j^{(\beta)}, \mathbf{x}_{j,\mathsf{priv}}^{(\beta)}\}_{\beta \in \{0,1\}}) : \quad \mathbf{c}_{j,i,\mathsf{fe}} : \quad (\ x_{j,i}^{(1)}, \mathbf{z}_j^{(1)}, \ \alpha_j \ , \mathbf{0}^{n_2}, \ 0 \ , 0, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \ ) \ .
\end{aligned}$$

For all key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,20}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,20}}) \\
&= [\![y_{\ell,i} x_{j,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{j,i}^{(1)} \rangle + \widetilde{R}_{\ell,i}\alpha_j + \widetilde{R}'_{\ell,i}\delta\alpha_j + \langle \mathbf{y}_{\ell,i,\mathsf{priv}}^{(1)}, \mathbf{x}_{j,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\mathsf{H}_{3,j,21}}, \mathsf{iCT}_{j,i}^{\mathsf{H}_{3,j,21}}).
\end{aligned}$$

We can show the indistinguishabilty between Hybrid $\mathsf{H}_{3,j,20}$ and Hybrid $\mathsf{H}_{3,j,21}$ through a reduction to the security of the underlying scheme Π_{sip}.

It holds that $\mathsf{H}_{3,j,21} \approx_s$ Hybrid $\mathsf{H}_{3,j}$ as $\widetilde{R}_{\ell,i}$, $\widetilde{R}'_{\ell,i}$ can be replaced by $\widetilde{r}_{\ell,i}$,$\widetilde{r}'_{\ell,i}$, respectively . This is a statistical change as $\widetilde{R}_{\ell,i}$ and $\widetilde{r}_{\ell,i}$ are statistically close as $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{R}_{\ell,i} = \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,i} = 0$. Similarly, the distributions $\widetilde{R}'_{\ell,i}$ and $\widetilde{r}'_{\ell,i}$ are also statistically close.

This completes the proof of Lemma 4. □

Lemma 6 *Hybrid 4 and Hybrid 5 are computationally indistinguishable if the underlying scheme Π_{sip} is function-hiding.*

The security follows from the fact that for all ciphertext queries $\kappa \in [Q_c]$ and key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\text{Hybrid 4}}, \mathsf{iCT}_{\kappa,i}^{\text{Hybrid 4}}) \\
&= [\![y_{\ell,i} x_{\kappa,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\kappa,i}^{(1)} \rangle + \widetilde{r}_{\ell,i}\alpha_\kappa + \delta \widetilde{r}'_{\ell,i}\alpha_j + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\text{Hybrid 5}}, \mathsf{iCT}_{\kappa,i}^{\text{Hybrid 5}}).
\end{aligned}$$

The reduction follows the same approach as proof of Lemma 2 on indistinguishability between Hybrid 1 and Hybrid 2.

Lemma 7 *Hybrid 5 and Hybrid 6 are computationally indistinguishable if the underlying scheme Π_{sip} is function-hiding.*

The security follows from the security of Π_{sip} scheme. For all ciphertext queries $\kappa \in [Q_c]$ and key queries $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\text{Hybrid 5}}, \mathsf{iCT}_{\kappa,i}^{\text{Hybrid 5}}) \\
&= [\![y_{\ell,i} x_{\kappa,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\kappa,i}^{(1)} \rangle + \widetilde{r}_{\ell,i}\alpha_\kappa + \delta \widetilde{r}'_{\ell,i}\alpha_\kappa + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\text{Hybrid 6}}, \mathsf{iCT}_{\kappa,i}^{\text{Hybrid 6}})
\end{aligned}$$

The reduction is similar to the proof of Lemma 2.

Lemma 8 *Hybrid 6 and Hybrid 7 are statistically indistinguishable.*

Proof. The distributions

$$\{r_{\ell,i} \leftarrow \mathbb{Z}_p : \sum_{i \in I_{\mathbf{y}_\ell}} r_{\ell,i} = 0\} \text{ and } \{\widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i} : \sum_{i \in I_{\mathbf{y}_\ell}} (\widetilde{r}'_{\ell,i} + \delta \widetilde{r}'_{\ell,i}) = 0, \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,k} = \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}'_{\ell,i} = 0\}$$

are statistically close as $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,i} + \delta \widetilde{r}'_{\ell,i} = 0$ and $\{\widetilde{r}_{\ell,i}, \widetilde{r}'_{\ell,i}\}_{i \in I_{\mathbf{y}_\ell}}, \delta$ are uniformly chosen over $\mathbb{Z}_p$ satisfying $\sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}_{\ell,i} = \sum_{i \in I_{\mathbf{y}_\ell}} \widetilde{r}'_{\ell,i} = 0$. □

Lemma 9 *Hybrid 7 and Hybrid 8 are computationally indistinguishable if the underlying scheme* Π_{sip} *is function-hiding.*

Proof. We prove the lemma through a reduction to the underlying IPFE scheme Π_{sip} similar to the proof of Lemma 2. For all $\kappa \in [Q_c]$, $\ell \in [Q_k]$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\text{Hybrid 7}}, \mathsf{iCT}_{\kappa,i}^{\text{Hybrid 7}}) \\
&= [\![y_{\ell,i} x_{\kappa,i}^{(1)} + \langle \mathbf{s}_{\ell,i}, \mathbf{z}_{\kappa,i}^{(1)} \rangle + r_{\ell,i} \alpha_\kappa + \langle \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(1)} \rangle]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,i}^{\text{Hybrid 8}}, \mathsf{iCT}_{\kappa,i}^{\text{Hybrid 8}}).
\end{aligned}$$

Thus, by Lemma 2 to Lemma 9, we show that $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ and $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 1)$ are computationally indistinguishable and our Π_{esi} scheme achieves sel-FH-IND security as per Definition 12. □

This completes the proof of Theorem 6. □

5 Attribute-Based Slotted UIPFE

In this section, we define the notion of *attribute-based slotted unbounded IPFE* (AB-sUIPFE) with the slot-specification $\mathcal{S} = \mathcal{S}_{\mathsf{pub}} \times \mathcal{S}_{\mathsf{priv}}$, where $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^*$ and $\mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{n_2}$ represent the elements in the public and private slots, respectively. The attribute-set space is denoted as $\mathcal{ATT}$, and the access policy space is represented by $\mathcal{P}$. Let $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ be a pairing group (see Definition 1) of prime order p.

Definition 13 An AB-sUIPFE scheme $\Pi_{\mathsf{asi}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$, defined over the slot-specification $\mathcal{S} = \mathcal{S}_{\mathsf{pub}} \times \mathcal{S}_{\mathsf{priv}}$, consists of the following five algorithms:

$\mathsf{Setup}(1^\lambda, 1^{n_2}) \to (\mathsf{MPK}, \mathsf{MSK})$: The setup algorithm takes as input the security parameter λ, and the length n_2 of the $\mathcal{S}_{\mathsf{priv}}$ part. It outputs the master public key MPK and the master secret key MSK.

$\mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}, \mathbf{y}_{\mathsf{priv}})]\!]_2, I_{\mathbf{y}}, \mathbb{A}) \to \mathsf{SK}$: The key generation algorithm takes as input MSK, the slot vector $(\mathbf{y}, \mathbf{y}_{\mathsf{priv}}) \in \mathcal{S}$ in the exponent of the group $\mathbb{G}_2$ with the associated index set $I_{\mathbf{y}}$ of $\mathbf{y}$ and an access structure $\mathbb{A} \in \mathcal{P}$. It outputs a secret key SK.

$\mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{x}_{\mathsf{priv}})]\!]_1, \mathsf{S}) \to \mathsf{CT}$: The encryption algorithm takes as input MSK, the slot vector $(\mathbf{x}, \mathbf{x}_{\mathsf{priv}}) \in \mathcal{S}$ in the exponent of the group $\mathbb{G}_1$, where $\mathbf{x} \in \mathbb{Z}_p^m$ (of arbitrary length) and an attribute set $\mathsf{S} \in \mathcal{ATT}$. It outputs the ciphertext CT.

$\mathsf{SlotEnc}(\mathsf{MPK}, [\![\mathbf{x}]\!]_1, \mathsf{S}) \to \mathsf{CT}$: The slot encryption algorithm takes as input MPK, the public slot vector $\mathbf{x} \in \mathcal{S}_{\mathsf{pub}}$ in the exponent of the group $\mathbb{G}_1$, where $\mathbf{x} \in \mathbb{Z}_p^m$ (of arbitrary length) and an attribute set $\mathsf{S} \in \mathcal{ATT}$. It outputs the ciphertext CT.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{CT}) \to [\![d]\!]_T \vee \perp$: The decryption algorithm takes as input SK, CT and outputs either a decrypted value $[\![d]\!]_T \in \mathbb{G}_T$ or a symbol $\perp$ indicating failure.

Correctness: For all $\lambda \in \mathbb{N}, (\mathbf{x}, \mathbf{x}_{\mathsf{priv}}), (\mathbf{y}, \mathbf{y}_{\mathsf{priv}}) \in \mathcal{S}$ such that $\mathbf{x} \in \mathbb{Z}_p^m, \mathbf{y} \in \mathbb{Z}_p^{|I_{\mathbf{y}}|}$ and $\mathbf{x}_{\mathsf{priv}}, \mathbf{y}_{\mathsf{priv}} \in \mathbb{Z}_p^{n_2}$ with $\mathbb{A}(\mathsf{S}) = 1 \wedge \mathcal{R}(\mathbf{x}, \mathbf{y}) = 1$, we require

$$\Pr\left[[\![d]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle_p + \langle \mathbf{x}_{\mathsf{priv}}, \mathbf{y}_{\mathsf{priv}} \rangle]\!]_T \ : \ \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_2}) \\ \mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}, \mathbf{y}_{\mathsf{priv}})]\!]_2, I_{\mathbf{y}}, \mathbb{A}) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{x}_{\mathsf{priv}})]\!]_1, \mathsf{S}) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Slot-mode correctness: For all $\mathbf{x} \in \mathcal{S}_{\mathsf{pub}}$, the following distributions must be identical:

$$\left\{ (\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) : (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_2}), \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{0}^{n_2})]\!]_1, \mathsf{S}) \right\},$$
$$\left\{ (\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) : (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_2}), \mathsf{CT} \leftarrow \mathsf{SlotEnc}(\mathsf{MPK}, [\![\mathbf{x}]\!]_1, \mathsf{S}) \right\}.$$

Definition 14 (Security of AB-sUIPFE) The $\Pi_{\mathsf{asi}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$ scheme is said to be xx-function-hiding-indistinguishability (xx-FH-IND)-based secure for $\mathsf{xx} \in \{\mathsf{sel}, \mathsf{adp}\}$ if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds

$$\mathsf{Adv}^{\mathsf{asi}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, 0) = 1 \right] - \Pr\left[\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, 1) = 1 \right] \right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\text{xx-FH-IND}}(\lambda, \beta) :}$
1: $n_2 \leftarrow \mathcal{A}(1^\lambda)$.
2: $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{n_2})$.
3: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KG},\beta}(\cdot), \mathcal{O}_{\mathsf{E},\beta}(\cdot)}(\mathsf{MPK})$.
4: output β'.

$\underline{\mathcal{O}_{\mathsf{KG},\beta}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}) :}$
output $\mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}_\ell, \mathbf{y}^{(\beta)}_{\ell,\mathsf{priv}})]\!]_2, I_{\mathbf{y}_\ell}, \mathbb{A})$.

$\underline{\mathcal{O}_{\mathsf{E},\beta}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa) :}$
output $\mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}})]\!]_1, \mathsf{S}_\kappa)$.

Here, $(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A})$ denotes the ℓ-th secret key query and $(\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}, \mathsf{S}_\kappa)_{\beta\in\{0,1\}}$ denotes the κ-th encryption queries where $|\mathbf{x}^{(0)}_\kappa| = |\mathbf{x}^{(1)}_\kappa| = m_\kappa$ (say). Let Q_k, Q_c be the numbers of queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ oracles respectively. Then for all $\ell \in [Q_k], \kappa \in [Q_c]$ with $\mathcal{R}(\mathbf{x}^{(0)}_\kappa, \mathbf{y}_\ell) = \mathcal{R}(\mathbf{x}^{(1)}_\kappa, \mathbf{y}_\ell) = 1$, and $\mathbb{A}(\mathsf{S}_\kappa) = 1$, it must holds that

$$[\![\langle \mathbf{x}^{(0)}_\kappa, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}} \rangle]\!]_T = [\![\langle \mathbf{x}^{(1)}_\kappa, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}^{(1)}_{\kappa,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}} \rangle]\!]_T \ .$$

– If $\mathsf{xx} = \mathsf{sel}$: Queries to $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ must be made before any queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$.
– If $\mathsf{xx} = \mathsf{adp}$: Queries to $\mathcal{O}_{\mathsf{KG},\beta}(\cdot)$, $\mathcal{O}_{\mathsf{E},\beta}(\cdot)$ can be made in any order.

5.1 Construction

Consider $\Pi_{\mathsf{sip}} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ be a bounded sIPFE scheme with $\mathcal{S}'_{\mathsf{pub}} = \mathbb{Z}_p^3$, $\mathcal{S}'_{\mathsf{priv}} = \mathbb{Z}_p^4$ and $\Pi_{\mathsf{esi}} = (\mathsf{eSetup}, \mathsf{eKeyGen}, \mathsf{eEnc}, \mathsf{eSlotEnc}, \mathsf{eDec})$ be an esUIPFE scheme with $\mathcal{S}''_{\mathsf{pub}} = \mathbb{Z}_p^* \times \mathbb{Z}_p$, $\mathcal{S}''_{\mathsf{priv}} = \mathbb{Z}_p^{n_2} \times \mathbb{Z}_p^3 \times \mathbb{Z}_p^{n_2}$. We present an AB-sUIPFE scheme $\Pi_{\mathsf{asi}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{SlotEnc}, \mathsf{Dec})$ with $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^*$, $\mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{n_2}$ for LSSS access structure. We discuss bounded sIPFE and LSSS access structure in Definitions 9 and 3 respectively.

$\mathsf{Setup}(1^\lambda, 1^{n_2})$: The setup algorithms takes as input the security parameter λ, private slot length n_2 and executes the following steps:

1. Generates $(\mathsf{iMPK}, \mathsf{iMSK}) \leftarrow \mathsf{iSetup}(1^\lambda), (\mathsf{eMPK}, \mathsf{eMSK}) \leftarrow \mathsf{eSetup}(1^\lambda, 1^{2n_2+3})$.[1]
2. Outputs the master public key $\mathsf{MPK} = (\mathsf{iMPK}, \mathsf{eMPK})$ and the master secret key $\mathsf{MSK} = (\mathsf{iMSK}, \mathsf{eMSK})$.

[1] Here, we note that $n_1 = 1$.

$\mathsf{KeyGen}(\mathsf{MSK}, [\![(\mathbf{y}, \mathbf{y}_{\mathsf{priv}})]\!]_2, I_{\mathbf{y}}, \mathbb{A})$: The key generation algorithm takes as input MSK, the slot vector $(\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}}, \mathbf{y}_{\mathsf{priv}})$ in the exponent power of $\mathbb{G}_2$, an access structure $\mathbb{A}$ and does the following steps:

1. Samples $a_0 \leftarrow \mathbb{Z}_p$ then use the secret sharing scheme based on $\mathbb{A}$ to create the shares $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ of a_0.
2. Defines the vectors $\mathbf{k}_{\mathsf{ab},j}$ and $\mathbf{k}_{\mathsf{fe}}$ as follows:

$$\begin{array}{lllllllll} \mathbf{k}_{\mathsf{ab},j} & = (& \pi_j(j,1), & a_j \cdot z, & 0, & 0, & 0, & 0 &) \\ \mathbf{k}_{\mathsf{fe}} & = (& \mathbf{y}, & a_0 \cdot z, & \mathbf{y}_{\mathsf{priv}}, & 0, & 0, & \mathbf{0}^{n_2} &) \end{array} \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ ,$$

   where $\pi_j \leftarrow \mathbb{Z}_p$ for all $j \in \mathsf{List\text{-}Att}(\mathbb{A})$ and $z \leftarrow \mathbb{Z}_p$.
3. Generates $\mathsf{iSK}_{\mathsf{ab},j} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\mathbf{k}_{\mathsf{ab},j}]\!]_2), \mathsf{eSK}_{\mathsf{fe}} \leftarrow \mathsf{eKeyGen}(\mathsf{eMSK}, [\![\mathbf{k}_{\mathsf{fe}}]\!]_2, I_{\mathbf{y}})$.
4. Outputs the secret key $\mathsf{SK} = \big(\{\mathsf{iSK}_{\mathsf{ab},j}\}_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}, \mathsf{eSK}_{\mathsf{fe}}\big)$.

$\mathsf{Enc}(\mathsf{MSK}, [\![(\mathbf{x}, \mathbf{x}_{\mathsf{priv}})]\!]_1, \mathsf{S})$: The encryption algorithm takes as input MSK, slot vector $(\mathbf{x} = (x_i)_{i \in [m]}, \mathbf{x}_{\mathsf{priv}})$ in the exponent power of $\mathbb{G}_1$, an attribute set S and performs the following:

1. Defines the vectors $\mathbf{c}_{\mathsf{ab},j}$ and $\mathbf{c}_{\mathsf{fe}}$ as follows:

$$\begin{array}{lllllllll} \mathbf{c}_{\mathsf{ab},j} & = (& \sigma_j(1,-j), & \psi, & 0, & 0, & 0, & 0 &) \\ \mathbf{c}_{\mathsf{fe}} & = (& \mathbf{x}, & \psi, & \mathbf{x}_{\mathsf{priv}}, & 0, & 0, & \mathbf{0}^{n_2} &) \end{array} \quad \forall j \in \mathsf{S} \ ,$$

   where $\sigma_j \leftarrow \mathbb{Z}_p$ for all $j \in \mathsf{S}$ and $\psi \leftarrow \mathbb{Z}_p$.
2. Generates $\mathsf{iCT}_{\mathsf{ab},j} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}, [\![\mathbf{c}_{\mathsf{ab},j}]\!]_1), \mathsf{eCT}_{\mathsf{fe}} \leftarrow \mathsf{eEnc}(\mathsf{eMSK}, [\![\mathbf{c}_{\mathsf{fe}}]\!]_1)$.
3. Outputs the ciphertext $\mathsf{CT} = (\{\mathsf{iCT}_{\mathsf{ab},j}\}_{j \in \mathsf{S}}, \mathsf{eCT}_{\mathsf{fe}})$.

$\mathsf{SlotEnc}(\mathsf{MPK}, [\![\mathbf{x}]\!]_1, \mathsf{S})$: The slot encryption algorithm takes as input MPK, public slot vector $\mathbf{x} = (x_i)_{i \in [m]}$ in the exponent power of $\mathbb{G}_1$, attribute set S and does the following:

1. Defines the vectors $\mathbf{c}_{\mathsf{ab},j}$ and $\mathbf{c}_{\mathsf{fe}}$ as follows:

$$\begin{array}{lllll} \mathbf{c}_{\mathsf{ab},j} & = (& \sigma_j(1,-j), & \psi &) \\ \mathbf{c}_{\mathsf{fe}} & = (& \mathbf{x}, & \psi &) \end{array} \quad \forall j \in \mathsf{S} \ ,$$

   where $\sigma_j \leftarrow \mathbb{Z}_p$ for all $j \in \mathsf{S}$ and $\psi \leftarrow \mathbb{Z}_p$.
2. Generates $\mathsf{iCT}_{\mathsf{ab},j} \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, [\![\mathbf{c}_{\mathsf{ab},j}]\!]_1), \mathsf{eCT}_{\mathsf{fe}} \leftarrow \mathsf{eSlotEnc}(\mathsf{eMPK}, [\![\mathbf{c}_{\mathsf{fe}}]\!]_1)$.
3. Outputs the ciphertext $\mathsf{CT} = (\{\mathsf{iCT}_{\mathsf{ab},j}\}_{j \in \mathsf{S}}, \mathsf{eCT}_{\mathsf{fe}})$.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{CT})$: The decryption algorithm takes as input SK, CT and proceeds as follows:

1. If there exists $A \subseteq \mathsf{S}$ and $A \in \mathbb{A}$, then compute the reconstruction vector $\mathbf{c} = (c_j)_j$ for the LSSS corresponding to A. Next, use the decryption algorithms of Π_{sip} and Π_{esi} to compute the following.

$$[\![\mu_j]\!]_T \leftarrow \mathsf{iDec}(\mathsf{iSK}_{\mathsf{ab},j}, \mathsf{iCT}_{\mathsf{ab},j}) \ \forall j \in A \text{ and}$$
$$[\![\mu]\!]_T = \prod_{j \in A} c_j [\![\mu_j]\!]_T, \quad [\![\nu]\!]_T \leftarrow \mathsf{eDec}(\mathsf{eSK}, \mathsf{eCT})$$

   Finally, it returns $[\![d]\!]_T$ where $[\![d]\!]_T = [\![\nu]\!]_T \cdot ([\![\mu]\!]_T)^{-1}$.
2. Otherwise, it returns $\perp$.

Correctness: From the correctness of Π_{sip} and Π_{esi}, with $\mathcal{R}(\mathbf{x}, \mathbf{y}) = 1$ and $\mathbb{A}(\mathsf{S}) = 1$, using $a_0 = \sum_{j \in A} c_j a_j$, we have

$$\mathsf{iDec}(\mathsf{iSK}_{\mathsf{ab},j}, \mathsf{iCT}_{\mathsf{ab},j}) = [\![\psi z a_j]\!]_T \text{ and } \prod_{j \in A} c_j [\![\psi z a_j]\!]_T = [\![\psi z a_0]\!]_T \ , \tag{1}$$

$$\mathsf{eDec}(\mathsf{eSK}_{\mathsf{fe}}, \mathsf{eCT}_{\mathsf{fe}}) = [\![\langle \mathbf{x}, \mathbf{y} \rangle_p + a_0 z \psi]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle_P + a_0 z \psi]\!]_T \ . \tag{2}$$

From Equations 1 and 2, we compute $[\![d]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle_p]\!]_T$.

Slot-mode Correctness: From the slot-mode correctness of Π_{esi}, we have

$$\mathsf{eEnc}(\mathsf{eMSK}, [\![\mathbf{u}||\mathbf{0}^{2n_2+3}]\!]_1) \equiv \mathsf{eSlotEnc}(\mathsf{eMPK}, [\![\mathbf{u}]\!]_1)$$

for all $\mathbf{u} \in \mathbb{Z}_p^* \times \mathbb{Z}_p$. Thus, we have slot-mode correctness for Π_{asi}.

5.2 Security Analysis

In Theorem 7, we present the security analysis of our AB-sUIPFE scheme, as described in Construction 5.1. We use the following masking lemma for the security analysis of our AB-sUIPFE.

Lemma 10 (Modified Masking Lemma) *Let $\mathbb{A}$ be an LSSS-realizable over a set of attributes $\mathsf{Att} \subseteq \mathbb{Z}_p$. Let $\mathsf{List\text{-}Att}(\mathbb{A})$ be the list of attributes appearing in $\mathbb{A}$ and $|\mathsf{List\text{-}Att}(\mathbb{A})| = P$. Let $S \subseteq \mathsf{Att}$ be a set of attributes. For two random integers $a_0, a_0' \leftarrow \mathbb{Z}_p$, we construct the random labelling $(a_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a_j')_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0'}(\mathbb{A})$. Consider $\Pi_{\mathsf{sip}} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iSlotEnc}, \mathsf{iDec})$ be an bounded sIPFE with slot-specification $\mathcal{S}'_{pub} = \mathbb{Z}_p^3$, $\mathcal{S}'_{priv} = \mathbb{Z}_p^4$ and $\Pi_{\mathsf{esi}} = (\mathsf{eSetup}, \mathsf{eKeyGen}, \mathsf{eEnc}, \mathsf{eSlotEnc}, \mathsf{eDec})$ be an esUIPFE scheme with slot-specification $\mathcal{S}''_{pub} = \mathbb{Z}_p^* \times \mathbb{Z}_p$, $\mathcal{S}''_{priv} = \mathbb{Z}_p^{n_2} \times \mathbb{Z}_p^3 \times \mathbb{Z}_p^{n_2}$. We define the following vectors:*

$$\begin{aligned}
\mathbf{k}_j^\beta &= (\ \ \pi_j(j,1),\ \ 0,\ \ a_j z,\ \ 0,\ \ 0,\ \ \beta \cdot a_j' yz/v_j\ \ ) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^\beta &= (\ \ \sigma_j(1,-j),\ \ 0,\ \ \psi,\ \ 0,\ \ 0,\ \ \beta \cdot \tau v_j x\ \ ) \quad \forall j \in S\ ,\\
\mathbf{k}_{\mathsf{root}}^\beta &= (\ \ \mathbf{0}^\star,\ \ 0,\ \ \mathbf{0}^{n_2},\ \ a_0 z,\ \ 0,\ \ \beta \cdot a_0' yz_i,\ \ \mathbf{0}^{n_2}\ \ )\ ,\\
\mathbf{c}_{\mathsf{root}}^\beta &= (\ \ \mathbf{0}^\star,\ \ 0,\ \ \mathbf{0}^{n_2},\ \ \psi,\ \ 0,\ \ \beta \cdot \tau x,\ \ \mathbf{0}^{n_2}\ \ )\ ,
\end{aligned}$$

where $x, y \in \mathbb{Z}_p$ and $\sigma_j, \pi_j, v_j, \tau, \psi \leftarrow \mathbb{Z}_p$. Here, $\mathbf{0}^\star$ represents an unbounded length vector containing all zeros. Then for all $(\mathsf{iMPK}, \mathsf{iMSK}) \leftarrow \mathsf{iSetup}(1^\lambda)$, $(\mathsf{eMPK}, \mathsf{eMSK}) \leftarrow \mathsf{eSetup}(1^\lambda, 1^{2n_2+3})$ and the DDH assumption holds in $\mathbb{G}_2$, the distributions $\{\mathsf{iSK}_{\mathbf{k}_j^\beta}\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$, $\{\mathsf{iCT}_{\mathbf{c}_j^\beta}\}_{j\in S}$, $\{\mathsf{eSK}_{\mathbf{k}_{\mathsf{root}}^\beta}\}, \{\mathsf{eCT}_{\mathbf{c}_{\mathsf{root}}^\beta}\}$ for $\beta \leftarrow \{0,1\}$ are computationally indistinguishable where

$$\begin{aligned}
\mathsf{iSK}_{\mathbf{k}_j^\beta} &= \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\mathbf{k}_j^\beta]\!]_2) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathsf{iCT}_{\mathbf{c}_j^\beta} &= \mathsf{iEnc}(\mathsf{iMSK}, [\![\mathbf{c}_j^\beta]\!]_1) \quad \forall j \in S\ ,\\
\mathsf{eSK}_{\mathbf{k}_{\mathsf{root}}^\beta} &= \mathsf{eKeyGen}(\mathsf{eMSK}, [\![\mathbf{k}_{\mathsf{root}}^\beta]\!]_2)\ ,\\
\mathsf{eCT}_{\mathbf{c}_{\mathsf{root}}^\beta} &= \mathsf{eEnc}(\mathsf{eMSK}, [\![\mathbf{c}_{\mathsf{root}}^\beta]\!]_1)\ .
\end{aligned}$$

This modified version of the Masking Lemma is adapted to our setting, originally proposed in [38].

Proof. We present the proof of our modified Masking Lemma 10 using the adversaries of Π_{sip} and Π_{esi} schemes. In the following, we propose several hybrids to show the distributions are computationally indistinguishable.

Hybrid 0: The ciphertext and the secret key components are generated for the challenge bit $\beta = 0$ using the following vectors:

$$\begin{aligned}
\mathbf{k}_j^\beta &= (\ \ \pi_j(j,1),\ \ 0,\ \ a_j z,\ \ 0,\ \ 0\ \ 0\ \ )\ \ \ \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^\beta &= (\ \ \sigma_j(1,-j),\ \ 0,\ \ \psi,\ \ 0,\ \ 0\ \ 0\ \ )\ \ \ \forall j \in S\ ,\\
\mathbf{k}_{\mathsf{root}}^\beta &= (\ \ \mathbf{0}^\star,\ \ 0,\ \ \mathbf{0}^{n_2},\ \ a_0 z,\ \ 0,\ \ 0,\ \ \mathbf{0}^{n_2}\ \ )\ ,\\
\mathbf{c}_{\mathsf{root}}^\beta &= (\ \ \mathbf{0}^\star,\ \ 0,\ \ \mathbf{0}^{n_2},\ \ \psi,\ \ 0,\ \ 0,\ \ \mathbf{0}^{n_2}\ \ )\ ,
\end{aligned}$$

where $\mathsf{iSK}_{\mathbf{k}_j^\beta} = \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\mathbf{k}_j^\beta]\!]_2)$; $\mathsf{iCT}_{\mathbf{c}_j^\beta} = \mathsf{iEnc}(\mathsf{iMSK}, [\![\mathbf{c}_j^\beta]\!]_1)$; $\mathsf{eSK}_{\mathbf{k}_{\mathsf{root}}^\beta} = \mathsf{eKeyGen}(\mathsf{eMSK}, [\![\mathbf{k}_{\mathsf{root}}^\beta]\!]_2)$ and $\mathsf{eCT}_{\mathbf{c}_{\mathsf{root}}^\beta} = \mathsf{eEnc}(\mathsf{eMSK}, [\![\mathbf{c}_{\mathsf{root}}^\beta]\!]_1)$.

Hybrid 1: Same as Hybrid 0 except that the secret key vector $\mathbf{k}^{\beta}_{\mathsf{root}}$ is modified as follows:

$$\begin{aligned}\mathbf{k}^{\beta}_{\mathsf{root}} &= (\ \mathbf{0}^{\star},\ 0,\ \mathbf{0}^{n_2},\ (a_0+xyb)z\ ,\ 0,\ 0,\ \mathbf{0}^{n_2}\ )\ ,\\ \mathbf{c}^{\beta}_{\mathsf{root}} &= (\ \mathbf{0}^{\star},\ 0,\ \mathbf{0}^{n_2},\ \psi,\ 0,\ 0,\ \mathbf{0}^{n_2}\ )\ ,\end{aligned}$$

where $a_0, z, b \leftarrow \mathbb{Z}_p$ and $x, y \in \mathbb{Z}_p$. Due to choice of a_0, b and z uniformly chosen from $\mathbb{Z}_p$, Hybrid 0 and Hybrid 1 are statistically indistinguishable.

Hybrid 2: Same as Hybrid 1 except that the secret key vector $\mathbf{k}^{\beta}_{j}$ is modified as follows:

$$\begin{aligned}\mathbf{k}^{\beta}_{j} &= (\ \pi_j(j,1),\ 0,\ (a_j+xybd_j)z\ ,\ 0,\ 0,\ 0\ ) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\ \mathbf{c}^{\beta}_{j} &= (\ \sigma_j(1,-j),\ 0,\ \psi,\ 0,\ 0,\ 0\ ) \quad \forall j \in \mathsf{S}\ ,\end{aligned}$$

where $(a_j)_j \leftarrow \Lambda_{a_0}(\mathbb{A}), (d_j)_j \leftarrow \Lambda_1(\mathbb{A}), b \leftarrow \mathbb{Z}_p$ and $x, y \in \mathbb{Z}_p$. The statistical indistinguishability between the hybrids follows from Claim 4.

Claim 4 Let $(b_j)_j \leftarrow \Lambda_{b_0}(\mathbb{A}), (d_j)_j \leftarrow \Lambda_1(\mathbb{A})$. Then for $x, y \in \mathbb{Z}_p$, $b \leftarrow \mathbb{Z}_p$, and $a_0 = b_0 + xyb$, the following holds

$$\mathcal{D}\left((a_j)_j, a_0\right) \approx_s \mathcal{D}\left((b_j + xybd_j)_j, b_0 + xyb\right) \text{ where } (a_j)_j \leftarrow \Lambda_{a_0}(\mathbb{A}).$$

Proof. By the linearity property of the linear secret sharing scheme,

$$\begin{aligned}(b_j + xybd_j)_j &\equiv (b_j)_j + (xybd_j)_j \equiv (b_j)_j + xyb(d_j)_j \equiv (b_j)_j + xyb(d_j)_j\\ &\approx_s \Lambda_{b_0}(\mathbb{A}) + xyb\Lambda_1(\mathbb{A}) \equiv \Lambda_{b_0}(\mathbb{A}) + \Lambda_{xyb}(\mathbb{A}) \equiv \Lambda_{b_0+xyb}(\mathbb{A})\end{aligned}$$

Thus, $(b_j + xybd_j)_j \approx_s \Lambda_{b_0+xyb}$ and from the claim, we know that $(a_j)_j \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $a_0 = b_0 + xyb$. Thus, then $(\Lambda_{a_0}(\mathbb{A}), a_0) \equiv (\Lambda_{b_0+xyb}(\mathbb{A}), b_0 + xyb) \approx_s ((b_j + xybd_j)_j, b_0 + xyb)$. □

Hybrid 3: This hybrid is the same as Hybrid 2 except that all the secret key and the ciphertext vectors are modified as follows:

$$\begin{aligned}\mathbf{k}^{\beta}_{j} &= (\ \pi_j(j,1),\ 0,\ a_jz\ ,\ 0,\ ybd_jz\ ,\ 0\ ) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\ \mathbf{c}^{\beta}_{j} &= (\ \sigma_j(1,-j),\ 0,\ \psi,\ 0,\ \psi x\ ,\ 0\ ) \quad \forall j \in \mathsf{S}\ ,\\ \mathbf{k}^{\beta}_{\mathsf{root}} &= (\ \mathbf{0}^{\star},\ 0,\ \mathbf{0}^{n_2},\ a_0z\ ,\ 0,\ ybz\ ,\ \mathbf{0}^{n_2}\ )\ ,\\ \mathbf{c}^{\beta}_{\mathsf{root}} &= (\ \mathbf{0}^{\star},\ 0,\ \mathbf{0}^{n_2},\ \psi,\ 0,\ \psi x\ ,\ \mathbf{0}^{n_2}\ )\ .\end{aligned}$$

The indistinguishability follows from the function-hiding property of the underlying Π_{sip} and Π_{esi} schemes.

Hybrid 4: Same as Hybrid 3 except that all the secret key and the ciphertext vectors are modified as follows:

$$\begin{aligned}\mathbf{k}^{\beta}_{j} &= (\ \pi_j(j,1),\ 0,\ \psi a_jz\ ,\ 0,\ \psi ybd_jz\ ,\ 0\ ) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\ \mathbf{c}^{\beta}_{j} &= (\ \sigma_j(1,-j),\ 0,\ 1\ ,\ 0,\ x\ ,\ 0\ ) \quad \forall j \in \mathsf{S}\ ,\\ \mathbf{k}^{\beta}_{\mathsf{root}} &= (\ \mathbf{0}^{\star},\ 0,\ \mathbf{0}^{n_2},\ \psi a_0z\ ,\ 0,\ \psi ybz\ ,\ \mathbf{0}^{n_2}\ )\ ,\\ \mathbf{c}^{\beta}_{\mathsf{root}} &= (\ \mathbf{0}^{\star},\ 0,\ \mathbf{0}^{n_2},\ 1\ ,\ 0,\ x\ ,\ \mathbf{0}^{n_2}\ )\ .\end{aligned}$$

The indistinguishability follows from the function-hiding property of the underlying Π_{sip} and Π_{esi} schemes.

Hybrid 5: Same as Hybrid 4 except that all the secret key vectors are modified as follows:

$$\begin{aligned}
\mathbf{k}_j^{\beta} &= (\quad \pi_j(j,1),\quad 0,\quad \psi a_j z,\quad 0,\quad hyd_j z,\quad 0\quad) \qquad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{k}_{\mathsf{root}}^{\beta} &= (\quad \mathbf{0}^{\star},\quad 0,\quad \mathbf{0}^{n_2},\quad \psi a_0 z,\quad 0,\quad hyz,\quad \mathbf{0}^{n_2}\quad)\ ,
\end{aligned}$$

where $h \leftarrow \mathbb{Z}_p$. The indistinguishability follows from Claim 5.

Claim 5 Hybrid 4 and Hybrid 5 are computationally indistinguishable if the DDH assumption holds over the group $\mathbb{G}_2$.

Proof. Let us assume that the challenger obtains an instance $(\mathbb{G}_2, [\![\psi]\!]_2, [\![b]\!]_2, [\![u_{\mathfrak{b}}]\!]_2)$ of DDH assumption over the group $\mathbb{G}_2$ where

$$u_{\mathfrak{b}} = \begin{cases} \psi b & \text{if } \mathfrak{b} = 0,\\ h \leftarrow \mathbb{Z}_p & \text{if } \mathfrak{b} = 1. \end{cases}$$

The challenger uses the DDH instance to traverse from Hybrid 4 to Hybrid 5.

Using the instances, the reduction sample π_j, z from $\mathbb{Z}_p$ and generates the secret keys components $\mathbf{k}_j^{\mathfrak{b}}$ and $\mathbf{k}_{\mathsf{root}}^{\mathfrak{b}}$ as follows:

$$\begin{aligned}
\mathbf{k}_j^{\mathfrak{b}} &= (\quad \pi_j(j,1),\quad 0,\quad \psi a_j z,\quad 0,\quad u_{\mathfrak{b}} \cdot yd_j z,\quad 0\quad) \qquad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{k}_{\mathsf{root}}^{\mathfrak{b}} &= (\quad \mathbf{0}^{\star},\quad 0,\quad \mathbf{0}^{n_2},\quad \psi a_0 z,\quad 0,\quad u_{\beta} yz,\quad \mathbf{0}^{n_2}\quad)\ .
\end{aligned}$$

To generate the ciphertext components, the reduction samples $\sigma_j \leftarrow \mathbb{Z}_p$, $x \in \mathbb{Z}_p$ and compute

$$\begin{aligned}
\mathbf{c}_j^{\mathfrak{b}} &= (\quad \sigma_j(1,-j),\quad 0,\quad 1,\quad 0,\quad x,\quad 0\quad) \qquad \forall j \in \mathsf{S}\ ,\\
\mathbf{c}_{\mathsf{root}}^{\mathfrak{b}} &= (\quad \mathbf{0}^{\star},\quad 0,\quad \mathbf{0}^{n_2},\quad 1,\quad 0,\quad x,\quad \mathbf{0}^{n_2}\quad)\ .
\end{aligned}$$

According to the DDH assumption, we have

$$(\mathbb{G}_2, [\![\psi]\!]_2, [\![b]\!]_2, [\![\psi b]\!]_2) \approx_c (\mathbb{G}_2, [\![\psi]\!]_2, [\![b]\!]_2, [\![h]\!]_2).$$

If $\mathfrak{b} = 0$, $u_{\beta} = [\![\psi b]\!]_2$, then the adversarial view is the same as Hybrid 4; otherwise for $\mathfrak{b} = 1$, $u_{\mathfrak{b}}$ is randomly chosen from the group $\mathbb{G}_2$ and hence the adversarial view is similar to Hybrid 5. Thus, we have Hybrid 4 $\approx_c$ Hybrid 5 via the DDH assumption. □

Hybrid 6: Same as Hybrid 5 except that all the secret key vectors are modified as follows:

$$\begin{aligned}
\mathbf{k}_j^{\beta} &= (\quad \pi_j(j,1),\quad 0,\quad \psi a_j z,\quad 0,\quad (\widehat{h} + b\psi) yd_j z,\quad 0\quad) \qquad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{k}_{\mathsf{root}}^{\beta} &= (\quad \mathbf{0}^{\star},\quad 0,\quad \mathbf{0}^{n_2},\quad \psi a_0 z,\quad 0,\quad (\widehat{h} + b\psi) yz,\quad \mathbf{0}^{n_2}\quad)\ ,
\end{aligned}$$

where $h = \widehat{h} + b\psi$. As h, b and ψ were uniformly chosen from $\mathbb{Z}_p$, it can be concluded that $\widehat{h} \leftarrow \mathbb{Z}_p$. Thus, Hybrid 5 and Hybrid 6 are statistically indistinguishable.

Hybrid 7: Same as Hybrid 6 except that all the secret key and the ciphertext vectors are modified as follows:

$$\begin{aligned}
\mathbf{k}_j^{\beta} &= (\quad \pi_j(j,1),\quad 0,\quad (a_j + xybd_j)z,\quad 0,\quad \widehat{h} yd_j z,\quad 0\quad) \qquad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^{\beta} &= (\quad \sigma_j(1,-j),\quad 0,\quad \psi,\quad 0,\quad x,\quad 0\quad) \qquad \forall j \in \mathsf{S}\ ,\\
\mathbf{k}_{\mathsf{root}}^{\beta} &= (\quad \mathbf{0}^{\star},\quad 0,\quad \mathbf{0}^{n_2},\quad (a_0 + xyb)z,\quad 0,\quad \widehat{h} yz,\quad \mathbf{0}^{n_2}\quad)\ ,\\
\mathbf{c}_{\mathsf{root}}^{\beta} &= (\quad \mathbf{0}^{\star},\quad 0,\quad \mathbf{0}^{n_2},\quad \psi,\quad 0,\quad x,\quad \mathbf{0}^{n_2}\quad)\ .
\end{aligned}$$

The indistinguishability follows from the function-hiding property of the underlying Π_{sip} and Π_{esi} schemes.

Hybrid 8: Same as Hybrid 7 except that all the secret key and the ciphertext vectors are modified as follows:

$$\begin{aligned}
\mathbf{k}_j^\beta &= (\ \pi_j(j,1),\ 0,\ a_j z,\ 0,\ a'_0 y d_j z,\ 0\ ) && \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^\beta &= (\ \sigma_j(1,-j),\ 0,\ \psi,\ 0,\ \tau x,\ 0\ ) && \forall j \in \mathsf{S}\ ,\\
\mathbf{k}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ a_0 z,\ 0,\ a'_0 y z,\ \mathbf{0}^{n_2}\ )\ ,\\
\mathbf{c}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ \psi,\ 0,\ \tau x,\ \mathbf{0}^{n_2}\ )\ ,
\end{aligned}$$

where $a_j = a_j + xybd_j, \widehat{h} = a'_0\tau$ such that a'_0 and τ are uniformly chosen from $\mathbb{Z}_p$. The statistical indistinguishability follows from the uniformly choice of a'_0 and τ from $\mathbb{Z}_p$.

Hybrid 9: Same as Hybrid 8 except that the ciphertext vector $\mathbf{c}_j^\beta$ is modified as follows:

$$\begin{aligned}
\mathbf{k}_j^\beta &= (\ \pi_j(j,1),\ 0,\ a_j z,\ 0,\ a'_0 y d_j z,\ 0\ ) && \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^\beta &= (\ \sigma_j(1,-j),\ 0,\ \psi,\ 0,\ \tau x,\ \tau x v_j\ ) && \forall j \in \mathsf{S}\ ,\\
\mathbf{k}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ a_0 z,\ 0,\ a'_0 y z,\ \mathbf{0}^{n_2}\ )\ ,\\
\mathbf{c}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ \psi,\ 0,\ \tau x,\ \mathbf{0}^{n_2}\ )\ .
\end{aligned}$$

The indistinguishability follows from the function-hiding property of the underlying Π_{sip} scheme.

Hybrid 10: Same as Hybrid 9 except that all the secret key and the ciphertext vectors are modified as follows:

$$\begin{aligned}
\mathbf{k}_j^\beta &= (\ \pi_j(j,1),\ 0,\ a_j z,\ 0,\ 0,\ a'_0 y d_j z/v_j\ ) && \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^\beta &= (\ \sigma_j(1,-j),\ 0,\ \psi,\ 0,\ \tau x,\ \tau x v_j\ ) && \forall j \in \mathsf{S}\ ,\\
\mathbf{k}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ a_0 z,\ 0,\ a'_0 y z,\ \mathbf{0}^{n_2}\ )\ ,\\
\mathbf{c}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ \psi,\ 0,\ \tau x,\ \mathbf{0}^{n_2}\ )\ .
\end{aligned}$$

The indistinguishability follows from the function-hiding property of the underlying Π_{sip} scheme and Lemma 1.

Hybrid 11: Same as Hybrid 10 except that the secret key vectors $\mathbf{k}_j^\beta$ are modified as follows:

$$\begin{aligned}
\mathbf{k}_j^\beta &= (\ \pi_j(j,1),\ 0,\ a_j z,\ 0,\ 0,\ a'_j y z/v_j\ ) && \forall j \in \mathsf{List\text{-}Att}(\mathbb{A})\ ,\\
\mathbf{c}_j^\beta &= (\ \sigma_j(1,-j),\ 0,\ \psi,\ 0,\ \tau x,\ \tau x v_j\ ) && \forall j \in \mathsf{S}\ ,\\
\mathbf{k}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ a_0 z,\ 0,\ a'_0 y z,\ \mathbf{0}^{n_2}\ )\ ,\\
\mathbf{c}_{\mathsf{root}}^\beta &= (\ \mathbf{0}^\star,\ 0,\ \mathbf{0}^{n_2},\ \psi,\ 0,\ \tau x,\ \mathbf{0}^{n_2}\ )\ ,
\end{aligned}$$

where $a'_j = a'_0 d_j$. The statistical indistinguishability follows as $(a'_j)_j \leftarrow \Lambda_{a'_0}(\mathbb{A})$ and $(d_j)_j \leftarrow \Lambda_1(\mathbb{A})$. This hybrid is the same as the distribution corresponding to the challenge bit $\beta = 1$. □

Theorem 7 *Our Π_{asi} scheme achieves sel-FH-IND security as per Definition 14 if the underlying schemes Π_{sip} and Π_{esi} are sel-FH-IND secure as per Definitions 10 and 12, respectively.*

Proof. We prove Theorem 7 through a sequence of hybrids. We represent the number of encryption and key generation queries by Q_c and Q_k, respectively. We briefly provide indistinguishability arguments of security hybrids in Fig. 3.

Hybrid 0. Same as the experiment $\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ where the adversary can query the following oracles. We represent the slots using dashed boxes, updated in the following hybrids. In the subsequent hybrids,

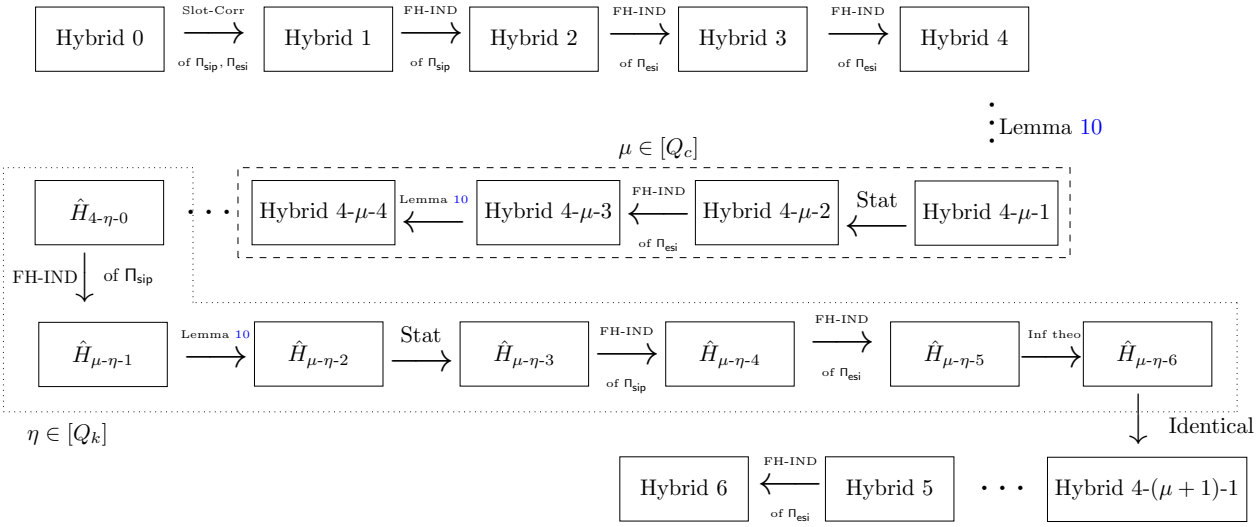

Figure 3: Outline of the security games for Theorem 7. Here, 'Stat' means statistically, 'Inf theo' is a shorthand for information-theoretically, 'Slot-Corr' is a shorthand for slot mode correctness and 'FH-IND' is a shorthand for function-hiding indistinguishability security.

we will only refer to the updated slots.

Encryption queries: On receiving the encryption queries of the form $(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa)$ with the attribute set $\mathsf{S}_\kappa \in \mathcal{ATT}$ and challenge vectors $(\mathbf{x}_\kappa^{(0)}, \mathbf{x}_\kappa^{(1)})$ of length m_κ with $(\mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(1)})$ of length n_2 from the adversary $\mathcal{A}$, the challenger computes the vectors $\mathbf{c}_{\kappa,\mathsf{ab},j}$ for $j \in \mathsf{S}_\kappa$ and vector $\mathbf{c}_{\kappa,\mathsf{fe}}$ and simulates the ciphertexts as follows:

$$\begin{aligned}
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{ab},j} = \quad & (\quad \sigma_{\kappa,j}(1,-j), \quad \boxed{\psi_\kappa,} \quad \boxed{\perp,} \quad \boxed{\perp,} \quad \boxed{\perp,} \quad \boxed{\perp} \quad) \quad \forall j \in \mathsf{S}_\kappa \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{fe}} = \quad & (\quad \boxed{\mathbf{x}_\kappa^{(0)},} \quad \boxed{\psi_\kappa,} \quad \boxed{\mathbf{x}_{\kappa,\mathsf{priv}}^{(0)},} \quad \boxed{\perp,} \quad \boxed{\perp,} \quad \boxed{\perp,} \quad \boxed{\perp} \quad) \ ,
\end{aligned}$$

where $\psi_\kappa, \sigma_{\kappa,j} \leftarrow \mathbb{Z}_p$.

Key Generation queries: On receiving ℓ-th functional query with access structure $\mathbb{A}$, and key vectors $(\mathbf{y}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)})$, the challenger samples $a_{\ell,0} \leftarrow \mathbb{Z}_p$, generate shares $(a_{\ell,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_{\ell,0}}(\mathbb{A})$ and computes the vectors $\mathbf{k}_{\ell,\mathsf{ab},j}$ for $j \in \mathsf{List\text{-}Att}(\mathbb{A})$ and $\mathbf{k}_{\ell,\mathsf{fe}}$ as follows:

$$\begin{aligned}
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{ab},j} = \quad & (\quad \pi_{\ell,j}(j,1), \quad a_{\ell,j}\cdot z_\ell, \quad \underbrace{\boxed{0,}}_{j-1} \quad \boxed{0,} \quad \boxed{0,} \quad \boxed{0} \quad) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ , \\
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{fe}} = \quad & (\quad \mathbf{y}_\ell, \quad a_{\ell,0}\cdot z_\ell, \quad \boxed{\mathbf{y}_{\ell,\mathsf{priv}}^{(0)},} \quad \boxed{0,} \quad \boxed{0,} \quad \boxed{0,} \quad \boxed{\mathbf{0}^{n_2}} \quad) \ ,
\end{aligned}$$

where $z_\ell, \pi_{\ell,j} \leftarrow \mathbb{Z}_p$.

Hybrid 1. For all $\kappa \in [Q_c]$, the vectors $\mathbf{c}_{\kappa,\mathsf{ab},j} \forall j \in \mathsf{S}_\kappa$ and $\mathbf{c}_{\kappa,\mathsf{fe}}$ are modified as follows.

$$\begin{aligned}
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{ab},j}: \quad & (\quad \psi_\kappa, \quad \boxed{0}, \quad \boxed{0}, \quad \boxed{0}, \quad \boxed{0} \quad) \quad \forall j \in \mathsf{S}_\kappa \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}_\kappa^{(\beta)}, \mathbf{x}_{\kappa,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{fe}}: \quad & (\quad \mathbf{x}_\kappa^{(0)}, \quad \psi_\kappa, \quad \mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}, \quad \boxed{0}, \quad \boxed{0}, \quad \boxed{0}, \quad \boxed{\mathbf{0}^{n_2}} \quad) \ .
\end{aligned}$$

The slot-mode correctness of the underlying schemes Π_{esi} and Π_{sip} guarantees that Hybrid 0 and Hybrid 1 are identically distributed.

Hybrid 2. The vectors $\mathbf{k}_{\ell,\mathsf{ab},j}, \mathbf{c}_{\kappa,\mathsf{ab},j}$ are modified for all $\ell \in [Q_k]$ and $\kappa \in [Q_c]$ as follows.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{ab},j}: \quad (\quad a_{\ell,j} z_\ell, \quad 0, \quad 0, \quad 0 \quad) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ ,$$
$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{ab},j}: \quad (\quad 0, \quad \psi_\kappa, \quad 0, \quad 0, \quad 0 \quad) \quad \forall j \in \mathsf{S}_\kappa \ .$$

Hybrid 3. For all $\ell \in [Q_k]$ and $\kappa \in [Q_c]$, the vectors $\mathbf{k}_{\ell,\mathsf{fe}}, \mathbf{c}_{\kappa,\mathsf{fe}}$ are modified as follows.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{fe}}: \quad (\quad \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \quad a_{\ell,0} \cdot z_\ell, \quad 0, \quad 0, \quad \mathbf{0}^{n_2} \quad) \ ,$$
$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{fe}}: \quad (\quad \mathbf{x}^{(0)}_\kappa, \quad 0, \quad \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}, \quad \psi_\kappa, \quad 0, \quad 0, \quad \mathbf{0}^{n_2} \quad) \ .$$

Hybrid 4. This hybrid is similar to Hybrid 3 except that the vectors $\mathbf{k}_{\ell,\mathsf{fe}}$ for all $\ell \in [Q_k]$ are modified as follows.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{fe}}: \quad (\quad \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \quad a_{\ell,0} \cdot z_\ell, \quad 0, \quad 0, \quad \mathbf{y}^{(1)}_{\ell,\mathsf{priv}} \quad) \ .$$

Hybrid 5. For all $\kappa \in [Q_c]$, we modify the following vectors.

$$\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{fe}}: \quad (\quad \mathbf{x}^{(1)}_\kappa, \quad 0, \quad \mathbf{0}^{n_2}, \quad \psi_\kappa, \quad 0, \quad 0, \quad \mathbf{x}^{(1)}_{\kappa,\mathsf{priv}} \quad) \ .$$

Hybrid 6. For all $\ell \in [Q_k]$ and $\kappa \in [Q_c]$, the vectors $\mathbf{k}_{\ell,\mathsf{fe}}, \mathbf{c}_{\kappa,\mathsf{fe}}$ are modified as follows:

$$\mathcal{O}_{\mathsf{KG},1}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{fe}}: \quad (\quad \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, \quad a_{\ell,0} \cdot z_\ell, \quad 0, \quad 0, \quad \mathbf{y}^{(0)}_{\ell,\mathsf{priv}} \quad) \ ,$$
$$\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}^{(\beta)}_\kappa, \mathbf{x}^{(\beta)}_{\kappa,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\kappa): \quad \mathbf{c}_{\kappa,\mathsf{fe}}: \quad (\quad \mathbf{x}^{(1)}_\kappa, \quad 0, \quad \mathbf{x}^{(1)}_{\kappa,\mathsf{priv}}, \quad \psi_\kappa, \quad 0, \quad 0, \quad \mathbf{0}^{n_2} \quad) \ .$$

Now, we can go back to $\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\mathsf{sel\text{-}IND}}(\lambda, 1)$ similar to the transformation from Hybrid 0 to Hybrid 4.

Lemma 11 *Hybrid 1 and Hybrid 2 are computationally indistinguishable if the underlying scheme Π_{sip} is function-hiding.*

Proof. We prove the above lemma by contradiction. Consider a PPT adversary $\mathcal{A}$ that can distinguish between the hybrids. We can use $\mathcal{A}$ to construct $\mathcal{B}$ that can break the sel-FH-IND security of Π_{sip}. On receiving key generation queries and encryption queries from $\mathcal{A}$, $\mathcal{B}$ generates the vectors eSK and eCT on their own as they have access to master secret key eMSK of the Π_{esi} scheme. $\mathcal{B}$ computes the rest of the vectors $\{\mathsf{iSK}_{\ell,\mathsf{ab},j}\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ for all $\ell \in [Q_k]$ by forwarding to the Π_{sip} challenger. For all $\ell \in [Q_k]$, $\kappa \in [Q_c]$ the vectors $\{\mathsf{iSK}_{\ell,\mathsf{ab},j}\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ and $\{\mathsf{iCT}_{\kappa,\mathsf{ab},i}\}_{i\in\mathsf{S}_\kappa}$ when the challenger samples $b = 0$ are computed as follows:

$$\mathsf{iSK}^{\text{Hybrid 1}}_{\ell,\mathsf{ab},j} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(\pi_{\ell,j}(j,1), a_{\ell,j} \cdot z_\ell, 0, 0, 0, 0)]\!]_2) = \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\widetilde{\mathbf{y}}^{(0)}_{\ell,j}]\!]_2) \text{ and}$$
$$\mathsf{iCT}^{\text{Hybrid 1}}_{\kappa,\mathsf{ab},i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}, [\![(\sigma_{\kappa,i}(1,-i), \psi_\kappa, 0, 0, 0, 0)]\!]_1) = \mathsf{iEnc}(\mathsf{iMSK}, [\![\widetilde{\mathbf{x}}^{(0)}_{\kappa,i}]\!]_1).$$

In the case of challenger sampling $b = 1$, the vectors are computed as follows:

$$\mathsf{iSK}^{\text{Hybrid 2}}_{\ell,\mathsf{ab},j} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, [\![(\pi_{\ell,j}(j,1), a_{\ell,j} \cdot z_\ell, a_{\ell,j} \cdot z_\ell, 0, 0, 0)]\!]_2) = \mathsf{iKeyGen}(\mathsf{iMSK}, [\![\widetilde{\mathbf{y}}^{(1)}_{\ell,j}]\!]_2) \text{ and}$$
$$\mathsf{iCT}^{\text{Hybrid 2}}_{\kappa,\mathsf{ab},i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}, [\![(\sigma_{\kappa,i}(1,-i), 0, \psi_k, 0, 0, 0)]\!]_1) = \mathsf{iEnc}(\mathsf{iMSK}, [\![\widetilde{\mathbf{x}}^{(1)}_{\kappa,i}]\!]_1).$$

For all queries $i \in \mathsf{S}_\kappa$, $j \in \mathsf{List\text{-}Att}(\mathbb{A})$, we have

$$\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,\mathsf{ab},j}^{\text{Hybrid 1}}, \mathsf{iCT}_{\kappa,\mathsf{ab},i}^{\text{Hybrid 1}}) \\
&= [\![\pi_{\ell,j} \cdot j \cdot \sigma_{\kappa,i} + \pi_{\ell,j} \cdot \sigma_{\kappa,i} \cdot -i + \psi_\kappa \cdot a_{\ell,j} \cdot z_\ell]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,\mathsf{ab},j}^{\text{Hybrid 2}}, \mathsf{iCT}_{\kappa,\mathsf{ab},i}^{\text{Hybrid 2}}).
\end{aligned}$$

Thus, $\mathcal{B}$ is an admissible adversary for the security of the Π_{sip} scheme. Thus, the advantage of $\mathcal{A}$ in distinguishing between Hybrid 1 and Hybrid 2 is the same as the advantage in distinguishing between the experiments $\mathsf{Expt}^{\mathsf{sip}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ and $\mathsf{Expt}^{\mathsf{sip}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 1)$. □

Lemma 12 *Hybrid 2 and Hybrid 3 are computationally indistinguishable if the underlying scheme Π_{esi} is function-hiding.*

Proof. We consider a PPT adversary $\mathcal{A}$ that can distinguish between the hybrids. We use $\mathcal{A}$ to construct $\mathcal{B}$ against the selective security of the underlying Π_{esi} scheme. In particular, if an adversary $\mathcal{A}$ can distinguish the hybrids, a PPT adversary $\mathcal{B}$ exists that can break the selective function-hiding security of the Π_{esi} scheme.

For the encryption queries from $\mathcal{A}$, $\mathcal{B}$ generates $\{\mathsf{iCT}_{\kappa,\mathsf{ab},j}\}_{j\in\mathsf{S}_\kappa}$ on their own as it has access to iMSK. The vectors eCT_κ for $\kappa \in [Q_c]$ are computed by querying the challenger.

$$\begin{aligned}
&\mathsf{eCT}_\kappa^{\text{Hybrid 2}} \leftarrow \mathsf{eEnc}(\mathsf{eMSK}, [\![(\mathbf{x}_\kappa^{(0)}, \psi_\kappa, \mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}, 0, 0, 0, \mathbf{0}^{n_2})]\!]_1) = \mathsf{eEnc}(\mathsf{eMSK}, [\![\widetilde{\mathbf{x}}_\kappa^{(0)}]\!]_1) \text{ and} \\
&\mathsf{eCT}_\kappa^{\text{Hybrid 3}} \leftarrow \mathsf{eEnc}(\mathsf{eMSK}, [\![(\mathbf{x}_\kappa^{(0)}, 0, \mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}, \psi_\kappa, 0, 0, \mathbf{0}^{n_2})]\!]_1) = \mathsf{eEnc}(\mathsf{eMSK}, [\![\widetilde{\mathbf{x}}_\kappa^{(1)}]\!]_1).
\end{aligned}$$

The algorithm $\mathcal{B}$ computes $\{\mathsf{iSK}_{\ell,\mathsf{ab},j}\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ themselves from iMSK. Then, $\mathcal{B}$ computes the key vectors eSK_ℓ by forwarding it to the challenger.

$$\begin{aligned}
&\mathsf{eSK}_\ell^{\text{Hybrid 2}} \leftarrow \mathsf{eKeyGen}(\mathsf{eMSK}, [\![(\mathbf{y}_\ell, a_{\ell,0} \cdot z_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} 0, 0, 0, \mathbf{0}^{n_2})]\!]_2) = \mathsf{eKeyGen}(\mathsf{eMSK}, [\![\widetilde{\mathbf{y}}_\ell^{(0)}]\!]_2) \text{ and} \\
&\mathsf{eSK}_\ell^{\text{Hybrid 3}} \leftarrow \mathsf{eKeyGen}(\mathsf{eMSK}, [\![(\mathbf{y}_\ell, a_{\ell,0} \cdot z_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, a_{\ell,0} \cdot z_\ell, 0, 0, \mathbf{0}^{n_2})]\!]_2) = \mathsf{eKeyGen}(\mathsf{eMSK}, [\![\widetilde{\mathbf{y}}_\ell^{(1)}]\!]_2).
\end{aligned}$$

We know that for $I_{\mathbf{y}_\ell} \subseteq [m_\kappa]$,

$$\begin{aligned}
&\mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 2}}, \mathsf{eCT}_\kappa^{\text{Hybrid 2}}) \\
&= [\![\langle \mathbf{x}_\kappa^{(0)}, \mathbf{y}_\ell \rangle_p + \psi_\kappa \cdot a_{\ell,0} \cdot z_\ell + \langle \mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle]\!]_T \\
&= \mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 3}}, \mathsf{eCT}_\kappa^{\text{Hybrid 3}}).
\end{aligned}$$

Therefore, $\mathcal{B}$ is an admissible adversary for Π_{esi} scheme. Thus, the advantage of $\mathcal{A}$ in distinguishing between Hybrid 2 and Hybrid 3 is the same as the advantage in distinguishing between the experiments $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ and $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 1)$. □

Lemma 13 *Hybrid 3 and Hybrid 4 are computationally indistinguishable if the underlying scheme Π_{esi} is function-hiding.*

The proof proceeds the same way as that of Lemma 12.

Lemma 14 *Hybrid 4 and Hybrid 5 are computationally indistinguishable if the underlying schemes $\Pi_{\mathsf{sip}}, \Pi_{\mathsf{esi}}$ are function-hiding.*

Proof. We prove the lemma 14 through a sequence of hybrids. We define the hybrids for every $\mu \in [Q_c]$ below. The Hybrid 4-0-4 ≡ Hybrid 4 and Hybrid 4-Q_c-4 ≡ Hybrid 5.

Hybrid 4-0-4. We can observe that Hybrid 4-0-4 is the same as Hybrid 4. We provide descriptions of the oracle below. We represent the slots using dashed boxes, updated in the subsequent hybrids to prove the indistinguishability between Hybrid 4 and Hybrid 5. In the sub-hybrids, we will only mention the updated slots.

$$
\begin{array}{lllllllllll}
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{ab},j} = & (& \pi_{\ell,j}(j,1), & a_{\ell,j}\cdot z_\ell, & a_{\ell,j} z_\ell, & 0, & \boxed{0}, & 0, &) & \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): & \mathbf{c}_{\mu,\mathsf{ab},j} = & (& \sigma_{\mu,j}(1,-j), & 0, & \psi_\mu, & 0, & \boxed{0}, & 0, &) & \forall j \in \mathsf{S}_\mu \ , \\
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{fe}} = & (& \mathbf{y}_\ell, & a_{\ell,0}\cdot z_\ell, & \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, & a_{\ell,0}\cdot z_\ell, & \boxed{0}, & 0, & \mathbf{y}^{(1)}_{\ell,\mathsf{priv}} &) \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): & \mathbf{c}_{\mu,\mathsf{fe}} = & (& \boxed{\mathbf{x}^{(0)}_\mu}, & 0, & \boxed{\mathbf{x}^{(0)}_{\mu,\mathsf{priv}}}, & \psi_\mu, & \boxed{0}, & 0, & \boxed{\mathbf{0}^{n_2}} &) \ .
\end{array}
$$

Hybrid 4-μ-1. This hybrid is similar to Hybrid 4-(μ-1)-4 except the vectors $\mathbf{c}_{\mu,\mathsf{ab},j}, \mathbf{c}_{\mu,\mathsf{fe}}, \mathbf{k}_{\ell,\mathsf{ab},j}, \mathbf{k}_{\ell,\mathsf{fe}}$ for $\ell \in [Q_k]$ are modified as follows. The vectors $\mathbf{c}_{\kappa,\mathsf{ab},j}, \mathbf{c}_{\kappa,\mathsf{fe}}$ for $\kappa \in [Q_c] \setminus \{\mu\}$ are the same as in Hybrid 4-(μ-1)-4. In the following, $\mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{I})$ and $\mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{II})$ represent the key components corresponding to $\mathbb{A}(\mathsf{S}_\mu) = 1$ and $\mathbb{A}(\mathsf{S}_\mu) = 0$ respectively.

$$
\begin{array}{lllll}
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{ab},j}: & (& a'_{\ell,j}\delta_\ell z_\ell / v_{\mu,j} &) \ \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): & \mathbf{c}_{\mu,\mathsf{ab},j}: & (& \tau_\mu v_{\mu,j} &) \ \forall j \in \mathsf{S}_\mu \ , \\
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{I}): & (& a'_{\ell,0}\delta_\ell z_\ell &) \ , \\
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{II}): & (& r'_{\ell,0}\delta_\ell z_\ell &) \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): & \mathbf{c}_{\mu,\mathsf{fe}}: & (& \mathbf{x}^{(0)}_\mu, \ \mathbf{x}^{(0)}_{\mu,\mathsf{priv}}, \ \tau_\mu, \ \mathbf{0}^{n_2} &) \ .
\end{array}
$$

where $a'_{\ell,0}, v_{\mu,j}, \tau_\mu, r'_{\ell,0} \leftarrow \mathbb{Z}_p$, $(a'_{\ell,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_{\ell,0}}(\mathbb{A})$ with $\delta_\ell = \langle \mathbf{x}^{(1)}_\mu, \mathbf{y}_\ell\rangle_p + \langle \mathbf{x}^{(1)}_{\mu,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}\rangle - \langle \mathbf{x}^{(0)}_\mu, \mathbf{y}_\ell\rangle_p - \langle \mathbf{x}^{(0)}_{\mu,\mathsf{priv}}, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}\rangle$. The indistinguishability between Hybrid 4-(μ-1)-4 and Hybrid 4-μ-1 is proven in Claim 7.

Hybrid 4-μ-2. We modify the vector $\mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{II})$ for all $\ell \in [Q_k]$ as following where $r''_{\ell,0} = r'_{\ell,0} + \frac{1}{z_\ell \tau_\mu}$.

$$
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{II}): \quad (\quad r''_{\ell,0}\delta_\ell z_\ell \quad) \ .
$$

Claim 6 Hybrid 4-μ-1 and Hybrid 4-μ-2 are statistically indistinguishable.

Proof. The following distributions

$$
\{r'_{\ell,0} : r'_{\ell,0} \leftarrow \mathbb{Z}_p\} \text{ and } \{r'_{\ell,0} + \frac{1}{z_\ell \tau_\mu} : r'_{\ell,0}, z_\ell, \tau_\mu \leftarrow \mathbb{Z}_p\}
$$

are statistically indistinguishable as $r'_{\ell,0}, z_\ell$ and τ_μ are uniformly distributed over $\mathbb{Z}_p$. Thus, $r'_{\ell,0} + \frac{1}{z_\ell \tau_\mu}$ is also uniformly distributed over $\mathbb{Z}_p$. Therefore, Hybrid 4-μ-1 and Hybrid 4-μ-2 are statistically indistinguishable. □

Hybrid 4-μ-3. This hybrid is the same as Hybrid 4-μ-2 except the vectors $\mathbf{k}_{\ell,\mathsf{fe}}, \mathbf{c}_{\mu,\mathsf{fe}}$. The modification is as follows:

$$
\begin{array}{lllll}
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{I}): & (& a'_{\ell,0}\delta_\ell z_\ell &) \ , \\
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}, I_{\mathbf{y}_\ell}, \mathbb{A}): & \mathbf{k}_{\ell,\mathsf{fe}}(\mathsf{II}): & (& r'_{\ell,0}\delta_\ell z_\ell &) \ , \\
\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): & \mathbf{c}_{\mu,\mathsf{fe}}: & (& \mathbf{x}^{(1)}_\mu, \ \mathbf{0}^{n_2}, \ \tau_\mu, \ \mathbf{x}^{(1)}_{\mu,\mathsf{priv}} &) \ .
\end{array}
$$

Hybrid 4-μ-2 and Hybrid 4-μ-3 are computationally indistinguishable if the underlying scheme Π_{esi} is function-hiding. The proof follows similar to Lemma 12 as we know that for $\mathbb{A}(\mathsf{S}_\mu) = 1$,

$$
\begin{aligned}
&\mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-2}}, \mathsf{eCT}_\mu^{\text{Hybrid 4-}\mu\text{-2}}) \\
&= [\![\langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle + \psi_\mu \cdot a_{\ell,0} \cdot z_\ell + \tau_\mu \cdot a'_{\ell,0} \cdot z_\ell \cdot \delta_\ell]\!]_T \\
&= [\![\langle \mathbf{x}_\mu^{(1)}, \mathbf{y}_\ell \rangle_p + \psi_\mu \cdot a_{\ell,0} \cdot z_\ell + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(1)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)} \rangle + \tau_\mu \cdot a'_{\ell,0} \cdot z_\ell \cdot \delta_\ell]\!]_T \\
&= \mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-3}}, \mathsf{eCT}_\mu^{\text{Hybrid 4-}\mu\text{-3}})
\end{aligned}
$$

as $[\![\langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell \rangle + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle]\!]_T = [\![\langle \mathbf{x}_\mu^{(1)}, \mathbf{y}_\ell \rangle + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(1)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)} \rangle]\!]_T$ from the security definition. In case of $\mathbb{A}(\mathsf{S}_\mu) = 0$,

$$
\begin{aligned}
&\mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-2}}, \mathsf{eCT}_\mu^{\text{Hybrid 4-}\mu\text{-2}}) \\
&= [\![\langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle + \psi_\mu \cdot a_{\ell,0} \cdot z_\ell + \tau_\mu \cdot r''_{\ell,0} \cdot z_\ell \cdot \delta_\ell]\!]_T \\
&= [\![\langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle + \psi_\mu \cdot a_{\ell,0} \cdot z_\ell + \tau_\mu \cdot (r'_{\ell,0} + \frac{1}{z_\ell \tau_\mu}) \cdot z_\ell \cdot \delta_\ell]\!]_T \\
&= [\![\langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle + \psi_\mu \cdot a_{\ell,0} \cdot z_\ell + \tau_\mu \cdot r'_{\ell,0} \cdot z_\ell \cdot \delta_\ell + \delta_\ell]\!]_T \\
&= [\![\langle \mathbf{x}_\mu^{(1)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(1)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)} \rangle + \psi_\mu \cdot a_{\ell,0} \cdot z_\ell + \tau_\mu \cdot r'_{\ell,0} \cdot z_\ell \cdot \delta_\ell]\!]_T \\
&= \mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-3}}, \mathsf{eCT}_\mu^{\text{Hybrid 4-}\mu\text{-3}})
\end{aligned}
$$

as $\delta_\ell = \langle \mathbf{x}_\mu^{(1)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(1)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)} \rangle - \langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell \rangle_p - \langle \mathbf{x}_{\mu,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle$. In case of $\kappa < \mu$ and $\kappa \in [Q_c]$, we have

$$
\begin{aligned}
&\mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-2}}, \mathsf{eCT}_\kappa^{\text{Hybrid 4-}\mu\text{-2}}) \\
&= [\![\langle \mathbf{x}_\kappa^{(1)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\kappa,\mathsf{priv}}^{(1)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)} \rangle + \psi_\kappa \cdot a_{\ell,0} \cdot z_\ell]\!]_T \\
&= \mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-3}}, \mathsf{eCT}_\kappa^{\text{Hybrid 4-}\mu\text{-3}})
\end{aligned}
$$

and for $\kappa > \mu$ and $\kappa \in [Q_c]$, we have

$$
\begin{aligned}
&\mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-2}}, \mathsf{eCT}_\kappa^{\text{Hybrid 4-}\mu\text{-2}}) \\
&= [\![\langle \mathbf{x}_\kappa^{(0)}, \mathbf{y}_\ell \rangle_p + \langle \mathbf{x}_{\kappa,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)} \rangle + \psi_\kappa \cdot a_{\ell,0} \cdot z_\ell]\!]_T \\
&= \mathsf{eDec}(\mathsf{eSK}_\ell^{\text{Hybrid 4-}\mu\text{-3}}, \mathsf{eCT}_\kappa^{\text{Hybrid 4-}\mu\text{-3}}).
\end{aligned}
$$

Thus, the advantage of $\mathcal{A}$ in distinguishing between Hybrid 4-μ-2 and Hybrid 4-μ-3 is the same as the advantage in distinguishing between the experiments $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ and $\mathsf{Expt}^{\mathsf{esi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 1)$.

Hybrid 4-μ-4. This hybrid is the same as Hybrid 4-μ-3 except for the following changes.

$$
\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{ab},j}: \quad (\ \boxed{0}\ ) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ , \\
&\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}_\mu^{(\beta)}, \mathbf{x}_{\mu,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{ab},j}: \quad (\ \boxed{0}\ ) \quad \forall j \in \mathsf{S}_\mu \ , \\
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\ell, \mathbf{y}_{\ell,\mathsf{priv}}^{(0)}, \mathbf{y}_{\ell,\mathsf{priv}}^{(1)}, I_{\mathbf{y}_\ell}, \mathbb{A}): \quad \mathbf{k}_{\ell,\mathsf{fe}}: \quad (\ \boxed{0}\ ) \ , \\
&\mathcal{O}_{\mathsf{E},1}(\{\mathbf{x}_\mu^{(\beta)}, \mathbf{x}_{\mu,\mathsf{priv}}^{(\beta)}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{fe}}: \quad (\ \mathbf{x}_\mu^{(1)}, \ \mathbf{0}^{n_2}, \ \boxed{0}, \ \mathbf{x}_{\mu,\mathsf{priv}}^{(1)}\ ) \ .
\end{aligned}
$$

The proof of indistinguishability follows the same way as Claim 7.

Claim 7 Hybrid 4-$(\mu-1)$-4 and Hybrid 4-μ-1 are computationally indistinguishable if Lemma 10 holds over the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and $\Pi_{\mathsf{sip}}, \Pi_{\mathsf{esi}}$ are function-hiding IPFE schemes.

Proof. We prove claim 7 through a sequence of hybrids $\hat{H}_{\mu-1,\eta,\omega}$ for $\eta \in [Q_k]$ and $\omega \in [6]$. We define $\hat{H}_{\mu-1,1,0}$ as Hybrid 4-(μ-1)-4 and $\hat{H}_{\mu-1,Q_k,6}$ as Hybrid 4-μ-1. In the following, we explicitly present the hybrid $\hat{H}_{\mu-1,1,0}$ once again, and then proceed to define the intermediate hybrids below. We represent the slots that are updated in the subsequent hybrids using dashed boxes. In the sub-hybrids, we will only mention the updated slots.

$\hat{H}_{\mu-1,1,0}$**:** This is the same as Hybrid 4-(μ-1)-4. For $\eta = 1$, the ciphertexts and the secret keys components are given below:

$$
\begin{array}{llccccccccc}
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}) & \mathbf{k}_{\eta,\mathsf{ab},j} = & (& \pi_{\eta,j}(j,1), & a_{\eta,j}\cdot z_\eta, & a_{\eta,j} z_\eta, & 0, & \boxed{0}, & \boxed{0} &) & \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu) & \mathbf{c}_{\mu,\mathsf{ab},j} = & (& \sigma_{\mu,j}(1,-j), & 0, & \psi_\mu, & 0, & \boxed{0}, & \boxed{0} &) & \forall j \in \mathsf{S}_\mu \ , \\
\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}) & \mathbf{k}_{\eta,\mathsf{fe}} = & (& \mathbf{y}_\eta, & a_{\eta,0}\cdot z_\eta, & \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, & a_{\eta,0}\cdot z_\eta, & \boxed{0}, & \boxed{0}, & \mathbf{y}^{(1)}_{\eta,\mathsf{priv}} &) \ , \\
\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu) & \mathbf{c}_{\mu,\mathsf{fe}} = & (& \mathbf{x}^{(0)}_\mu, & 0, & \mathbf{x}^{(0)}_{\mu,\mathsf{priv}}, & \psi_\mu, & \boxed{0}, & \boxed{0}, & \mathbf{0}^{n_2} &) \ .
\end{array}
$$

$\hat{H}_{\mu-1,\eta,1}$**:** In case of $\eta = 1$, sample $\tau_\mu, v_{\mu,j} \leftarrow \mathbb{Z}_p$ for all $j \in \mathsf{S}_\mu$. In other cases, this hybrid is the same as $\hat{H}_{\mu-1,\eta-1,5}$.

$$
\begin{aligned}
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{ab},j}: \quad (\quad \boxed{\tau_\mu v_{\mu,j}}, \quad 0 \quad) \quad \forall j \in \mathsf{S}_\mu \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{fe}}: \quad (\quad \boxed{\tau_\mu}, \quad 0 \quad) \ .
\end{aligned}
$$

The indistinguishability proof follows similar to the proofs of Lemma 11 and Lemma 12. It follows from the fact that for all $\ell \in [Q_k]$, $\kappa \in [Q_c]$,

$$
\begin{aligned}
&\mathsf{eDec}(\mathsf{eSK}_\ell^{\hat{H}_{\mu-1,1,0}}, \mathsf{eCT}_\kappa^{\hat{H}_{\mu-1,1,0}}) \\
&= [\![\langle \mathbf{x}^{(0)}_\kappa, \mathbf{y}_\ell\rangle_p + \langle \mathbf{x}^{(0)}_{\kappa,\mathsf{priv}}, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}\rangle + \psi_\kappa \cdot a_{\ell,0} \cdot z_\ell]\!]_T \\
&= \mathsf{eDec}(\mathsf{eSK}_\ell^{\hat{H}_{\mu-1,1,1}}, \mathsf{eCT}_\kappa^{\hat{H}_{\mu-1,1,1}})
\end{aligned}
$$

and for all queries $i \in \mathsf{S}_\kappa$, $j \in \mathsf{List\text{-}Att}(\mathbb{A})$, we have

$$
\begin{aligned}
&\mathsf{iDec}(\mathsf{iSK}_{\ell,\mathsf{ab},j}^{\text{Hybrid } \hat{H}_{\mu-1,1,0}}, \mathsf{iCT}_{\kappa,\mathsf{ab},i}^{\text{Hybrid } \hat{H}_{\mu-1,1,0}}) \\
&= [\![\pi_{\ell,j}\cdot j\cdot\sigma_{\kappa,i} + \pi_{\ell,j}\cdot\sigma_{\kappa,i}\cdot -i + \psi_\kappa \cdot a_{\ell,j}\cdot z_\ell]\!]_T \\
&= \mathsf{iDec}(\mathsf{iSK}_{\ell,\mathsf{ab},j}^{\text{Hybrid } \hat{H}_{\mu-1,1,1}}, \mathsf{iCT}_{\kappa,\mathsf{ab},i}^{\text{Hybrid } \hat{H}_{\mu-1,1,1}}).
\end{aligned}
$$

$\hat{H}_{\mu-1,\eta,2}$**:** This is the same as $\hat{H}_{\mu-1,\eta,1}$ except for the changes below. Generate the vectors as following where $a'_{\eta,0}, v'_{\eta,j}, \tau'_\eta \leftarrow \mathbb{Z}_p$, $(a'_{\eta,j})_{j\in\mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_{\eta,0}}(\mathbb{A})$ with $\delta_\eta = \langle \mathbf{x}^{(1)}_\mu, \mathbf{y}_\eta\rangle_p + \langle \mathbf{x}^{(1)}_{\mu,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}\rangle - \langle \mathbf{x}^{(0)}_\mu, \mathbf{y}_\eta\rangle_p - \langle \mathbf{x}^{(0)}_{\mu,\mathsf{priv}}, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}\rangle$.

$$
\begin{aligned}
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}): \quad \mathbf{k}_{\eta,\mathsf{ab},j}: \quad (\quad 0, \quad \boxed{a'_{\eta,j}\delta_\eta z_\eta / v'_{\eta,j}} \quad) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{ab},j}: \quad (\quad \tau_\mu v_{\mu,j}, \boxed{\tau'_\eta v'_{\eta,j}} \quad) \quad \forall j \in \mathsf{S}_\mu \ , \\
&\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}): \quad \mathbf{k}_{\eta,\mathsf{fe}}: \quad (\quad 0, \boxed{a'_{\eta,0}\delta_\eta z_\eta} \quad) \ , \\
&\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{fe}}: \quad (\quad \tau_\mu, \boxed{\tau'_\eta} \quad) \ .
\end{aligned}
$$

Claim 8 $\hat{H}_{\mu-1,1,1}$ and $\hat{H}_{\mu-1,1,2}$ are computationally indistinguishable if Lemma 10 holds over the groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

Proof. We prove the above lemma through a reduction to the Lemma 10 with vectors $\mathbf{k}_{\eta,\mathsf{ab},j}, \mathbf{c}_{\mu,\mathsf{ab},j}$, $\mathbf{k}_{\eta,\mathsf{fe}}, \mathbf{c}_{\mu,\mathsf{fe}}$ set to $\mathbf{k}^{\beta}_{j}, \mathbf{c}^{\beta}_{j}, \mathbf{k}^{\beta}_{\mathsf{root}}, \mathbf{c}^{\beta}_{\mathsf{root}}$, respectively. The variables in the masking lemma are set as $x = 1$ and $y = \delta_\eta$. □

$\hat{H}_{\mu-1,\eta,3}$**:** For all $j \in \mathsf{S}_\mu$, set $v'_{\eta,j} = v_{\mu,j}$ and $\tau'_\eta = \tau_\mu$. This is a statistical modification.

$\hat{H}_{\mu-1,\eta,4}$**:** We modify the vectors $\mathbf{k}_{\eta,\mathsf{ab},j}, \mathbf{c}_{\mu,\mathsf{ab},j}$, $\mathbf{k}_{\eta,\mathsf{fe}}, \mathbf{c}_{\mu,\mathsf{fe}}$ as below. All the other vectors remain the same as in $\hat{H}_{\mu-1,\eta,3}$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}): \quad \mathbf{k}_{\eta,\mathsf{ab},j}: \quad (\ a'_{\eta,j}\delta_\eta z_\eta / v_j, \quad 0\ ) \quad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \ ,$$
$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{ab},j}: \quad (\ \tau_\mu v_{\mu j}, \quad 0\ ) \quad \forall j \in \mathsf{S}_\mu \ ,$$

$\hat{H}_{\mu-1,\eta,3}$ and $\hat{H}_{\mu-1,\eta,4}$ are computationally indistinguishable if the underlying schemes Π_{sip} are function-hiding. The proof is similar to the proofs of Lemma 11.

$\hat{H}_{\mu-1,\eta,5}$**:** We modify the vectors $\mathbf{k}_{\eta,\mathsf{fe}}, \mathbf{c}_{\mu,\mathsf{fe}}$ as below. All the other vectors remain the same as in $\hat{H}_{\mu-1,\eta,4}$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}): \quad \mathbf{k}_{\eta,\mathsf{fe}}: \quad (\ a'_{\eta,0}\delta_\eta z_\eta, \quad 0\ ) \ ,$$
$$\mathcal{O}_{\mathsf{E},0}(\{\mathbf{x}^{(\beta)}_\mu, \mathbf{x}^{(\beta)}_{\mu,\mathsf{priv}}\}_{\beta\in\{0,1\}}, \mathsf{S}_\mu): \quad \mathbf{c}_{\mu,\mathsf{fe}}: \quad (\ \tau_\mu, \quad 0\ ) \ .$$

$\hat{H}_{\mu-1,\eta,4}$ and $\hat{H}_{\mu-1,\eta,5}$ are computationally indistinguishable if the underlying scheme Π_{esi} are function-hiding. The proof is similar to the proofs of Lemma 12.

$\hat{H}_{\mu-1,\eta,6}$**:** This is the same as $\hat{H}_{\mu-1,\eta,5}$ except that when $\mathbb{A}(\mathsf{S}_\mu) = 0$, we define the key vector $\mathbf{k}_{\eta,\mathsf{fe}}(\mathsf{II})$ as following where $r'_{\eta,0} \leftarrow \mathbb{Z}_p$. This hybrid is information-theoretically indistinguishable from $\hat{H}_{\mu-1,\eta,5}$.

$$\mathcal{O}_{\mathsf{KG},0}(\mathbf{y}_\eta, \mathbf{y}^{(0)}_{\eta,\mathsf{priv}}, \mathbf{y}^{(1)}_{\eta,\mathsf{priv}}, I_{\mathbf{y}_\eta}, \mathbb{A}): \quad \mathbf{k}_{\eta,\mathsf{fe}}(\mathsf{II}): \quad (\ r'_{\eta,0}\delta_\eta z_\eta, \quad 0\ ) \ .$$

Note that for $\eta \in [Q_k]$, $\hat{H}_{\mu-1,\eta,6} \equiv \hat{H}_{\mu-1,\eta+1,1}$ and $\hat{H}_{\mu-1,Q_k,6} \equiv$ Hybrid 4-μ-1 by the definition of $\hat{H}_{\mu-1,\eta,6}$ and Hybrid 4-μ-1, respectively. This completes the proof of Claim 7. □

This concludes the proof of Lemma 14. □

Lemma 15 *Hybrid 5 and Hybrid 6 are computationally indistinguishable if the underlying scheme* Π_{esi} *is function-hiding.*

The proof follows the same way as that of Lemma 12.

This completes the proof of Theorem 7. □

6 Multi-Client Attribute-Based UIPFE

In this section, we define the *multi-client unbounded FE* (MC-UFE) scheme over the key space $\mathcal{K}$, message space $\mathcal{M}$ and label space $\mathcal{L}$ for functionality $f : (\mathcal{K}^*)^n \times (\mathcal{M}^* \times \mathcal{L})^n \to \mathcal{Z}$ having n users in the system.

Definition 15 An MC-UFE scheme $\Pi_{\mathsf{mcf}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of following four algorithms:

$\mathsf{Setup}(1^\lambda, n) \to (\{\mathsf{EK}_k\}_{k\in[n]}, \mathsf{MSK})$: The algorithm takes as input security parameter λ, total number of users in the system n and outputs encryption keys EK_k for each user $k \in [n]$ and the master secret key MSK.

$\mathsf{KeyGen}(\mathsf{MSK}, (\mathsf{Key}_{k,j})_{j\in I_k, k\in[n]}) \to \mathsf{SK}$: The key generation algorithm takes as input MSK, and a key space object $(\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]}$ with the associated index sets I_k. It outputs a secret key SK.

$\mathsf{Enc}(\mathsf{EK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k}, L_k) \to \mathsf{CT}_k$: The algorithm takes as input k-th party's EK_k, a message $(\mathsf{Msg}_{k,j})_{j\in I'_k}$ with the associated index set I'_k and a label L_k. It outputs a ciphertext CT_k.

$\mathsf{Dec}(\mathsf{SK}, \{\mathsf{CT}_k\}_{k\in[n]}) \to \zeta \vee \perp$: This algorithm takes as input SK, $\{\mathsf{CT}_k\}_{k\in[n]}$ and outputs either ζ or the special symbol $\perp$ indicating failure.

Correctness: For all $\lambda \in \mathbb{N}, (\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]} \in (\mathcal{K}^*)^n$, and for all $k \in [n]$ $(\mathsf{Msg}_{k,j})_{j\in I'_k} \in \mathcal{M}^*$, $L_k \in \mathcal{L}$, we have

$$\Pr\left[\begin{array}{l|l} \zeta = f((\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]}, & (\mathsf{EK}_k, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, n) \\ \quad \{(\mathsf{Msg}_{k,j})_{j\in I'_k}, L_k\}_{k\in[n]}) & \mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, (\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]}) \\ & \mathsf{CT}_k \leftarrow \mathsf{Enc}(\mathsf{EK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k}, L_k) \\ & \zeta \leftarrow \mathsf{Dec}(\mathsf{SK}, \{\mathsf{CT}_k\}_{k\in[n]}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

if for all $k_1, k_2 \in [n], L_{k_1} = L_{k_2}$.

Definition 16 (Security of MC-UFE) The $\Pi_{\mathsf{mcf}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be xx-yy-indistinguishability (xx-yy-IND) secure for $\mathrm{xx} \in \{\mathsf{sel}, \mathsf{adp}\}, \mathrm{yy} \in \{\mathsf{any}, \mathsf{pos}^+\}$ if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds

$$\mathsf{Adv}^{\mathsf{mcf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda) = \left|\Pr\left[\mathsf{Expt}^{\mathsf{mcf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mcf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{mcf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{mcf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, \beta):}$
1: $(\{\mathsf{EK}_k\}_{k\in[n]}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, n)$.
2: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Corr}}(\cdot), \mathcal{O}_{\mathsf{KG}}(\cdot), \mathcal{O}_{\mathsf{E}}(\cdot), \mathcal{O}_{\mathsf{LoR},\beta}(\cdot)}(1^\lambda, n)$.
3: Output β' if condition $(*)$ is satisfied.

$\underline{\mathcal{O}_{\mathsf{Corr}}(\cdot):}$
output EK_k.

$\underline{\mathcal{O}_{\mathsf{KG}}((\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]}):}$
output $\mathsf{KeyGen}(\mathsf{MSK}, (\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]})$.

$\underline{\mathcal{O}_{\mathsf{E}}(k, (\mathsf{Msg}_{k,j})_{j\in I'_k}, L_k):}$
output $\mathsf{Enc}(\mathsf{EK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k}, L_k)$.

$\underline{\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k}, L_k):}$
output $\mathsf{Enc}(\mathsf{EK}_k, (\mathsf{Msg}^\beta_{k,j})_{j\in I'_k}, L_k)$.

Let $\mathcal{CS}$ be the set of all inputs $k \in [n]$ for which $\mathcal{A}$ makes queries to $\mathcal{O}_{\mathsf{Corr}}(\cdot)$ and $\mathcal{HS} = [n] \setminus \mathcal{CS}$. The condition $(*)$ is that if there exist two messages satisfying

$$f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in[n]}, \{k, (\mathsf{Msg}^0_{k,j})_{j\in I'_k}, L_k\}_{k\in[n]}) \quad \neq f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in[n]}, \{k, (\mathsf{Msg}^1_{k,j})_{j\in I'_k}, L_k\}_{k\in[n]})$$

then at least one of the following should *not* hold

- for all $k \in [n]$, $[\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k}, L_k)$ is queried or $\mathcal{O}_{\mathsf{E}}(k, (\mathsf{Msg}_{k,j})_{j\in I'_k}, L_k)$ with $(\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j} = \mathsf{Msg}_{k,j})_{j\in I'_k}$ is queried] or $[(\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j} = \mathsf{Msg}_{k,j})_{j\in I'_k}$ and $k \in \mathcal{CS}]$.
- $\mathcal{O}_{\mathsf{KG}}(\cdot)$ was queried on $(\mathsf{Key}_{k,j})_{j\in I_k,k\in[n]}$.

– If xx = sel: Queries to $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot), \mathcal{O}_{\mathsf{Corr}}(\cdot), \mathcal{O}_{\mathsf{E}}(\cdot)$ must be made in one shot before any queries to $\mathcal{O}_{\mathsf{KG}}(\cdot)$.
– If yy = pos: for any user $k \in [n]$ and $L \in \mathcal{L}$, if $Q_{k,L} > 0$, then for any user $k' \in \mathcal{HS}$, $Q_{k',L} > 0$ where $Q_{k,L}$ denotes the number of ciphertext queries to the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ in of the form $(k, *, *, L)$. In other words, for any label, either the adversary makes no left-right encryption query or makes at least one left-right encryption query for each $k' \in \mathcal{HS}$.

In the *one-time label security*, all the queries to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle should be in one label L and no queries to the $\mathcal{O}_{\mathsf{E}}(\cdot)$ oracle will be possible with the same label L.

Definition 17 (MC-UFE for AB-IP) A multi-client attribute-based UIPFE (MC-AB-UIPFE) is a particular class of MC-UFE where $\mathcal{K}^* = \mathbb{Z}_p^* \times \mathcal{P}$, and $\mathcal{M}^* = \mathbb{Z}_p^* \times \mathcal{ATT}$ such that $\mathcal{P}$ and $\mathcal{ATT}$ represent the access policy and attribute spaces respectively. The function f is defined as follows: for the message components $\mathsf{Msg}_k = (\mathbf{x}_k, \mathsf{S}_k) \in \mathcal{M}^*$, the key components $\mathsf{Key} = (\mathbf{y}_k, \mathbb{A})_{k\in[n]} \in (\mathcal{K}^*)^n$ and $\mathbf{x}_k, \mathbf{y}_k$ are associated with the index sets I_k and I'_k,

$$f((\mathsf{Key}_{k,j})_{j\in I_k, k\in[n]}, \{(\mathsf{Msg}_{k,j}, L_k)_{j\in I'_k}\}_{k\in[n]}) = \begin{cases} \sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k \rangle_p & \text{if following conditions holds} \\ \perp & \text{otherwise.} \end{cases}$$

The conditions in $(\star)$ define as follows:

- $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_k) = 1 \wedge \mathbb{A}(\mathsf{S}_k) = 1$ for all $k \in [n]$.
- for all $k_1, k_2 \in [n]$, $L_{k_1} = L_{k_2}$.

6.1 Construction

Consider $\Pi_{\mathsf{asi}} = (\mathsf{aSetup}, \mathsf{aKeyGen}, \mathsf{aEnc}, \mathsf{aSlotEnc}, \mathsf{aDec})$ be an AB-sUIPFE scheme with slot-specification $\mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{\widetilde{m}+1}$, $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^*$ and $\mathsf{PRF}^{\mathsf{seed}} : \mathcal{L} \to \mathbb{Z}_p^{\widetilde{m}}$ be a family of pseudorandom function with $\mathsf{seed} \in \mathcal{K}_{\mathsf{prf}}$ where $\mathcal{K}_{\mathsf{prf}}, \mathcal{L}$ be pseudorandom key space and the label space for any security parameter λ. Note that, our proposed MC-AB-UIPFE only involves the aEnc algorithm to encrypt the slot-specified message vector using a corresponding master secret key. In the following, we present our MC-AB-UIPFE scheme $\Pi_{\mathsf{mcai}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ for LSSS access structure. We discuss the PRF and the LSSS access structure in Definitions 4 and 3, respectively.

$\mathsf{Setup}(1^\lambda, n)$: The setup algorithm takes the security parameter λ with the total number of user n in the system as input and executes the following steps:

1. Generates $(\mathsf{aMPK}_k, \mathsf{aMSK}_k) \leftarrow \mathsf{aSetup}(1^\lambda, 1^{\widetilde{m}+1})$ for all $k \in [n]$.
2. Samples $\mathsf{seed}_{k,\iota} \leftarrow K_\lambda$ for all $k, \iota \in [n]$ with $\mathsf{seed}_{k,\iota} = \mathsf{seed}_{\iota,k}$ for $\iota < k$.
3. Outputs encryption key $\mathsf{EK}_k = (\mathsf{aMSK}_k, \{\mathsf{seed}_{\iota,k}\}_{\iota\neq k})$ and the master secret key $\mathsf{MSK} = \{\mathsf{aMSK}_k\}_{k\in[n]}$.

$\mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y} = (\mathbf{y}_k)_{k\in[n]}, \{I_{\mathbf{y}_k}\}_{k\in[n]}, \mathbb{A})$: The key generation algorithm takes as input MSK, the access structure $\mathbb{A}$ and a key vector $\mathbf{y} = (\mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \cdots \parallel \mathbf{y}_n)$ where each $\mathbf{y}_k$ is associated with the index set $I_{\mathbf{y}_k}$ for all $k \in [n]$. It works as follows:

1. Samples $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_p^{\widetilde{m}}$.
2. Generates $\mathsf{aSK}_k \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\mathbf{y}_k, \boldsymbol{\alpha}, 0)]\!]_2, I_{\mathbf{y}_k}, \mathbb{A})$ for all $k \in [n]$.
3. Outputs the secret key $\mathsf{SK} = \{\mathsf{aSK}_k\}_{k\in[n]}$.

$\mathsf{Enc}(\mathsf{EK}_k, \mathbf{x}_k, L, \mathsf{S}_k)$: The encryption algorithm takes as input k-th user's EK_k, a message vector $\mathbf{x}_k = (x_{k,i})_{i\in[m_k]}$ of an arbitrary length m_k, a label L with an attribute set S_k and proceeds to do the following steps:

1. Computes $\mathbf{s}_k = \sum_{\iota\neq k} (-1)^{\iota<k} \mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L)$.
2. Generates $\mathsf{aCT}_k \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\mathbf{x}_k, \mathbf{s}_k, 0)]\!]_1, \mathsf{S}_k)$.
3. Outputs the ciphertext $\mathsf{CT}_k = \mathsf{aCT}_k$.

$\mathsf{Dec}(\mathsf{SK}, \{\mathsf{CT}_k\}_{k\in[n]})$: The decryption algorithm takes as input SK, CT_k and performs the following steps:

1. Returns either $[\![d]\!]_T \leftarrow \prod_{k\in[n]} \mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k)$ or $\perp$.

Correctness: If $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_k) = 0 \vee \mathbb{A}(\mathsf{S}_k) = 0$ for any $k \in [n]$, outputs $\perp$. Otherwise, from the correctness of Π_{asi}, we have

$$\mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k) = [\![\langle \mathbf{x}_k, \mathbf{y}_k \rangle_p + \langle \mathbf{s}_k, \boldsymbol{\alpha} \rangle]\!]_T \ . \tag{3}$$

From Equation 3, we compute

$$[\![d]\!]_T = \prod_{k\in[n]} \mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k) = [\![\sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k\rangle_p + \langle \mathbf{s}_k, \boldsymbol{\alpha}\rangle]\!]_T = [\![\sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k\rangle_p]\!]_T \ .$$

6.2 Security Analysis

In Theorem 8, we present the security analysis of our MC-AB-UIPFE scheme, as described in Construction 6.1.

Theorem 8 *Our* Π_{mcai} *scheme achieves selective indistinguishability (sel-IND) security as per Definition 16 if the underlying* Π_{asi} *is selectively secure as per Definition 14 and the MDDH assumption holds in* G.

Proof. Suppose $\mathcal{A}$ be a PPT adversary against the sel-FH-IND security of our MC-AB-UIPFE scheme. We construct an algorithm $\mathcal{B}$ for breaking underlying Π_{mcai} scheme that uses $\mathcal{A}$ as a subroutine. Let $\{\mathsf{PRF}^{\mathsf{seed}}\} : \mathcal{L}_\lambda \to \mathbb{Z}_p^{\widetilde{m}}$ be a family of pseudorandom function. In the following, we consider a series of hybrids to prove Theorem 8. We provide a brief indistinguishable arguments of security hybrids in Fig. 4.

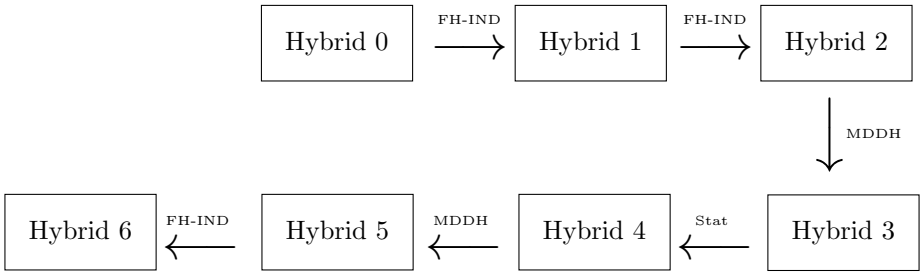

Figure 4: Outline of the security games for Theorem 8. Here, 'Stat' means statistically, and 'FH-IND' is a shorthand for the function-hiding indistinguishability security of Π_{asi}.

Hybrid 0. This hybrid is the same as the real security game where the challenge ciphertext is the encryption for the challenge bit $\beta = 0$ as described in Definition 16 of sel-pos-IND security model. In the following, we describe the oracles that the adversary $\mathcal{A}$ can queried during the security experiment. We represent the slots using dashed boxes, which are updated in the following hybrid steps.

- **Corruption queries:** The adversary $\mathcal{A}$ first submits the corrupted users set $\mathcal{C}$ to the challenger $\mathcal{B}$ and returns each encryption keys EK_k corresponding to the user index $k \in \mathcal{C}$.
- **Left or right oracle queries:** On receiving the μ-th query tuple to the oracle $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ for the tuple $(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k})$ with $k \in \mathcal{HS}$, the challenger simulates the challenge ciphertext as given below.

$$\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k}) : \quad \mathsf{aCT}^{(0)}_{\mu,k} = \quad \mathsf{aEnc}(\quad \mathsf{aMSK}_k, \quad [\![(\boxed{\mathbf{x}^{(0)}_{\mu,k},} \quad \boxed{\mathbf{s}_k,} \quad \boxed{0})]\!]_1 \quad \mathsf{S}_{\mu,k} \quad) \ ,$$

  where $\mathsf{CT}^{(0)}_{\mu,k} = \mathsf{aCT}^{(0)}_{\mu,k}$, $\mathbf{s}_k = \sum_{\iota\neq k}(-1)^{\iota<k}\mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L)$ and the μ-th challenge messages $\{(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k})\}_{k\in[n]}$ of length m_k for each user $k \in [n]$.
- **Encryption oracle queries:** As dictated in the security Definition 16, the adversary can only query with respect to any label $L'(\neq L)$ for $k \in [n]$ and the messages $\{\mathbf{x}_{\mu',k}\}_{k\in[n]}$ with an attribute set $\mathsf{S}_{\mu',k}$ and generates the ciphertext as given below.

$$\mathcal{O}_{\mathsf{E}}(k, \mathbf{x}_k, L', \mathsf{S}'_k) : \quad \mathsf{aCT}_{\mu',k} = \quad \mathsf{aEnc}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{x}_{\mu',k}, \quad \mathbf{s}'_{\mu',k}, \quad 0)]\!]_1 \quad \mathsf{S}'_{\mu',k} \quad) \ ,$$

  where $\mathsf{CT}_{\mu',k} = \mathsf{aCT}_{\mu',k}$ and $\mathbf{s}'_k = \sum_{\iota\neq k}(-1)^{\iota<k}\mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L')$.

– **Key generation queries:** For the ℓ-th functional key corresponding to the access structure $\mathbb{A}$ and the key vector $\mathbf{y}_\ell = (\mathbf{y}_{\ell,k})_{k\in[n]}$ with each non-empty index set $I_{\mathbf{y}_{\ell,k}}$ for all $k \in [n]$, the simulator generates the secret key components $\mathsf{SK}_{\ell,k}$ as in the following.

$$\mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad 0)]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ ,$$

where $\mathsf{SK}_\ell = \{\mathsf{aSK}_{\ell,k}\}_k$ with $\boldsymbol{\alpha}_\ell \leftarrow \mathbb{Z}_p$.

Hybrid 1. This game is the same as Hybrid 0 except that the generated challenge ciphertext and the secret key components using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ are modified as follows.

$$\begin{aligned}\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k}): \quad \mathsf{aCT}^{(0)}_{\mu,k} = \quad \mathsf{aEnc}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{x}^{(0)}_{\mu,k}, \quad \mathbf{0}, \quad 1)]\!]_1, \quad \mathsf{S}_{\mu,k} \quad) \ , \\ \mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle)]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ .\end{aligned}$$

Hybrid 2. This game is the same as Hybrid 1 except that the generated challenge ciphertext and the secret key components for $\mathbb{A}(\mathsf{S}_{\mu,k}) = 1 \wedge k \in \mathcal{HS}$ using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ are modified below.

$$\begin{aligned}\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k}): \quad \mathsf{aCT}^{(1)}_{\mu,k} = \quad \mathsf{aEnc}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{x}^{(1)}_{\mu,k}, \quad \mathbf{0}, \quad 1)]\!]_1, \quad \mathsf{S}_{\mu,k} \quad) \ , \\ \mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle + \delta_{1,\ell,k})]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ ,\end{aligned}$$

where $\delta_{1,\ell,k} = \langle \mathbf{x}^{(0)}_{1,k} - \mathbf{x}^{(1)}_{1,k}, \mathbf{y}_{\ell,k}\rangle_p$ and $(\mathbf{x}^{(0)}_{\mu,k}, , \mathbf{x}^{(1)}_{\mu,k})$ is the pair of challenge messages in the μ-th query to $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ of the form $(k, *, *, *, L)$ for $k \in \mathcal{HS}$.

Hybrid 3. This game is the same as Hybrid 2 except that challenge ciphertext and the secret key components for $\mathbb{A}(\mathsf{S}_{\mu,k}) = 1 \wedge k \in \mathcal{HS}$ are generated as follows.

$$\mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad v_{\ell,k} + \delta_{1,\ell,k})]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ ,$$

where $\delta_{1,\ell,k} = \langle \mathbf{x}^{(0)}_{1,k} - \mathbf{x}^{(1)}_{1,k}, \mathbf{y}_{\ell,k}\rangle_p$ and $\sum_{k\in\mathcal{HS}} v_{\ell,k} + \sum_{k\in\mathcal{CS}} \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle = 0$.

Hybrid 4. This game is the same as Hybrid 3 except that the generated challenge ciphertext and the secret key components for $\mathbb{A}(\mathsf{S}_{\mu,k}) = 1 \wedge k \in \mathcal{HS}$ using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ are given below.

$$\mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad v_{\ell,k})]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ ,$$

where $\sum_{k\in\mathcal{HS}} v_{\ell,k} + \sum_{k\in\mathcal{CS}} \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle = 0$.

Hybrid 5. This game is the same as Hybrid 4 except that the generated challenge ciphertext and the secret key components for $\mathbb{A}(\mathsf{S}_{\mu,k}) = 1$ using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ are given below.

$$\mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle)]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ ,$$

where $\mathbf{s}_k = \sum_{\iota\neq k} (-1)^{\iota<k} \mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L)$ such that $\sum_{k\in[n]} \mathbf{s}_k = 0$.

Hybrid 6. This game is the same as Hybrid 5 except that the generated challenge ciphertext and the secret key components for $\mathbb{A}(\mathsf{S}_{\mu,k}) = 1$ using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ are given below.

$$\begin{aligned}\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k}): \quad \mathsf{aCT}^{(1)}_{\mu,k} = \quad \mathsf{aEnc}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{x}^{(1)}_{\mu,k}, \quad \mathbf{s}_k, \quad 0)]\!]_1, \quad \mathsf{S}_{\mu,k} \quad) \ , \\ \mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A}): \quad \mathsf{aSK}_{\ell,k} = \quad \mathsf{aKeyGen}(\quad \mathsf{aMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boldsymbol{\alpha}_\ell, \quad 0)]\!]_2, \quad I_{\mathbf{y}_{\ell,k}}, \quad \mathbb{A} \quad) \ ,\end{aligned}$$

where $\mathbf{s}_k = \sum_{\iota \neq k} (-1)^{\iota < k} \mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L)$. This hybrid is the same as the real security game with sel-IND model for $\beta = 1$ in Definition 16. Thanks to Lemma 16 to Lemma 21, we can conclude the proof of Theorem 8.

Lemma 16 *Hybrid 0 and Hybrid 1 are computationally indistinguishable if the underlying scheme* Π_{asi} *is function-hiding.*

Proof. We consider a PPT adversary $\mathcal{A}$ against sel-FH-IND security of the MC-AB-UIPFE scheme. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ against the sel-FH-IND security of the underlying Π_{asi} scheme. In particular, we show that if $\mathcal{A}$ is able to break the sel-FH-IND security of MC-AB-UIPFE, then there is a PPT adversary $\mathcal{B}$ which will break the selective function-hiding security of the Π_{asi} scheme.

For μ-th ciphertext and the for all $\mathsf{aCT}_{\mu,k}$'s that the adversary obtains as a reply to the query of the form $\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k})$ and all components aSK_k's for all $k \in [n] \wedge \mathcal{R}(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k}) = 1$, that it obtains as a reply to the query of the form $\mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A})$.

In Hybrid 0, the challenger replies Π_{asi} components using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ as follows:

$$\mathsf{aCT}^{\text{Hybrid 0}}_{\mu,k} \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{s}_k, 0)]\!]_1, \mathsf{S}_{\mu,k}) = \mathsf{aEnc}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{x}}^{(0)}_{\mu,k}]\!]_1, \mathsf{S}_{\mu,k}) \text{ and}$$
$$\{\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k} \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\mathbf{y}_{\ell,k}, \boldsymbol{\alpha}_\ell, 0)]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A}) = \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{y}}^{(0)}_{\ell,k}]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A})\}_{k\in[n]}.$$

In Hybrid 1, the challenger replies Π_{asi} components using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ as follows:

$$\mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k} \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\mathbf{x}^{(0)}_{\mu,k}, 0, 1)]\!]_1, \mathsf{S}_{\mu,k}) = \mathsf{aEnc}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{x}}^{(1)}_{\mu,k}]\!]_1, \mathsf{S}_{\mu,k}) \text{ and}$$
$$\{\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k} \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\mathbf{y}_{\ell,k}, \alpha_\ell, \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle)]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A}) = \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{y}}^{(1)}_{\ell,k}]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A})\}_{k\in[n]}$$

where $\widetilde{\mathbf{x}}^{(b)}_{\mu,k} = (\mathbf{x}^{'(b)}_{\mu,k}, \mathbf{x}^{(b)}_{\mu,\mathsf{priv}})$ with $\mathbf{x}^{'(0)}_{\mu,k} = \mathbf{x}^{(0)}_{\mu,k} = \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{x}^{(0)}_{\mu,\mathsf{priv}} = (\mathbf{s}_k, 0), \mathbf{x}^{(1)}_{\mu,\mathsf{priv}} = (\mathbf{0}, 1)$ and $\widetilde{\mathbf{y}}^{(b)}_{\ell,k} = (\mathbf{y}_{\ell,k}, \mathbf{y}^{(b)}_{\ell,\mathsf{priv}})$ with $\mathbf{y}^{(0)}_{\ell,\mathsf{priv}} = (\boldsymbol{\alpha}_\ell, 0), \mathbf{y}^{(1)}_{\ell,\mathsf{priv}} = (\boldsymbol{\alpha}_\ell, \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle)$. Now, we have to show that

$$\mathsf{aDec}(\mathsf{aSK}^{\text{Hybrid 0}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 0}}_{\mu,k}) = \mathsf{aDec}(\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k}) \quad \text{for all } k \in [n], \mathcal{R}(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k}) = 1$$

holds for all key queries that made by $\mathcal{B}$. Using aDec for $k \in [n], \mathcal{R}(\mathbf{x}^{(0)}_k, \mathbf{y}_{\ell,k}) = 1$, we get

$$\begin{aligned}
&\mathsf{aDec}(\mathsf{eSK}^{\text{Hybrid 0}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 0}}_{\mu,k}) \\
&= [\![\langle \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle + 0]\!]_T \\
&= [\![\langle \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + \langle \mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}\rangle]\!]_T \\
&= [\![\langle \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + \boldsymbol{\alpha}_\ell \cdot 0 + 1 \cdot \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle]\!]_T \\
&= [\![\langle \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + \langle \mathbf{x}^{(1)}_{\mu,k}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}\rangle]\!]_T \\
&= \mathsf{aDec}(\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k}) \text{ for all } k \in [n], \mathcal{R}(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k}) = 1.
\end{aligned}$$

Therefore, $\mathcal{B}$ is an admissible adversary for the sel-IND security game of Π_{asi}. Thus, the advantage of $\mathcal{A}$ in distinguishing between Hybrid 0 and Hybrid 1 is exactly the same as the advantage in distinguishing between the experiments $\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\mathsf{sel\text{-}FH\text{-}IND}}(\lambda, 0)$ and $\mathsf{Expt}^{\mathsf{asi}}_{\mathcal{A},\mathsf{sel\text{-}IND}}(\lambda, 1)$. This completes the proof of Lemma 16. □

Lemma 17 *Hybrid 1 and Hybrid 2 are computationally indistinguishable if the underlying scheme* Π_{asi} *is function-hiding.*

Proof. The proof follows similarly as Lemma 16 using the function-hiding security of Π_{asi} scheme. We consider a PPT adversary $\mathcal{A}$ against sel-pos-IND security of the MC-AB-UIPFE scheme. We use $\mathcal{A}$ to

construct an adversary $\mathcal{B}$ against the security of the underlying Π_{asi} scheme. In particular, we show that if $\mathcal{A}$ is able to break the sel-pos-IND security of MC-AB-UIPFE, then there is a PPT adversary $\mathcal{B}$ which will break the selective function-hiding security of the Π_{asi} scheme.

For μ-th ciphertext and the for all $\mathsf{aCT}_{\mu,k}$'s that the adversary obtains as a reply to the query of the form $\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, L, \mathsf{S}_{\mu,k})$ and all components aSK_k's for all $k \in [n] \wedge \mathcal{R}(\mathbf{x}^{(b)}_{\mu,k}, \mathbf{y}_{\ell,k}) = 1, b = 1, 2$ that it obtains as a reply to the query of the form $\mathcal{O}_{\mathsf{KG}}(\mathbf{y}_\ell, \{I_{\ell,k}\}_k, \mathbb{A})$.

In Hybrid 1, the challenger replies Π_{asi} components using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ as follows:

$$\mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k} \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\mathbf{x}^{(0)}_{\mu,k}, 0, 1)]\!]_1, \mathsf{S}_{\mu,k}) = \mathsf{aEnc}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{x}}^{(1)}_{\mu,k}]\!]_1, \mathsf{S}_{\mu,k}) \text{ and}$$
$$\{\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k} \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\mathbf{y}_{\ell,k}, \alpha_\ell, \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle)]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A}) = \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{y}}^{(1)}_{\ell,k}]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A})\}_{k\in[n]}$$

In Hybrid 2, the challenger replies Π_{asi} components using the oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ and $\mathcal{O}_{\mathsf{KG}}(\cdot)$ as follows:

$$\mathsf{aCT}^{\text{Hybrid 2}}_{\mu,k} \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\mathbf{x}^{(1)}_{\mu,k}, 0, 1)]\!]_1, \mathsf{S}_{\mu,k}) = \mathsf{aEnc}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{x}}^{(1)}_{\mu,k}]\!]_1, \mathsf{S}_{\mu,k}) \text{ and}$$
$$\{\mathsf{aSK}^{\text{Hybrid 2}}_{\ell,k} \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\mathbf{y}_{\ell,k}, \alpha_\ell, \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle + \delta_{1,\ell,k})]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A}) = \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![\widetilde{\mathbf{y}}^{(1)}_{\ell,k}]\!]_2, I_{\mathbf{y}_{\ell,k}}, \mathbb{A})\}_{k\in[n]}$$

where $\widetilde{\mathbf{x}}^{(b)}_{\mu,k} = (\mathbf{x}^{'(b)}_{\mu,k}, \mathbf{x}^{(b)}_{\mu,\mathsf{priv}})$ with $\mathbf{x}^{'(0)}_{\mu,k} = \mathbf{x}^{(0)}_{\mu,k}; \mathbf{x}^{'(1)}_{\mu,k} = \mathbf{x}^{(1)}_{\mu,k}, \mathbf{x}^{(0)}_{\mu,\mathsf{priv}} = (\mathbf{0}, 1), \mathbf{x}^{(1)}_{\mu,\mathsf{priv}} = (0, 1)$ and $\widetilde{\mathbf{y}}^{(b)}_{\ell,k} = (\mathbf{y}_{\ell,k}, \mathbf{y}^{(b)}_{\ell,\mathsf{priv}})$ with $\mathbf{y}^{(0)}_{\ell,\mathsf{priv}} = (\boldsymbol{\alpha}_\ell, \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle), \mathbf{y}^{(1)}_{\ell,\mathsf{priv}} = (\boldsymbol{\alpha}_\ell, \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle + \delta_{1,\ell,k})$. Now, we have to show that

$$\mathsf{aDec}(\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k}) = \mathsf{aDec}(\mathsf{aSK}^{\text{Hybrid 2}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 2}}_{\mu,k}) \quad \text{for } k \in [n], \mathcal{R}(\mathbf{x}^{(b)}_{\mu,k}, \mathbf{y}_{\ell,k}) = 1, b = 1, 2$$

holds and all key queries that made by $\mathcal{B}$. From the admissible conditions of MCAB-UIPFE scheme and function-hiding security of Π_{asi} for each user k, we get the following constraints:

$$\langle \mathbf{x}^{(\beta)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p - \langle \mathbf{x}^{(\beta)}_{1,k}, \mathbf{y}_{\ell,k}\rangle_p = \langle \mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p - \langle \mathbf{x}^{(0)}_{1,k}, \mathbf{y}_{\ell,k}\rangle_p \quad \text{for all } \mu \in [Q_{c,k,L}]; \beta = 1, 2 \tag{4}$$

where $(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k})$ are the μ-th challenge ciphertext query to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle with the label L for the user $k \in [n]$. The above inclusion follows from the fact that the adversary can learn $\langle \mathbf{x}^{(\beta)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p - \langle \mathbf{x}^{(\beta)}_{1,k}, \mathbf{y}_{\ell,k}\rangle_p$ from challenge queries whenever $\mathbb{A}(\mathsf{S}_{\mu,k}) = 1$. It was observed in [6]. Using aDec for $k \in [n], \mathcal{R}(\mathbf{x}^{(0)}_k, \mathbf{y}_{\ell,k}) = 1$, we get

$$\begin{aligned}
&\mathsf{aDec}(\mathsf{eSK}^{\text{Hybrid 1}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k}) \\
&= [\![\langle \mathbf{x}^{'(0)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + 0 + \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle]\!]_T \\
&= [\![\langle \mathbf{x}^{'(0)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + \langle \mathbf{x}^{(0)}_{\mathsf{priv}}, \mathbf{y}^{(0)}_{\ell,\mathsf{priv}}\rangle]\!]_T \\
&= [\![\langle \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + 1 \cdot \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle + \delta_{1,\mu,k}]\!]_T \text{ from Equation 4} \\
&= [\![\langle \mathbf{x}^{'(1)}_{\mu,k}, \mathbf{y}_{\ell,k}\rangle_p + \langle \mathbf{x}^{(1)}_{\mathsf{priv}}, \mathbf{y}^{(1)}_{\ell,\mathsf{priv}}\rangle]\!]_T \\
&= \mathsf{aDec}(\mathsf{aSK}^{\text{Hybrid 1}}_{\ell,k}, \mathsf{aCT}^{\text{Hybrid 1}}_{\mu,k}) \text{ for all } k \in [n], \mathcal{R}(\mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k}) = 1.
\end{aligned}$$

□

Lemma 18 *Hybrid 2 and Hybrid 3 are computationally indistinguishable if the $\mathrm{MDDH}_{\widetilde{m}}$ assumption holds over the bilinear group G.*

Proof. We would like to prove that

$$\{[\![\boldsymbol{\alpha}_\ell]\!]_2, \{[\![\langle \boldsymbol{\alpha}_\ell, \mathbf{s}_k\rangle]\!]_2\}_{k\in\mathcal{HS}}\}_{\ell\in Q_{[\mathsf{Key}]}} \approx_c \{[\![\boldsymbol{\alpha}_\ell]\!]_2, \{[\![v_{\ell,k}]\!]_2\}_{k\in\mathcal{HS}}\}_{\ell\in Q_{[\mathsf{Key}]}} \tag{5}$$

where $Q_{[\mathsf{Key}]}$ is the number of secret key queries by the adversary to the oracle $\mathcal{O}_{\mathsf{KG}}(\cdot)$. For $k \in \mathcal{HS}$, $\boldsymbol{\alpha}_\ell \leftarrow \mathbb{Z}_p^{\widetilde{m}}$, $\mathbf{s}_k \leftarrow \mathbb{Z}_p^{\widetilde{m}}$ such that

$$\sum_{k\in\mathcal{HS}} \mathbf{s}_k + \sum_{\iota\in\mathcal{CS}} (-1)^{\iota<k} \mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L) = \mathbf{0} \implies \sum_{k\in\mathcal{HS}} v_{\ell,k} + \sum_{\iota\in\mathcal{CS}} \langle (-1)^{\iota<k} \mathsf{PRF}^{\mathsf{seed}_{\iota,k}}(L), \mathbf{s}_\iota \rangle = 0.$$

The above indistinguishability of Equation 5 can be shown using the following $\text{MDDH}_{\widetilde{m}}$ instances:

$$\{[\![\mathbf{A}]\!]_2, [\![\mathbf{A}\mathbf{t}_1]\!]_2, [\![\mathbf{A}\mathbf{t}_2]\!]_2, \dots [\![\mathbf{A}\mathbf{t}_d]\!]_2\} \approx_c \{[\![\mathbf{A}]\!]_2, [\![\mathbf{r}_1]\!]_2, [\![\mathbf{r}_2]\!]_2, \dots [\![\mathbf{r}_d]\!]_2\} \tag{6}$$

where $d > 1$, $m \in \mathbb{N}$ and $\mathbf{c} \in \mathbb{Z}_p^{\widetilde{m}}$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{m\times\widetilde{m}}$ and $\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_d \leftarrow \mathbb{Z}_p^{\widetilde{m}}$ satisfying $\sum_{\jmath\in[d]} \mathbf{t}_\jmath = \mathbf{c}$, which implies $\sum_{\jmath\in[d]} \mathbf{r}_\jmath = \mathbf{A}\mathbf{c}$. Thus the above relation in Equation 6 can be written as

$$\{[\![\mathbf{A}]\!]_2, [\![\mathbf{A}\mathbf{t}_1]\!]_2, [\![\mathbf{A}\mathbf{t}_2]\!]_2, \dots, [\![\mathbf{A}\mathbf{c} - \sum_{\jmath\in[d-1]} \mathbf{A}\mathbf{m}_\jmath]\!]_2\} \approx_c \{[\![\mathbf{A}]\!]_2, [\![\mathbf{r}_1]\!]_2, [\![\mathbf{r}_2]\!]_2, \dots, [\![\mathbf{A}\mathbf{c} - \sum_{\jmath\in[d-1]} \mathbf{r}_\jmath]\!]_2\}$$

using the similar $d-1$ folds $\text{MDDH}_{\widetilde{m}}$ assumption, we get that

$$\{[\![\mathbf{A}]\!]_2, [\![\mathbf{A}\mathbf{t}_1]\!]_2, [\![\mathbf{A}\mathbf{t}_2]\!]_2, \dots, [\![\mathbf{A}\mathbf{t}_{d-1}]\!]_2\} \approx_c \{[\![\mathbf{A}]\!]_2, [\![\mathbf{r}_1]\!]_2, [\![\mathbf{r}_2]\!]_2, \dots, [\![\mathbf{r}_{d-1}]\!]_2\}$$

□

Lemma 19 *Hybrid 3 and Hybrid 4 are identically distributed.*

Proof. From the admissibilty condition of the MC-AB-UIPFE, we have

$$\sum_{k\in\mathcal{HS}} \langle \mathbf{x}^{(0)}_{1,k,i}, \mathbf{y}_{\ell,k,i} \rangle_p = \sum_{k\in\mathcal{HS}} \langle \mathbf{x}^{(1)}_{1,k,i}, \mathbf{y}_{\ell,k,i} \rangle_p \ .$$

In both Hybrid 3 and Hybrid 4, $\{v_{\ell,k}\}_{k\in\mathcal{HS}}$ and $\{v_{\ell,k} + \delta_{1,\ell,k}\}_{k\in\mathcal{HS}}$ are randomly distributed over $\mathbb{Z}_p$. Also, note that

$$\sum_{k\in\mathcal{HS}} v_{\ell,k} = \sum_{k\in\mathcal{HS}} v_{\ell,k} + \delta_{1,\ell,k} = -\sum_{k\in\mathcal{CS}} \langle \alpha_\ell, \mathbf{s}_k \rangle \ ,$$

Therefore, Hybrid 3 ≡ hybrid 4. □

Lemma 20 *Hybrid 4 and Hybrid 5 are computationally indistinguishable if the $\text{MDDH}_{\widetilde{m}}$ assumption holds over the bilinear group G.*

Proof of the above Lemma follows similarly as Lemma 18.

Lemma 21 *Hybrid 5 and Hybrid 6 are computationally indistinguishable if the underlying scheme Π_{asi} is function-hiding.*

Proof of the above lemma follows similarly as Lemma 16.

This completes the proof of Theorem 8. □

7 Dynamic Decentralized UIPFE

In this section, we define the *dynamic decentralized unbounded FE* (DD-UFE) scheme over key space $\mathcal{K}$, message space $\mathcal{M}$, and set of identities $\mathcal{ID}$ for functionality $f : L(\mathcal{ID} \times \mathcal{K}^*) \times L(\mathcal{ID} \times \mathcal{M}^*) \to \mathcal{Z}$ where $L(S)$ to denote the set of finite lists of elements from S.

Definition 18 A DD-UFE scheme $\Pi_{\mathsf{ddf}} = (\mathsf{GlobalSetup}, \mathsf{LocalSetup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of following five algorithms:

$\mathsf{GlobalSetup}(1^\lambda) \to \mathsf{PP}$: The global setup algorithm takes as input security parameter λ and outputs a public parameter PP. Those parameters are implicit arguments to all the other algorithms.

$\mathsf{LocalSetup}(\mathsf{PP}) \to (\mathsf{PK}_k, \mathsf{MSK}_k)$: The local setup algorithm takes as input public parameter PP and outputs a local public parameter PK_k and a master secret key MSK_k for $k \in \mathcal{ID}$. The following three algorithms implicitly take PK_k.

$\mathsf{KeyGen}(\mathsf{MSK}_k, \{(\mathsf{Key}_{k,j})_{j\in I_k}\}_k) \to \mathsf{SK}_k$: The key generation algorithm takes as input MSK_k, and a key space object $(\mathsf{Key}_{k,j})_{j\in I_k}$ with the associated index set I_k. It outputs a private key SK_k.

$\mathsf{Enc}(\mathsf{MSK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k}) \to \mathsf{CT}_k$: The encryption algorithm takes as input MSK_k, and a message $(\mathsf{Msg}_{k,j})_{j\in I'_k}$ with the associated index set I'_k. It outputs a ciphertext CT_k.

$\mathsf{Dec}(\{\mathsf{SK}_k\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}}) \to \zeta \vee \perp$: The decryption algorithm takes as input $\{\mathsf{SK}_k\}_{k\in\mathcal{U}_{\mathsf{Key}}}$, $\{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}}$ where $\mathcal{U}_{\mathsf{Key}}, \mathcal{U}_{\mathsf{Msg}} \subseteq \mathcal{ID}$ are any sets. It outputs either ζ or a special symbol $\perp$ indicating failure.

Correctness: For all $\lambda \in \mathbb{N}, \mathcal{U}_{\mathsf{Key}}, \mathcal{U}_{\mathsf{Msg}} \subseteq \mathcal{ID}, \{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{i\in\mathcal{U}_{\mathsf{Key}}} \in L(\mathcal{ID}\times\mathcal{K}^*), \{k, (\mathsf{Msg}_{k,j})_{j\in I'_k}\}_{i\in\mathcal{U}_{\mathsf{Msg}}} \in L(\mathcal{ID}\times\mathcal{M}^*)$, the following must hold

$$\Pr\left[\begin{array}{l|l} \zeta = f\big(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in\mathcal{U}_{\mathsf{Key}}}, & \mathsf{PP} \leftarrow \mathsf{GlobalSetup}(1^\lambda) \\ \quad \{k, (\mathsf{Msg}_{k,j})_{j\in I'_k}\}_{k\in\mathcal{U}_{\mathsf{Msg}}}\big) & (\mathsf{PK}_k, \mathsf{MSK}_k) \leftarrow \mathsf{LocalSetup}(\mathsf{PP}) \\ & \mathsf{SK}_k \leftarrow \mathsf{KeyGen}(\mathsf{MSK}_k, \{(\mathsf{Key}_{k,j})_{j\in I_k}\}_k) \\ & \mathsf{CT}_k \leftarrow \mathsf{Enc}(\mathsf{MSK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k}) \\ & \zeta \leftarrow \mathsf{Dec}(\{\mathsf{SK}_k\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 19 (Security of DD-UFE) The $\Pi_{\mathsf{ddf}} = (\mathsf{GlobalSetup}, \mathsf{LocalSetup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be xx-yy-*indistinguishability* (xx-yy-IND) (xx $\in \{\mathsf{sel}, \mathsf{adp}\}$, yy $\in \{\mathsf{sym}, \mathsf{asym}\}$) secure if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that the following holds

$$\mathsf{Adv}^{\mathsf{ddf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda) = \left|\Pr\left[\mathsf{Expt}^{\mathsf{ddf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{ddf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{ddf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{ddf}}_{\mathcal{A},\text{xx-yy-IND}}(\lambda, \beta):}$
1: $\mathsf{PP} \leftarrow \mathsf{GlobalSetup}(1^\lambda)$.
2: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{HonGen}}(\cdot), \mathcal{O}_{\mathsf{Corr}}(\cdot), \mathcal{O}_{\mathsf{KG}}(\mathsf{MSK},\cdot), \mathcal{O}_{\mathsf{E}}(\cdot), \mathcal{O}_{\mathsf{LoR},\beta}(\cdot)}(\mathsf{PP})$.
3: Output β' if condition $(*)$ is satisfied.

$\underline{\mathcal{O}_{\mathsf{Corr}}(k):}$
output MSK_k.

$\underline{\mathcal{O}_{\mathsf{HonGen}}(k):}$
output PK_k.

$\underline{\mathcal{O}_{\mathsf{KG}}(\{(\mathsf{Key}_{k,j})_{j\in I_k}\}_k):}$
output $\mathsf{KeyGen}(\mathsf{MSK}_k, (\mathsf{Key}_{k,j})_{j\in I_k})$.

$\underline{\mathcal{O}_{\mathsf{E}}(k, (\mathsf{Msg}_{k,j})_{j\in I'_k}):}$
output $\mathsf{Enc}(\mathsf{MSK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k})$.

$\underline{\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k}):}$
output $\mathsf{Enc}(\mathsf{MSK}_k, (\mathsf{Msg}^\beta_{k,j})_{j\in I'_k})$.

Let $\mathcal{Q}, \mathcal{CS}$ be the sets of all inputs $k \in \mathcal{ID}$ for which the adversary makes queries to the oracles $\mathcal{O}_{\mathsf{HonGen}}(\cdot)$ and $\mathcal{O}_{\mathsf{Corr}}(\cdot)$ respectively, and $\mathcal{HS} = \mathcal{Q} \setminus \mathcal{CS}$. The condition $(*)$ is that if there exist two subsets of identities $\mathcal{U}_{\mathsf{Key}}, \mathcal{U}_{\mathsf{Msg}} \subseteq \mathcal{Q}$ satisfying

$$f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in\mathcal{U}_K}, \{k, (\mathsf{Msg}^0_{k,j})_{j\in I'_k}\}_{k\in\mathcal{U}_{\mathsf{Msg}}}) \quad \neq f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \{k, (\mathsf{Msg}^1_{k,j})_{j\in I'_k}\}_{k\in\mathcal{U}_{\mathsf{Msg}}})$$

then at least one of the following should *not* hold:

- for all $k \in \mathcal{U}_{\mathsf{Msg}}$, $[\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k})$ or $\mathcal{O}_{\mathsf{E}}(k, (\mathsf{Msg}_{k,j})_{j\in I'_k})$ with $(\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j} = \mathsf{Msg}_{k,j})_{j\in I'_k}$ is queried] or $[(\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j} = \mathsf{Msg}_{k,j})_{j\in I'_k}$ and $k \in \mathcal{CS}]$.

- for all $k \in \mathcal{U}_{\mathsf{Key}}$, $[\mathcal{O}_{\mathsf{KG}}(k, (\mathsf{Key}_{k,j})_{j\in I_k})$ is queried] or $[k \in \mathcal{CS}]$.

– For xx = sel: Generates the $\mathcal{CS}$ set in one shot before queries to all oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ or $\mathcal{O}_{\mathsf{E}}(\cdot)$ or $\mathcal{O}_{\mathsf{KG}}(\cdot)$.
– For yy = sym: for $i \in \mathcal{CS}$, the queries $\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k})$ must satisfy $\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j}$.

Definition 20 (DD-UFE for IP) A dynamic decentralized unbounded IPFE (DD-UIPFE) is a particular class of DD-UFE where $\mathcal{ID} = \{0,1\}^*, \mathcal{K}^* = \mathbb{Z}_p^*$, and $\mathcal{M}^* = \mathbb{Z}_p^* \times \mathcal{L}$ such that $\mathcal{L}$ represents the label space. The function f is defined as follows: for the message components $\mathsf{Msg}_k = (\mathbf{x}_k, L_k) \in \mathcal{M}^*$, the key components $\mathsf{Key}_k = (\{\mathbf{y}_k\}_k) \in \mathcal{K}^*$ and $\mathbf{x}_k$ and $\mathbf{y}_k$ are associated with the index sets I_k and I'_k,

$$f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{\mathcal{U}_{\mathsf{Key}}}, \{k, (\mathsf{Msg}_{k,j})_{j\in I'_k}\}_{\mathcal{U}_{\mathsf{Msg}}}) = \begin{cases} \sum_{k\in\mathcal{U}} \langle \mathbf{x}_k, \mathbf{y}_k \rangle_p & \text{if } (\star) \text{ holds} \\ \perp & \text{otherwise.} \end{cases}$$

The conditions in $(\star)$ define as follows:

– $\mathcal{U}_{\mathsf{Msg}} = \mathcal{U}_{\mathsf{Key}} = \mathcal{U}$ and for all $k \in \mathcal{U}$, $\mathcal{U}_{\mathsf{Key},k} = \mathcal{U}_{\mathsf{Msg},k} = \mathcal{U}$.
– $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_k) = 1$ for all $k \in \mathcal{U}$.
– for all $k_1, k_2 \in \mathcal{U}$, $L_{k_1} = L_{k_2}$.

7.1 Construction

Let $\Pi_{\mathsf{esi}} = (\mathsf{eSetup}, \mathsf{eKeyGen}, \mathsf{eEnc}, \mathsf{eSlotEnc}, \mathsf{eDec})$ be an esUIPFE with $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^*, \mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{m+1}$ considering $n_1 = 0$ and $n_2 = m + 1$, and $\Pi_{\mathsf{aone}} = (\mathsf{aoGlobalSetup}, \mathsf{aoLocalSetup}, \mathsf{aoEnc}, \mathsf{aoDec})$ be an AoNE scheme, $\Pi_{\mathsf{nike}} = (\mathsf{nSetup}, \mathsf{nKeyGen}, \mathsf{nKeyshared})$ be a NIKE scheme, $\mathsf{PRF}_1^{\mathsf{seed}} : 2^{\mathcal{ID}} \times \mathcal{L} \to \mathbb{Z}_p^m, \mathsf{PRF}_2^{\mathsf{seed}} : 2^{\mathcal{ID}} \to \mathbb{Z}_p^m$ be the families of PRF functions with the key space $\mathcal{K}_{\mathsf{prf}_1}, \mathcal{K}_{\mathsf{prf}_2}$ respectively, $\mathcal{ID}$ be the identity space and a hash function $H : \{0,1\}^* \to \mathbb{G}_2^m$ is treated as the random oracle. We discuss AoNE, NIKE, PRF in Definitions 7, 5 and 4, respectively. Note that, our proposed DD-UIPFE only involve the eEnc algorithm to encrypt the slot-specified message vector using corresponding master secret key. We present our DD-UIPFE scheme $\Pi_{\mathsf{ddi}} = (\mathsf{GlobalSetup}, \mathsf{LocalSetup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in following.

$\mathsf{GlobalSetup}(1^\lambda)$: The setup algorithm takes input the security parameter λ and executes the following steps:

1. Generates $\mathsf{nPP} \leftarrow \mathsf{nSetup}(1^\lambda), \mathsf{aoPP} \leftarrow \mathsf{aoSetup}(1^\lambda)$.
2. Outputs the public parameter $\mathsf{PP} = (\mathsf{nPP}, \mathsf{aoPP})$.

$\mathsf{LocalSetup}(\mathsf{PP})$: The local setup algorithm takes input PP with a user identity $k \in \mathcal{ID}$, and runs the following steps:

1. Generates $(\mathsf{nPK}_k, \mathsf{nSK}_k) \leftarrow \mathsf{nKeyGen}(\mathsf{nPP}), (\mathsf{aoPK}_k, \mathsf{aoSK}_k) \leftarrow \mathsf{aoKeyGen}(\mathsf{aoPP})$.
2. Chooses $\mathsf{seed}_{k,2} \leftarrow \mathcal{K}_{\mathsf{prf}_2}$.
3. Outputs the public key $\mathsf{PK}_k = (\mathsf{nPK}_k, \mathsf{aoPK}_k)$ and the master secret key $\mathsf{MSK}_k = (\mathsf{nSK}_k, \mathsf{aoSK}_k, \mathsf{seed}_{k,2})$ for $k \in \mathcal{ID}$.

$\mathsf{KeyGen}(\mathsf{MSK}_k, \{\mathbf{y}_k = (y_{k,\iota})_{\iota\in I_{\mathbf{y}_k}}, I_{\mathbf{y}_k}\}_{k\in\mathcal{U}_{\mathsf{Key},k}})$: The key generation algorithm takes as input MSK_k and $\mathsf{Key}_k = (\{\mathbf{y}_k\}_k, \{I_{\mathbf{y}_k}\}_k, \mathcal{U}_{\mathsf{Key},k})$ and performs the following steps:

1. Computes $\mathsf{rt}_k \leftarrow \mathsf{PRF}_2^{\mathsf{seed}_{k,2}}(\mathcal{U}_{\mathsf{Key},k})$.
2. Runs $(\mathsf{eMPK}_k, \mathsf{eMSK}_k) \leftarrow \mathsf{eSetup}(1^\lambda, \mathsf{rt}_k)$.
3. Computes $\mathsf{eSK}_k \leftarrow \mathsf{eKeyGen}(\mathsf{eMSK}_k, [\![(\mathbf{y}_k, \boldsymbol{\alpha}, 0)]\!]_2, I_{\mathbf{y}_k})$ with $H(\{\mathbf{y}_k\}_k, \mathcal{U}_{\mathsf{Key},k}) = [\![\boldsymbol{\alpha}]\!]_2$.
4. Generates $\mathsf{aoCT}_k \leftarrow \mathsf{aoEnc}(\mathsf{aoSK}_k, (\mathsf{eSK}_k, \mathcal{U}_{\mathsf{Key},k}, \{\mathbf{y}_k\}_k))$.
5. Outputs the secret key $\mathsf{SK}_k = (\mathsf{aoCT}_k, \mathcal{U}_{\mathsf{Key},k}, \{\mathbf{y}_k\}_k, L_k)$.

$\mathsf{Enc}(\mathsf{MSK}_k, \mathbf{x}_k = (x_{k,i})_{i\in[m_k]}, \mathcal{U}_{\mathsf{Msg},k}, L_k)$: The encryption algorithm takes as input MSK_k, and $\mathsf{Msg}_k = (\mathbf{x}_k, \mathcal{U}_{\mathsf{Msg},k}, L_k)$ and proceeds as follows:

1. Runs $(\mathsf{eMPK}_k, \mathsf{eMSK}_k) \leftarrow \mathsf{eSetup}(1^\lambda, \mathsf{rt}_k)$ where $\mathsf{rt}_k \leftarrow \mathsf{PRF}_2^{\mathsf{seed}_{k,2}}(\mathcal{U}_{\mathsf{Msg},k})$.

2. Generates $\mathsf{seed}_{k,\nu} \leftarrow \mathsf{nKeyShared}(\mathsf{nSK}_k, \mathsf{nPK}_\nu)$ for all $\nu \in \mathcal{U}_{\mathsf{Msg},k} \setminus \{k\}$.
3. Computes $\mathbf{s}_k = \sum_{\nu \in \mathcal{U}_{\mathsf{Msg},k} \setminus \{k\}} (-1)^{\nu<k} \mathsf{PRF}_1^{\mathsf{seed}_{k,\nu},1}(\mathcal{U}_{\mathsf{Msg},k}, L_k)$.
4. Generates $\mathsf{aoCT}_k \leftarrow \mathsf{aoEnc}(\mathsf{aoSK}_k, (\mathsf{eCT}_k, \mathcal{U}_{\mathsf{Msg},k}, L_k))$ where $\mathsf{eCT}_k \leftarrow \mathsf{eEnc}(\mathsf{eMSK}_k, [\![(\mathbf{x}_k, \mathbf{s}_k, 0)]\!]_1)$.
5. Outputs the ciphertext $\mathsf{CT}_k = (\mathsf{aoCT}_k, \mathcal{U}_{\mathsf{Msg},k}, L_k)$.

$\mathsf{Dec}(\{\mathsf{SK}_k\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}})$: The decryption algorithm takes as input $\{\mathsf{SK}_k\}_{k\in\mathcal{U}_{\mathsf{key}}}$, $\{\mathsf{CT}_k\}_{k\in\mathcal{U}_{\mathsf{Msg}}}$ such that $\mathcal{U} = \mathcal{U}_{\mathsf{Key}} = \mathcal{U}_{\mathsf{Msg}}$ and performs the following steps:

1. For all $k \in \mathcal{U}$, computes $\widetilde{\mathsf{eSK}_k} \leftarrow \mathsf{aoDec}(\mathsf{aoCT}_k)$ and $\widetilde{\mathsf{eCT}_k} \leftarrow \mathsf{aoDec}(\mathsf{aoCT}'_k)$.
2. Generates $[\![\xi_k]\!]_T \leftarrow \mathsf{eDec}(\widetilde{\mathsf{eSK}_k}, \widetilde{\mathsf{eCT}_k})$ for all $k \in \mathcal{U}$.
3. Outputs $[\![d]\!]_T = \prod_{k\in\mathcal{U}} [\![\xi_k]\!]_T$
4. If eDec returns $\perp$, outputs $\perp$.

Correctness: Firstly, we observe that if $\mathcal{U}_{\mathsf{Key}} = \mathcal{U}_{\mathsf{Msg}} = \mathcal{U}$, $L_k = L_{\mathsf{Msg}}$ for all $k \in \mathcal{U}$, where L_{Msg} is any label in $\mathcal{L}$ and $\{\mathbf{y}_k = (y_{k,\iota})_{\iota\in I_{\mathbf{y}_k}}, I_{\mathbf{y}_k}\}_{k\in\mathcal{U}_{\mathsf{Key},k}}$ is same in all the ciphertexts input to the decryption algorithm, then

- From Π_{aone} correctness, we have $\mathsf{eSK}_{\mathsf{fe},k} = \widetilde{\mathsf{eSK}_{\mathsf{fe},k}}$, $\mathsf{eCT}_{\mathsf{fe},k} = \widetilde{\mathsf{eCT}_{\mathsf{fe},k}}$, for all $k \in \mathcal{U}$.
- For all $k \in \mathcal{U}$, the computation $H(\{\mathbf{y}_k\}_k, \mathcal{U}_{\mathsf{Key},k}) = [\![\boldsymbol{\alpha}]\!]_2$ remains same.

From the correctness of Π_{nike}, we have $\mathsf{seed}_{k,\nu} = \mathsf{seed}_{\nu,k}$ and $\sum_{k\in\mathcal{U}} \mathbf{s}_k = \mathbf{0}$. Now applying the correctness of Π_{esi} with $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_k) = 1$ for all $k \in \mathcal{U}$, we get

$$\mathsf{eDec}(\widetilde{\mathsf{eSK}_k}, \widetilde{\mathsf{eCT}_k}) = [\![\langle \mathbf{x}_k, \mathbf{y}_k\rangle_p + \langle \mathbf{s}_k, \boldsymbol{\alpha}\rangle]\!]_T \text{ for all } k \in \mathcal{U}.$$

Therefore, $$[\![d]\!]_T = \prod_{k\in\mathcal{U}} [\![\langle \mathbf{x}_k, \mathbf{y}_k\rangle_p + \langle \mathbf{s}_k, \boldsymbol{\alpha}\rangle]\!]_T = [\![\sum_{k\in\mathcal{U}} \langle \mathbf{x}_k, \mathbf{y}_k\rangle_p]\!]_T.$$

7.2 Security Analysis

In Theorem 9, we present the security analysis of our DD-UIPFE scheme, as described in Construction 7.1.

Theorem 9 *Our Π_{ddi} scheme achieves sel-sym-IND security as per the Definition 20 if $\mathsf{PRF}_1, \mathsf{PRF}_2$ are pseudo-random functions, Π_{nike} and Π_{aone} are IND-secure protocols and Π_{esi} is function-hiding.*

Proof. We prove the above theorem through a sequence of hybrids. We describe the hybrids below. We represent the slots that are updated in the subsequent hybrids using dashed boxes.

Hybrid 0: This game is the same as $\mathsf{Exp}^{\mathsf{ddi}}_{\mathcal{A},\mathsf{sel\text{-}IND}}(\lambda, 0)$. The adversary $\mathcal{A}$ has access to the following oracles.

- **Corruption queries:** The adversary $\mathcal{A}$ submits the corrupted user index $k \in \mathcal{ID}$ to challenger $\mathcal{B}$ and the challenger returns keys $\mathsf{MPK}_k, \mathsf{MSK}_k$ corresponding to the user k.
- **Left or right oracle queries:** On receiving index $k \in \mathcal{ID}$, identity set $\mathcal{U}_{\mathsf{Msg}}$, label L_{Msg} and the challenge messages $\{\mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}\}_{k\in\mathcal{U}_{\mathsf{Msg}}}$ of length m_k, the challenger simulates the challenge ciphertexts $\mathsf{CT}_k^{(0)} = (\mathsf{aoCT}_k, \mathcal{U}_{\mathsf{Msg},k}, L_{\mathsf{Msg}})$ using the following component:

$$\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}}): \quad \mathsf{eCT}_k^{(0)} = \mathsf{eEnc}(\quad \mathsf{eMSK}_k, \quad [\![(\boxed{\mathbf{x}_k^{(0)}}, \quad \boxed{\mathbf{s}_k}, \quad \boxed{0})]\!]_1 \quad)$$

where $\mathsf{aoCT}_k \leftarrow \mathsf{aoEnc}(\mathsf{aoMSK}_k, (\mathsf{eCT}_k^{(0)}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}}))$, $\mathbf{s}_k = \sum_{\nu\in\mathcal{U}_{\mathsf{Msg},k}\setminus\{k\}} (-1)^{\nu<k} \mathsf{PRF}_1^{\mathsf{seed}_{k,\nu},1}(\mathcal{U}_{\mathsf{Msg},k}, L_{\mathsf{Msg}})$.

– **Encryption oracle queries:** On receiving index $k \in \mathcal{ID}$, message vector $\mathbf{x}_k$, identity set $\mathcal{U}_{\mathsf{Msg},k}$ and label L_{Msg}, the challenger generates the queried ciphertexts $\mathsf{CT}_k = (\mathsf{aoCT}_k, \mathcal{U}_{\mathsf{Msg},k}, L'_{\mathsf{Msg}})$ using the following component:

$$\mathcal{O}_{\mathsf{E}}(k, \mathbf{x}_k, \mathcal{U}_{\mathsf{Msg}}, L'_{\mathsf{Msg}}): \quad \mathsf{eCT}'_k = \quad \mathsf{eEnc}(\quad \mathsf{eMSK}_k, \quad [\![(\mathbf{x}_k, \quad \mathbf{s}'_k, \quad 0)]\!]_1 \quad)$$

where $\mathsf{aoCT}_k \leftarrow \mathsf{aoEnc}(\mathsf{aoMSK}_k, (\mathsf{eCT}'_k, \mathcal{U}_{\mathsf{Msg}}, L'_{\mathsf{Msg}}))$ and and $\mathbf{s}'_k = \sum_{\nu \in \mathcal{U}_{\mathsf{Msg},k} \setminus \{k\}} (-1)^{\nu<k} \mathsf{PRF}_1^{\mathsf{seed}_{k,\nu,1}}(\mathcal{U}_{\mathsf{Msg},k}, L'_{\mathsf{Msg}})$.

– **Key generation oracle queries:** For ℓ-th functional key corresponding to the access structure $\mathbb{A}$, index $k \in \mathcal{ID}$, key vector $\mathbf{y}_\ell = (\mathbf{y}_{\ell,k})_{k \in \mathcal{U}_{\mathsf{Key}}}$, the challenger generates the secret key $\mathsf{SK}_{\ell,k} = (\mathsf{aoCT}_{\ell,k}, , \mathcal{U}_{\mathsf{Key}}, \{\mathbf{y}_{\ell,k}\}_k)$ using the following components:

$$\mathcal{O}_{\mathsf{KG}}(k, \{\mathbf{y}_{\ell,k}\}_k, \mathcal{U}_{\mathsf{Key}}): \quad \mathsf{eSK}_{\ell,k} = \quad \mathsf{eKeyGen}(\quad \mathsf{eMSK}_k, \quad [\![(\mathbf{y}_{\ell,k}, \quad \boxed{\boldsymbol{\alpha}_\ell}, \quad \boxed{0})]\!]_2 \quad)$$

Here, $\mathsf{aoCT}_{\ell,k} \leftarrow \mathsf{aoEnc}(\mathsf{aoMSK}_k, (\mathsf{eSK}_{\ell,k}, \mathcal{U}_{\mathsf{Key}}, \{\mathbf{y}_{\ell,k}\}_k))$ and $H(\{\mathbf{y}_{\ell,k}\}_k, \mathcal{U}_{\mathsf{Key},k}) = [\![\boldsymbol{\alpha}_\ell]\!]_2$.

Hybrid 1: In this hybrid, we modify the incomplete LoR queries. The queries of the form $(k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}})$ to oracles $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ or $\mathcal{O}_{\mathsf{KG}}(\cdot)$ is said to be *incomplete* with respect to $(\mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}})$ if there exists an index $k' \in \mathcal{U}_{\mathsf{Msg}} \cap \mathcal{HS}$ such that no $\mathcal{O}_{\mathsf{E}}(k', \mathbf{x}_{k'}, \mathcal{U}_{\mathsf{Msg}}, L'_{\mathsf{Msg}})$ or $\mathcal{O}_{\mathsf{LoR},\beta}(k', \mathbf{x}_{k'}^{(0)}, \mathbf{x}_{k'}^{(1)}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}})$ queries are made. In case of such incomplete queries, the challenge ciphertext $\mathsf{CT}_k^{(0)} = (\mathsf{aoCT}_k, \mathcal{U}_{\mathsf{Msg},k}, L_{\mathsf{Msg}})$ is computed as follows:

$$\mathsf{aoEnc}(\mathsf{aoMSK}_k, (\boxed{0}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}})) \rightarrow \mathsf{aoCT}_k.$$

The indistinguishability follows from the security of the Π_{aone}.

Hybrid 2 We handle incomplete key queries in this game. A key query $(k, \{\mathbf{y}_{\ell,k}\}_{k \in \mathcal{U}_{\mathsf{Key}}}, \mathcal{U}_{\mathsf{Key}})$ is said to be *incomplete* if there exists an $k' \in \mathcal{U}_{\mathsf{Key}} \cap \mathcal{HS}$ such that there is no key query of the form $(k', \{\mathbf{y}_{\ell,k}\}_{k \in \mathcal{U}_{\mathsf{Key}}}, \mathcal{U}_{\mathsf{Key}})$. For all incomplete key queries, the secret key $\mathsf{SK}_{\ell,k} = (\mathsf{aoCT}_{\ell,k}, , \mathcal{U}_{\mathsf{Key},k}, \{\mathbf{y}_{\ell,k}\}_k)$ are computed as follows:

$$\mathsf{aoEnc}(\mathsf{aoMSK}_k, (\boxed{0}, \mathcal{U}_{\mathsf{Key}}, \{\mathbf{y}_{\ell,k}\}_{k \in \mathcal{U}_{\mathsf{Key}}})) \rightarrow \mathsf{aoCT}_{\ell,k}.$$

The indistinguishability follows from the security of Π_{aone}.

Hybrid 3: This game is the same as the experiment $\mathsf{Exp}^{\mathsf{ddi}}_{\mathcal{A},\mathsf{sel\text{-}IND}}(\lambda, 1)$ for the challenge bit $\beta = 1$. The challenger generates the challenge ciphertext $\mathsf{CT}_k^{(0)} = (\mathsf{aoCT}_k, \mathcal{U}_{\mathsf{Msg},k}, L_{\mathsf{Msg}})$ using the following component:

$$\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}}): \quad \mathsf{eCT}_k^{(1)} = \quad \mathsf{eEnc}(\quad \mathsf{eMSK}_k, \quad [\![(\boxed{\mathbf{x}_k^{(1)}}, \quad \mathbf{s}_k, \quad 0)]\!]_1 \quad)$$

where $\mathbf{s}_k = \sum_{\nu \in \mathcal{U}_{\mathsf{Msg},k} \setminus \{k\}} (-1)^{\nu<k} \mathsf{PRF}_1^{\mathsf{seed}_{k,\nu,1}}(\mathcal{U}_{\mathsf{Msg},k}, L_{\mathsf{Msg}})$.

Lemma 22 *Hybrid 2 and Hybrid 3 are computationally indistinguishable if the $MDDH_m$ assumption holds over the bilinear group G.*

Proof. We prove the above lemma through a series of hybrids. Let q_u be the total number of ID-sets with complete LoR queries. Let $\{\mathcal{U}_1, \cdots, \mathcal{U}_{q_u}\}$ be some fixed ordering of the ID sets with an upper bound Q_u on q_u. We define the sub-hybrids $\widetilde{\mathrm{H}}_\varrho^0$ as follows,

$\widetilde{\mathrm{H}}^0_\varrho$: For $\varrho \in [Q_u] \cup \{0\}$, this is the same as Hybrid 2 except that for every complete LoR query, for $k \in \mathcal{HS}$, the challenger sets

$$\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_{\mathsf{Msg}}, L_{\mathsf{Msg}}): \quad \mathsf{eCT}_k^{(0)} = \begin{cases} \mathsf{eEnc}(\ \mathsf{eMSK}_k,\ [\![(\ \mathbf{x}_k^{(1)},\ \mathbf{s}_k,\ 0)]\!]_1\ ) & \text{if } \mathcal{U}_{\mathsf{Msg}} \in \{\mathcal{U}_1, \cdots, \mathcal{U}_\varrho\} \\ \mathsf{eEnc}(\ \mathsf{eMSK}_k,\ [\![(\mathbf{x}_k^{(0)},\ \mathbf{s}_k,\ 0)]\!]_1\ ) & \text{if } \mathcal{U}_{\mathsf{Msg}} \in \{\mathcal{U}_{\varrho+1}, \cdots, \mathcal{U}_{q_u}\}\ . \end{cases}$$

Consider $\mathcal{U}_\varrho = \{\perp\}$ for $\varrho > q_u$. Observe that, Hybrid 2 $\equiv \widetilde{\mathrm{H}}^0_0$ and $\widetilde{\mathrm{H}}^0_{Q_u} \equiv$ Hybrid 3. In the following, we show that $\widetilde{\mathrm{H}}^0_{\varrho-1} \approx_c \widetilde{\mathrm{H}}^0_\varrho$.

Claim 9 For $\varrho \in [Q_u]$, $\widetilde{\mathrm{H}}^0_{\varrho-1}$ and $\widetilde{\mathrm{H}}^0_\varrho$ are computationally indistinguishable.

Proof. To show this, assume $L^1_{\mathcal{U}_\varrho}, \cdots, L^v_{\mathcal{U}_\varrho}$ be the labels queried on the ID set $\mathcal{U}_\varrho$ and Q_L be the upper bound on v. We introduce a series of hybrids $\widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta}$ where $\vartheta \in [Q_L]$ based on the complete query of the form $(\star, \star, \star, \mathcal{U}_\varrho, \star)$.

$\widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta}$: For $\vartheta \in [Q_L]$, this is the same as $\widetilde{\mathrm{H}}^0_{\varrho-1}$ except that for every complete query of the form $(k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_\varrho, L)$ to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$, for $k \in \mathcal{HS}$, the challenger sets

$$\mathcal{O}_{\mathsf{LoR},\beta}(k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_\varrho, L): \quad \mathsf{eCT}_k^{(0)} = \begin{cases} \mathsf{eEnc}(\ \mathsf{eMSK}_k,\ [\![(\ \mathbf{x}_k^{(1)},\ \mathbf{s}_k,\ 0)]\!]_1\ ) & \text{if } L \in \{L^1_{\mathcal{U}_\varrho}, \cdots, L^\vartheta_{\mathcal{U}_\varrho}\} \\ \mathsf{eEnc}(\ \mathsf{eMSK}_k,\ [\![(\mathbf{x}_k^{(0)},\ \mathbf{s}_k,\ 0)]\!]_1\ ) & \text{if } L \in \{L^{\vartheta+1}_{\mathcal{U}_\varrho}, \cdots, L^{Q_L}_{\mathcal{U}_\varrho}\}\ . \end{cases}$$

We define another hybrid $\widetilde{\mathrm{H}}^0_{\varrho-1,0} \equiv \widetilde{\mathrm{H}}^0_{\varrho-1}$. Observe that $\widetilde{\mathrm{H}}^0_{\varrho-1,Q_L} \equiv \widetilde{\mathrm{H}}^0_\varrho$. We have to prove $\widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta-1} \approx_c \widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta}$ to complete the cycle of hybrids.

Claim 10 For $\vartheta \in [Q_L], \widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta-1}$ and $\widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta}$ are computationally indistinguishable.

Proof. We define the identity set $\mathcal{U}_\varrho^{\mathcal{HS}} = \mathcal{HS} \cap \mathcal{U}_\varrho = \{u_1, \ldots, u_w\}$ with Q_w as upper bound on w. In the following, we consider the sequence of hybrids $\overline{\mathrm{H}}^0_\eta$ for $(\eta \in [Q_w] \cup \{0\})$ based on the each complete encryption query of the form $(u_k, \mathbf{x}_{u_k}^{(0)}, \mathbf{x}_{u_k}^{(1)}, \mathcal{U}_\varrho, L^\vartheta_{\mathcal{U}})$ and each complete secret key query of the form $(u_k, \{\mathbf{y}_k\}_{k \in \mathcal{U}_\varrho}, \mathcal{U}_\varrho)$.

$\overline{\mathrm{H}}^0_\eta$ for $(\eta \in [Q_w] \cup \{0\})$: Same as hybrid $\widetilde{\mathrm{H}}^0_{\varrho-1,\vartheta-1}$ except for all users $u_k \in \mathcal{U}_\varrho^{\mathcal{HS}}$, satisfying $I_{\mathbf{y}_{\ell,u_k}} \subseteq [m_{u_k}]$, the queried key and the LoR ciphertext components corresponding to the underlying Π_{esi} scheme are changed as follows:

$$\mathcal{O}_{\mathsf{KG}}(u_k, \{\mathbf{y}_k\}_{k \in \mathcal{U}_\varrho}, \mathcal{U}_\varrho): \quad \mathsf{eSK}_{\ell,u_k} = \begin{cases} \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ 0)]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k < w \\ \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \sum_{\iota \in [\eta]} \delta_{u_\iota, L_{\mathcal{U}_\varrho^\vartheta}})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k = w \end{cases},$$

$$\mathcal{O}_{\mathsf{LoR},\beta}(u_k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_\varrho, L^\vartheta_{\mathcal{U}}): \quad \mathsf{eCT}_k^{(0)} = \begin{cases} \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ [\![(\ \mathbf{x}_{u_k}^{(1)},\ \mathbf{s}_{u_k},\ 0)]\!]_1\ ) & \text{if } k \le \eta \\ \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{x}_{u_k}^{(0)},\ \mathbf{s}_{u_k},\ 0)]\!]_1\ ) & \text{if } \eta < k < w \\ \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{x}_{u_k}^{(0)},\ \mathbf{s}_{u_k},\ 1\ )]\!]_1\ ) & \text{if } k = w \end{cases}$$

where $\delta_{u_\iota, L_{\mathcal{U}_\varrho^\vartheta}} = \langle \mathbf{x}_{u_\iota}^{1,(1)}, \mathbf{y}_{\ell,u_\iota} \rangle - \langle \mathbf{x}_{u_\iota}^{1,(0)}, \mathbf{y}_{\ell,u_\iota} \rangle$ and the superscript 1 represent the first LoR queries are of the form $(u_\iota, \star, \star, \mathcal{U}_\varrho, L^\vartheta_{\mathcal{U}_\varrho})$. From the admissibility conditions of the Π_{ddi}, we have

- Let $Q_{c,u_\iota,\mathcal{U}_\varrho,L_{\mathcal{U}_\varrho^\vartheta}}$ be the number of the ciphertext queries of the form $(u_\iota, \star, \star, \mathcal{U}_\varrho, L^\vartheta_{\mathcal{U}_\varrho})$ and consider $\delta_{u_\iota, L_{\mathcal{U}_\varrho^\vartheta}} = \langle \mathbf{x}_{u_\iota}^{\tau,(1)}, \mathbf{y}_{\ell,u_\iota} \rangle - \langle \mathbf{x}_{u_\iota}^{\tau,(0)}, \mathbf{y}_{\ell,u_\iota} \rangle$ for all $\tau \in [Q_{c,u_\iota,\mathcal{U}_\varrho,L_{\mathcal{U}_\varrho^\vartheta}}]$.

– Also $\sum_{u_\iota \in \mathcal{U}_\varrho^{\mathcal{HS}}} \delta_{u_\iota, L_{\mathcal{U}_\varrho}^\vartheta} = 0$.

From the function-hiding security of Π_{esi}, we have $\overline{\mathrm{H}}_0^0 \approx_c \widetilde{\mathrm{H}}_{\varrho-1,\vartheta-1}^0$. In both the hybrids, the complete key queries for the tuple $(u_w, \{\mathbf{y}_{\ell,k}\}_{k\in\mathcal{U}_\varrho}, \mathcal{U}_\varrho)$ is of the form

$$\mathcal{O}_{\mathsf{KG}}(u_w, \{\mathbf{y}_{\ell,k}\}_{k\in\mathcal{U}_\varrho}, \mathcal{U}_\varrho): \quad \mathsf{eSK}_{\ell,u_k} = \quad \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ \ [\![(\mathbf{y}_{\ell,u_k},\ \ \boldsymbol{\alpha}_\ell,\ \ \ 0)]\!]_2,\ \ I_{\mathbf{y}_{\ell,u_k}}\ \ )$$

and the complete ciphertext queries $(u_w, \mathbf{x}_{u_w}^{(0)}, \mathbf{x}_{u_w}^{(1)}, \mathcal{U}_\varrho, L_{\mathcal{U}_\varrho}^\vartheta)$ is of the form

$$\mathcal{O}_{\mathsf{LoR},\beta}(u_w, \mathbf{x}_{u_w}^{(0)}, \mathbf{x}_{u_w}^{(1)}, \mathcal{U}_\varrho, L_{\mathcal{U}_\varrho}^\vartheta): \quad \mathsf{eCT}_k^{(0)} = \begin{cases} \mathsf{eEnc}(\ \mathsf{eMSK}_{u_w},\ \ [\![(\mathbf{x}_{u_w}^{(0)},\ \ \mathbf{s}_{u_w},\ \ 1)]\!]_1\ \ ) & \text{in } \overline{\mathrm{H}}_0^0 \\ \mathsf{eEnc}(\ \mathsf{eMSK}_{u_w},\ \ [\![(\mathbf{x}_{u_w}^{(0)},\ \ \mathbf{s}_{u_w},\ \ 0)]\!]_1\ \ ) & \text{in } \widetilde{\mathrm{H}}_{\varrho-1,\vartheta-1}^0. \end{cases}$$

Thus, the indistinguishability follows from function-hiding argument of the underlying Π_{esi} scheme.

By similar arguments, we can show that $\overline{\mathrm{H}}_{Q_w}^0 \approx_c \widetilde{\mathrm{H}}_{\varrho-1,\vartheta}^0$. To complete the cycle between all the subsequent hybrids, we now show $\overline{\mathrm{H}}_{\eta-1}^0 \approx_c \overline{\mathrm{H}}_\eta^0$ through the following claim.

Claim 11 For all $\eta \in [Q_w]$, the hybrids $\overline{\mathrm{H}}_{\eta-1}^0$ and $\overline{\mathrm{H}}_\eta^0$ are computationally indistinguishable.

Proof. To show the indistinguishability of the above hybrids for $\eta \in [Q_w]$, we define the following series of hybrids namely $\overline{\mathrm{H}}_{\eta-1,1}^0$ to $\overline{\mathrm{H}}_{\eta-1,7}^0$.

$\overline{\mathrm{H}}_{\eta-1,1}^0$**:** For every complete challenge query to the oracle $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$, uniformly choose $\mathsf{seed}_{u_w,u_\eta} \leftarrow \mathcal{K}_2$ satisfying $\mathsf{seed}_{u_w,u_\eta} = \mathsf{seed}_{u_\eta,u_w}$ instead of generating these using $\mathsf{nKeyShared}$ algorithm.

For $u_w, u_\eta \in \mathcal{HS}$, from the security of Π_{nike}, we have

$$\mathcal{A}\Big(\{\mathsf{nSK}_{\ell,k}\}_{k\in\mathcal{CS}}, \mathsf{seed}_{u_w,u_\eta} \leftarrow \mathsf{nKeyShared}\Big) \approx_c \mathcal{A}\Big(\{\mathsf{nSK}_{\ell,k}\}_{k\in\mathcal{CS}} : \mathsf{seed}_{u_w,u_\eta} \leftarrow \mathcal{K}_2\Big)$$

Thus, the indistinguishability of hybrids $\overline{\mathrm{H}}_{\eta-1}^0$ and $\overline{\mathrm{H}}_{\eta-1,1}^0$ follows the security of Π_{nike}.

$\overline{\mathrm{H}}_{\eta-1,2}^0$**:** The vectors $\mathbf{s}_{u_\eta}, \mathbf{s}_{u_w}$ are modified in this hybrid as follows:

$$\mathbf{s}_{u_\eta} = \sum_{i\in\mathcal{U}_\varrho, k\notin\{u_\eta,u_w\}} (-1)^{k<u_\eta} \mathsf{PRF}_1^{\mathsf{seed}_{u_\eta,k,1}}(\mathcal{U}_\varrho, L_{\mathcal{U}_\varrho}^\vartheta) \ + \mathbf{t}_{u_\eta,u_w} \ ,$$
$$\mathbf{s}_{u_w} = \sum_{k\in\mathcal{U}_\varrho, k\notin\{u_\eta,u_w\}} (-1)^{k<u_w} \mathsf{PRF}_1^{\mathsf{seed}_{u_w,k,1}}(\mathcal{U}_\varrho, L_{\mathcal{U}_\varrho}^\vartheta) \ - \mathbf{t}_{u_\eta,u_w} \ ,$$

where $\mathbf{t}_{u_\eta,u_w} \leftarrow \mathbb{Z}_p^m$. The indistinguishability follows from the security of PRF_1.

$\overline{\mathrm{H}}_{\eta-1,3}^0$**:** The secret key and the ciphertext components corresponding to the complete queries for the user $u_k \in \mathcal{HS}$ for $k \in \{\eta, w\}$ with $\mathcal{R}(\mathbf{x}_{u_k}, \mathbf{y}_{\ell,u_k}) = 1$ are modified as follows.

$$\mathcal{O}_{\mathsf{KG}}(u_k, \{\mathbf{y}_k\}_{k\in\mathcal{U}_\varrho}, \mathcal{U}_\varrho): \quad \mathsf{eSK}_{\ell,u_k} = \begin{cases} \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ \ [\![(\mathbf{y}_{\ell,u_k},\ \ \boldsymbol{\alpha}_\ell,\ \ -\langle\boldsymbol{\alpha}_\ell, \mathbf{t}_{u_\eta,u_w}\rangle)]\!]_2,\ \ I_{\mathbf{y}_{\ell,u_k}}\ \ ) & \text{if } k = \eta \\ \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ \ [\![(\mathbf{y}_{\ell,u_k},\ \ \boldsymbol{\alpha}_\ell,\ \ \sum_{\iota\in[\eta-1]} \delta_{u_\iota, L_{\mathcal{U}_\varrho^\vartheta}} + \langle\boldsymbol{\alpha}_\ell, \mathbf{t}_{u_\eta,u_w}\rangle)]\!]_2,\ \ I_{\mathbf{y}_{\ell,u_k}}\ \ ) & \text{if } k = w \end{cases},$$

$$\mathcal{O}_{\mathsf{LoR},\beta}(u_k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_\varrho, L_{\mathcal{U}}^\vartheta): \quad \mathsf{eCT}_k^{(0)} = \begin{cases} \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ \ [\![(\mathbf{x}_{u_k}^{(0)},\ \ \mathbf{s}_{u_k} + \mathbf{t}_{u_\eta,u_w},\ \ 1)]\!]_1\ \ ) & \text{if } k = \eta \\ \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ \ [\![(\mathbf{x}_{u_k}^{(0)},\ \ \mathbf{s}_{u_k} - \mathbf{t}_{u_\eta,u_w}\ \ 1)]\!]_1\ \ ) & \text{if } k = w. \end{cases}$$

The indistinguishability follows from the function-hiding security of Π_{esi} scheme. Since for $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_{\ell,k}) = 1, k = \eta$, we have

$$\begin{aligned}\mathsf{eDec}(\mathsf{eSK}_{\ell,k}^{\overline{\mathrm{H}}^0_{\eta-1,2}}, \mathsf{eCT}_k^{\overline{\mathrm{H}}^0_{\eta-1,2}}) &= [\![\langle \mathbf{x}_{u_k}^{(0)}, \mathbf{y}_{\ell,u_k}\rangle + \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_{u_k}\rangle]\!]_T \\ &= [\![\langle \mathbf{x}_{u_k}^{(0)}, \mathbf{y}_{\ell,u_k}\rangle + \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_{u_k} + \mathbf{t}_{u_\eta,u_w}\rangle - \langle \boldsymbol{\alpha}_\ell, \mathbf{t}_{u_\eta,u_w}\rangle]\!]_T \\ &= \mathsf{eDec}(\mathsf{eSK}_{\ell,k}^{\overline{\mathrm{H}}^0_{\eta-1,3}}, \mathsf{eCT}_k^{\overline{\mathrm{H}}^0_{\eta-1,3}}).\end{aligned}$$

For $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_{\ell,k}) = 1, k = w$, we have

$$\begin{aligned}\mathsf{eDec}(\mathsf{eSK}_{\ell,k}^{\overline{\mathrm{H}}^0_{\eta-1,2}}, \mathsf{eCT}_k^{\overline{\mathrm{H}}^0_{\eta-1,2}}) &= [\![\langle \mathbf{x}_{u_k}^{(0)}, \mathbf{y}_{\ell,u_k}\rangle + \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_{u_k}\rangle + \sum_{\iota\in[\eta-1]} \delta_{u_\iota, L_{\mathcal{U}_\varrho^\vartheta}}]\!]_T \\ &= [\![\langle \mathbf{x}_{u_k}^{(0)}, \mathbf{y}_{\ell,u_k}\rangle + \langle \boldsymbol{\alpha}_\ell, \mathbf{s}_{u_k} - \mathbf{t}_{u_\eta,u_w}\rangle + \sum_{\iota\in[\eta-1]} \delta_{u_\iota, L_{\mathcal{U}_\varrho^\vartheta}} + \langle \boldsymbol{\alpha}_\ell, \mathbf{t}_{u_\eta,u_w}\rangle]\!]_T \\ &= \mathsf{eDec}(\mathsf{eSK}_{\ell,k}^{\overline{\mathrm{H}}^0_{\eta-1,3}}, \mathsf{eCT}_k^{\overline{\mathrm{H}}^0_{\eta-1,3}}).\end{aligned}$$

$\overline{\mathrm{H}}^0_{\eta-1,4}$**:** Same as the previous hybrid except that the secret key components corresponding to the complete key queries for $k \in \{\eta, w\}$ with $\mathcal{R}(\mathbf{x}_{u_k}, \mathbf{y}_{\ell,u_k}) = 1$ is modified as follows.

$$\mathcal{O}_{\mathsf{KG}}(u_k, \{\mathbf{y}_k\}_{k\in\mathcal{U}_\varrho}, \mathcal{U}_\varrho): \ \mathsf{eSK}_{\ell,u_k} = \begin{cases} \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \ -t_{u_\eta,u_w})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k=\eta \\ \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \sum_{\iota\in[\eta-1]}\delta_{u_\iota,L_{\mathcal{U}_\varrho^\vartheta}} + t_{u_\eta,u_w})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k=w. \end{cases}$$

For the complete key queries of the form $(\star, \mathcal{U}_\varrho, L^\vartheta_{\mathcal{U}_\varrho})$ associated with the key vectors $\{\{\mathbf{y}_k^1\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \{\mathbf{y}_k^2\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \cdots, \{\mathbf{y}_k^{Q_{\mathsf{key}}}\}_{k\in\mathcal{U}_{\mathsf{Key}}}\}$, we replace value $\langle \boldsymbol{\alpha}_\ell, \mathbf{t}_{u_\eta,u_w}\rangle$ with a random $t_{u_\eta,u_w} \leftarrow \mathbb{Z}_p$ where $\{\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^{Q_{\mathsf{key}}}\}$ be the set of corresponding hash values generated by hash H over the key vectors $\{\{\mathbf{y}_k^1\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \{\mathbf{y}_k^2\}_{k\in\mathcal{U}_{\mathsf{Key}}}, \cdots, \{\mathbf{y}_k^{Q_{\mathsf{key}}}\}_{k\in\mathcal{U}_{\mathsf{Key}}}\}$. Here, we consider Q_{key} be the maximum number of key queries by the adversary $\mathcal{A}$. The indistinguishability follows from MDDH_m assumption over the bilinear group G.

$\overline{\mathrm{H}}^0_{\eta-1,5}$**:** Same as the previous hybrid except that the secret key components corresponding to the complete key queries for $k \in \{\eta, w\}$ with $\mathcal{R}(\mathbf{x}_{u_k}, \mathbf{y}_{\ell,u_k}) = 1$ is modified as follows.

$$\mathcal{O}_{\mathsf{KG}}(u_k, \{\mathbf{y}_k\}_{k\in\mathcal{U}_\varrho}, \mathcal{U}_\varrho): \ \mathsf{eSK}_{\ell,u_k} = \begin{cases} \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \ -t'_{u_\eta,u_w} - \delta_{u_\eta,L_{\mathcal{U}_\varrho^\vartheta}})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k=\eta \\ \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \sum_{\iota\in[\ \eta\ ]}\delta_{u_\iota,L_{\mathcal{U}_\varrho^\vartheta}} + t'_{u_\eta,u_w})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k=w \end{cases}$$

where we implicitly set $t'_{u_\eta,u_w} = t_{u_\eta,u_w} + \delta_{u_\eta,L_{\mathcal{U}_\varrho^\vartheta}}$. As t_{u_η,u_w} is distributed uniformly random over $\mathbb{Z}_p$, the hybrids are statistically indistinguishable.

$\overline{\mathrm{H}}^0_{\eta-1,6}$**:** The secret key and the ciphertext components corresponding to the complete queries for the user $u_k \in \mathcal{HS}$ for $k = \eta$ with $\mathcal{R}(\mathbf{x}_{u_k}, \mathbf{y}_{\ell,u_k}) = 1$ are modified as follows.

$$\mathcal{O}_{\mathsf{KG}}(u_k, \{\mathbf{y}_k\}_{k\in\mathcal{U}_\varrho}, \mathcal{U}_\varrho): \ \mathsf{eSK}_{\ell,u_k} = \begin{cases} \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \ -t'_{u_\eta,u_w})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k=\eta \\ \mathsf{eKeyGen}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{y}_{\ell,u_k},\ \boldsymbol{\alpha}_\ell,\ \sum_{\iota\in[\eta]}\delta_{u_\iota,L_{\mathcal{U}_\varrho^\vartheta}} + t'_{u_\eta,u_w})]\!]_2,\ I_{\mathbf{y}_{\ell,u_k}}\ ) & \text{if } k=w \end{cases},$$

$$\mathcal{O}_{\mathsf{LoR},\beta}(u_k, \mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathcal{U}_\varrho, L^\vartheta_{\mathcal{U}}): \ \mathsf{eCT}_k^{(1)} = \begin{cases} \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ [\![(\ \mathbf{x}_{u_k}^{(1)},\ \mathbf{s}_{u_k} + \mathbf{t}_{u_\eta,u_w},\ 1,)]\!]_1\ ) & \text{if } k=\eta \\ \mathsf{eEnc}(\ \mathsf{eMSK}_{u_k},\ [\![(\mathbf{x}_{u_k}^{(0)},\ \mathbf{s}_{u_k} - \mathbf{t}_{u_\eta,u_w},\ 1)]\!]_1\ ) & \text{if } k=w \end{cases}.$$

The indistinguishability follows from the function-hiding security of Π_{esi}.

Now, we undo the changes in the previous hybrids to get to hybrid $\overline{\mathrm{H}}_{\eta}^{0}$. Therefore, Claim 11 holds. □

This also concludes the proof of Claim 10 and Claim 9. □ □

We reach Hybrid 3 when we loop over $\eta \in [Q_w]$. Hence, Lemma 22 holds. □
This completes the proof of Theorem 9. □

Acknowledgements. The first author acknowledges partial support from the Swiss Government Excellence Scholarship (ESKAS) under Personal ESKAS-No: 2024.0100. We also extend our gratitude to the anonymous reviewers for their valuable comments and suggestions.

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. Kobe, Japan (Dec 8–12, 2019). doi: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. Beijing, China (Apr 14–17, 2019). doi:10.1007/978-3-030-17259-6_5

[3] Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 733–751. Gaithersburg, MD, USA (Mar 30 – Apr 1, 2015). doi:10.1007/978-3-662-46447-2_33

[4] 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. Santa Barbara, CA, USA (Aug 19–23, 2018). doi:10.1007/978-3-319-96884-1_20

[5] 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. Daejeon, South Korea (Dec 7–11, 2020). doi:10.1007/978-3-030-64840-4_16

[6] 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. Paris, France (Apr 30 – May 4, 2017). doi:10.1007/978-3-319-56620-7_21

[7] Abdalla, M., Gong, J., Wee, H.: Functional encryption for attribute-weighted sums from k-Lin. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 685–716. Santa Barbara, CA, USA (Aug 17–21, 2020). doi:10.1007/978-3-030-56784-2_23

[8] 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. Raleigh, NC, USA (Nov 8–11, 2021). doi:10.1007/978-3-030-90453-1_8

[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. Chicago, IL, USA (Nov 7–10, 2022). doi: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. Santa Barbara, CA, USA (Aug 14–18, 2016). doi:10.1007/978-3-662-53015-3_12

[11] Agrawal, S., Maitra, M., Vempati, N.S., Yamada, S.: Functional encryption for Turing machines with dynamic bounded collusion from LWE. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 239–269. Virtual Event (Aug 16–20, 2021). doi:10.1007/978-3-030-84259-8_9

[12] 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. Santa Barbara, CA, USA (Aug 20–24, 2023). doi:10.1007/978-3-031-38551-3_15

[13] Agrawal, S., Yadav, A., Yamada, S.: Multi-input attribute based encryption and predicate encryption. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part I. LNCS, vol. 13507, pp. 590–621. Santa Barbara, CA, USA (Aug 15–18, 2022). doi:10.1007/978-3-031-15802-5_21

[14] Ananth, P.V., Sahai, A.: Functional encryption for Turing machines. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part I. LNCS, vol. 9562, pp. 125–153. Tel Aviv, Israel (Jan 10–13, 2016). doi:10.1007/978-3-662-49096-9_6

[15] Beimel, A.: Secure schemes for secret sharing and key distribution. Ph.D. thesis, Technion - Israel Institute of Technology, Israel (1996), https://technion.primo.exlibrisgroup.com/permalink/972TEC_INST/q1jq5o/alma990021768270203971

[16] Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Providence, RI, USA (Mar 28–30, 2011). doi:10.1007/978-3-642-19571-6_16

[17] Cho, C., Döttling, N., Garg, S., Gupta, D., Miao, P., Polychroniadou, A.: Laconic oblivious transfer and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 33–65. Santa Barbara, CA, USA (Aug 20–24, 2017). doi:10.1007/978-3-319-63715-0_2

[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. Brisbane, Queensland, Australia (Dec 2–6, 2018). doi:10.1007/978-3-030-03329-3_24

[19] 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. Santa Barbara, CA, USA (Aug 17–21, 2020). doi:10.1007/978-3-030-56784-2_25

[20] Chotard, J., Dufour-Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with repetition for inner product (2018), https://eprint.iacr.org/2018/1021

[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. Rio de Janeiro, Brazil (Mar 25–29, 2018). doi:10.1007/978-3-319-76581-5_9

[22] Datta, P., Pal, T.: (Compact) adaptively secure FE for attribute-weighted sums from k-lin. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 434–467. Singapore (Dec 6–10, 2021). doi:10.1007/978-3-030-92068-5_15

[23] 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. Atlanta, GA, USA (May 7–10, 2023). doi:10.1007/978-3-031-31368-4_21

[24] Datta, P., Pal, T., Takashima, K.: Compact FE for unbounded attribute-weighted sums for logspace from SXDH. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part I. LNCS, vol. 13791, pp. 126–159. Taipei, Taiwan (Dec 5–9, 2022). doi:10.1007/978-3-031-22963-3_5

[25] Dowerah, U., Dutta, S., Mitrokotsa, A., Mukherjee, S., Pal, T.: Unbounded predicate inner product functional encryption from pairings. Journal of Cryptology **36**(3), 29 (Jul 2023). doi:10.1007/s00145-023-09458-2

[26] Dufour Sans, E., Pointcheval, D.: Unbounded inner-product functional encryption with succinct keys. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 19. LNCS, vol. 11464, pp. 426–441. Bogota, Colombia (Jun 5–7, 2019). doi:10.1007/978-3-030-21568-2_21

[27] 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. Santa Barbara, CA, USA (Aug 18–22, 2013). doi:10.1007/978-3-642-40084-1_8

[28] Francati, D., Friolo, D., Malavolta, G., Venturi, D.: Multi-key and multi-input predicate encryption from learning with errors. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part III. LNCS, vol. 14006, pp. 573–604. Lyon, France (Apr 23–27, 2023). doi:10.1007/978-3-031-30620-4_19

[29] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS. pp. 40–49. IEEE Computer Society Press, Berkeley, CA, USA (Oct 26–29, 2013). doi:10.1109/FOCS.2013.13

[30] Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Functional encryption without obfuscation. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 480–511. Tel Aviv, Israel (Jan 10–13, 2016). doi:10.1007/978-3-662-49099-0_18

[31] Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F., Sahai, A., Shi, E., Zhou, H.: Multi-input functional encryption. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 578–602, Copenhagen, Denmark, (May 11–15, 2014). doi:10.1007/978-3-642-55220-5_32

[32] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 89–98. ACM Press, Alexandria, Virginia, USA (Oct 30 – Nov 3, 2006). doi:10.1145/1180405.1180418, available as Cryptology ePrint Archive Report 2006/309

[33] Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. In: Khuller, S., Williams, V.V. (eds.) 53rd ACM STOC. pp. 60–73. ACM Press, Virtual Event, Italy (Jun 21–25, 2021). doi:10.1145/3406325.3451093

[34] Li, Y., Wei, J., Guo, F., Susilo, W., Chen, X.: Robust decentralized multi-client functional encryption: Motivation, definition, and inner-product constructions. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part V. LNCS, vol. 14442, pp. 134–165. Guangzhou, China (Dec 4–8, 2023). doi:10.1007/978-981-99-8733-7_5

[35] 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. Kobe, Japan (Dec 8–12, 2019). doi:10.1007/978-3-030-34618-8_18

[36] Lin, H., Luo, J.: Compact adaptively secure ABE from k-Lin: Beyond NC^1 and towards NL. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 247–277. Zagreb, Croatia (May 10–14, 2020). doi:10.1007/978-3-030-45727-3_9

[37] Lin, H., Luo, J.: Succinct and adaptively secure ABE for ABP from k-Lin. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part III. LNCS, vol. 12493, pp. 437–466. Daejeon, South Korea (Dec 7–11, 2020). doi:10.1007/978-3-030-64840-4_15

[38] 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. Taipei, Taiwan (Dec 5–9, 2022). doi:10.1007/978-3-031-22963-3_4

[39] Nguyen, K., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with public inputs and strong security. Cryptology ePrint Archive, Paper 2024/740 (2024), https://eprint.iacr.org/2024/740

[40] Nguyen, K., Pointcheval, D., Schädlich, R.: Decentralized multi-client functional encryption with strong security. Cryptology ePrint Archive (2024), https://eprint.iacr.org/2024/764

[41] 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. Santa Barbara, CA, USA (Aug 15–19, 2010). doi:10.1007/978-3-642-14623-7_11

[42] 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. Beijing, China (Dec 2–6, 2012). doi:10.1007/978-3-642-34961-4_22

[43] Okamoto, T., Takashima, K.: Dual pairing vector spaces and their applications. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **98-A**(1), 3–15 (2015). doi:10.1587/TRANSFUN.E98.A.3

[44] O'Neill, A.: Definitional issues in functional encryption. Cryptology ePrint Archive (2010), https://eprint.iacr.org/2010/556

[45] 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. Atlanta, GA, USA (May 7–10, 2023). doi:10.1007/978-3-031-31368-4_22

[46] Tomida, J.: Unbounded quadratic functional encryption and more from pairings. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part III. LNCS, vol. 14006, pp. 543–572. Lyon, France (Apr 23–27, 2023). doi:10.1007/978-3-031-30620-4_18

[47] Tomida, J., Takashima, K.: Unbounded inner product functional encryption from bilinear maps. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 609–639. Brisbane, Queensland, Australia (Dec 2–6, 2018). doi:10.1007/978-3-030-03329-3_21

[48] Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Taormina, Italy (Mar 6–9, 2011). doi:10.1007/978-3-642-19379-8_4

[49] Waters, B.: A punctured programming approach to adaptively secure functional encryption. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 678–697. Santa Barbara, CA, USA (Aug 16–20, 2015). doi:10.1007/978-3-662-48000-7_33

[50] Wee, H.: Functional encryption for quadratic functions from k-lin, revisited. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 210–228. Durham, NC, USA (Nov 16–19, 2020). doi:10.1007/978-3-030-64375-1_8

A Multi-Input Attribute-Based UIPFE

In the section, we mainly focus on the *multi-input unbounded FE* (MI-UFE) which is a particular form of MC-UFE assuming all the clients use the same label. Thus the syntax of MI-UFE follows from the Definition 15 by ignoring the encryption algorithm from MC-UFE. Subsequently, we discuss the standard security of MI-UFE in Definition 21 and later we instantiate MI-UFE over attribute-based IPFE with wildcard attributes in Definition 22.

Definition 21 (Security for MI-UFE) The MI-UFE scheme $\Pi_{\mathsf{mif}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be xx-indistinguishability (xx-IND) (xx $\in \{\mathsf{sel}, \mathsf{adp}\}$) secure if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that the following holds

$$\mathsf{Adv}^{\mathsf{mif}}_{\mathcal{A},\text{xx-IND}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{mif}}_{\mathcal{A},\text{xx-IND}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mif}}_{\mathcal{A},\text{xx-IND}}(\lambda, 1) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{mif}}_{\mathcal{A},\text{xx-IND}}(\lambda, \beta)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{mif}}_{\mathcal{A},\text{xx-IND}}(\lambda, \beta) :}$

1: $(\mathsf{PP}, \mathsf{EK}_k, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, n)$.
2: $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Corr}}(\cdot), \mathcal{O}_{\mathsf{KG}}(\cdot), \mathcal{O}_{\mathsf{E}}(\cdot), \mathcal{O}_{\mathsf{LoR},\beta}(\cdot)}(\mathsf{PP})$.
3: Output β' if condition $(*)$ is satisfied.

$\underline{\mathcal{O}_{\mathsf{Corr}}(k) :}$

output MSK_k.

$\underline{\mathcal{O}_{\mathsf{KG}}(\{(\mathsf{Key}_{k,j})_{j\in I_k}\}_k) :}$

output $\mathsf{KeyGen}(\mathsf{MSK}, (\mathsf{Key}_{k,j})_{j\in I_k})$.

$\underline{\mathcal{O}_{\mathsf{E}}(k, (\mathsf{Msg}_{k,j})_{j\in I'_k}) :}$

output $\mathsf{Enc}(\mathsf{EK}_k, (\mathsf{Msg}_{k,j})_{j\in I'_k})$.

$\underline{\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k}) :}$

output $\mathsf{Enc}(\mathsf{EK}_k, (\mathsf{Msg}^\beta_{k,j})_{j\in I'_k})$.

Let $\mathcal{CS}$ be the sets of all inputs $k \in \mathcal{ID}$ for which the adversary makes queries to the oracles $\mathcal{O}_{\mathsf{Corr}}(\cdot)$ and $\mathcal{HS} = [n] \setminus \mathcal{CS}$. The condition $(*)$ is that if there exist two messages satisfying

$$f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in[n]}, \{k, (\mathsf{Msg}^0_{k,j})_{j\in I'_k}\}_{k\in[n]}) = f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in[n]}, \{k, (\mathsf{Msg}^1_{k,j})_{j\in I'_k}\}_{k\in[n]}).$$

- for all $k \in [n]$, $[\mathcal{O}_{\mathsf{LoR},\beta}(k, (\mathsf{Msg}^0_{k,j}, \mathsf{Msg}^1_{k,j})_{j\in I'_k})$ or $\mathcal{O}_{\mathsf{E}}(k, (\mathsf{Msg}_{k,j})_{j\in I'_k})$ with $(\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j} = \mathsf{Msg}_{k,j})_{j\in I'_k}$ is queried] or $[(\mathsf{Msg}^0_{k,j} = \mathsf{Msg}^1_{k,j} = \mathsf{Msg}_{k,j})_{j\in I'_k}$ and $k \in \mathcal{CS}]$.
- for all $k \in [n]$, $[\mathcal{O}_{\mathsf{KG}}(k, (\mathsf{Key}_{k,j})_{j\in I_k})$ is queried] or $[k \in \mathcal{CS}]$.

– If xx = sel: Queries to $\mathcal{O}_{\mathsf{Corr}}(\cdot)$ and $\mathcal{O}_{\mathsf{E}}(\cdot)$ in one shot. That is, adversary submits $(\mathcal{CS}, k, \mathbf{x}^{(0)}_k, \mathbf{x}^{(1)}_k)$ and obtains the response $(\{\mathsf{EK}_k\}, \{\mathsf{Enc}(\mathsf{EK}_k, \mathbf{x}^{(\beta)})\})$. Only after the one-shot query, the adversary can query $\mathcal{O}_{\mathsf{KG}}(\cdot)$ oracle.

Definition 22 (MI-UFE for AB-IP) A multi-input attribute-based UIPFE (MI-AB-UIPFE) is a class of MI-UFE where $\mathcal{K}^* = \mathbb{Z}^*_p \times \mathcal{P}$, and $\mathcal{M}^* = \mathbb{Z}^*_p \times \mathcal{ATT} \cup \{\diamond\}$ such that $\mathcal{P}$ and $\mathcal{ATT}$ represent the access policy and attribute spaces, respectively. Here, $\{\diamond\}$ represents the wildcard attributes. The function f is defined as follows: for the message components $\mathsf{Msg}_k = (\mathbf{x}_k, \mathsf{S}_k) \in \mathcal{M}^*$, the key components $\mathsf{Key} = (\mathbf{y} = \{\mathbf{y}_k\}_k, \mathbb{A}) \in \mathcal{K}^*$ and $\mathbf{x}_k, \mathbf{y}_k$ are associated with the index sets I_k and I'_k,

$$f(\{k, (\mathsf{Key}_{k,j})_{j\in I_k}\}_{k\in[n]}, \{k, (\mathsf{Msg}_{k,j})_{j\in I'_k}\}_{k\in[n]}) = \begin{cases} \sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k \rangle_p & \text{if } (\star) \text{ holds} \\ \perp & \text{otherwise.} \end{cases}$$

The conditions in $(\star)$ is define as follows:

– $(\mathcal{R}(\mathbf{x}_k, \mathbf{y}_k) = 1 \wedge \mathbb{A}(\mathsf{S}_k) = 1$ for all $k \in [n]) \vee \mathbb{A}(\diamond) = 1$.

Definition 23 (Security of Weak MI-AB-UIPFE) We say that an MI-AB-UIPFE scheme is secure against *legitimate* keys if the scheme is secure against adversaries that satisfy the condition defined below in addition to the conditions defined in Definition 21. Let $(\mathcal{CS}, \{k, (\mathbf{x}^{(0)}_{\mu,k}, \mathbf{x}^{(1)}_{\mu,k}, \mathsf{S}_{\mu,k})\}_{\mu\in[Q_{c,k}],k\in[n]}, \{\mathbf{y}_\ell, \mathbb{A}_\ell\}_{\ell\in[Q_{\mathsf{key}}]})$ be the query of the adversary, where Q_{key} is the number of queries to $\mathcal{O}_{\mathsf{KG}}(\cdot)$ and $Q_{c,k}$ be the numbers of queries of the forms of $(k, *, *)$ to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle. For $\ell \in [Q_{\mathsf{key}}]$, we say that the key components $(\mathbf{y}_\ell, \mathbb{A}_\ell)$ is *legitimate* if for all $k \in \mathcal{HS}$, there must exists $\mu'_k \in [Q_{c,k}]$ such that $\mathbb{A}_\ell(\mathsf{S}_{\mu'_k}) = 1$. In security against legitimate keys, $(\mathbf{y}_\ell, \mathbb{A}_\ell)$ must be legitimate for all $\ell \in [Q_{\mathsf{key}}]$. In contrast, we say that an MI-AB-UIPFE satisfies security against any keys if the scheme is secure against adversaries that follows just the condition defined in Definition 21.

A.1 Security against Legitimate Keys

In this section, we analyze the MI-AB-UIPFE scheme in the context of legitimate keys. Due to the weak security model, we refer this scheme as *weak* MI-AB-UIPFE. This construction is inspired by the MC-AB-UIPFE scheme from the Construction 6.1, where we assume $\mathbf{s}_k = \mathbf{1} = (1, 1, \ldots, 1)$, generated from label L. The detailed construction is provided below.

A.1.1 Construction

Consider $\Pi_{\mathsf{asi}} = (\mathsf{aSetup}, \mathsf{aKeyGen}, \mathsf{aEnc}, \mathsf{aSlotEnc}, \mathsf{aDec})$ be an AB-sUIPFE scheme with $\mathcal{S}_{\mathsf{pub}} = \mathbb{Z}_p^*$, $\mathcal{S}_{\mathsf{priv}} = \mathbb{Z}_p^{\widetilde{m}+1}$. Note that, our proposed weak MI-AB-UIPFE scheme only involves the aEnc algorithm to encrypt the slot-specified message vector using a corresponding master secret key. In the following, we present our weak MI-AB-UIPFE scheme $\Pi_{\mathsf{wmiai}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ for LSSS access structure.

$\mathsf{Setup}(1^\lambda, n)$: The setup algorithm takes the security parameter λ with the total number of user n in the system as input and executes the following steps:

1. Generates $(\mathsf{aMPK}_k, \mathsf{aMSK}_k) \leftarrow \mathsf{aSetup}(1^\lambda, 1^{\widetilde{m}+1})$ for all $k \in [n]$.
2. Outputs k-th party's encryption key $\mathsf{EK}_k = \mathsf{aMSK}_k$ for $k \in [n]$ and the master secret key $\mathsf{MSK} = \{\mathsf{aMSK}_k\}_{k\in[n]}$.

$\mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y} = (\mathbf{y}_k)_{k\in[n]}, \{I_{\mathbf{y}_k}\}_k, \mathbb{A})$: The key generation algorithm takes as input MSK, the access structure $\mathbb{A}$ and a key vector $\mathbf{y} = (\mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \cdots \parallel \mathbf{y}_n)$ where each $\mathbf{y}_k$ is associated with the index set $I_{\mathbf{y}_k}$ for all $k \in [n]$. It does as follows:

1. Samples $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^{\widetilde{m}}$ such that $\sum_{k\in[n]} \mathbf{r}_k = \mathbf{0}$.
2. Generates $\mathsf{aSK}_k \leftarrow \mathsf{aKeyGen}(\mathsf{aMSK}_k, [\![(\mathbf{y}_k, \mathbf{r}_k, 0)]\!]_2, I_{\mathbf{y}_k}, \mathbb{A})$ for all $k \in [n]$.
3. Outputs the secret key $\mathsf{SK} = \{\mathsf{aSK}_k\}_{k\in[n]}$.

$\mathsf{Enc}(\mathsf{EK}_k, \mathbf{x}_k, \mathsf{S}_k)$: The encryption algorithm takes as input EK_k, a message vector $\mathbf{x}_k = (x_{k,i})_{i\in[m_k]}$ with an attribute set S_k and proceeds as following:

1. Generates $\mathsf{aCT}_k \leftarrow \mathsf{aEnc}(\mathsf{aMSK}_k, [\![(\mathbf{x}_k, \mathbf{1}, 0)]\!]_1, \mathsf{S}_k)$ where $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{Z}_p^{\widetilde{m}}$.
2. Outputs the ciphertext $\mathsf{CT}_k = \mathsf{aCT}_k$.

$\mathsf{Dec}(\mathsf{SK}, \{\mathsf{CT}_k\}_{k\in[n]})$: The decryption algorithm takes as input SK, $\{\mathsf{CT}_k\}_{k\in[n]}$ and performs the following steps:

1. Returns $[\![d]\!]_T \leftarrow \prod_{k\in[n]} \mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k)$ or $\perp$.

Correctness: If there exists any $k \in [n]$ such that $\mathcal{R}(\mathbf{x}_k, \mathbf{y}_k) = 0 \vee \mathbb{A}(\mathsf{S}_k) = 0$, output $\perp$. Otherwise, from the correctness of Π_{asi}, we have

$$\mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k) = [\![\langle \mathbf{x}_k, \mathbf{y}_k \rangle_p + \langle \mathbf{r}_k, \mathbf{1} \rangle]\!]_T \quad \text{for all } k \in [n]. \tag{7}$$

From Equation 7, we compute

$$[\![d]\!]_T = \prod_{k\in[n]} \mathsf{aDec}(\mathsf{aSK}_k, \mathsf{aCT}_k) = [\![\sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k \rangle_p + \langle \mathbf{r}_k, \mathbf{1} \rangle]\!]_T = [\![\sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k \rangle_p]\!]_T.$$

A.1.2 Security

Theorem 10 *Our Π_{wmiai} scheme achieves sel-FH-IND security against legitimate keys as per Definition 23 if the underlying Π_{asi} is sel-FH-IND secure as per Definition 14.*

(*Proof Sketch*). In the instantiation of the MC-UFE security from Definition 16, over AB-UIPFE with one-label security, it is observed that all $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle queries for each honest user $k \in \mathcal{HS}$ must occur at the same label, and no further $\mathcal{O}_{\mathsf{E}}(\cdot)$ oracle queries can be made at the same label L. Consequently, for each honest user k, the adversary $\mathcal{A}$ can submit multiple challenge ciphertext queries to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle. The security constraints ensure that the queried secret keys of the form $(\mathbf{y}_\ell, \mathbb{A}_\ell)$ must satisfy the admissibility condition whenever $\mathbb{A}_\ell(\mathsf{S}_{\mu,k}) = 1$ for all $k \in \mathcal{HS}$ where μ is the number of challenge ciphertext query to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ with the label L.

If we consider this security in the context of MI-AB-UIPFE, then we have to ignore the label, and the secret key query of the form $\{(\mathbf{y}_\ell, \mathbb{A}_\ell)\}_{\ell \in [Q_{\mathsf{key}}]}$ will satisfy

$$\sum_{k \in \mathcal{HS}} \langle \mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k} \rangle_p = \sum_{k \in \mathcal{HS}} \langle \mathbf{x}^{(1)}_{\mu,k}, \mathbf{y}_{\ell,k} \rangle_p \text{ whenever } \mathbb{A}_\ell(\mathsf{S}_{\mu,k}) = 1 \text{ for all } k \in \mathcal{HS}$$

with the constraints $(*)$ of Definition 16 for all $\mu \in [Q_{c,k}]$. Due to the admissible conditions and the multiple challenge ciphertext query to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle, we can compute

$$\begin{aligned}
\mu' &= \min \left\{ \mu \in [Q_{c,k}] : \left(\sum_{k \in \mathcal{HS}} \langle \mathbf{x}^{(0)}_{\mu,k}, \mathbf{y}_{\ell,k} \rangle_p = \sum_{k \in \mathcal{HS}} \langle \mathbf{x}^{(1)}_{\mu,k}, \mathbf{y}_{\ell,k} \rangle_p \right) \forall k \in \mathcal{HS} \right\} \\
&= \min \left\{ \mu \in [Q_{c,k}] : \mathbb{A}_\ell(\mathsf{S}_{\mu,k}) = 1 \ \forall k \in \mathcal{HS} \right\}
\end{aligned}$$

From this above inclusion, it can concluded that for all $k \in \mathcal{HS}$, and all the secret key query, there exist a μ'-th ciphertext query to the $\mathcal{O}_{\mathsf{LoR},\beta}(\cdot)$ oracle in of the form $(k, *, *)$ such that $\mathbb{A}_\ell(\mathsf{S}_{\mu',k}) = 1$ holds. Thus the above MI-AB-UIPFE scheme is secure against legitimate keys. □

A.2 Security against Any Keys

A.2.1 Construction

Let $\Pi_{\mathsf{wmiai}} = (\mathsf{wSetup}, \mathsf{wKeyGen}, \mathsf{wEnc}, \mathsf{wDec})$ be a weak MI-AB-UIPFE scheme with wildcard attribute and the security against legitimate keys as per Definition 23, $\Pi_{\mathsf{abe}} = (\mathsf{abSetup}, \mathsf{abKeyGen}, \mathsf{abEnc}, \mathsf{abDec})$ be a CP-ABE scheme of Lin and Luo [37] (as ABPs capture monotone span programs) for LSSS access structure and a secret sharing scheme $\Pi_{\mathsf{ss}} = (\mathsf{Share}, \mathsf{Rec})$. Then, the MI-AB-UIPFE scheme $\Pi_{\mathsf{miai}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ for LSSS access structure is constructed as follows achieves security against any keys.

$\mathsf{Setup}(1^\lambda, 1^n)$: The setup algorithm takes as input the security parameter λ, the number of user n and executes the following steps:

1. Generates $(\mathsf{wPP}, \{\mathsf{wEK}_k\}_{k \in [n]}, \mathsf{wMSK}) \leftarrow \mathsf{wSetup}(1^\lambda, 1^n)$.
2. For all $k \in [n]$, generates $(\mathsf{abMPK}_k, \mathsf{abMSK}_k) \leftarrow \mathsf{abSetup}(1^\lambda)$.
3. Outputs the public parameter $\mathsf{PP} = (\mathsf{wPP}, \{\mathsf{abMPK}_k\}_{k \in [n]})$, the encryption key $\mathsf{EK}_k = (\mathsf{wEK}_k, \mathsf{abMSK}_k)$ and the master secret key $\mathsf{MSK} = \mathsf{wMSK}$.

$\mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y} = (\mathbf{y}_k)_{k \in [n]}, \{I_{\mathbf{y}_k}\}_k, \mathbb{A})$: The key generation algorithm takes input as MSK, the access structure $\mathbb{A}$ and a key vector $\mathbf{y} = (\mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \cdots \parallel \mathbf{y}_n)$ where each $\mathbf{y}_k$ is associated with the index set $I_{\mathbf{y}_k}$ for all $k \in [n]$. It works as follows:

1. Generates $\mathsf{wSK} \leftarrow \mathsf{wKeyGen}(\mathsf{wMSK}, \mathbf{y}, \mathbb{A})$.
2. Generates $(\sigma_1, \ldots, \sigma_n) \leftarrow \mathsf{Share}(\mathsf{wSK}, n)$ where $\sigma_k = \{\sigma_{k,i}\}_{i \in I_{\mathbf{y}_k}}$.
3. Compute $\mathsf{abCT}_k \leftarrow \{\mathsf{abCT}_{k,i} = \mathsf{abEnc}(\mathsf{abMPK}_k, \sigma_{k,i}, \mathbb{A})\}_{i \in I_{\mathbf{y}_k}}$ for all $k \in [n]$.
4. Outputs the secret key $\mathsf{SK} = \{\mathsf{abCT}_k\}_{k \in [n]}$.

$\mathsf{Enc}(\mathsf{EK}_k, \mathbf{x}_k, \mathsf{S}_k)$: The encryption algorithm takes as input the k-th encryption key EK_k, a message vector $\mathbf{x}_k = (x_{k,i})_{i\in[m_k]}$ with an attribute set S_k and proceeds to do the following steps:

1. Generates $\mathsf{wCT}_k \leftarrow \mathsf{wEnc}(\mathsf{wEK}_k, \mathbf{x}_k, \mathsf{S}_k)$ and $\mathsf{abSK}_k \leftarrow \mathsf{abKeyGen}(\mathsf{abMSK}_k, \mathsf{S}_k)$.
2. Outputs the ciphertext $\mathsf{CT}_k = (\mathsf{wCT}_k, \mathsf{abSK}_k)$.

$\mathsf{Dec}(\mathsf{SK}, \{\mathsf{CT}_k\}_{k\in[n]})$: The decryption algorithm takes input the secret key SK, the ciphertext $\{\mathsf{CT}_k\}_k$ and proceeds as follows:

1. If there exists $k \in [n]$ such that $\mathbb{A}(\mathsf{S}_k) = 0$, then outputs $\perp$.
2. Otherwise, computes $\sigma'_k \leftarrow \{\sigma'_{k,i} = \mathsf{abDec}(\mathsf{abSK}_k, \mathsf{abCT}_{k,i})\}_{i\in I_{\mathbf{y}_k}}$ for $k \in [n]$.
3. Computes $\mathsf{wSK}' \leftarrow \mathsf{Rec}(\sigma'_1, \ldots, \sigma'_n)$.
4. Computes $[\![d]\!]_T \leftarrow \mathsf{wDec}(\mathsf{wSK}', \{\mathsf{wCT}_k\}_{k\in[n]})$.

Correctness: From the correctness of Π_{abe} for wildcard attributes, $\sigma'_1, \ldots, \sigma'_n$ are vaild shares of wmiSK, and from the correctness of the Π_{wmiai}, we get

$$[\![d]\!]_T = [\![\sum_{k\in[n]} \langle \mathbf{x}_k, \mathbf{y}_k\rangle_p]\!]_T \text{ whenever } \mathbb{A}(\mathsf{S}_k) = 1 \text{ for all } k \in [n].$$

A.2.2 Security Analysis

In Theorem 11, we present the security analysis of MI-AB-UIPFE against any keys, as described above.

Theorem 11 *Our Π_{miai} scheme achieves sel-IND security against any keys as per Definition 21 if Π_{wmiai} has security against legitimate keys, Π_{abe} is selectively secure, and Π_{ss} scheme is secure.*

Proof. To prove the security against any keys of the scheme provided in Section A.2 , we consider the following series of hybrids.

Hybrid 0. Same as real hybrid. We consider a secret key as an illegitimate key if all the combinations of the challenge ciphertexts decrypt to $\perp$. More explicitly, for each illegitimate secret key corresponding to $(\mathbf{y} = (\mathbf{y}_k)_{k\in[n]}, \mathbb{A})$, there exist $k' \in \mathcal{HS}$ such that $\mathbb{A}(\mathsf{S}_{k'}) = 0$ where $\mathsf{S}_{k'}$ is the associated attribute set of ciphertext query for the honest slot k'.

Hybrid 1. Same as Hybrid 0 except the responses of the illegitimate secret key queries. The $\mathsf{abCT}_{k'}$ in SK is now generated as

$$\mathsf{abCT}_{k',i} \leftarrow \mathsf{abEnc}(\mathsf{abMPK}_{k'}, 0^m, \mathbb{A})$$

where m is the bit-length of the each share $\sigma'_{k,i}s$. Due to the CP-ABE security of Lin and Luo [48] Π_{abe} scheme, hybrid 0 and hybrid 1 are computationally indistinguishable.

Hybrid 2. Same as Hybrid 1 except the responses of the illegitimate secret key queries. The secret shares σ_k's are now generated as

$$\sigma_k = \{\sigma_{k,i} \leftarrow \{0,1\}^m\}_{i\in I_{\mathbf{y}_k}}.$$

Due to the security of Π_{ss} scheme, the hybrid 1 and hybrid 2 are identically distributed.

Hybrid 3. Same as Hybrid 1 except the responses of the challenge ciphertexts. For all the ciphertext queries, the challenger replies

$$\mathsf{Enc}(\mathsf{EK}_k, \mathbf{x}_k^{(1)}, \mathsf{S}_k).$$

The indistinguishability follows directly from the security of Π_{wmiai} scheme.

Thus, by the above hybrids, we see that the adversary has no information about the challenge bit β. This completes the proof of Theorem 11. □