Multi-Client Attribute-Based and Predicate Encryption, Revisited

Robert Schädlich

DIENS, École normale supérieure, PSL University, CNRS, Inria
robert.schaedlich@ens.fr

Abstract

Multi-client Attribute-Based Encryption (ABE) is a generalization of key-policy ABE where attributes can be independently encrypted across several ciphertexts w.r.t. labels, and a joint decryption of these ciphertexts is possible if and only if (1) all ciphertexts share the same label, and (2) the combination of attributes satisfies the policy of the decryption key. All encryptors have their own secret key and security is preserved even if some of them are known to the adversary.

Very recently, Pointcheval et al. (TCC 2024) presented a semi-generic construction of MC-ABE for restricted function classes, e.g., NC^0 and constant-threshold policies. We identify an abstract criterion common to all their policy classes which suffices to present the construction in a fully black-box way and allows for a slight strengthening of the supported policy classes. The construction of Pointcheval et al. is based on pairings. We additionally provide a new lattice-based instantiation from (public-coin) evasive LWE.

Furthermore, we revisit existing constructions for policies that can be viewed as a conjunction of local policies (one per encryptor). Existing constructions from MDDH (Agrawal et al., CRYPTO 2023) and LWE (Francati et al., EUROCRYPT 2023) do not support encryption w.r.t. different labels. We show how this feature can be included at the cost of additionally relying on random oracles. Notably, the security model of Francati et al. additionally guarantees attribute-hiding but does not capture collusions. Our new construction is also attribute-hiding and provides resilience against any polynomially bounded number of collusions which must be fixed at the time of setup.

This is the full version of [Sch25], DOI 10.1007/978-3-031-91832-2_2, published by Springer in the proceedings of PKC 2025.

Contents

1 Introduction **3**
- 1.1 Related Work . . . 3
- 1.2 Our Results . . . 4
- 1.3 Relation to Witness Encryption . . . 5

2 Technical Overview **5**
- 2.1 MC-ABE for Succinctly Enumerable Function Classes . . . 7
- 2.2 MC-ABE for Conjunction Policies . . . 10

3 Preliminaries **14**
- 3.1 Notational Conventions . . . 14
- 3.2 Computational Models . . . 15
- 3.3 Pairing Groups . . . 15
- 3.4 Lattices . . . 16
- 3.5 Exact Linear Garbling . . . 16
- 3.6 Noisy Linear Garbling . . . 17
- 3.7 Multi-Client Attribute-Based Encryption . . . 19
- 3.8 Attribute-Based Inner-Product Functional Encryption . . . 20
- 3.9 Slotted Identity-Based Inner-Product Functional Encryption . . . 23
- 3.10 Attribute-Based Encryption . . . 25
- 3.11 Other Preliminaries . . . 26

4 MC-ABE for Succinctly Enumerable Function Classes **26**
- 4.1 Construction from Slotted IPFE . . . 28
- 4.2 Proof of Security . . . 30
- 4.3 Construction from Evasive IPFE . . . 32
- 4.4 Proof of Security . . . 34

5 MC-ABE for Conjunction Predicates **37**
- 5.1 Construction . . . 38
- 5.2 Proof of Security – Function-Hiding Case . . . 39
- 5.3 Proof of Security – Non-Function-Hiding Case . . . 41

A AB-MIFE for Inner Products Against Bounded Collusions **49**
- A.1 Definitions for MIFE with Corruptions . . . 49
- A.2 Construction of AB-MIFE Against Legitimate Keys . . . 50
- A.3 Upgrade to Any Keys . . . 52

1 Introduction

Attribute-Based Encryption. Attribute-based encryption (ABE) [SW05, GPSW06] is an elevated version of traditional public-key encryption that enables fine-grained access control on encrypted data. A ciphertext CT_x encrypting a message μ is generated with respect to a public attribute x, while a secret decryption key DK_f is generated with respect to a policy f. The decryption key DK_f is authorized to decrypt the ciphertext CT_x if and only if the attribute x satisfies the policy f, i.e., $f(x) = 0$. Security requires resilience to collusion attacks. That is, a set of decryption keys $\{\mathsf{DK}_{f_i}\}_i$ does not disclose any information about μ unless one of the keys DK_{f_i} is individually authorized to decrypt. A strengthening of this notion, traditionally referred to as Predicate Encryption (PE) [BW07, SBC+07], requires ciphertexts to not only hide messages but also their associated attributes.

Decentralized Encryption. Until recently, ABE and PE were solely studied in the centralized setting, i.e., the considered policies f have arity one. The notions of Multi-Input ABE (MI-ABE) [BJK+18, AYY22] and Multi-Input PE (MI-PE) [FFMV23] overcome this limitation by considering n encryptors who each encrypt their inputs $(x_0, \mu_0), \ldots, (x_{n-1}, \mu_{n-1})$ independently using uncorrelated random coins. The key generator provides decryption keys for arity n functions f, and a joint decryption recovers $(\mu_0, \ldots, \mu_{n-1})$ if and only if $f(x_0, \ldots, x_{n-1}) = 0$. As in the single-input case, the security model of MI-ABE only guarantees to hide the message encoded in a ciphertext while MI-PE additionally hides the associated attribute.

A very recent work [PS24] initiates the formal study of Multi-Client ABE (MC-ABE) and Multi-Client PE (MC-PE) by adding the usual features of Multi-Client Functional Encryption (MCFE) compared to Multi-Input Functional Encryption (MIFE) [GGG+14]. First, MC-ABE separates the secret keys of the slots and guarantees security even if some of them are known to the adversary. Second, encryption proceeds with respect to a label and a joint decryption of ciphertexts is possible only if all of them share the same label.[2]

1.1 Related Work

The notion of MI-ABE had been studied first by Brakerski et al. [BJK+18] as a new pathway for achieving witness encryption. However, they did not consider strong security notions nor did they provide any constructions. In [AYY22], Agrawal et al. provided the first constructions for 2-input ABE for NC^1 from LWE and a nonstandard assumption on pairings. They also gave heuristic constructions for 2-input ABE for P and 3-input ABE for NC^1. Additionally, they gave a compiler that lifts a constant-input ABE scheme to a PE scheme for the same arity and policy class, using a sophisticated nesting technique of lockable obfuscation which can be based on LWE. In an independent work, Francati et al. [FFMV23] built MI-PE for *conjunctions* of (bounded) polynomial-depth circuits from LWE.[3] Notably, they can support a polynomial number of inputs. Furthermore, when restricting to constant arity, they provide a construction that remains secure under user corruptions. On the negative side, neither of their MI-PE constructions can be proven secure under collusions. Agrawal et al. [ARYY23] presented a constant-arity MI-ABE for NC^1 whose security is based on (private-coin) evasive LWE. When additionally assuming tensor LWE, they can upgrade their scheme to support arbitrary policies in P. A follow-up work [AKY24] removes the dependency on tensor LWE and enables any polynomial arity. Very recently, Pointcheval and Schädlich [PS24] provided pairing-based constructions for several restricted function classes including NC^0 and constant-threshold policies. They can support polynomial arity and prove security in the stronger MC-ABE model. As a side result, they also show how to capture all of NC^1 when restricting to one-time labels. Furthermore, they generalize the PE compiler of [AYY22] to the multi-client setting. In a concurrent work [CLW25], Cini et al. present a MC-ABE construction for P from (public-coin) evasive LWE and prove security in the one-time label setting against a restricted form of corruptions.

[2] We use the term "multi-client" as a generalization of "multi-input" by allowing multiple uses of labels. In contrast, a weaker notion of multi-client has been considered in the literature [CDG+18] where each label can be used only once, thus it does not imply the corresponding multi-input notion. We call this weaker form *one-time* labels.

[3] A policy f is said to be a *conjunction* of a policy class $\mathcal{F}$ if there exist policies $f_0, \ldots, f_{n-1} \in \mathcal{F}$ such that $f(x_0, \ldots, x_{n-1}) = f_0(x_0) \wedge \cdots \wedge f_{n-1}(x_{n-1})$.

MI-ABE and MI-PE can also be viewed as a special case of the more general MIFE primitive. MIFE has been the subject of extensive studies resulting in a large body of works with various trade-offs between expressiveness, security, underlying assumptions and efficiency, e.g., [GGG+14, AJ15, AGRW17, DOT18, ACF+18, CDG+18, Tom19, ABKW19, ABG19, LT19, AGT21a, AGT22]. As MIFE for NC^1 is known to imply indistinguishability Obfuscation (iO) [BGI+01, GGH+13], it remains an important area of research to build MIFE schemes for simpler function classes from assumptions not known to imply iO. Some of these function classes still encompass MI-ABE. Specifically, Nguyen et al. [NPP22] build the first attribute-based MCFE with one-time labels for inner products, where policies are conjunctions of Linear Secret Sharings (LSS), implying MC-ABE for the same function class. Their construction uses pairings and security is proven under the SXDH assumption. In a very recent follow-up [NPP24], the authors allow reusing the same label for several private inputs (i.e. the inner-product vectors) and prove security in a strong model: adversaries can submit adaptive oracle queries and the admissibility condition excludes fewer attacks than previous works. In [ATY23], Agrawal et al. presented the first attribute-based MIFE for Attribute-Weighted Sums (AWS). The supported policies are conjunctions of Arithmetic Branching Programs (ABPs) for a polynomial number of slots. Their construction uses pairings and is proven secure under the matrix DDH assumption (MDDH).

1.2 Our Results

We provide new MC-ABE constructions for previously unknown combinations of policy classes and computational assumptions. Please see Table 1 for a comparison with known results. For succinctness, the table does not include MC-ABE constructions that support only one-time labels.

MC-ABE Beyond Conjunctions. Very recently, [PS24] presented a semi-generic MC-ABE scheme for several restricted function classes that, nevertheless, cannot be viewed as conjunctions. We revisit their construction and enhance it in the following directions.

- **Abstract Function Class.** We first identify an abstract criterion that we dub *succinct enumerability*. It is common to all the function classes considered in [PS24] and implicitly required by their construction to work properly. While the formal definition of succinct enumerability seems quite artificial, it enables a fully black-box presentation of the construction. It also helps us to identify slightly stronger function classes that were not considered in [PS24]. Notably, the following classes are succinctly enumerable:
  - **Constant Bit Dependency.** Succinct enumerability captures all functions with constant bit dependency on slots $1,\ldots,n-1$, but arbitrary dependency on the 0-th slot. This includes NC^0 (considered in [PS24]) as a special case since NC^0 policies have a constant bit dependency on *all* slots. In addition, succinct enumerability captures some special NC^1 predicates where most of the relevant bits (in particular, more than constant) are part of the 0-th input and the bit dependency on the other $n-1$ inputs is still constant.
  - **Polynomial-Size Accepting Sets.** Succinct enumerability also captures policies with a polynomial-size accepting set on slots $1,\ldots,n-1$ for each slot-0 input. As a special case, this includes constant-threshold policies (considered in [PS24]) as these policies have *globally* poly-size accepting sets.
- **Lattice-Based Construction.** The construction of [PS24] uses slotted Inner-Product Functional Encryption (IPFE) as a building block which is only known to exist assuming MDDH on pairings [LL20]. To obtain a lattice-based scheme, we instantiate the construction from another IPFE primitive called evasive IPFE [HLL24]. In this way, we obtain MC-ABE for succinctly enumerable function classes from (public-coin) evasive LWE. We note that the previous work [ARYY23] presented MI-ABE from (private-coin) evasive LWE for bounded-depth circuits. While their function class is certainly much stronger than ours, their construction considers the weaker MI-ABE model and can only deal with constant arity.

MC-ABE for Conjunctions. We present the first constructions for conjunction policies that provide all features of MC-ABE.

- **MC-ABE for Conjunctions of ABPs from MDDH.** We first revisit the AB-MIFE scheme for AWS by [ATY23]. When ignoring the AWS functionality, this gives an MI-ABE scheme supporting conjunctions of ABPs. Notably, the construction is already secure against corruptions. We modify the scheme to additionally enable encryption with respect to labels, thus obtaining a full-fledged MC-ABE scheme. Security is proven under the MDDH assumption on pairings (as in [ATY23]) and additionally relies on the ROM. On a technical level, our construction is a compiler that turns any (single-input) Attribute-Based IPFE (AB-IPFE) scheme for a policy class $\mathcal{F}$ with function-hiding security into an MC-ABE for conjunctions of $\mathcal{F}$.

- **Bounded-Collusion Secure MC-ABE for Conjunctions of P from LWE.** In the second step, we attempt to provide a construction that does not use pairings and is secure under LWE. Since lattice-based IPFE schemes with function-hiding security are unlikely[4] to exist, an MC-ABE scheme from LWE can only rely on an AB-IPFE with non-function-hiding security. In fact, the only existing AB-IPFE scheme with security from LWE [LLW21] is only secure against bounded collusions. We therefore modify the previous compiler to turn non-function-hiding, bounded-collusion secure AB-IPFE into bounded-collusion secure MC-ABE. When instantiated with the AB-IPFE of [LLW21], this yields a bounded-collusion secure MC-ABE for conjunctions of P. As a side result, we also obtain the first bounded-collusion secure AB-MIFE for inner products assuming the hardness of LWE.

All our MC-ABE constructions can be plugged into the compiler of [PS24] which converts constant-input MC-ABE into constant-input MC-PE for the same policy class. In particular, this yields the first constant-input, bounded-collusion secure MC-PE for conjunctions of P and improves on the MC-PE of [FFMV23] (no collusions, no labels) at the cost of relying on random oracles.

1.3 Relation to Witness Encryption

A *witness encryption* (WE) scheme for an NP relation $\mathcal{R}$ defined over a language $\mathcal{L}$ allows a sender to efficiently encrypt a message μ with respect to a problem instance x. A receiver holding a witness w can recover the message μ if $(x,w) \in \mathcal{R}$. Security requires that ciphertexts for messages μ^0 and μ^1 are computationally indistinguishable if $x \notin \mathcal{L}$.

In [BJK+18], the authors show that n-input ABE for a policy class $\mathcal{F}$ implies WE for relations with length n witnesses whose verification algorithm is in $\mathcal{F}$. If $n = \mathrm{poly}(\lambda)$ and $\mathcal{F} = \mathsf{P}$, we obtain WE for all NP relations. But since there are NP relations that can be verified in NC^1 (e.g., 3-SAT), even MI-ABE for NC^1 policies would imply WE for certain NP relations. Unfortunately, our constructions fail to imply WE because conjunction policies and classes of succinctly enumerable functions are not powerful enough to verify an NP language. The work [FFMV23] mentions an interesting alternative pathway towards WE. If the MI-ABE is secure under corruptions, then a two-input scheme for conjunctions of some policy class $\mathcal{F}$ implies WE for any relation whose verification algorithm lies in $\mathcal{F}$. Importantly, the conversion of [FFMV23] works only if the first slot has a wildcard while the second slot does not. However, as previous works on conjunction policies [FFMV23, ATY23], our construction runs into the problem that all slots have wildcards.

2 Technical Overview

We start by recalling the definition of MC-ABE. The 0-th client (a.k.a. the "encryptor") runs an algorithm Enc which takes as input a label L_0, an attribute x_0 and a message μ to create a ciphertext CT_{L_0,x_0}. The other

[4] Strictly speaking, the existence of lattice-based IPFE schemes with function-hiding security is not proven to be impossible, but there are strong structural limitations ruling out most natural constructions [Ünа20, TÜ24].

Work	Arity	Attribute[1]	Collusion	Corruption	Labels	Policy Class	Assumptions, (Model)
[AYY22]	2	private	✓	✗	✗	NC^1	KOALA[2], LWE
[ARYY23]	const	private	✓	✗	✗	NC^1	private-coin evasive LWE, LWE
						P	private-coin evasive LWE, tensor LWE, LWE
[AKY24]	const	private	✓	✗	✗	P	private-coin evasive LWE
	poly						non-uniform evasive LWE[3], LWE
[PS24]	poly	public	✓	✓	✓	NC^0 or const threshold	SXDH
	const	private					SXDH, LWE
Sec. 4	poly	public	✓	✓	✓	succinctly enumerable	MDDH or [LWE, public-coin evasive LWE]
Sec. 4 + [PS24]	const	private					LWE, [MDDH or public-coin evasive LWE]
[FFMV23]	poly	private	✗	✗	✗	conjunctions of P	LWE
	const			✓			
[ATY23]	poly	public	✓	✓	✗	conjunctions of NC^1	MDDH
[ATY23] + [PS24]	const	private					MDDH, LWE
Sec. 5	poly	public	✓	✓	✓	conjunctions of NC^1	MDDH, LWE, (ROM)
Sec. 5 + [PS24]	const	private					
Sec. 5	poly	public	poly[4]	✓	✓	conjunctions of P	LWE, (ROM)
Sec. 5 + [PS24]	const	private					

[1] Public attributes correspond to ABE and private attributes to PE.

[2] KOALA is a nonstandard knowledge type assumption on pairings.

[3] Non-uniform κ-evasive LWE is a strengthening of (private-coin) evasive LWE.

[4] The scheme supports any polynomial number of collusions, but the number must be fixed at the time of setup.

Table 1: Comparison with existing works in MI-ABE and MI-PE

clients $1,\dots,n-1$ (a.k.a. the "attribute key generators") run an algorithm $\mathsf{AKeyGen}$ which takes only a label L_i and an attribute x_i to generate a decryption key DK_{L_i,x_i}. Policy decryption keys DK_f for a policy f are generated by a central authority which runs an algorithm $\mathsf{PKeyGen}$. CT_{L_0,x_0} can be decrypted using $\{\mathsf{DK}_{L_i,x_i}\}_i$ and DK_f if $L_0 = \cdots = L_{n-1}$ and $f(x_0,\dots,x_{n-1}) = 0$. For security, we require the usual ciphertext indistinguishability against collusion attacks under corruptions. We summarize the syntax as follows:

$$\begin{aligned} \text{Client } 0: &\quad \mathsf{Enc}(\mathsf{MPK}, L, x_0, \mu) \to \mathsf{CT}_{x_0} \\ \text{Client } i \in [n-1]: &\quad \mathsf{AKeyGen}(\mathsf{SK}_i, L, x_i) \to \mathsf{DK}_{x_i} \\ \text{Authority}: &\quad \mathsf{PKeyGen}(\mathsf{MSK}, f) \to \mathsf{DK}_f \end{aligned}$$

For completeness, we mention that MC-ABE can also be considered in an n-message setting where not only the 0-th but all slots encrypt a message. In [AYY22], it was shown that a single-message MI-ABE scheme can be generically lifted to an n-message scheme by running n single-message schemes in parallel with rotated slots. The technique generalizes to MC-ABE.

2.1 MC-ABE for Succinctly Enumerable Function Classes

Ingredients to our Constructions. Let $\mathbb{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, p)$ be a pairing group. For $i \in \{1,2,t\}$ and $a \in \mathbb{Z}_p$, we write $[\![a]\!]_i = g_i^a$ and use additive notations for the group operations.

- An *identity-based IPFE* (IB-IPFE) scheme based on $\mathbb{G}$ enables the generation of ciphertexts $\mathsf{iCT}(i, [\![\mathbf{x}]\!]_1)$ associated with identities i and vectors $\mathbf{x} \in \mathbb{Z}_p^N$ encoded in $\mathbb{G}_1$, and the generation of (secret) decryption keys $\mathsf{iDK}(i', [\![\mathbf{y}]\!]_2)$ for identities i' and vectors $\mathbf{y} \in \mathbb{Z}_p^N$ encoded in $\mathbb{G}_2$. Decryption of $\mathsf{iCT}(i, [\![\mathbf{x}]\!]_1)$ with $\mathsf{iDK}(i', [\![\mathbf{y}]\!]_2)$ reveals the inner product $[\![\mathbf{x}\mathbf{y}^\top]\!]_t$ of $\mathbf{x}$ and $\mathbf{y}$ encoded in $\mathbb{G}_t$ if $i = i'$ and hides all other information about $\mathbf{x}$ whereas $[\![\mathbf{y}]\!]_2$ is usually public. If $i \neq i'$, then no information about $\mathbf{x}$ is revealed.

- In the same vein, *identity-based encryption* (IBE) allows creating ciphertexts $\mathsf{idCT}(i, \mu)$ associated with an identity i for a message μ, and decryption keys $\mathsf{idDK}(i')$ associated with an identity i'. Decryption is possible if and only if $i = i'$.

- A *linear secret sharing* (LSS) scheme allows to decompose a secret $s \in \mathbb{Z}_p$ into shares

$$\{s_{j,b}\}_{j\in[k], b\in\{0,1\}} \leftarrow \mathsf{Share}(s, f)$$

  with respect to a policy $f\colon \{0,1\}^k \to \{0,1\}$ such that s can be reconstructed from a subset of shares $\{s_{j,\mathbf{x}[j]}\}_{j\in[k]}$ for $\mathbf{x} \in \{0,1\}^k$ if and only if $f(\mathbf{x}) = 0$. For this reconstruction, there exists an efficient algorithm $\mathsf{FindCoef}(f, \mathbf{x})$ that outputs coefficients $\omega_1, \ldots, \omega_k$ such that $\sum_{j\in[k]} \omega_j \mathbf{s}_{j,\mathbf{x}[j]} = s$.

Recap: KP-ABE from LSS Schemes and Pairing-Based IPFE. Our starting point is a well-known technique of constructing ABE from LSS schemes and pairing-based IPFE. To encrypt a message μ under an attribute vector $\mathbf{x} \in \{0,1\}^k$, one picks a random scalar $r \xleftarrow{\$} \mathbb{Z}_p$ and encodes it in IPFE ciphertexts in a way that reflects the choice of $\mathbf{x}$: for each $j \in [k]$, encrypt $(r, 0)$ under identity j if $\mathbf{x}[j] = 0$ and $(0, r)$ otherwise. To generate a key for a policy f, one generates secret shares $\{s_{j,b}\}_{j\in[k], b\in\{0,1\}} \leftarrow \mathsf{Share}(s, f)$ for $s \xleftarrow{\$} \mathbb{Z}_p$ and encodes them in the IPFE secret keys such that IPFE decryption recovers the shares $\{r \cdot s_{j,\mathbf{x}[j]}\}_{j\in[k]}$ corresponding to $\mathbf{x}$ randomized by r.

$$\left.\begin{array}{ll} \mathsf{ABE.CT}_{\mathbf{x}}\colon & \mathsf{iCT}(0, [\![r, \mu]\!]_1), \left\{\mathsf{iCT}(j, [\![0^{\mathbf{x}[j]}, r, 0^{1-\mathbf{x}[j]}]\!]_1)\right\}_{j\in[k]} \\ \mathsf{ABE.DK}_f\colon & \mathsf{iDK}(0, [\![s, 1]\!]_2), \left\{\mathsf{iDK}(j, [\![s_{j,0}, s_{j,1}]\!]_2)\right\}_{j\in[k]} \end{array}\right\} \implies \begin{array}{l} [\![r \cdot s + \mu]\!]_t, \\ \left\{[\![r \cdot s_{j,\mathbf{x}[j]}]\!]_t\right\}_{j\in[k]} \end{array}$$

IPFE decryption yields target group encodings of $r \cdot s + \mu$ and $r \cdot \mathbf{s}_{j,\mathbf{x}[j]}$ for all $j \in [k]$. If $f(\mathbf{x}) = 0$, one can run $\mathsf{FindCoef}(f, \mathbf{x}) \to \{\omega_j\}_j$ and recover the message μ encoded in $\mathbb{G}_t$:

$$[\![r \cdot s + \mu]\!]_t - \sum_{j\in[k]} \omega_j [\![r \cdot s_{j,\mathbf{x}[j]}]\!]_t = [\![r \cdot s + \mu]\!]_t - \left[\!\!\left[r \cdot \sum_{j\in[k]} \omega_j s_{j,\mathbf{x}[j]} \right]\!\!\right]_t = [\![\mu]\!]_t \,.$$

MC-ABE for Short Inputs. To turn the above KP-ABE into a MC-ABE scheme, we have to distribute the generation of the ciphertext $\mathsf{CT}_{\mathbf{x}}$. We decompose $\mathbf{x} = (\mathbf{x}_0, \ldots, \mathbf{x}_{n-1})$ where $\mathbf{x}_0, \ldots, \mathbf{x}_{n-1} \in \mathbb{Z}_p^k$. To begin with, let us assume that the total length of the input is short, i.e., $kn = O(\log \lambda)$. In this case, the encryptor, who is given $\mathbf{x}_0$ as input, can generate an independent KP-ABE ciphertext for each possible choice of $\mathbf{x}_{\geq 1} = (\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in \{0,1\}^{(n-1)k}$. Since $2^{kn} = \mathrm{poly}(\lambda)$, the total number of ciphertexts to generate is still polynomial. If the encryptor provided all these ciphertexts without further protection, there would be no need for attribute decryption keys from the other slots since decryption is already possible. Therefore, instead of providing these ciphertexts for the other slots "in the clear", the encryptor hides them with an additional layer of identity-based encryption

on top. Specifically, each IPFE ciphertext $\mathsf{iCT}((i,j), [\![0^{\mathbf{x}_i[j]}, r, 0^{1-\mathbf{x}_i[j]}]\!]_1)$ is encrypted with respect to the identity $(L, \mathbf{x}_i)$ using the public key of client i, where L denotes the label with respect to which we want to encrypt. Correspondingly, client $i \in [n-1]$ can issue identity-based decryption keys $\mathsf{idDK}_i(L, \mathbf{x}_i)$ to remove the IBE layer added by client 0, thereby controlling for which attributes $\mathbf{x}_i$ the IPFE ciphertexts are disclosed. This results in the following construction:

$$\mathsf{CT}_{\mathbf{x}_0}: \left\{ \begin{array}{l} \mathsf{iCT}\big([\![0, r_{\mathbf{x}_{\geq 1}}, \mu]\!]_1\big), \\ \big\{\mathsf{iCT}\big((0,j), [\![0^{\mathbf{x}_0[j]}, r_{\mathbf{x}_{\geq 1}}, 0^{1-\mathbf{x}_0[j]}]\!]_1\big)\big\}^{j\in[k]}, \\ \big\{\mathsf{idCT}_i\big((L,\mathbf{x}_i), \mathsf{iCT}\big((i,j), [\![0^{\mathbf{x}_i[j]}, r_{\mathbf{x}_{\geq 1}}, 0^{1-\mathbf{x}_i[j]}]\!]_1\big)\big)\big\}_{i\in[n-1]}^{j\in[k]} \end{array} \right\}_{\mathbf{x}_{\geq 1} := (\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in \{0,1\}^{(n-1)k}}$$

$$\mathsf{DK}_{\mathbf{x}_i}: \mathsf{idDK}_i(L, \mathbf{x}_i)$$

$$\mathsf{DK}_f: \mathsf{iDK}_0([\![s, 1]\!]_2), \big\{\mathsf{iDK}\big((i,j), [\![s_{i,j,0}, s_{i,j,1}]\!]_2\big)\big\}_{i\in[0;n-1]}^{j\in[k]}$$

Here, $r_{\mathbf{x}_{\geq 1}} \xleftarrow{\$} \mathbb{Z}_p$ is a fresh random scalar for each $\mathbf{x}_{\geq 1} \in \{0,1\}^{(n-1)k}$. Given $\mathsf{CT}_{\mathbf{x}_0}$ and $\mathsf{DK}_{\mathbf{x}_1}, \ldots, \mathsf{DK}_{\mathbf{x}_{n-1}}$, the decryptor first runs IBE decryption to recover the IPFE ciphertexts corresponding to $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$. In the second step, using the IPFE keys in DK_f, it runs IPFE decryption to obtain $[\![r_{\mathbf{x}_{\geq 1}} \cdot s + \mu]\!]_t$ and $\{[\![r_{\mathbf{x}_{\geq 1}} \cdot s_{j,\mathbf{x}_i[j]}]\!]_t\}_{i\in[0;n-1], j\in[k]}$. Finally, it is possible to recover $[\![\mu]\!]_t$ using LSS reconstruction as in the KP-ABE case.

Succinct Enumerability. This naive MC-ABE construction becomes inefficient if k or n are chosen too large. The authors of [PS24] observe however that the size of the ciphertexts can be kept polynomial for particular function classes even when $k, n = \mathrm{poly}(\lambda)$. As an intuitive example, let us consider NC^0 policies. The output of a function that can be computed by an NC^0 circuit depends only on a set $\mathfrak{s} \subseteq [0; n-1] \times [k]$ of size $\tau = O(1)$ out of the total of kn inputs. As the remaining $kn - \tau$ inputs can be chosen arbitrarily without changing the output, it suffices to consider vectors $\mathbf{x}_{\geq 1}$ that are 0 outside $\mathfrak{s}$. There are $\binom{kn}{\tau} \leq (kn)^\tau = \mathrm{poly}(\lambda)$ possible sets $\mathfrak{s}$ and for each choice we only need to consider $2^\tau = O(1)$ vectors $\mathbf{x}_{\geq 1}$. So the total size of the ciphertexts remains polynomial.

Let us take a closer look at the concrete properties that make the construction work even for $k, n = \mathrm{poly}(\lambda)$. Initially, the encryptor needs to pick a subset $V \subseteq \{0,1\}^{[n-1]\times[k]}$ for which it provides independent ciphertexts. This set must be of polynomial size and cannot depend on inputs $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$ or a policy f as these values are unknown to the encryptor. To guarantee correctness, the set V must be chosen in such a way that, for each combination of attributes $(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1})$ and function f such that $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = 0$, there exists an element $\mathbf{v} \in V$ that coincides with $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$ on a "sufficiently large" subset of the indices $[n-1] \times [k]$ such that we can recover enough shares to run the LSS reconstruction algorithm. In a bit more detail, to make sure that reconstruction works even without holding a share for each coordinate, we view the policies as functions with a restricted domain by ignoring all inputs that do not have an impact on the output. Note that these restricted domains may be different for each function of the class. While this domain restriction guarantees correctness, we must be careful not to violate security. In particular, we have to make sure that we do not give the adversary the opportunity to change the output of the function when restricting the domain too much.

This leads us to the following condition dubbed *succinct enumerability* which is sufficient for a policy class to be captured by our construction. Let us introduce the following notations. We set $X_0 = \{0\}$, $X_1 = \{1\}$ and $X_\star = \{0,1\}$. Given a vector $\mathbf{v} \in \{0,1,\star\}^k$, we denote $X_{\mathbf{v}} = X_{\mathbf{v}[1]} \times \cdots \times X_{\mathbf{v}[k]}$. Furthermore, for concatenated vectors $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in \{0,1,\star\}^{[n-1]\times[k]}$, we write

$$X_{\mathbf{v}} = X_{\mathbf{v}_1} \times \cdots \times X_{\mathbf{v}_{n-1}} \qquad \text{and} \qquad J_{\mathbf{v}} = \big\{(i,j) \in [n-1] \times [k] : \mathbf{v}_i[j] \neq \star\big\}.$$

On an intuitive level, the set $X_{\mathbf{v}}$ contains all $\mathbf{x}_{\geq 1} = (\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in \{0,1\}^{[n-1]\times[k]}$ that coincide with $\mathbf{v}$ in the $\{0,1\}$-coordinates and have an arbitrary value in $\mathbf{v}$'s $\star$-coordinates. $J_{\mathbf{v}}$ contains the indices of all non-$\star$ coordinates of $\mathbf{v}$. A policy class $\mathcal{F}$ is said to be efficiently enumerable if there exists a fixed set $V \subseteq \{0,1,\star\}^{[n-1]\times[k]}$ of polynomial size with the following properties. For each policy $f \in \mathcal{F}$ there exists an index set $J_f \subseteq [n-1] \times [k]$ such that

1. for each $(\mathbf{x}_0, \mathbf{x}_{\geq 1}) \in \{0,1\}^{\{0\}\times[k]} \times \{0,1\}^{[n-1]\times[k]}$, if $f(\mathbf{x}_0, \mathbf{x}_{\geq 1}) = 0$ then there exists $\mathbf{v} \in V$ such that $J_{\mathbf{v}} = J_f$ and $\mathbf{x}_{\geq 1} \in X_{\mathbf{v}}$, and

2. for each $\mathbf{x}_0 \in \{0,1\}^{\{0\}\times[k]}$ and $\mathbf{v} \in V$, if $J_{\mathbf{v}} = J_f$ then the function f is constant on the set $\{(\mathbf{x}_0, \mathbf{x}_{\geq 1}) : \mathbf{x}_{\geq 1} \in X_{\mathbf{v}}\}$.

The first condition is needed for correctness to ensure that we can always recover a set of shares sufficient for reconstruction. The second property is used during the security proof to justify the restriction of the policy f to the smaller domain J_f. When adapting the construction from the NC^0 case to a general succinctly enumerable function class, we obtain the following scheme.

Construction 1 (MC-ABE for Succintly Enumerable Policy Classes – Informal)**.** Let $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ be a pairing-based IB-IPFE scheme and $\mathsf{idFE} = (\mathsf{idSetup}, \mathsf{idEnc}, \mathsf{idKeyGen}, \mathsf{idDec})$ be an IBE scheme.

$\mathsf{Setup}(1^\lambda)$**:** Sample $\mathsf{iMSK} \leftarrow \mathsf{iSetup}(1^\lambda)$ and $\mathsf{idMSK}_i \leftarrow \mathsf{idSetup}(1^\lambda)$ for $i \in [n-1]$. Then output $\mathsf{MPK} = (\mathsf{iMPK}, \{\mathsf{idMPK}_i\}_{i\in[n-1]})$, $\{\mathsf{SK}_i = \mathsf{idMSK}_i\}_{i\in[n-1]}$ and $\mathsf{MSK} = \mathsf{iMSK}$.

$\mathsf{Enc}(\mathsf{MPK}, L, \mathbf{x}_0, [\![\mu]\!]_\mathsf{t})$ Sample $[\![r^{\mathbf{v}}]\!]_1 \xleftarrow{\$} \mathbb{G}_1^\kappa$ for each $\mathbf{v} \in V$, set $\mathbf{v}_0 = \mathbf{x}_0$ and run

$$\begin{aligned}
\left\{\mathsf{iCT}^{\mathbf{v}}_0 \leftarrow \mathsf{iEnc}\left(\mathsf{iMPK}, 0, [\![r^{\mathbf{v}}, \mu]\!]_1\right)\right\}^{\mathbf{v}\in V} \\
\left\{\mathsf{iCT}^{\mathbf{v}}_{i,j} \leftarrow \mathsf{iEnc}\left(\mathsf{iMPK}, (i,j), [\![0^{\mathbf{v}_i[j]}, r^{\mathbf{v}}, 0^{(1-\mathbf{v}_i[j])}]\!]_1\right)\right\}^{\mathbf{v}=(\mathbf{v}_1,\ldots,\mathbf{v}_{n-1})\in V}_{(i,j)\in J_{\mathbf{v}}\cup(\{0\}\times[k])} \\
\left\{\mathsf{idCT}^{\mathbf{v}}_{i,j} \leftarrow \mathsf{idEnc}\left(\mathsf{idMPK}_i, (L, \mathbf{v}_i), \mathsf{iCT}^{\mathbf{v}}_{i,j}\right)\right\}^{\mathbf{v}=(\mathbf{v}_1,\ldots,\mathbf{v}_{n-1})\in V}_{(i,j)\in J_{\mathbf{v}}} .
\end{aligned}$$

Output $\mathsf{CT} = \left([\![d]\!]_\mathsf{t}, \left\{\mathsf{iCT}^{\mathbf{v}}_0\right\}^{\mathbf{v}\in V}, \left\{\mathsf{iCT}^{\mathbf{v}}_{0,j}\right\}^{\mathbf{v}\in V}_{j\in[k]}, \left\{\mathsf{idCT}^{\mathbf{v}}_{i,j}\right\}^{\mathbf{v}\in V}_{(i,j)\in J_{\mathbf{v}}}\right)$.

$\mathsf{AKeyGen}(\mathsf{SK}_i, L, \mathbf{x}_i)$ Define $V_{\mathbf{x}_i} = \left\{\mathbf{v}_i : (\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in V, \mathbf{x}_i \in X_{\mathbf{v}_i}\right\}$ and run

$$\left\{\mathsf{idDK}^{\mathbf{v}_i}_i \leftarrow \mathsf{idKeyGen}\left(\mathsf{idMSK}_i, (L, \mathbf{v}_i)\right)\right\}^{\mathbf{v}_i\in V_{\mathbf{x}_i}} .$$

Output $\mathsf{DK}_i = \{\mathsf{idDK}^{\mathbf{v}_i}_i\}^{\mathbf{v}_i\in V_{\mathbf{x}_i}}$.

$\mathsf{PKeyGen}(\mathsf{MSK}, f)$ Sample $s \xleftarrow{\$} \mathbb{Z}_p^\kappa$, generates shares $\{s_{i,j,b}\}_{(i,j)\in J_f, b\in\{0,1\}} \leftarrow \mathsf{Share}(f, s)$ and run

$$\begin{aligned}
\mathsf{iDK}_0 &\leftarrow \mathsf{iKeyGen}\left(\mathsf{iMSK}, 0, [\![s, 1]\!]_2\right) \\
\mathsf{iDK}_{i,j} &\leftarrow \mathsf{iKeyGen}\left(\mathsf{iMSK}, (i,j), [\![s_{i,j,0}, s_{i,j,1}]\!]_2\right) \qquad \text{for each } (i,j) \in J_f \cup (\{0\}\times[k]) .
\end{aligned}$$

Output $\mathsf{DK}_f = (\mathsf{iDK}_0, \{\mathsf{iDK}_{i,j}\}_{(i,j)\in J_f})$.

$\mathsf{Dec}(\mathsf{DK}_f, \{\mathsf{DK}_i\}_{i\in[n-1]}, \mathsf{CT})$ Pick $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in V$ such that $J_{\mathbf{v}} = J_f$ and $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in X_{\mathbf{v}}$. Run

$$\begin{aligned}
\left\{\mathsf{iCT}^{\mathbf{v}}_{i,j} \leftarrow \mathsf{idDec}\left(\mathsf{idDK}^{\mathbf{v}_i}_i, \mathsf{idCT}^{\mathbf{v}}_{i,j}\right)\right\}_{(i,j)\in J_{\mathbf{v}}} \\
\left\{[\![d_{i,j}]\!]_\mathsf{t} \leftarrow \mathsf{iDec}\left(\mathsf{iDK}_{i,j}, \mathsf{iCT}^{\mathbf{v}}_{i,j}\right)\right\}_{(i,j)\in J_{\mathbf{v}}\cup(\{0\}\times[k])} \\
[\![d_0]\!]_\mathsf{t} \leftarrow \mathsf{iDec}\left(\mathsf{iDK}_0, \mathsf{iCT}^{\mathbf{v}}_0\right) .
\end{aligned}$$

Run LSS reconstruction on the set $\{[\![d_{i,j}]\!]_\mathsf{t}\}_{(i,j)\in J_{\mathbf{v}}\cup(\{0\}\times[k])}$ to find $[\![d']\!]_\mathsf{t} = [\![r\cdot s]\!]_\mathsf{t}$ and output $[\![d_0 - d']\!]_\mathsf{t}$.

The first property of succinctly enumerable function classes guarantees that there exists $\mathbf{v} \in V$ such that $J_{\mathbf{v}} = J_f$ and $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in X_{\mathbf{v}}$. This implies that all idFE ciphertexts $\{\mathsf{idCT}^{\mathbf{v}}_{i,j}\}_{(i,j)\in J_{\mathbf{v}}}$ and secret keys $\{\mathsf{idDK}^{\mathbf{v}}_{i,j}\}_{(i,j)\in J_{\mathbf{v}}}$ used in the first step of decryption are contained in CT_L and $\{\mathsf{DK}_i\}_{i\in[n-1]}$, respectively. It then follows that $d_0 = \mu + r\cdot s$ and $d_{i,j} = r\cdot s_{i,j,\mathbf{x}_i[j]}$. If $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = 0$, the LSS reconstruction correctly recovers $[\![r\cdot s]\!]_\mathsf{t}$ and $[\![\mu]\!]_\mathsf{t}$ is revealed.

Security. The security proof follows a well-established technique. We use a primitive called slotted IPFE [LL20] which is a mix between public-key and private-key IPFE that provides standard security on the public part and additionally hides the function vectors on the private part. While the real scheme only uses the public part, the private part helps us to move the scalars r from the message vectors encoded in $\mathbb{G}_1$ into a hidden coordinate of the function vectors in $\mathbb{G}_2$. The only values that the adversary learns after this transformation are

- encodings $[\![s]\!]_2$ of random scalars $s \xleftarrow{\$} \mathbb{Z}_p$ sampled during the key generation and their corresponding shares $\{[\![s_{i,j,b}]\!]_2\}_{i \in J_f \cup (\{0\} \times [k])}$, and
- encodings of the form $[\![r \cdot s + \mu]\!]_2$ (iFE indentity 0) and $[\![r \cdot s_{i,j,\mathbf{x}_i[j]}]\!]_2$ (iFE identity (i,j)) for random scalars $r \xleftarrow{\$} \mathbb{Z}_p$ sampled during the encryption of the challenge message.

Importantly, nothing about r is leaked in $\mathbb{G}_1$. So we can rely on the DDH assumption to obtain a fresh looking mask $[\![\widetilde{s}]\!]_t \approx_c [\![r \cdot s]\!]_t$ with fresh shares $[\![\widetilde{s}_{i,j}]\!]_t \approx_c [\![\{[\![s_{i,j,\mathbf{x}_i[j]}]\!]_2\}_{i \in J_f \cup (\{0\} \times [k])}]\!]_2$ for each combination of r and s. Then we can exploit the one-time, information-theoretic security provided by the LSS scheme to replace the individual secret sharings with random values: by the admissibility of the adversary and the second item of the succinct enumerability property, there does not exist any pair (r,s) such that the adversary has sufficient information about a subset of shares that allows her to recover the mask $[\![r \cdot s]\!]_t$. Instead, they look uniformly random and, thus, perfectly hide the challenge message.

Lattice-Based Construction. Slotted IPFE is only known from the MDDH assumption on pairings. A lattice-based instantiation of Construction 1 must therefore employ a different IPFE primitive. The crucial step in the above sketch of the security proof happens when applying the DDH assumption: at this point, we switch from *one* set of secret shares that appears in the decryption results with all ciphertexts to *many* independent sharings, one for each ciphertext. In the lattice setting, we need an IPFE primitive which allows us to perform a similar step while relying on LWE. Very recently, Hsieh et al. [HLL24] introduced a new variant of IPFE called *evasive* IPFE for precisely this purpose. Using this new IPFE primitive and a noisy variant of LSS schemes also introduced in that work, everything follows quite readily. Notably, we can rely on a restricted security requirement of evasive IPFE in which case security can be reduced to public-coin evasive LWE. We omit further details here and refer the reader to Section 4.3.

2.2 MC-ABE for Conjunction Policies

The work of Agrawal et al. [ATY23], henceforth referred to as ATY, presents an AB-MIFE for the Attribute-Weighted Sums (AWS) functionality. We recall their construction in the special case of inner products, i.e. the functionality $F\colon (\mathcal{X} \times \mathbb{Z}_p^d)^n \to \mathbb{G}_t$ specified by $((f_0,\mathbf{v}_0),\dots,(f_{n-1},\mathbf{v}_{n-1})) \in (\mathcal{F} \times \mathbb{Z}_p^d)^n$ and defined as

$$F((x_0,\mathbf{u}_0),\dots,(x_{n-1},\mathbf{u}_{n-1})) = \begin{cases} \sum_{i \in [0;n-1]} [\![\mathbf{u}_i \mathbf{v}_i^\top]\!]_t & \text{if } f_i(x_i) = 0 \text{ for all } i \in [0;n-1]\,, \\ \bot & \text{otherwise}\,. \end{cases}$$

The construction's main building block is an AB-IPFE scheme $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ with (partially) function-hiding security. Specifically, $\mathsf{iCT} \leftarrow \mathsf{iEnc}(\mathsf{MSK}, x, \mathbf{u}')$ takes as input the master secret key MSK, a public attribute $x \in \mathcal{X}$ and a vector $\mathbf{u}' \in \mathbb{Z}_p^{d+1}$; $\mathsf{iDK} \leftarrow \mathsf{iKeyGen}(\mathsf{MSK}, f, \mathbf{v}')$ takes MSK, a policy $f \in \mathcal{F}$ and a vector $\mathbf{v}' \in \mathbb{Z}_p^{d+1}$. If $f(x) = 0$, then $\mathsf{iDec}(\mathsf{iDK}, \mathsf{iCT})$ returns $\mathbf{u}' \cdot (\mathbf{v}')^\top$ (but reveals nothing else about $\mathbf{u}'$ and $\mathbf{v}'$). Conversely, if $f(x) \neq 0$, then $\mathbf{u}'$ and $\mathbf{v}'$ are completely hidden.

Recap: The AB-MIFE of ATY. In the AB-MIFE construction of ATY, the setup algorithm provides each slot $i \in [0;n-1]$ with their own secret key $\mathsf{SK}_i := \mathsf{iMSK}_i \leftarrow \mathsf{iSetup}(1^\lambda)$ and defines the corresponding master secret key $\mathsf{MSK} = \{\mathsf{iMSK}_i\}_{i \in [0;n-1]}$. Given $(x_i, \mathbf{u}_i)$, the encryptor for slot $i \in [0;n-1]$ runs $\mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_i, \mathbf{u}_i')$ for $\mathbf{u}_i' = (\mathbf{u}_i, 1)$ and outputs $\mathsf{CT}_i := \mathsf{iCT}_i$. Given $\{(f_i, \mathbf{v}_i)\}_{i \in [0;n-1]}$, the key generator samples $t_1,\dots,t_{n-1} \xleftarrow{\$} \mathbb{Z}_p$,

sets $t_0 = -\sum_{i\in[n-1]} t_i$, runs $\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, f_i, \mathbf{v}'_i)$ for $\mathbf{v}'_i = (\mathbf{v}_i, t_i)$ and outputs $\mathsf{DK} = \{\mathsf{iDK}_i\}_{i\in[0;n-1]}$. If $f_i(x_i) = 0$ for all $i \in [0; n-1]$, decryption outputs

$$\sum_{i\in[0;n-1]} \mathsf{iDec}(\mathsf{iDK}_i, \mathsf{iCT}_i) = \sum_{i\in[0;n-1]} [\![\mathbf{u}_i \mathbf{v}_i^\top]\!]_t \,,$$

where the equality follows from the fact that $\sum_{i\in[0;n-1]} t_i = 0$. The random element t_i is used during the security proof to hide partial decryption values $\mathbf{u}_i\mathbf{v}_i^\top$. Importantly, this scheme is only secure if the adversary does not possess so-called illegitimate keys. A decryption key for a function $F = ((f_0, \mathbf{v}_0), \ldots, (f_{n-1}, \mathbf{v}_{n-1}))$ is said to be *illegitimate* if, for some slot i, the adversary did not query for a slot-i ciphertext encoding $(x_i, \star)$ such that $f_i(x_i) = 0$. We refer the reader to ATY for a detailed discussion why the scheme becomes insecure in the presence of illegitimate keys. Importantly for us, ATY presents a generic compiler to remove this restriction from the security model and achieve security against any keys.

From MI-ABE to MC-ABE. The scheme of ATY directly implies an MI-ABE for conjunction policies. Let $d = 1$. Given a message $[\![\mu]\!]_t \in \mathbb{G}_t$, the encryptor (slot 0) samples a random masking term $r \xleftarrow{\$} \mathbb{Z}_p$, encrypts $\mathsf{iCT}_0 \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, (r, 1))$ and outputs $\mathsf{CT} := (\mathsf{iCT}_0, [\![c]\!]_t = [\![r + \mu]\!]_t)$. The attribute key generators (slots $i \in [n-1]$) proceed in the same way as in the AB-MIFE scheme except that they always set $\mathbf{u}_i = 0$. If $f_i(x_i) = 0$ for all $i \in [0; n-1]$, then iFE decryption reveals $\sum_{i\in[0;n-1]} \mathsf{iDec}(\mathsf{iDK}_i, \mathsf{iCT}_i) = [\![r]\!]_t$, and we can recover $[\![\mu]\!]_t = [\![c - r]\!]_t$. We note that this yields an MI-ABE in the secret-key setting, where the encryptor takes a secret key SK_0 instead of a public key MPK. As the security relies on the *function-hiding* property of iFE, it is not possible to replace iFE with a public-key AB-IPFE which might allow to disclose the encryption key for slot 0.[5]

We next discuss how to lift this MI-ABE to MC-ABE. Fortunately for us, the AB-MIFE of ATY (and thus the implied MI-ABE) are already secure against corruptions of secret keys. So the only missing feature towards MC-ABE is managing different labels. For the case of security against legitimate keys, such a scheme can be obtained by generalizing known techniques for building MCFE for inner products [ABG19] to the attribute-based setting. Specifically, we use a fresh set of random scalars $t_{L,0}, \ldots, t_{L,n-1} \xleftarrow{\$} \mathbb{Z}_p$ for each label L and move them from the key vectors $\mathbf{v}'_i$ into the ciphertext vectors $\mathbf{u}'_i$, i.e., $\mathbf{u}'_i = (\mathbf{u}_i, t_{L,i})$. Correspondingly, key generation samples a random scalar $s \xleftarrow{\$} \mathbb{Z}_p$ and places it in the coordinates facing $t_{L,i}$, i.e., $\mathbf{v}'_i = (\mathbf{v}_i, s)$. The scalars $\{t_{L,i}\}_{i\in[0;n-1]}$ can be simulated "on the fly" by using a pseudorandom function PRF with key space $\mathcal{K}$. For each $i, j \in [0; n-1]$, $i < j$, sample $K_{i,j} \xleftarrow{\$} \mathcal{K}$ and set $K_{i,j} = K_{j,i}$ for $j > i$. Then we let

$$t_{L,i} = \sum_{j\in[0;n-1]\setminus\{i\}} (-1)^{(j<i)} \mathsf{PRF}_{K_{i,j}}(L) \,.$$

It is not hard to see that $\sum_{i\in[0;n-1]} t_{L,i} = 0$ for any label L. Pseudorandomness follows from the security of PRF.

Security Against Any Keys. While this AB-MCFE scheme trivially follows from existing constructions, it only provides security against legitimate keys. The actual hurdle lies in achieving security against illegitimate keys because the ATY compiler from the AB-MIFE setting is not able to handle different labels. We do not know how to deal with this issue in the general case of AB-MCFE. Let us therefore focus on how to deal with illegitimate keys in the special case of MC-ABE.

In this case, the legitimacy condition simplifies as there is only one slot that encrypts a message and thus receives left-or-right queries during the security game. Specifically, a key for a policy described by $(f_0, \ldots, f_{n-1})$ is regarded illegitimate in the spirit of the prior definition if there exists a slot $i \in [n-1]$ and a label L such that

1. $\mathcal{A}$ does obtain a ciphertext for (L, x_0, μ) such that $f_0(x_0) = 0$, but
2. $\mathcal{A}$ does not obtain an attribute decryption key DK_i for L and an attribute x_i such that $f_i(x_i) = 0$.

[5] In more detail, if the adversary held an IPFE encryption key, she could always generate ciphertexts of all unit vectors by herself which allow recovering the coordinates of the key vector one by one. Therefore, function-hiding IPFE cannot exist in the public-key setting.

In this case, the adversary is supposed to learn nothing about μ as globally the policy is not satisfied. Looking at our construction, however, we can observe that this is not true. To see why, let us consider a two-client scheme where an adversary obtains ciphertexts of (L, x_0, μ_1) and (L, x_0, μ_2) for slot 0 (denoted by $\mathsf{CT}_1 = (\mathsf{iCT}_1, [\![c_1]\!]_t = [\![\mu_1 + r_1]\!]_t)$ and $\mathsf{CT}_2 = (\mathsf{iCT}_2, [\![c_2]\!]_t = [\![\mu_2 + r_2]\!]_t)$) and a policy decryption key for $F = (f_0, f_1)$ denoted by $\mathsf{SK}_F = \{\mathsf{iDK}_i\}_{i\in\{0,1\}}$ such that $f_0(x_0) = 0$. In this case, the adversary should not obtain any information about μ_1 and μ_2 since she does not possess an attribute decryption key $\mathsf{DK}_{L,1}$ satisfying f_1. However, the adversary can compute

$$[\![r_\Delta]\!]_t = \mathsf{iDec}(\mathsf{iDK}_0, \mathsf{iCT}_1) - \mathsf{iDec}(\mathsf{iDK}_0, \mathsf{iCT}_2) = [\![(st_{L,0} + r_1) - (st_{L,0} + r_2)]\!]_t = [\![r_1 - r_2]\!]_t \,,$$

where we recall that iCT_j, for $j \in \{1,2\}$, encrypts a vector $(t_{L,0}, r_j)$ and iDK_0 encodes a vector $(s, 1)$. Thus, the adversary can learn $[\![c_1 - c_2 - r_\Delta]\!]_t = [\![\mu_1 - \mu_2]\!]_t$. This is leakage we need to avoid.

At this point, our observation is that even though the adversary can learn certain relations between the (potentially many) masking terms r, the resulting linear system is always underdetermined. More precisely, an analysis of this linear system shows that the marginal distribution of each r is still uniform. Therefore, we can use a random oracle to turn the dependent random variables into independent variables which can perfectly hide the messages μ_1 and μ_2. The resulting scheme can be summarized as follows. It provides selective security in the ROM assuming MDDH. Note that we can simplify the generation of the elements $t_{L,i}$ as there is only one message-encrypting slot in MC-ABE.

Construction 2 (MC-ABE from Function-Hiding IPFE in ROM – Informal)**.** Let $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ be an AB-IPFE scheme with function-hiding security on the inner-product vectors, PRF be a pseudorandom function with key space $\mathcal{K}$, and $H\colon \mathbb{G}_t \to \mathbb{G}_t$ be a hash function modeled as a random oracle.

$\mathsf{Setup}(1^\lambda)$**:** Sample $\mathsf{iMSK}_i \leftarrow \mathsf{iSetup}(1^\lambda)$ for $i \in [0; n-1]$, and $K_i \xleftarrow{\$} \mathcal{K}$ for $i \in [n-1]$. Output $\mathsf{SK}_0 := (\mathsf{iMSK}_0, \{K_i\}_{i\in[n-1]})$, $\{\mathsf{SK}_i := (\mathsf{iMSK}_i, K_i)\}_{i\in[n-1]}$ and $\mathsf{MSK} := \{\mathsf{iMSK}_i\}_{i\in[0;n-1]}$.

$\mathsf{Enc}(\mathsf{SK}_0, L, x_0, [\![\mu]\!]_t)$**:** Sample $r \xleftarrow{\$} \mathbb{Z}_p$, set $t_{L,0} = -\sum_{i\in[n-1]} \mathsf{PRF}_{K_i}(L)$, run $\mathsf{iCT}_0 \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, [\![t_{L,0}, r]\!]_1)$ and output $\mathsf{CT} := (\mathsf{iCT}_0, [\![c]\!]_t = [\![\mu]\!]_t + H([\![r]\!]_t))$.

$\mathsf{AKeyGen}(\mathsf{SK}_i, L, x_i)$**:** Set $t_{L,i} = \mathsf{PRF}_{K_i}(L)$, run $\mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, [\![t_{L,i}, 0]\!]_1)$ and output $\mathsf{DK}_i := \mathsf{iCT}_i$.

$\mathsf{PKeyGen}(\mathsf{MSK}, f = (f_0, \ldots, f_{n-1}))$**:** Run $\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, f_i, [\![s, 1]\!]_2)$ for each $i \in [0; n-1]$, where $s \xleftarrow{\$} \mathbb{Z}_p$. Output $\mathsf{DK}_f := \{\mathsf{iDK}_i\}_{i\in[0;n-1]}$.

$\mathsf{Dec}(\mathsf{DK}_f, \{\mathsf{DK}_i\}_{i\in[n-1]}, \mathsf{CT})$**:** For $i \in [0; n-1]$, run $[\![d_i]\!]_t \leftarrow \mathsf{iDec}(\mathsf{iDK}_i, \mathsf{iCT}_i)$. Output $[\![c]\!]_t - H(\sum_{i\in[0;n-1]} [\![d_i]\!]_t)$.

Security in the Random Oracle Model. Let us analyze why the random oracle H suffices to prove security against illegitimate keys. Suppose the adversary obtains a policy decryption key $\mathsf{DK}_f := \{\mathsf{iDK}_i\}_{i\in[0;n-1]}$ for some policy f and ciphertexts $\mathsf{CT}_j := (\mathsf{iCT}_{j,0}, [\![c_j]\!]_t = [\![\mu]\!]_t + H([\![r_j]\!]_t))$ for $j \in [q]$, where $r_j \xleftarrow{\$} \mathbb{Z}_p$ is sampled for the reply to the j-th encryption query $\mathsf{QEnc}(L, x_{j,0}, [\![\mu_j]\!]_t)$ with respect to a label L such that $f(x_j) = 0$. Roughly speaking, this allows the adversary to learn the values $[\![r'_j]\!]_t = [\![st_{L,0} + r_j]\!]_t = [\![-s\sum_{i\in[n-1]} t_{L,i} + r_j]\!]_t$ by running iFE decryption. Hence, for proving security we must argue that the masks $H([\![r_j]\!]_t)$ are pseudorandom even if the adversary learns the terms $[\![r'_j]\!]_t$. By item 2, it follows that there exists a slot $i \in [n-1]$ such that the adversary does not learn $[\![st_{L,i}]\!]_t$, and we note that $[\![st_{L,i}]\!]_t \approx_c [\![s']\!]_t$ for $s' \xleftarrow{\$} \mathbb{Z}_p$ under the DDH assumption. So it suffices to show that the following distributions are indistinguishable

$$\{([\![s' + r_j]\!]_t, H([\![r_j]\!]_t))\}_{j\in[q]} \approx_c \{([\![s' + r_j]\!]_t, [\![y_j]\!]_t)\}_{j\in[q]},$$

where $y_1, \ldots, y_q \xleftarrow{\$} \mathbb{Z}_p$. For a single query to the random oracle, the probability to have asked an r_j is bounded by q/p. This allows the adversary to exclude up to q^2 points using the constraints on $\{s' + r_j\}_{j\in[q]}$. Thus, after $t = \mathrm{poly}(\lambda)$ queries, the probability to have queried the random oracle on one of the r_j is at most $qt/(p - q^2 t)$ which is negligible in λ.

Lattice-Based Construction. We now focus on a construction that does not use pairings and can be proven secure assuming LWE. To our knowledge, there is currently no lattice-based AB-MIFE scheme for inner products in the literature. We therefore go back to the original AB-MIFE of ATY and attempt to build a lattice-version for this functionality first. Subsequently, we can follow the same path towards MC-ABE as described in the pairing-based setting since these ideas do not exploit the underlying group or pairing structure.

The biggest hurdle in the lattice-based setting is that we can no longer rely on AB-IPFE with function-hiding security, so the ATY compiler from (single-input) AB-IPFE to AB-MIFE cannot be instantiated anymore. As observed by ATY, existing MIFE constructions for inner products (without access control) can roughly be classified into two categories depending on which property of the employed IPFE primitive is used for the security proof: (1) ciphertext homomorphism, e.g., [AGRW17, CDG+18, ACF+18, ABG19], and (2) function-hiding security, e.g., [DOT18, Tom19, AGT21b, SV23]. Here, ciphertext homomorphism means that both message space and ciphertext space form a group, and taking the sum of two ciphertexts encrypting messages $\mathbf{u}_1$ and $\mathbf{u}_2$ yields a ciphertext for $\mathbf{u}_1 + \mathbf{u}_2$. In the case of AB-MIFE, we evaluate non-linear policies such as ABPs on the public input. It is therefore hopeless trying to equip an AB-IPFE scheme with ciphertext homomorphism. This rules out all lattice-based MIFE constructions for inner products that we could potentially generalize to the attribute-based setting.[6] It therefore seems inevitable to somehow mimic the proof technique of the function-hiding case even though we cannot rely on function-hiding security.

To our knowledge, the only existing AB-IPFE scheme in the lattice setting was presented by Lai et al. in [LLW21] and is only secure against *bounded collusions*, i.e., the adversary obtains only a polynomial number Q of decryption keys with Q being fixed at the time of setup. Given that even in the single-input case the best known construction is only secure against bounded collusions, we restrict ourselves to the same level of security in the multi-input case.

It turns out that this limitation helps a lot. To see why, we recall that the AB-MIFE of ATY encrypts $(x_i, \mathbf{u}_i)$ by running $\mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_i, \mathbf{u}'_i)$ for the vector $\mathbf{u}'_i = (\mathbf{u}_i, 1)$. To generate a decryption key for policy-vector pairs $\{(f_i, \mathbf{v}_i)\}_{i\in[0;n-1]}$, one samples $t_0, \ldots, t_{n-1} \xleftarrow{\$} \mathbb{Z}_p$ such that $\sum_{i\in[0;n-1]} t_i = 0$ and computes $\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, f_i, \mathbf{v}'_i)$ for all $i \in [0; n-1]$ for the vectors $\mathbf{v}'_i = (\mathbf{v}_i, t_i)$. Suppose that the adversary submits a slot-i encryption query $\mathsf{QEnc}(i, x_i, \mathbf{u}^0_i, \mathbf{u}^1_i)$ for an attribute x_i and message vectors $(\mathbf{u}^0_i, \mathbf{u}^1_i)$ along with a key generation query $\mathsf{QKeyGen}(\{(f_{i'}, \mathbf{v}_{i'})\}_{i'\in[0;n-1]})$ for policy-vector pairs $(f_{i'}, \mathbf{v}_{i'})$. If $f_i(x_i) = 0$, then the adversary can learn $\mathbf{u}'_i(\mathbf{v}'_i)^\top = \mathbf{u}_i\mathbf{v}_i^\top + t_i$, where the random scalar t_i is used to hide the local inner product $\mathbf{u}_i\mathbf{v}_i$ from the adversary. More precisely, during the security proof the scalar t_i is used to compensate for the difference between the left and the right inner product $\delta_i := \mathbf{u}^0_i\mathbf{v}_i^\top - \mathbf{u}^1_i\mathbf{v}_i^\top$ in slot i. Since t_i is placed in the key vectors, this compensation step requires function-hiding security. Without function-hiding, the only choice is to flip the construction and put the scalar t_i in the ciphertexts instead. (Note that we already did this in our pairing-based MC-ABE scheme for different reasons). While this seemingly solves the issue with the lack of function-hiding security, it immediately leads to a new problem. This is because the value δ_i does not only depend on the slot index i, but also on the vector $\mathbf{v}_i$, i.e., δ_i can be different for every key query. In the pairing-based setting, this issue can be dealt with by placing a random scalar s in the key vector in the coordinate facing t_i. Under a DDH-style assumption, we can obtain a different randomization of t_i for each key. But we do not have an analogous assumption in the lattice-based setting. So we cannot randomize t_i during the security proof anymore.

Security Against Bounded Collusions. This is the point where the restriction to a fixed number Q of decryption keys comes to the rescue as it limits the total number of different δ_i's. It is therefore possible to embed sufficiently many independent random scalars $t^1_i, t^2_i, \ldots$ into the ciphertext at the time of encryption. Naively, we could use Q different t_i's, one for each decryption key. But this would require a stateful key generation algorithm which maintains a counter to keep track of which t_i's were already used for previously issued keys. To avoid this, we rely on a probabilistic argument using cover-free sets, inspired by [Agr17, LLW21]. Intuitively,

[6] The lattice-based construction of Libert and Titiu [LT19] is somewhat different in the sense that it does not really fit in one of the two categories. However, we do not know how to enhance their construction with access control.

for a fixed $Q = \text{poly}(\lambda)$, we can choose κ, τ such that independently sampled subsets $\Delta_1, \ldots, \Delta_Q \subset [\kappa]$ of size τ satisfy $\Delta_i \setminus (\bigcup_{j \in [Q] \setminus \{i\}} \Delta_j) \neq \varnothing$ for all i with high probability.

We can use this result in the following way. During encryption, we replace $t_i \xleftarrow{\$} \mathbb{Z}_p$ with a vector $\mathbf{t}_i$ of dimension κ, i.e. we let $\mathbf{u}'_i = (\mathbf{u}_i, \mathbf{t}_i)$. During key generation, we sample a random subset $\Delta \subset [\kappa]$, define the vector $\mathbf{e}_\Delta = \sum_{k \in \Delta} \mathbf{e}_k \in \mathbb{Z}_p^\kappa$ and set $\mathbf{v}'_i = (\mathbf{v}_i, \mathbf{e}_\Delta)$. For the security proof, we use the fact that for each key $j \in [Q]$, there exists $k_j \in [\kappa]$ such that $\mathbf{t}_i[k_j]$ appears *only* when decrypting a ciphertext with the j-th decryption key. Hence, we can use $\mathbf{t}_i[k_j]$ to compensate for the value δ_i with respect to the j-th decryption key. The resulting AB-MIFE scheme can be summarized as follows:

Construction 3 (AB-MIFE Against Bounded Collusions – Informal)**.** Let $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ be an AB-IPFE scheme with standard security.

$\mathsf{Setup}(1^\lambda)$**:** Sample $\mathsf{iMSK}_i \leftarrow \mathsf{iSetup}(1^\lambda)$ and $\mathbf{t}_i \xleftarrow{\$} \mathbb{Z}_p^\kappa$ for $i \in [0; n-1]$ such that $\sum_{i \in [0; n-1]} \mathbf{t}_i = \mathbf{0}$. Output $\{\mathsf{SK}_i := (\mathsf{iMSK}_i, \mathbf{t}_i)\}_{i \in [0; n-1]}$ and $\mathsf{MSK} := \{\mathsf{iMSK}_i\}_{i \in [0; n-1]}$.

$\mathsf{Enc}(\mathsf{SK}_i, x_i, \mathbf{u}_i)$**:** Run $\mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, (\mathbf{u}_i, \mathbf{t}_i))$ and output $\mathsf{CT}_i := \mathsf{iCT}_i$.

$\mathsf{KeyGen}(\mathsf{MSK}, \{(f_i, \mathbf{v}_i)\}_{i \in [0; n-1]})$**:** Sample $\Delta \subseteq [\kappa]$ of size τ and run $\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, f_i, (\mathbf{v}_i, \mathbf{e}_\Delta))$ for each $i \in [0; n-1]$. Output $\mathsf{DK} := \{\mathsf{iDK}_i\}_{i \in [0; n-1]}$.

$\mathsf{Dec}(\mathsf{DK}_f, \{\mathsf{DK}_i\}_{i \in [n-1]}, \mathsf{CT})$**:** For $i \in [0; n-1]$, run $d_i \leftarrow \mathsf{iDec}(\mathsf{iDK}_i, \mathsf{iCT}_i)$ and output $d = \sum_{i \in [0; n-1]} d_i$.

A detailed description of the AB-MIFE can be found in Section A. Applying the same modification as in the pairing-based case yields our MC-ABE for conjunction policies from LWE.

3 Preliminaries

3.1 Notational Conventions

Let $\lambda \in \mathbb{N}$ be the security parameter. Except in the definitions, we will suppress λ in subscripts for brevity. A nonnegative function $\varepsilon \colon \mathbb{N} \to \mathbb{R}$ is negligible if $\varepsilon(\lambda) = O(\lambda^{-n})$ for all $n \in \mathbb{N}$. An algorithm is said to be *efficient* if it runs in probabilistic polynomial time (PPT) in the security parameter.

To avoid confusion, we always write vectors $\mathbf{v}$ and matrices $\mathbf{A}$ in boldface and use uppercase letters for the latter. Scalars s are written in italics. Unless otherwise stated, all vectors $\mathbf{v}$ are viewed as row vectors. The corresponding column vector is denoted by $\mathbf{v}^\top$.

Security Experiments and Distributions. Let $\mathbf{Exp}$ be an interactive *experiment* that interacts with an algorithm $\mathcal{A}$ (called the *adversary*), depends on the security parameter λ and has binary outcome. We also refer to such objects as *games* or *hybrids*. We let "$\mathbf{Exp}_{\mathcal{A}}(1^\lambda) \to 1$" denote the event that the outcome of running $\mathbf{Exp}$ with $\mathcal{A}$ on security parameter λ is 1. For two experiments $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$, we define the distinguishing advantage of $\mathcal{A}$ against the tuple $(\mathbf{Exp}^0, \mathbf{Exp}^1)$ as

$$\mathbf{Adv}_{\mathbf{Exp}^0, \mathbf{Exp}^1, \mathcal{A}}(\lambda) := \left| \Pr\left[\mathbf{Exp}^1_{\mathcal{A}}(1^\lambda) \to 1\right] - \Pr\left[\mathbf{Exp}^0_{\mathcal{A}}(1^\lambda) \to 1\right] \right| .$$

We write $\mathbf{Exp}^0 \approx_c \mathbf{Exp}^1$ if the experiments are *computationally indistinguishable*, i.e., their distinguishing advantage is negligible for all efficient adversaries $\mathcal{A}$. We write $\mathbf{Exp}^0 \approx_s \mathbf{Exp}^1$ if the experiments are *statistically indistinguishable*, i.e., their distinguishing advantage is negligible for all (even unbounded) adversaries. We write $\mathbf{Exp}^0 \equiv \mathbf{Exp}^1$ if the experiments are *identically distributed*, i.e., their distinguishing advantage is 0 for all (even unbounded) adversaries. By default, the term *indistinguishable* refers to computational indistinguishability.

More general, the same notations can be used for sequences of distributions. Let $D^0 = \{D^0_\lambda\}_{\lambda \in \mathbb{N}}$ and $D^1 = \{D^1_\lambda\}_{\lambda \in \mathbb{N}}$ be two sequences of distributions. For $b \in \{0, 1\}$, we define $\mathbf{Exp}^b_{\mathcal{A}}(1^\lambda)$ as follows: sample $x \xleftarrow{\$} D^b_\lambda$, run $\mathcal{A}(1^\lambda, x)$ and use the output of $\mathcal{A}$ as the outcome of the experiment. Then we write $D^0 \approx_c D^1$ (resp. $D^0 \approx_s D^1$, $D^0 \equiv D^1$) if $\mathbf{Exp}^0_{\mathcal{A}} \approx_c \mathbf{Exp}^1_{\mathcal{A}}$ (resp. $\mathbf{Exp}^0_{\mathcal{A}} \approx_s \mathbf{Exp}^1_{\mathcal{A}}$, $\mathbf{Exp}^0_{\mathcal{A}} \equiv \mathbf{Exp}^1_{\mathcal{A}}$).

Sets and Indexing. We denote by $\mathbb{Z}$ and $\mathbb{N}$ the sets of integers and natural numbers (positive integers). For integers m and n, we write $[m;n]$ to denote the set $\{z \in \mathbb{Z} : m \le z \le n\}$ and let $[n] := [1;n]$. For a prime number p, $\mathbb{Z}_p$ denotes the finite field of integers modulo p. For a finite set S, we let 2^S denote the power set of S.

To index a vector or the columns of a matrix, we write $\mathbf{v}[i]$ and $\mathbf{A}[j]$. In contrast, objects of some collection that is not regarded as a vector or matrix are indexed using subscripts (or superscripts in some cases). For instance, $\mathbf{v}_i$ represents a vector, not a component of some vector. If i runs through some index set $[n]$, it means that there are n vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. If the n objects are scalars (or not explicitly vectors), we will write $v_1, \ldots, v_n$ instead.

For convenience, objects might be indexed by arbitrary sets, not just integers. For finite sets $\mathfrak{s}, A$, we write $A^{\mathfrak{s}} := \{(\mathbf{v}[i])_{i\in\mathfrak{s}} : \mathbf{v}[i] \in A\}$ for the set of vectors whose entries are in A and indexed by $\mathfrak{s}$, e.g., $\mathbb{Z}_p^{[n]}$ is just $\mathbb{Z}_p^n$. Suppose $\mathfrak{s}_1, \mathfrak{s}_2$ are two index sets with $\mathfrak{s}_1 \subseteq \mathfrak{s}_2$. For a vector $\mathbf{v} \in \mathbb{Z}_p^{\mathfrak{s}_2}$, we denote by $\mathbf{u} = \mathbf{v}|_{\mathfrak{s}_1}$ its canonical projection onto $\mathbb{Z}_p^{\mathfrak{s}_1}$, i.e., $\mathbf{u} \in \mathbb{Z}_p^{\mathfrak{s}_1}$ and $\mathbf{u}[i] = \mathbf{v}[i]$ for all $i \in \mathfrak{s}_1$. Conversely, for any vector $\mathbf{u} \in \mathbb{Z}_p^{\mathfrak{s}_1}$, we write $\mathbf{v} = \mathbf{u}|^{\mathfrak{s}_2}$ for its zero-extension into $\mathbb{Z}_p^{\mathfrak{s}_2}$, i.e., $\mathbf{v} \in \mathbb{Z}_p^{\mathfrak{s}_2}$ and $\mathbf{v}[i] = \mathbf{u}[i]$ if $i \in \mathfrak{s}_1$, and $\mathbf{v}[i] = 0$ if $i \in \mathfrak{s}_2 \setminus \mathfrak{s}_1$.

3.2 Computational Models

Arithmetic Branching Programs (ABP). An ABP $f\colon \mathbb{Z}_p^n \to \mathbb{Z}_p$ is defined by a tuple $(V, E, s, t, p, n, \sigma)$ consisting of a directed acyclic graph (V, E) with two distinguished vertices $s, t \in V$, a prime p, an arity n and a labelling function $\sigma\colon E \to \mathcal{F}^{\mathsf{aff}}$, where $\mathcal{F}^{\mathsf{aff}}$ contains all affine functions $g\colon \mathbb{Z}_p^n \to \mathbb{Z}_p$. Let P be the set of all paths from s to t. The output of the ABP on input $\mathbf{x} \in \mathbb{Z}_p^n$ is defined as $f(\mathbf{x}) = \sum_{p\in P} \prod_{e\in p} \sigma(e)(\mathbf{x})$. We refer to $|V| + |E|$ as the size of f and denote by $\mathcal{F}_n^{\mathsf{abp}}$ the class of all functions $f\colon \mathbb{Z}_p^n \to \mathbb{Z}_p$ that can be computed by an ABP.

Bounded-Depth (Arithmetic) Circuits. We denote by $\mathcal{F}_{n,d}^{\mathsf{circ}}$ the class of functions $f\colon \{0,1\}^n \to \{0,1\}$ such that there exists a binary circuit C (using only NAND gates) of depth no more than d satisfying $C(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^n$.

3.3 Pairing Groups

Our constructions use a sequence of pairing groups

$$\mathbb{G} = \{\mathbb{G}_\lambda = (\mathbb{G}_{\lambda,1}, \mathbb{G}_{\lambda,2}, \mathbb{G}_{\lambda,\mathsf{t}}, g_{\lambda,1}, g_{\lambda,2}, g_{\lambda,\mathsf{t}}, e_\lambda, p_\lambda)\}_{\lambda\in\mathbb{N}} \ ,$$

where $\mathbb{G}_{\lambda,1}$ (resp. $\mathbb{G}_{\lambda,2}$, $\mathbb{G}_{\lambda,\mathsf{t}}$) is a cyclic group of order p_λ generated by $g_{\lambda,1}$ (resp. $g_{\lambda,2}$, $g_{\lambda,\mathsf{t}}$), and $e_\lambda\colon \mathbb{G}_{\lambda,1} \times \mathbb{G}_{\lambda,2} \to \mathbb{G}_{\lambda,\mathsf{t}}$ is the pairing operation satisfying $e_\lambda(g_{\lambda,1}^a, g_{\lambda,2}^b) = g_{\lambda,\mathsf{t}}^{ab}$ for all integers a, b. The group operations and the pairing map are required to be efficiently computable. Following the implicit notation in [EHK+13], we write $[\![a]\!]_i$ to denote $g_{\lambda,i}^a$ for $i \in \{1,2,\mathsf{t}\}$. This notation extends component-wise to matrices and vectors having entries in $\mathbb{Z}_p$. Equipped with these notations, group operations are written additively and the pairing operation multiplicatively, e.g., $[\![\mathbf{A}]\!]_1 - \mathbf{B}[\![\mathbf{C}]\!]_1\mathbf{D} = [\![\mathbf{A} - \mathbf{BCD}]\!]_1$ and $[\![\mathbf{A}]\!]_1[\![\mathbf{B}]\!]_2 = [\![\mathbf{AB}]\!]_\mathsf{t}$. We rely on the MDDH assumption defined in [EHK+13].

Definition 4 (MDDH Assumption)**.** Let $\alpha \in \{1,2,\mathsf{t}\}$, $k \ge 1$ an integer constant and $\ell = \ell(\lambda), q = q(\lambda)$ some polynomials. For a sequence of pairing groups $\mathbb{G}$, the $\mathsf{MDDH}_{k,\ell}^q$ assumption holds in $\mathbb{G}_\alpha$ if

$$\left\{\left(1^\lambda, [\![\mathbf{A}]\!]_\alpha, [\![\mathbf{AS}]\!]_\alpha\right)\right\}_{\lambda\in\mathbb{N}} \approx_c \left\{\left(1^\lambda, [\![\mathbf{A}]\!]_\alpha, [\![\mathbf{U}]\!]_\alpha\right)\right\}_{\lambda\in\mathbb{N}}$$

for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{p_\lambda}^{\ell\times k}$, $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_{p_\lambda}^{k\times q}$ and $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_{p_\lambda}^{\ell\times q}$.

In [EHK+13], it was shown that k-Lin implies $\mathsf{MDDH}_{k,k+1}^1$ which in turn implies $\mathsf{MDDH}_{k,\ell}^q$ for all $\ell > k$ and $q \ge 1$ with a tight security reduction. For convenience, we might refer to the $\mathsf{MDDH}_{k,k+1}^1$ assumption by MDDH_k.

3.4 Lattices

Let $n, m \geq 1$ and $p \geq 2$ be integers. Given $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$ and $\mathbf{b} \in \mathbb{Z}_p^n$ such that there exists $\mathbf{k}^* \in \mathbf{Z}^m$ satisfying $\mathbf{A}\mathbf{k}^* = \mathbf{b}$, we define the lattice coset

$$\Lambda_{\mathbf{b}}^{\perp}(\mathbf{A}) = \{\mathbf{k} \in \mathbb{Z}^m : \mathbf{A}\mathbf{k} = \mathbf{b}\} = \mathbf{k}^* + \Lambda_{\mathbf{0}}^{\perp}(\mathbf{A})$$

Let Λ' be any lattice coset and $\sigma \geq 0$. We write $\mathcal{D}_{\Lambda',\sigma}$ to denote the discrete Gaussian distribution [MP12] over Λ' with width σ. For $B > 0$, we let $\mathcal{D}_{\mathbb{Z},\sigma,\leq B}$ be the truncated $\mathcal{D}_{\mathbb{Z},\sigma}$ distribution which is sampled by first sampling $x \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}$, then returning x if $|x| \leq B$, and 0 otherwise. We will use the following lemmas.

Lemma 5 (Truncation)**.** *For all $\sigma \geq 1$ and $\varepsilon \geq 2$,*

$$\Pr[|x| > \sigma\sqrt{\varepsilon} : x \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}] \leq 2^{-\varepsilon} ,$$

so $\mathcal{D}_{\mathbb{Z},\sigma,\leq\sigma\sqrt{\varepsilon}}$ is $2^{-\varepsilon}$-close to $\mathcal{D}_{\mathbb{Z},\sigma}$.

Lemma 6 (Noise Flooding)**.** *For all $y \in \mathbb{Z}$ and $\sigma \geq 2^{\varepsilon+6}y$, it holds that $(y + \mathcal{D}_{\mathbb{Z},\sigma})$ is $2^{-\varepsilon}$-close to $\mathcal{D}_{\mathbb{Z},\sigma}$.*

We recall the LWE assumption introduced in [Reg05].

Definition 7 (LWE Assumption)**.** Let $n = n(\lambda), m = m(\lambda) \leq \mathrm{poly}(\lambda)$, $p = p(\lambda)$, $\sigma = \sigma(\lambda)$. The $\mathsf{LWE}_{n,m,p,\sigma}$ assumption states that

$$\{(1^\lambda, \mathbf{A}, \mathbf{s}\mathbf{A} + \mathbf{e})\}_{\lambda \in \mathbb{N}} \approx_c \{(1^\lambda, \mathbf{A}, \mathbf{u})\}_{\lambda \in \mathbb{N}} ,$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z},\sigma}^m$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^m$.

3.5 Exact Linear Garbling

We recall the definition of a linear secret sharing (LSS) scheme [BL90, Bei96]. Let $m, n \in \mathbb{N}$ and p be a prime number. We denote $\mathbf{e}_1 = (1, 0, \ldots, 0) \in \mathbb{Z}_p^m$ the first unit-vector in $\mathbb{Z}_p^n$. A LSS scheme over $\mathbb{Z}_p$ for a function $f\colon \{0,1\}^n \to \{0,1\}$ is specified by a share generating matrix $\mathbf{M} \in \mathbb{Z}_p^{m \times n}$ which satisfies the following condition for all inputs $\mathbf{x} \in \{0,1\}^n$:[7]

$$f(\mathbf{x}) = 0 \iff \mathbf{e}_1^\top \in \mathsf{span}\{\mathbf{M}[i] : \mathbf{x}[i] = 1\} \ . \tag{1}$$

Lewko and Waters [LW11] presented an LSS scheme for all monotone access structures. We will use this fact in the following way.

Lemma 8 (Exact Linear Garbling for NC^1)**.** *There exist four efficient algorithms* $(\mathsf{Gen}, \mathsf{Share}, \mathsf{Rec}, \mathsf{Sim})$ *with the following properties.*

- ***Syntax.*** *On input $f \in \mathsf{NC}^1$, $f\colon \{0,1\}^n \to \{0,1\}$, $\mathsf{Gen}(f)$ outputs $\mathbf{M} \in \mathbb{Z}_p^{m \times 2n}$ and*

$$\mathsf{Share}(f, \mathbf{s}; \mathbf{t}) = (\mathbf{s}, \mathbf{t}) \cdot (\mathbf{M} \otimes \mathbf{I}_k)$$
$$\mathsf{Sim}(f; \mathbf{t}) = (\mathbf{0}, \mathbf{t}) \cdot (\mathbf{M} \otimes \mathbf{I}_k) \ ,$$

  where $\mathbf{s} \in \mathbb{Z}_p^k$, $\mathbf{t} \in \mathbb{Z}_p^{k(m-1)}$ and m is linear in the size of f.

- ***Correctness.*** *For all $\mathbf{x} \in \{0,1\}^n$ with $f(\mathbf{x}) = 0$, $\mathsf{Rec}(f, \mathbf{x})$ outputs a vector $\mathbf{d}_{f,\mathbf{x}} \in \mathbb{Z}_p^n$ such that for all $\mathbf{r}, \mathbf{s} \in \mathbb{Z}_p^k$ and $\mathbf{t} \in \mathbb{Z}_p^{k(m-1)}$,*

$$\mathbf{p}_{f,\mathbf{x},\mathbf{r},\mathbf{s},\mathbf{t}} \cdot \mathbf{d}_{f,\mathbf{x}}^\top = \mathbf{s} \cdot \mathbf{r}^\top \ ,$$

  where we parse $(\mathbf{s}_1^0, \ldots, \mathbf{s}_n^0, \mathbf{s}_1^1, \ldots, \mathbf{s}_n^1) \leftarrow \mathsf{Share}(f, \mathbf{s})$[8] such that $\mathbf{s}_i^\beta \in \mathbb{Z}_p^k$ for $i \in [n]$ and $\beta \in \{0,1\}$, and $\mathbf{p}_{f,\mathbf{x},\mathbf{r},\mathbf{s},\mathbf{t}} = (\mathbf{s}_1^{\mathbf{x}[1]}, \ldots, \mathbf{s}_n^{\mathbf{x}[n]}) \cdot (\mathbf{I}_n \otimes \mathbf{r}^\top)$. If $f(\mathbf{x}) = 1$, then $\mathsf{Rec}(f, x)$ outputs $\perp$.

[7] For notational convenience, we consider $\mathbf{x} \in \{0,1\}^n$ with coordinates being indexed by elements in $[n]$. But this is w.l.o.g., and we may use arbitrary index sets $\mathfrak{s}$.

[8] For notational convenience, we may also parse the output of $\mathsf{Share}(f, \mathbf{s})$ as a set $\{\mathbf{s}_i^\beta\}_{i \in [n]}^{\beta \in \{0,1\}}$.

- ***Security.*** *For all* $f \in \mathsf{NC}^1$, $\mathbf{s} \in \mathbb{Z}_p^k$ *and* $\mathbf{x} \in \{0,1\}^n$, *if* $f(\mathbf{x}) = 1$ *then*

$$\begin{aligned}&\left\{(\mathbf{s}_1^{\mathbf{x}[1]}, \ldots, \mathbf{s}_n^{\mathbf{x}[n]}) : (\mathbf{s}_1^0, \ldots, \mathbf{s}_n^0, \mathbf{s}_1^1, \ldots, \mathbf{s}_n^1) \leftarrow \mathsf{Share}(f, \mathbf{s}; \mathbf{t})\right\} \\ &\approx_s \left\{(\mathbf{s}_1^{\mathbf{x}[1]}, \ldots, \mathbf{s}_n^{\mathbf{x}[n]}) : (\mathbf{s}_1^0, \ldots, \mathbf{s}_n^0, \mathbf{s}_1^1, \ldots, \mathbf{s}_n^1) \leftarrow \mathsf{Sim}(f, \mathbf{t})\right\} \ ,\end{aligned}$$

  where the randomness is taken over $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{k(m-1)}$.

Proof. Given a policy $f\colon \{0,1\}^n \to \{0,1\}$ computable by an NC^1 circuit, we construct the corresponding monotone policy $f_{\mathsf{mon}}\colon \{0,1\}^{2n} \to \{0,1\}$ as follows: first, we view f as a Boolean formula consisting of (fan-in 1) $\neg$ gates and (fan-in 2) $\wedge$ and $\vee$ gates. Then, using De Morgan laws, we push the $\neg$ gates to the leaves such that all internal nodes consist only of $\wedge$ and $\vee$ gates, while leaves are labeled by either attributes or their negations. In other words, we can transform each $f \in \mathsf{NC}^1$ over $\{0,1\}^n$ into a function $f_{\mathsf{mon}}\colon \{0,1\}^{2n} \to \{0,1\}$ described by a monotone Boolean formula over the same attributes and their negations, i.e., we have $f(\mathbf{x}) = f_{\mathsf{mon}}(\mathbf{x}_{\mathsf{mon}})$ for all $\mathbf{x} \in \{0,1\}^n$ and $\mathbf{x}_{\mathsf{mon}} = (\mathbf{x}, 1^k - \mathbf{x}) \in \{0,1\}^{2n}$.

Correctness. Using an LSS scheme for monotone formulas, the implementation of Gen and Rec is immediate.

- Given $f \in \mathsf{NC}^1$, the algorithm $\mathsf{Gen}(f)$ constructs f_{mon} and outputs a share generating matrix $\mathbf{M} \in \mathbb{Z}_p^{m \times 2n}$ for f_{mon}.
- Given $f \in \mathsf{NC}^1$ and $x \in \{0,1\}^n$ such that $f(\mathbf{x}) = 0$, $\mathsf{Rec}(f, x)$ outputs a coefficient vector $\mathbf{d}_{f,\mathbf{x}} \in \mathbb{Z}_p^n$ such that $\sum_{i \in [n]} \mathbf{d}_{f,\mathbf{x}}[i] \cdot \mathbf{M}[i + n \cdot \mathbf{x}[i]] = \mathbf{e}_1^\top$. The existence of such a vector is guaranteed by Equality (1).

Then we have

$$\begin{aligned}\mathbf{p}_{f,\mathbf{x},\mathbf{r},\mathbf{s},\mathbf{t}} \cdot \mathbf{d}_{f,\mathbf{x}}^\top &= (\mathbf{s}_1^{\mathbf{x}[1]}, \ldots, \mathbf{s}_n^{\mathbf{x}[n]}) \cdot (\mathbf{I}_n \otimes \mathbf{r}^\top) \cdot \mathbf{d}_{f,\mathbf{x}} \\ &= (\mathbf{s}, \mathbf{t})(\mathbf{I}_m \otimes \mathbf{r}^\top) \cdot \sum_{i \in [n]} \mathbf{d}_{f,\mathbf{x}} \cdot \mathbf{M}[i + n \cdot \mathbf{x}[i]] \\ &= (\mathbf{s}, \mathbf{t})(\mathbf{I}_m \otimes \mathbf{r}^\top) \cdot \mathbf{e}_1^\top = \mathbf{s} \cdot \mathbf{r}^\top \ .\end{aligned}$$

Security. Consider $f \in \mathsf{NC}^1$ and $\mathbf{x} \in \{0,1\}^n$ such that $f(\mathbf{x}) = 1$. Let $M_{\mathbf{x}} = \mathsf{span}\{\mathbf{M}[i + n \cdot \mathbf{x}[i]]\}_{i \in [n]}$ be the vector space spanned by the columns of $\mathbf{M}$ corresponding to the shares $\{\mathbf{s}_i^{\mathbf{x}[i]}\}_{i \in [n]}$. Furthermore, let $M_{\mathbf{x}}^\perp$ denote the orthogonal complement of $M_{\mathbf{x}}$ and $A_{\mathbf{x}}^\perp$ the affine subspace of $M_{\mathbf{x}}^\perp$ containing all vectors whose first coordinate is 1. Since $f(\mathbf{x}) = 1$, Equality (1) gives that $\mathbf{e}_1 \notin M_{\mathbf{x}}$ which implies that $A_{\mathbf{x}}^\perp$ is nonempty, i.e., there exists $(1, \mathbf{u}) \in A_{\mathbf{x}}^\perp$. For every $\mathbf{s} \in \mathbb{Z}_p^k$ and random $\mathbf{t} \in \mathbb{Z}_p^{k(m-1)}$, we have

$$\begin{aligned}&\left\{(\mathbf{s}, \mathbf{t}) \cdot (\mathbf{M}[i + n \cdot \mathbf{x}[i]] \otimes \mathbf{I}_k)\right\}_{i \in [n]} \\ &\equiv \left\{(\mathbf{s}, \mathbf{t} + \mathbf{u} \otimes \mathbf{s}) \cdot (\mathbf{M}[i + n \cdot \mathbf{x}[i]] \otimes \mathbf{I}_k)\right\}_{i \in [n]} \\ &= \left\{\big((\mathbf{0}, \mathbf{t}) + (1, \mathbf{u}) \otimes \mathbf{s}\big) \cdot (\mathbf{M}[i + n \cdot \mathbf{x}[i]] \otimes \mathbf{I}_k)\right\}_{i \in [n]} \\ &= \left\{(\mathbf{0}, \mathbf{t}) \cdot (\mathbf{M}[i + n \cdot \mathbf{x}[i]] \otimes \mathbf{I}_k) + \big((1, \mathbf{u}) \cdot (\mathbf{M}[i + n \cdot \mathbf{x}[i]])\big) \otimes \mathbf{s}\right\}_{i \in [n]} \\ &= \left\{(\mathbf{0}, \mathbf{t}) \cdot (\mathbf{M}[i + n \cdot \mathbf{x}[i]] \otimes \mathbf{I}_k)\right\}_{i \in [n]} \ ,\end{aligned}$$

where the first distribution corresponds to the output of Share whereas the last distribution corresponds to the output of Sim. The first equivalence follows from the fact that $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{k(m-1)}$ perfectly masks $\mathbf{u} \otimes \mathbf{s}$, and the last equality since $(1, \mathbf{u}) \in M_{\mathbf{x}}^\perp$ and $\mathbf{M}[i + n \cdot \mathbf{x}[i]] \in M_{\mathbf{x}}$. □

3.6 Noisy Linear Garbling

We recall the definition of noisy linear garblings from [HLL24] in the variant for binary inputs.

Definition 9 (Noisy Linear Garbling)**.** Let $\{p_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be sequences of prime numbers and function families, respectively, where each $f \in \mathcal{F}_\lambda\colon \{0,1\}^n \to \{0,1\}$. A noisy linear garbling for $\mathcal{F}$ consists of the following three efficient algorithms.

$\mathsf{Setup}(1^\lambda) \to \mathsf{PP}$**:** On input the security parameter 1^λ, this algorithm outputs some public parameters PP.

$\mathsf{GenF}(\mathsf{PP}, f) \to (1^{n_f}, \mathbf{w}_{\mathsf{out}}, \{\mathbf{W}_i^0, \mathbf{W}_i^\times\}_{i\in[n]}, \mathbf{T}, \mathbf{R})$**:** On input public parameters PP and a function $f \in \mathcal{F}_\lambda$, this algorithm outputs a deterministic dimension 1^{n_f} of garbling randomness, a secret vector $\mathbf{w}_{\mathsf{out}}$, label functions $\{\mathbf{W}_{i,j}^0, \mathbf{W}_{i,j}^\times\}_{i\in[n]}$, a garbled table $\mathbf{T}$, and some reusable information $\mathbf{R}$. Here, $\mathbf{w}_{\mathsf{out}}$, $\{\mathbf{W}_{i,j}^\times\}_{i\in[n]}$, and $\mathbf{T}$ are n_f-row matrices over $\mathbb{Z}$ and their shapes, including n_f, are fully determined by PP and f.

$\mathsf{Eval}(\mathsf{PP}, f, \mathbf{R}, \mathbf{x}, \{\mathbf{w}_i\}_{i\in[n]}, \mathbf{t}) \to w_{\mathsf{out}}$**:** On input the public parameters, the function f, reusable information $\mathbf{R}$, input $\mathbf{x} \in \{0,1\}^n$, a set of randomized labels $\{\mathbf{w}_i\}_{i\in[n]}$, and the randomized garbled table, this algorithm outputs an approximation of the randomized secret w_{out} if $f(\mathbf{x}) = 0$. Here, w_{out}, $\{\mathbf{w}_i\}_{i\in[n]}$ and $\mathbf{t}$ are over $\mathbb{Z}_{p_\lambda}$.

Correctness. Let $B_{\mathsf{in}} = B_{\mathsf{in}}(\lambda), B_{\mathsf{out}} = B_{\mathsf{out}}(\lambda) \in \mathbb{N}$. A noisy linear garbling is $(B_{\mathsf{in}}, B_{\mathsf{out}})$-correct if for all $\lambda \in \mathbb{N}$, functions $f \in \mathcal{F}_\lambda$, inputs $\mathbf{x} \in \{0,1\}^n$ such that $f(\mathbf{x}) = 0$, and error vectors $e_{\mathsf{out}}, \mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{e}_{\mathsf{t}} \in [-B_{\mathsf{in}}; B_{\mathsf{in}}]^\star$ (of suitable dimensions), it holds that

$$\Pr\left[e \in [-B_{\mathsf{out}}; B_{\mathsf{out}}] \;\middle|\; \begin{array}{l} \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{n_f};\ \mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda) \\ \left(1^{n_f}, \mathbf{w}_{\mathsf{out}}, \{\mathbf{W}_{i,j}^0, \mathbf{W}_{i,j}^\times\}_{i\in[n]}, \mathbf{T}, \mathbf{R}\right) \leftarrow \mathsf{GenF}(\mathsf{PP}, f) \\ w_{\mathsf{out}} \leftarrow \mathsf{Eval}\left(\mathsf{PP}, f, \mathbf{R}, \mathbf{x}, \left\{\mathbf{s}(\mathbf{W}_i^0 + \mathbf{x}[i]\mathbf{W}_i^\times) + \mathbf{e}_i\right\}_{i\in[n]}, \mathbf{sT} + \mathbf{e}_{\mathsf{t}}\right) \\ e \coloneqq w_{\mathsf{out}} - (\mathbf{sw}_{\mathsf{out}}^\top + e_{\mathsf{out}}) \end{array} \right] = 1 \ .$$

Shortness. Let $B_{\mathsf{short}} = B_{\mathsf{short}}(\lambda) \in \mathbb{N}$. A noisy linear garbling scheme is B_{short}-short if for all $\lambda \in \mathbb{N}$, functions $f \in \mathcal{F}_\lambda$ and inputs $\mathbf{x} \in \{0,1\}^n$ such that $f(\mathbf{x}) = 1$, it holds that

$$\Pr\left[\begin{array}{l} \|\mathbf{w}_{\mathsf{out}}\|, \|\mathbf{T}\| \le B_{\mathsf{short}}, \\ \|\mathbf{W}_i^0 + \mathbf{x}[i]\mathbf{W}_i^\times\| \le B_{\mathsf{short}} \text{ for all } i \in [n] \end{array} \;\middle|\; \begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda) \\ \left(1^{n_f}, \mathbf{w}_{\mathsf{out}}, \{\mathbf{W}_{i,j}^0, \mathbf{W}_{i,j}^\times\}_{i\in[n]}, \mathbf{T}, \mathbf{R}\right) \leftarrow \mathsf{GenF}(\mathsf{PP}, f) \end{array} \right] = 1 \ .$$

Security. Security requires that the randomized secret, labels and garbled table are pseudorandom if $f(\mathbf{x}) = 1$.

Definition 10 (Security)**.** A noisy linear garbling scheme NLG is $\mathsf{GenNoise}$-secure, where $\mathsf{GenNoise}$ is an efficient algorithm with suitable input/output formats, if $\mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}real}}(1^\lambda) \approx_c \mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}sim}}(1^\lambda)$ for all PPT adversaries $\mathcal{A}$, where $\mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}real}}$ and $\mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}sim}}$ proceeds as follows.

- **Setup.** Launch $\mathcal{A}(1^\lambda)$, sample r_{Setup}, run $\mathsf{PP}, \leftarrow \mathsf{Setup}(1^\lambda; r_{\mathsf{Setup}})$ and send r_{Setup} to $\mathcal{A}$.
- **Challenge.** Upon $\mathcal{A}$ submitting a function $f \in \mathcal{F}_\lambda$ and an input $\mathbf{x} \in \{0,1\}^n$, sample r_{GenF} and run

  $$(1^{n_f}, \mathbf{w}_{\mathsf{out}}, \{\mathbf{W}_i^0, \mathbf{W}_i^\times\}_{i\in[n]}, \mathbf{T}, \mathbf{R}) \leftarrow \mathsf{GenF}(\mathsf{PP}, f; r_{\mathsf{GenF}}) \ .$$

  Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{n_f}$, run

  $$(e_{\mathsf{out}}, \mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{e}_{\mathsf{t}}) \leftarrow \mathsf{GenNoise}(\mathsf{PP}, f, \mathbf{w}_{\mathsf{out}}, \{\mathbf{W}_i^0, \mathbf{W}_i^\times\}_{i\in[n]}, \mathbf{T}, \mathbf{R}, \mathbf{x}) \ ,$$

  and compute

  $$\begin{array}{lllll} \text{in } \mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}real}}\text{:} & w_{\mathsf{out}} \leftarrow \mathbf{sW}_{\mathsf{out}}^\top \ , & \mathbf{t} \leftarrow \mathbf{sT} + \mathbf{e}_{\mathsf{t}} \ , & \left\{\mathbf{w}_i \leftarrow \mathbf{s}(\mathbf{W}_i^0 + \mathbf{x}[i]\mathbf{W}_i^\times) + \mathbf{e}_i\right\}_{i\in[n]} \ , \\ \text{in } \mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}sim}}\text{:} & w_{\mathsf{out}} \xleftarrow{\$} \mathbb{Z}_p \ , & \mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^\star \ , & \left\{\mathbf{w}_i \xleftarrow{\$} \mathbb{Z}_p^\star\right\}_{i\in[n]} \ , \end{array}$$

  where $\mathbf{t}$ and $\{\mathbf{w}_i\}_{i\in[n]}$ in $\mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}sim}}$ are of the same dimensions as their counterparts in $\mathbf{Exp}_{\mathsf{NLG},\mathcal{A}}^{\mathsf{nlg\text{-}real}}$. Return $(r_{\mathsf{GenF}}, w_{\mathsf{out}}, \{\mathbf{w}_i\}_{i\in[n]}, \mathbf{t})$ to $\mathcal{A}$.
- **Guess.** The adversary outputs a guess $\alpha \in \{\mathsf{real}, \mathsf{sim}\}$. The outcome of the experiment is α if $f(\mathbf{x}) = 1$. Otherwise, the outcome is set to 0.

3.7 Multi-Client Attribute-Based Encryption

We recall the definition of MC-ABE. Recent papers [AYY22, FFMV23, ARYY23, PS24] consider a varying syntax for MI-ABE and MC-ABE. Our definition follows [ARYY23, PS24].

Public-Key Setting. We first discuss the public-key case.

Definition 11 (Public-Key Syntax)**.** Let $n = n(\lambda)$ be a polynomial and $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$, $\mathcal{L} = \{\mathcal{L}_\lambda\}_{\lambda\in\mathbb{N}}$, $P = \{P_\lambda\}_{\lambda\in\mathbb{N}}$, $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$, and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be sequences of message spaces, label spaces, predicates, ciphertext attribute spaces and key attribute spaces, respectively, where $P_\lambda\colon \mathcal{X}_\lambda^{[0;n-1]} \times \mathcal{Y}_\lambda \to S_\lambda$ and S_λ is a strict superset of $\{0\}$.

An MC-ABE scheme for $\mathcal{M}$, $\mathcal{L}$, and P consists of five efficient algorithms:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{MPK}, \mathsf{MSK}, \{\mathsf{SK}_i\}_{i\in[n-1]})$**:** On input the security parameter 1^λ, this algorithm outputs a pair of master public key MPK and master secret key MSK as well as a set of secret keys $\{\mathsf{SK}_i\}_{i\in[n-1]}$.

$\mathsf{Enc}(\mathsf{MPK}, L, x_0, \mu) \to \mathsf{CT}_L$**:** On input the master public key MPK, a label $L \in \mathcal{L}_\lambda$, an attribute $x_0 \in \mathcal{X}_\lambda$ and a message $\mu \in \mathcal{M}_\lambda$, this algorithm outputs a ciphertext CT_L. We assume that CT_L implicitly includes x_0.

$\mathsf{AKeyGen}(\mathsf{SK}_i, L, x_i) \to \mathsf{DK}_{L,i}$**:** On input a secret key SK_i for some $i \in [n-1]$, a label $L \in \mathcal{L}_\lambda$ and an attribute $x_i \in \mathcal{X}_\lambda$, this algorithm outputs a decryption key $\mathsf{DK}_{L,i}$. We assume that $\mathsf{DK}_{L,i}$ implicitly includes x_i.

$\mathsf{PKeyGen}(\mathsf{MSK}, y) \to \mathsf{DK}_y$**:** On input the master secret key MSK and an attribute $y \in \mathcal{Y}_\lambda$, this algorithm outputs a decryption key DK_y. We assume that DK_y implicitly includes a description of y.

$\mathsf{Dec}(\mathsf{DK}_y, \{\mathsf{DK}_{L,i}\}_{i\in[n-1]}, \mathsf{CT}_L) \to \mu' \vee \bot$**:** On input a decryption key DK_y for a policy $y \in \mathcal{Y}_\lambda$, a set of attribute decryption keys $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$ generated with respect to some label $L \in \mathcal{L}_\lambda$ and a ciphertext CT_L created with respect to the same label L, this algorithm outputs an element $\mu' \in \mathcal{M}_\lambda$ or $\bot$.

An MC-ABE scheme is *correct* if for every $\lambda, n \in \mathbb{N}$, label $L \in \mathcal{L}_\lambda$, message $\mu \in \mathcal{M}_\lambda$ and attributes $x_0, \ldots, x_{n-1} \in \mathcal{X}_\lambda$, $y \in \mathcal{Y}_\lambda$ such that $P_\lambda(x_0, \ldots, x_{n-1}, y) = 1$, it holds that

$$\Pr\left[\mu' = \mu \;\middle|\; \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}, \{\mathsf{SK}_i\}_{i\in[n-1]}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{CT}_L \leftarrow \mathsf{Enc}(\mathsf{MPK}, L, x_0, \mu) \\ \forall i \in [n-1]\colon \mathsf{DK}_{L,i} \leftarrow \mathsf{AKeyGen}(\mathsf{SK}_i, L, x_i) \\ \mathsf{DK}_y \leftarrow \mathsf{PKeyGen}(\mathsf{MSK}, y) \\ \mu' := \mathsf{Dec}(\mathsf{DK}_y, \{\mathsf{DK}_{L,i}\}_{i\in[n-1]}, \mathsf{CT}_L) \end{array}\right] = 1$$

We define IND-CPA security for MC-ABE in the public-key setting.

Definition 12 (Public-Key Security for MC-ABE)**.** Let $\mathsf{xxx} \in \{\mathsf{adap}, \mathsf{sel}, \mathsf{stat}\}$. An MC-ABE scheme ABE is xxx-secure against Q collusions if $\mathbf{Exp}^{\text{mc-abe-0}}_{\mathsf{ABE},\mathcal{A}}(1^\lambda) \approx_c \mathbf{Exp}^{\text{mc-abe-1}}_{\mathsf{ABE},\mathcal{A}}(1^\lambda)$ for all PPT adversaries $\mathcal{A}$, where $\mathbf{Exp}^{\text{mc-abe-}b}_{\mathsf{ABE},\mathcal{A}}$ for $b \in \{0,1\}$ proceeds as follows.

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ to receive a set $\mathcal{C} \subseteq [n-1]$ of corrupted slots, run $(\mathsf{MPK}, \mathsf{MSK}, \{\mathsf{SK}_i\}_{i\in[n-1]}) \leftarrow \mathsf{Setup}(1^\lambda)$ and send MPK to $\mathcal{A}$. Initialize an empty set $\mathcal{Q}_{\mathsf{akey}} = \varnothing$ and an empty multiset $\mathcal{Q}_{\mathsf{pkey}} = \varnothing$.
- **Query I.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ has two options.
  - $\mathsf{QAKeyGen}(i, L, x_i)$: upon $\mathcal{A}$ submitting a slot index $i \in [n-1]$, a label $L \in \mathcal{L}_\lambda$ and an attribute $x_i \in \mathcal{X}_\lambda$, run $\mathsf{DK}_i \leftarrow \mathsf{AKeyGen}(\mathsf{SK}_i, L, x_i)$ and return DK_i to $\mathcal{A}$. Add (i, L, x_i) to $\mathcal{Q}_{\mathsf{akey}}$.
  - $\mathsf{QPKeyGen}(y)$: upon $\mathcal{A}$ submitting an attribute $y \in \mathcal{Y}_\lambda$, run $\mathsf{DK}_y \leftarrow \mathsf{PKeyGen}(\mathsf{MSK}, y)$ and return DK_y to $\mathcal{A}$. Add y to $\mathcal{Q}_{\mathsf{pkey}}$.

- **Challenge.** Upon $\mathcal{A}$ submitting a label $L \in \mathcal{L}_\lambda$, a challenge attribute $x_0^* \in \mathcal{X}_\lambda$ and messages $\mu^0, \mu^1 \in \mathcal{M}_\lambda$, run $\mathsf{CT}_L \leftarrow \mathsf{Enc}(\mathsf{MPK}, L, x_0^*, \mu^b)$ and return CT_L to $\mathcal{A}$.
- **Query II.** Same as Query I.
- **Guess.** The adversary outputs a bit $b' \in \{0,1\}$. The outcome of the experiment is b' if $\mathcal{A}$ satisfies the following admissibility condition. Otherwise, the outcome is set to 0. The adversary $\mathcal{A}$ is admissible if the following conditions are satisfied.
  - $|\mathcal{Q}_{\mathsf{pkey}}| \leq Q$,
  - For all $y \in \mathcal{Q}_{\mathsf{pkey}}$ and $x_1, \ldots, x_{n-1} \in \mathcal{X}_\lambda$ such that $(i, L, x_i) \in \mathcal{Q}_{\mathsf{akey}}$ for all $i \in [n-1] \setminus \mathcal{C}$, it holds that $P_\lambda(x_0, \ldots, x_{n-1}, y) \neq 0$.
  - if $\mathsf{xxx} = \mathsf{sel}$, then the adversary submits all queries to $\mathsf{QAKeyGen}$ and the challenge query in one shot. Subsequently, the adversary can query $\mathsf{QPKeyGen}$.
  - If $\mathsf{xxx} = \mathsf{stat}$, then the adversary submits all queries in one shot.

If we do not explicitly specify Q, then the adversary can submit any number of queries to $\mathsf{QPKeyGen}$ and we say that ABE is secure against *unbounded* collusions.

Secret-Key Setting. Some of our constructions are presented in the secret-key setting. In this case, the setup algorithm outputs a secret key SK_0 that is used for encryption instead of a master public key MPK. The security definition is the same as in the public-key setting except for the following changes.

- The Setup phase does not provide the adversary with MPK and initializes an additional set $\mathcal{Q}_{\mathsf{enc}}$.
- The phases Query I, Challenge and Query II are replaced by a single phase Query during which the adversary has access to the following three oracles.
  - $\mathsf{QEnc}(L, x_0, \mu^0, \mu^1)$: upon $\mathcal{A}$ submitting a label $L \in \mathcal{L}_\lambda$, an attribute $x_0 \in \mathcal{X}_\lambda$ and messages $\mu^0, \mu^1 \in \mathcal{M}_\lambda$, run $\mathsf{CT}_L \leftarrow \mathsf{Enc}(\mathsf{SK}_0, L, x_0, \mu^b)$ and return CT_L to $\mathcal{A}$. Add (L, x_0, μ^0, μ^1) to $\mathcal{Q}_{\mathsf{enc}}$.
  - $\mathsf{QAKeyGen}(i, L, x_i)$: same as in the public-key case.
  - $\mathsf{QPKeyGen}(y)$: same as in the public-key case.
- The second item of the admissibility condition changes to: for all $y \in \mathcal{Q}_{\mathsf{pkey}}$, $L \in \mathcal{L}_\lambda$, $(L, x_0, \mu^0, \mu^1) \in \mathcal{Q}_{\mathsf{enc}}$ and $x_1, \ldots, x_{n-1} \in \mathcal{X}_\lambda$ such that $(i, L, x_i) \in \mathcal{Q}_{\mathsf{akey}}$ for all $i \in [n-1] \setminus \mathcal{C}$, it holds that $P_\lambda(x_0, \ldots, x_{n-1}, y) \neq 0$ or $\mu^0 = \mu^1$.

Note that public-key security implies secret-key security. For this reason, we preferably present our constructions in the public-key setting and only resort to secret-key security if necessary.

Transformation to MC-PE. In contrast to MC-ABE, MC-PE security is stronger in the secret-key setting (see e.g. [AYY22, FFMV23, PS24] for details). The work [PS24] presents a compiler that turns any constant-arity, public-key MC-ABE into a constant-arity, secret-key MC-PE. Inspecting their compiler, we observe that the security proof does not rely on the fact that the employed MC-ABE scheme satisfies public-key security. We can therefore use their compiler to also convert secret-key MC-ABE into secret-key MC-PE.

3.8 Attribute-Based Inner-Product Functional Encryption

We recall the syntax of attribute-based IPFE (AB-IPFE) and define security in the classical [ABDP15, ALS16, ACGU20, LLW21] and evasive [HLL24] setting. In this work, we consider evasive IPFE only in the variant *without* structured noises in which case we can adapt the syntax to match that of classical IPFE schemes.

Definition 13 (Syntax of AB-IPFE)**.** Let $\{p_\lambda\}_{\lambda\in\mathbb{N}}$, $\{P_\lambda\}_{\lambda\in\mathbb{N}}$, $\{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be sequences of prime numbers, predicates, ciphertext attribute spaces and key attribute spaces, respectively. An AB-IPFE scheme consists of four efficient algorithms.

$\mathsf{Setup}(1^\lambda, 1^N) \to (\mathsf{MPK}, \mathsf{MSK})$**:** On input the security parameter 1^λ and the dimension of the vectors 1^N, this algorithm outputs a pair of a master public and a master secret key $(\mathsf{MPK}, \mathsf{MSK})$.

$\mathsf{Enc}(\mathsf{MPK}, x, \mathbf{u}) \to \mathsf{CT}$**:** On input a master public key MPK, a ciphertext attribute $x \in \mathcal{X}_\lambda$ and a vector $\mathbf{u} \in \mathbb{Z}_{p_\lambda}^N$, this algorithm outputs a ciphertext CT for $\mathbf{u}$ with respect to x. We assume that CT implicitly contains x.

$\mathsf{KeyGen}(\mathsf{MSK}, y, \mathbf{v}) \to \mathsf{DK}$**:** On input a master secret key MSK, a key attribute $y \in \mathcal{Y}_\lambda$ and a vector $\mathbf{v} \in \mathbb{Z}_{p_\lambda}^N$, this algorithm outputs a decryption key DK for $\mathbf{v}$ with respect to y. We assume that DK implicitly contains y.

$\mathsf{Dec}(\mathsf{DK}, \mathsf{CT}) \to d \vee \perp$**:** On input a decryption key DK and a ciphertext CT, this algorithm outputs an element $d \in \mathbb{Z}_{p_\lambda}$ or the special symbol $\perp$ indicating failure of decryption.

We say that iFE is a *secret-key* AB-IPFE scheme if Setup outputs only a master secret key MSK which is passed to both Enc and KeyGen. Furthermore, we say that iFE is *based on a pairing group* $\mathbb{G} = \{\mathbb{G}_\lambda = (\mathbb{G}_{\lambda,1}, \mathbb{G}_{\lambda,2}, \mathbb{G}_{\lambda,\mathsf{t}}, g_{\lambda,1}, g_{\lambda,2}, g_{\lambda,\mathsf{t}}, e_\lambda, p_\lambda)\}_{\lambda\in\mathbb{N}}$ if Enc takes vectors $[\![\mathbf{u}]\!]_1$ encoded in $\mathbb{G}_{\lambda,1}$, KeyGen takes vectors $[\![\mathbf{v}]\!]_2$ encoded in $\mathbb{G}_{\lambda,2}$ and Dec outputs elements $[\![d]\!]_\mathsf{t} \in \mathbb{G}_{\lambda,\mathsf{t}}$.

Correctness. Let $B = B(\lambda, p_\lambda, N)$ be a function with values in $\mathbb{N}$. An AB-IPFE scheme is B-correct if for all $\lambda, N \in \mathbb{N}$, all attributes $x \in \mathcal{X}_\lambda$ and $y \in \mathcal{Y}_\lambda$ such that $P_\lambda(x, y) = 0$, and all vectors $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_{p_\lambda}^N$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{DK}, \mathsf{CT}) = \mathbf{u}\cdot\mathbf{v}^\top + e \text{ for some } e \in [-B; B] \;\middle|\; \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, x, \mathbf{u}) \\ \mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, y, \mathbf{v}) \end{array}\right] = 1 \ .$$

If we do not explicitly specify B, then it defaults to the constant zero function, i.e., perfect correctness.

Batch Notations. It is convenient to consider Enc, KeyGen and Dec in batches. Consider $x \in \mathcal{X}_\lambda$, $\mathbf{y} \in \mathcal{Y}_\lambda$ and matrices

$$\mathbf{U} = \begin{bmatrix} — & \mathbf{u}_1 & — \\ & \vdots & \\ — & \mathbf{u}_I & — \end{bmatrix} \qquad \mathbf{V} = \begin{bmatrix} — & \mathbf{v}_1 & — \\ & \vdots & \\ — & \mathbf{v}_J & — \end{bmatrix} .$$

We write

$$\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MPK}, x, \mathbf{U}) \qquad \text{and} \qquad \mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, y, \mathbf{V})$$

for $\mathsf{CT} = \{\mathsf{CT}_i\}_{i\in[I]}$ and $\mathsf{DK} = \{\mathsf{DK}_j\}_{j\in[J]}$, where

$$\begin{aligned} &\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MPK}, x, \mathbf{u}_i) \qquad \text{for all } i \in [I] \ , \\ &\mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, y, \mathbf{v}_j) \qquad \text{for all } j \in [J] \ , \end{aligned}$$

and $\mathsf{Dec}(\mathsf{DK}, \mathsf{CT})$ refers to the matrix

$$\begin{bmatrix} \mathsf{Dec}(\mathsf{DK}_1, \mathsf{CT}_1) & \cdots & \mathsf{Dec}(\mathsf{DK}_J, \mathsf{CT}_1) \\ \vdots & \ddots & \vdots \\ \mathsf{Dec}(\mathsf{DK}_1, \mathsf{CT}_I) & \cdots & \mathsf{Dec}(\mathsf{DK}_J, \mathsf{CT}_I) \end{bmatrix}$$

which is (approximately) $\mathbf{U}\cdot\mathbf{V}^\top$ if $P_\lambda(x, y) = 0$.

Concrete Predicates. We consider AB-IPFE for three concrete predicates.

- **Identity-Based IPFE (IB-IPFE).** Let $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda\in\mathbb{N}}$ be a sequence of identity spaces. For each $\lambda \in \mathbb{N}$, we let $\mathcal{X}_\lambda = \mathcal{Y}_\lambda = \mathcal{I}_\lambda$ and $P_\lambda(x,y) = 0$ if $x = y$, and 1 otherwise.
- **Arithmetic Branching Programs.** Let $\{p_\lambda\}_{\lambda\in\mathbb{N}}$ be a sequence of prime numbers. For each $\lambda \in \mathbb{N}$, we let $\mathcal{Y} = \mathcal{F}_n^{\mathsf{abp}}$, $\mathcal{X} = \mathbb{Z}_{p_\lambda}^n$ and $P_\lambda(x,y) = y(x)$.
- **Bounded-Depth Circuits.** For each $\lambda \in \mathbb{N}$, we let $\mathcal{Y}_\lambda = \mathcal{F}_{n,d}^{\mathsf{circ}}$, $\mathcal{X}_\lambda\{0,1\}^n$ and $P_\lambda(x,y) = y(x)$.

Classical Security. We define IND-CPA security.

Definition 14 (Classical Security of AB-IPFE)**.** Let $\mathsf{xxx} \in \{\mathsf{adp},\mathsf{stat}\}$. An AB-IPFE scheme iFE is secure against Q collusions if $\mathbf{Exp}^{\mathsf{ab\text{-}ipfe\text{-}0}}_{\mathsf{iFE},\mathcal{A}}(1^\lambda) \approx_c \mathbf{Exp}^{\mathsf{ab\text{-}ipfe\text{-}1}}_{\mathsf{iFE},\mathcal{A}}(1^\lambda)$ for all PPT adversaries $\mathcal{A}$, where $\mathbf{Exp}^{\mathsf{ab\text{-}ipfe\text{-}}b}_{\mathsf{iFE},\mathcal{A}}$ for $b \in \{0,1\}$ proceeds as follows.

- **Setup.** Launch $\mathcal{A}(1^\lambda)$, run $(\mathsf{MPK},\mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and send MPK to $\mathcal{A}$. Initialize an empty multi-set $\mathcal{Q}_{\mathsf{key}} = \varnothing$.
- **Query I.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ submits a query $\mathsf{QKeyGen}(y,\mathbf{v})$, where $y \in \mathcal{Y}_\lambda$ and $\mathbf{v} \in \mathbb{Z}_{p_\lambda}^N$. Upon this query, run $\mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK},y,\mathbf{v})$ and return DK to $\mathcal{A}$. Add $(y,\mathbf{v})$ to $\mathcal{Q}_{\mathsf{key}}$.
- **Challenge.** Upon $\mathcal{A}$ submitting a challenge attribute x^* and two vectors $\mathbf{u}^0,\mathbf{u}^1 \in \mathbb{Z}_{p_\lambda}^N$, compute $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MPK},x^*,\mathbf{u}^b)$ and return CT to $\mathcal{A}$.
- **Query II.** Same as Query I.
- **Guess.** The adversary outputs a bit $b' \in \{0,1\}$. The outcome of the experiment is b' if $\mathcal{A}$ satisfies the following admissibility condition. Otherwise, the outcome is set to 0. The adversary $\mathcal{A}$ is admissible if
  - $|\mathcal{Q}_{\mathsf{key}}| \le Q$,
  - for all $(y,\mathbf{v}) \in \mathcal{Q}_{\mathsf{key}}$, if $P_\lambda(x^*,y) = 0$ then $\mathbf{u}^0\mathbf{v}^\top = \mathbf{u}^1\mathbf{v}^\top$, and
  - if $\mathsf{xxx} = \mathsf{stat}$, then $\mathcal{A}$ submits all queries to $\mathsf{QKeyGen}$ and the challenge query in one shot.

If we do not explicitly specify Q, then the adversary can submit any number of queries to $\mathsf{QKeyGen}$ and we say that iFE is secure against *unbounded* collusions.

Assuming the hardness of LWE, there exists an AB-IPFE scheme for bounded-depth circuits that is secure against $\mathrm{poly}(\lambda)$ collusions [LLW21].

Function-Hiding Security. In the secret-key, pairing-based setting, we can define the stronger notion of function-hiding security which differs from classical security (Definition 14) in the following way.

- The Setup phase does not provide the adversary with MPK and initializes a second set $\mathcal{Q}_{\mathsf{enc}} = \varnothing$.
- The phases Query I, Challenge and Query II are replaced with a single phase Query during which the adversary has access to the following two oracles.
  - $\mathsf{QEnc}(x,[\![\mathbf{u}^0]\!]_1,[\![\mathbf{u}^1]\!]_1)$: upon $\mathcal{A}$ submitting an attribute $x \in \mathcal{X}_\lambda$ and two vectors $[\![\mathbf{u}^0]\!]_1,[\![\mathbf{u}^1]\!]_1 \in \mathbb{G}_{\lambda,1}^N$, run $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK},x,[\![\mathbf{u}^b]\!]_1)$ and return CT to $\mathcal{A}$. Add $(x,[\![\mathbf{u}^0]\!]_1,[\![\mathbf{u}^1]\!]_1)$ to $\mathcal{Q}_{\mathsf{enc}}$.
  - $\mathsf{QKeyGen}(y,[\![\mathbf{v}^0]\!]_2,[\![\mathbf{v}^1]\!]_2)$: upon $\mathcal{A}$ submitting an attribute $y \in \mathcal{Y}_\lambda$ and two vectors $[\![\mathbf{v}^0]\!]_2,[\![\mathbf{v}^1]\!]_2 \in \mathbb{G}_{\lambda,2}^N$, run $\mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK},y,[\![\mathbf{v}^b]\!]_2)$ and return DK to $\mathcal{A}$. Add $(y,[\![\mathbf{v}^0]\!]_2,[\![\mathbf{v}^1]\!]_2)$ to $\mathcal{Q}_{\mathsf{key}}$.
- The second item of the admissibility condition changes to: for all tuples $(x,[\![\mathbf{u}^0]\!]_1,[\![\mathbf{u}^1]\!]_1) \in \mathcal{Q}_{\mathsf{enc}}$ and $(y,[\![\mathbf{v}^0]\!]_2,[\![\mathbf{v}^1]\!]_2) \in \mathcal{Q}_{\mathsf{key}}$, if $P_\lambda(x,y) = 0$ then $[\![\mathbf{u}^0]\!]_1[\![(\mathbf{v}^0)^\top]\!]_2 = [\![\mathbf{u}^1]\!]_1[\![(\mathbf{v}^1)^\top]\!]_2$.

Assuming MDDH on a pairing group $\mathbb{G}$, there exists an AB-IPFE scheme for ABP predicates based on $\mathbb{G}$ that is function-hiding secure against unbounded collusions (special case of [ATY23]).

Restricted Evasive Security for IB-IPFE. For the special case of IB-IPFE we define very selective evasive security against restricted samplers following [HLL24].

Definition 15 (Restricted Evasive Security of IB-IPFE)**.** Let $\mathcal{S} = (\mathcal{S}_{\mathsf{V}}, \mathcal{S}_{\mathsf{U}})$ be two algorithms with the following syntax.

- Given randomness r_{pub}, $\mathcal{S}_{\mathsf{V}}(1^\lambda; r_{\mathsf{pub}})$ outputs

$$1^N, \mathsf{ID} \subseteq \mathcal{I},\ \{(1^{I_{\mathsf{id}}},\ 1^{J_{\mathsf{id}}},\ \mathbf{V}_{\mathsf{id}} \in \mathbb{Z}_p^{J_{\mathsf{id}} \times N})\}_{\mathsf{id} \in \mathsf{ID}},\ \sigma_{\mathsf{pre}} \geq 0\,.$$

- Given r_{pub} and additional randomness r_{pri}, $\mathcal{S}_{\mathsf{U}}(1^\lambda, r_{\mathsf{pub}}; r_{\mathsf{pri}})$ proceeds in the following two steps.

  1. Use a fixed deterministic algorithm $\mathcal{S}_{\mathsf{A0}}$ to compute

$$(1^{n'}, \{\mathbf{A}_{\mathsf{id},0,i}\}_{\mathsf{id} \in \mathsf{ID}, i \in [I_{\mathsf{id}}]}) \leftarrow \mathcal{S}_{\mathsf{A0}}(1^\lambda, r_{\mathsf{pub}})\,,$$

  where $\mathbf{A}_{\mathsf{id},0,i} \in \mathbb{Z}_p^{n' \times N}$ for all $\mathsf{id} \in \mathsf{ID}$, $i \in [I_{\mathsf{id}}]$, and the values of ID and $\{I_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}}$ agree with the output of $\mathcal{S}_{\mathsf{V}}(1^\lambda; r_{\mathsf{pub}})$.

  2. Use r_{pri} to sample $\mathbf{d}_0 \xleftarrow{\$} \mathbb{Z}_p^{n'}$ and output $\{\mathbf{U}_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}}$, where

$$\mathbf{U}_{\mathsf{id}} = \begin{bmatrix} — & \mathbf{u}_{\mathsf{id},1} & — \\ & \vdots & \\ — & \mathbf{u}_{\mathsf{id},I_{\mathsf{id}}} & — \end{bmatrix} \in \mathbb{Z}_p^{I_{\mathsf{id}} \times N}\,,$$

  and $\mathbf{u}_{\mathsf{id},i} = \mathbf{d}_0 \mathbf{A}_{\mathsf{id},0,i}$ for all $\mathsf{id} \in \mathsf{ID}$ and $i \in [I_{\mathsf{id}}]$.

Suppose $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and, for all $\mathsf{id} \in \mathsf{ID}$,

$$\begin{aligned} \mathbf{E}_{\mathsf{id}} &\leftarrow \mathcal{D}_{\mathbb{Z},\sigma_{\mathsf{pre}}}^{I_{\mathsf{id}} \times J_{\mathsf{id}}}\,, & \mathsf{CT}_{\mathsf{id}}^{*} &\leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathsf{id}, \mathbf{U}_{\mathsf{id}})\,, \\ \mathbf{\Delta}_{\mathsf{pre},\mathsf{id}} &\xleftarrow{\$} \mathbb{Z}_p^{I_{\mathsf{id}} \times J_{\mathsf{id}}}\,, & \mathsf{CT}_{\mathsf{id}}^{\$} &\leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathsf{id}, \mathbf{U}_{\$,\mathsf{id}})\,, \\ \mathbf{U}_{\$,\mathsf{id}} &\xleftarrow{\$} \mathbb{Z}_p^{I_{\mathsf{id}} \times N}\,, & \mathsf{DK}_{\mathsf{id}} &\leftarrow \mathsf{KeyGen}(\mathsf{MSK}, \mathsf{id}, \mathbf{V}_{\mathsf{id}})\,. \end{aligned}$$

$\mathcal{S}$ has *pseudorandom noisy inner products*, denoted $\mathsf{IPFEsec}_{\mathsf{pre}}^{\mathcal{S}}$, if

$$\left\{\left(1^\lambda, r_{\mathsf{pub}}, \{\mathbf{U}_{\mathsf{id}} \mathbf{V}_{\mathsf{id}}^\top + \mathbf{E}_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}}\right)\right\}_{\lambda \in \mathbb{N}} \approx_c \left\{\left(1^\lambda, r_{\mathsf{pub}}, \{\mathbf{\Delta}_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}}\right)\right\}_{\lambda \in \mathbb{N}}\,.$$

The IB-IPFE scheme iFE is $\mathcal{S}$-secure, denoted $\mathsf{IPFEsec}_{\mathsf{post}}^{\mathcal{S}}$, if

$$\left\{\left(1^\lambda, r_{\mathsf{pub}}, \mathsf{MPK}, \{\mathsf{CT}_{\mathsf{id}}^{*}, \mathsf{DK}_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}}\right)\right\}_{\lambda \in \mathbb{N}} \approx_c \left\{\left(1^\lambda, r_{\mathsf{pub}}, \mathsf{MPK}, \{\mathsf{CT}_{\mathsf{id}}^{\$}, \mathsf{DK}_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}}\right)\right\}_{\lambda \in \mathbb{N}}\,.$$

The scheme is $\sigma_{\mathsf{pre},0}$-secure if it is $\mathcal{S}$-secure for all efficient $\mathcal{S}$ such that $\sigma_{\mathsf{pre}} \leq \sigma_{\mathsf{pre},0}$ and $\mathsf{IPFEsec}_{\mathsf{pre}}^{\mathcal{S}}$ holds.

Assuming evasive LWE, there exists an IB-IPFE scheme satisfying very selective evasive security against restricted samplers [HLL24].

3.9 Slotted Identity-Based Inner-Product Functional Encryption

We recall the definition of slotted IPFE from [LL20] in the identity-based setting.

Definition 16 (Slotted IB-IPFE)**.** Let $\mathbb{G} = \{\mathbb{G}_\lambda = (\mathbb{G}_{\lambda,1}, \mathbb{G}_{\lambda,2}, \mathbb{G}_{\lambda,\mathsf{t}}, g_{\lambda,1}, g_{\lambda,2}, g_{\lambda,\mathsf{t}}, e_\lambda, p_\lambda)\}_{\lambda \in \mathbb{N}}$ be a sequence of pairing groups and $\{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ a sequence of identity spaces. A slotted IPFE scheme based on $\mathbb{G}$ consists of five efficient algorithms:

$\mathsf{Setup}(1^\lambda, \mathfrak{s}_{\mathsf{pub}}, \mathfrak{s}_{\mathsf{pri}}) \to (\mathsf{MPK}, \mathsf{MSK})$**:** On input the security parameter and two disjoint index sets, the public slot $\mathfrak{s}_{\mathsf{pub}}$ and the private slot $\mathfrak{s}_{\mathsf{pri}}$, this algorithm outputs a pair of a master public and a master secret key $(\mathsf{MPK}, \mathsf{MSK})$. We denote the whole index set by $\mathfrak{s} := \mathfrak{s}_{\mathsf{pub}} \cup \mathfrak{s}_{\mathsf{pri}}$.

$\mathsf{Enc}(\mathsf{MSK}, i, [\![\mathbf{u}]\!]_1) \to \mathsf{CT}$**:** On input a master secret key MSK, an identity $i \in \mathcal{I}_\lambda$ and an encoding of a vector $\mathbf{u} \in \mathbb{Z}_{p_\lambda}^{\mathfrak{s}}$ in $\mathbb{G}_{\lambda,1}$, this algorithm outputs a ciphertext CT for $\mathbf{u}$.

$\mathsf{KeyGen}(\mathsf{MSK}, i', [\![\mathbf{v}]\!]_2) \to \mathsf{DK}$**:** On input a master secret key MSK, an identity $i' \in \mathcal{I}_\lambda$ and an encoding of a vector $\mathbf{v} \in \mathbb{Z}_{p_\lambda}^{\mathfrak{s}}$ in $\mathbb{G}_{\lambda,2}$, this algorithm outputs a decryption key DK for $\mathbf{v}$.

$\mathsf{Dec}(\mathsf{DK}, \mathsf{CT}) \to [\![d]\!]_{\mathsf{t}} \vee \perp$**:** On input a decryption key DK and a ciphertext CT, this algorithm outputs an element $[\![d]\!]_{\mathsf{t}} \in \mathbb{G}_{\lambda,\mathsf{t}}$ or the special symbol $\perp$ indicating failure of decryption.

$\mathsf{SlotEnc}(\mathsf{MPK}, i, [\![\mathbf{u}_{\mathsf{pub}}]\!]_1) \to \mathsf{CT}$**:** On input a master public key MPK, an identity $i \in \mathcal{I}_\lambda$ and an encoding of a message vector $\mathbf{u}_{\mathsf{pub}} \in \mathbb{Z}_{p_\lambda}^{\mathfrak{s}_{\mathsf{pub}}}$ in $\mathbb{G}_{\lambda,1}$, this algorithm outputs a ciphertext for the vector $\mathbf{u} = \mathbf{u}_{\mathsf{pub}}|^{\mathfrak{s}} \in \mathbb{Z}_{p_\lambda}^{\mathfrak{s}}$.

Correctness. A slotted IB-IPFE scheme satisfies *decryption correctness* if for all $\lambda \in \mathbb{N}$, all disjoint index sets $\mathfrak{s}_{\mathsf{pub}}, \mathfrak{s}_{\mathsf{pri}}$, all identities $i \in \mathcal{I}_\lambda$ and all vectors $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_{p_\lambda}^{\mathfrak{s}}$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{DK}, \mathsf{CT}) = [\![\mathbf{u}\mathbf{v}^\top]\!]_{\mathsf{t}} \;\middle|\; \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, \mathfrak{s}_{\mathsf{pub}}, \mathfrak{s}_{\mathsf{pri}}) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, i, [\![\mathbf{u}]\!]_1) \\ \mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, i, [\![\mathbf{v}]\!]_2) \end{array}\right] = 1 \ .$$

Furthermore, we say that a slotted IB-IPFE scheme satisfies *slot-mode correctness* if for all $\lambda \in \mathbb{N}$, all disjoint index sets $\mathfrak{s}_{\mathsf{pub}}, \mathfrak{s}_{\mathsf{pri}}$, all identities $i \in \mathcal{I}_\lambda$ and $\mathbf{u}_{\mathsf{pub}} \in \mathbb{Z}_p^{\mathfrak{s}_{\mathsf{pub}}}$, the following distributions $\mathcal{D}_0, \mathcal{D}_1$ are identical:

$$\mathcal{D}_0 = \left\{(\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) \;\middle|\; \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, \mathfrak{s}_{\mathsf{pub}}, \mathfrak{s}_{\mathsf{pri}}) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, i, [\![\mathbf{u}_{\mathsf{pub}}|^{\mathfrak{s}}]\!]_1) \end{array}\right\} \ ,$$

$$\mathcal{D}_1 = \left\{(\mathsf{MPK}, \mathsf{MSK}, \mathsf{CT}) \;\middle|\; \begin{array}{l} (\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, \mathfrak{s}_{\mathsf{pub}}, \mathfrak{s}_{\mathsf{pri}}) \\ \mathsf{CT} \leftarrow \mathsf{SlotEnc}(\mathsf{MPK}, i, [\![\mathbf{u}_{\mathsf{pub}}]\!]_1) \end{array}\right\} \ ,$$

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

Security. We define adaptive function-hiding IND-CPA security.

Definition 17 (Function-Hiding Security)**.** A slotted IB-IPFE scheme iFE satisfies function-hiding security if $\mathbf{Exp}_{0,\mathcal{A}}^{\text{sl-ipfe-}b}(1^\lambda) \approx_c \mathbf{Exp}_{1,\mathcal{A}}^{\text{sl-ipfe-}b}(1^\lambda)$ for all PPT adversaries $\mathcal{A}$, where $\mathbf{Exp}_{\mathsf{iFE},\mathcal{A}}^{\text{sl-ipfe-}b}$ for $b \in \{0,1\}$ proceeds as follows.

- **Setup.** Launch $\mathcal{A}(1^\lambda)$, generate $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and send MPK to $\mathcal{A}$. Initialize two empty sets $\mathcal{Q}_{\mathsf{enc}}, \mathcal{Q}_{\mathsf{key}} = \varnothing$.
- **Query.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ has two options.
  - $\mathsf{QEnc}(i, [\![\mathbf{u}^0]\!]_1, [\![\mathbf{u}^1]\!]_1)$: upon $\mathcal{A}$ submitting an attribute $i \in \mathcal{I}_\lambda$ and two vectors $[\![\mathbf{u}^0]\!]_1, [\![\mathbf{u}^1]\!]_1 \in \mathbb{G}_{\lambda,1}^{\mathfrak{s}}$, run $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MSK}, i, [\![\mathbf{u}^b]\!]_1)$ and return CT to $\mathcal{A}$. Add $(i, [\![\mathbf{u}^0]\!]_1, [\![\mathbf{u}^1]\!]_1)$ to $\mathcal{Q}_{\mathsf{enc}}$.
  - $\mathsf{QKeyGen}(i', [\![\mathbf{v}^0]\!]_2, [\![\mathbf{v}^1]\!]_2)$: upon $\mathcal{A}$ submitting an attribute $i' \in \mathcal{I}_\lambda$ and two vectors $[\![\mathbf{v}^0]\!]_2, [\![\mathbf{v}^1]\!]_2 \in \mathbb{G}_{\lambda,2}^{\mathfrak{s}}$, run $\mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, i', [\![\mathbf{v}^b]\!]_2)$ and return DK to $\mathcal{A}$. Add $(i', [\![\mathbf{v}^0]\!]_2, [\![\mathbf{v}^1]\!]_2)$ to $\mathcal{Q}_{\mathsf{key}}$.
- **Guess.** The adversary outputs a bit $b' \in \{0,1\}$. The outcome of the experiment is b' if $\mathcal{A}$ satisfies the following admissibility condition. Otherwise, the outcome is set to 0. The adversary $\mathcal{A}$ is admissible if for all $(i, [\![\mathbf{u}^0]\!]_1, [\![\mathbf{u}^1]\!]_1) \in \mathcal{Q}_{\mathsf{enc}}$ and $(i', [\![\mathbf{v}^0]\!]_2, [\![\mathbf{v}^1]\!]_2) \in \mathcal{Q}_{\mathsf{key}}$ it holds that $[\![\mathbf{v}_0|_{\mathfrak{s}_{\mathsf{pub}}}]\!]_2 = [\![\mathbf{v}_1|_{\mathfrak{s}_{\mathsf{pub}}}]\!]_2$ and, if $i = i'$, then $[\![\mathbf{u}^0]\!]_1 [\![(\mathbf{v}^0)^\top]\!]_2 = [\![\mathbf{u}^1]\!]_1 [\![(\mathbf{v}^1)^\top]\!]_2$.

There exists a (non-identity-based) slotted IPFE scheme based on $\mathbb{G}$ which can be proven secure under the MDDH assumption on $\mathbb{G}$. The construction is based on a sequence of works [ALS16, Wee17, LV16, Lin17] and has been described explicitly in [LL20]. In this work, we use slotted IB-IPFE only for identity spaces of polynomial size. Slotted IB-IPFE for such small identity spaces can be obtained naively by running multiple slotted IPFE instance in parallel, one for each identity.

3.10 Attribute-Based Encryption

We recall the definition of Attribute-Based Encryption (ABE).

Definition 18 (Attribute-Based Encryption)**.** Let $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$, $P = \{P_\lambda\}_{\lambda\in\mathbb{N}}$, $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be sequences of message spaces, predicates, ciphertext attribute spaces and key attribute spaces, respectively. An ABE scheme for $\mathcal{M}$, P, $\mathcal{X}$ and $\mathcal{Y}$ consists of four efficient algorithms:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{MPK}, \mathsf{MSK})$**:** On input the security parameter, this algorithm outputs a pair of a master public key MPK and a master secret key MSK.

$\mathsf{Enc}(\mathsf{MPK}, x, \mu) \to \mathsf{CT}$**:** On input a master public key MPK, a ciphertext attribute $x \in \mathcal{X}_\lambda$ and a message $\mu \in \mathcal{M}_\lambda$, this algorithm outputs a ciphertext CT for μ created with respect to x.

$\mathsf{KeyGen}(\mathsf{MSK}, y) \to \mathsf{DK}$**:** On input a master secret key MSK and a key attribute $y \in \mathcal{Y}_\lambda$, this algorithm outputs a decryption key DK for y.

$\mathsf{Dec}(\mathsf{DK}, \mathsf{CT}) \to \mu' \vee \bot$**:** On input a decryption key DK and a ciphertext CT, this algorithm outputs an element $\mu' \in \mathcal{M}_\lambda$ or $\bot$.

Correctness. An ABE scheme is said to be correct if for all $\lambda \in \mathbb{N}$, all attributes $x \in \mathcal{X}_\lambda$ and $y \in \mathcal{Y}_\lambda$ such that $P_\lambda(x, y) = 0$, and all messages $\mu \in \mathcal{M}_\lambda$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{DK},\mathsf{CT}) = \mu \;\middle|\; \begin{array}{l}(\mathsf{MPK},\mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)\\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MPK}, x, \mu)\\ \mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, y)\end{array}\right] = 1 \ ,$$

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

Security. We define adaptive IND-CPA security.

Definition 19 (Security)**.** An ABE scheme ABE is secure if $\mathbf{Exp}^{\mathsf{abe\text{-}0}}_{\mathsf{ABE},\mathcal{A}}(1^\lambda) \approx_c \mathbf{Exp}^{\mathsf{abe\text{-}1}}_{\mathsf{ABE},\mathcal{A}}(1^\lambda)$ for all PPT adversaries $\mathcal{A}$, where $\mathbf{Exp}^{\mathsf{abe}\text{-}b}_{\mathsf{ABE},\mathcal{A}}$ for $b \in \{0,1\}$ proceeds as follows.

- **Setup.** Launch $\mathcal{A}(1^\lambda)$, run $(\mathsf{MPK},\mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and send MPK to $\mathcal{A}$. Initialize an empty set $\mathcal{Q} = \varnothing$.
- **Query I.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ submits a query $\mathsf{QKeyGen}(y)$ where $y \in \mathcal{Y}_\lambda$. Upon this query, run $\mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, y)$ and return DK to $\mathcal{A}$. Add y to $\mathcal{Q}$.
- **Challenge.** Upon $\mathcal{A}$ submitting a challenge attribute x^* and two messages $\mu^0, \mu^1 \in \mathcal{M}_\lambda$, run $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{MPK}, x^*, \mu^b)$ and return CT to $\mathcal{A}$.
- **Query II.** Same as Query I.
- **Guess.** The adversary outputs a bit $b' \in \{0,1\}$. The outcome of the experiment is b' if $P_\lambda(x^*, y) \neq 0$ for all $y \in \mathcal{Q}$. Otherwise, the outcome is set to 0.

We mostly use ABE for the equality predicate, i.e., Identity-Based Encryption (IBE). There exist various IBE schemes in the group-based setting, e.g., [Wat09, CLL+13, JR17], and in the lattice-based setting, e.g., [GPV08, ABB10]. We also rely on a CP-ABE for bounded-depth circuits in which case $\mathcal{X}_\lambda = \mathcal{F}^{\mathsf{circ}}_{n,d}$, $\mathcal{Y}_\lambda = \mathbb{Z}_{p_\lambda}$ (for a prime p_λ) and $P_\lambda(x,y) = x(y)$ for all $\lambda \in \mathbb{N}$. Assuming the hardness of LWE, there exists a CP-ABE for bounded-depth circuits in the *secret-key* setting [AY20].

3.11 Other Preliminaries

We recall the following lemma from [LLW21].

Lemma 20 ([LLW21], Lemma 5.4)**.** *Let $\kappa = Q\tau\varepsilon^2$ and $\tau = \Theta(\varepsilon)$. There exists an efficient sampling algorithm, denoted $\mathsf{SamplerSet}(\kappa, Q, \tau)$, with the following properties:*

1. *The sampler always outputs a set $\Delta \subset [\kappa]$ with cardinality τ.*
2. *For independent samples $\Delta_1, \ldots, \Delta_Q$ from $\mathsf{SamplerSet}(\kappa, Q, \tau)$, the sets are cover-free with probability $(1 - 2^{-\Omega(\varepsilon)})$, i.e., for all $i \in [Q]$,*

$$\Pr\left[\Delta_i \setminus \left(\bigcup_{j \in [Q]\setminus\{i\}} \Delta_j\right) \neq \varnothing\right] \geq 1 - 2Q \cdot 2^{-\Omega(\varepsilon)} \ .$$

4 MC-ABE for Succinctly Enumerable Function Classes

Our constructions in this section capture all function classes that satisfy a specific property which we call succinct enumerability.

Succinct Enumerability. We define the following notations:

- Let $X_0 = \{0\}$, $X_1 = \{1\}$ and $X_\star = \{0,1\}$. For a vector $\mathbf{v} \in \{0,1,\star\}^k$, we let

$$X_{\mathbf{v}} = X_{\mathbf{v}[1]} \times \cdots \times X_{\mathbf{v}[k]} \ .$$

- For concatenated vectors $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in \{0,1,\star\}^{[n-1]\times[k]}$, we write

$$X_{\mathbf{v}} = X_{\mathbf{v}_1} \times \cdots \times X_{\mathbf{v}_{n-1}} \qquad \text{and} \qquad J_{\mathbf{v}} = \{(i,j) \in [n-1] \times [k] : \mathbf{v}_i[j] \neq \star\} \ .$$

Definition 21 (Succinct Enumerability)**.** Consider a policy class $\mathcal{F}$ containing policies $f\colon \{0,1\}^{[0;n-1]\times[k]} \to \{0,1\}$. $\mathcal{F}$ is called succinctly enumerable if there exists a set $V \subseteq \{0,1,\star\}^{[n-1]\times[k]}$ of polynomial size such that for all $f \in \mathcal{F}$ there exists a set $J_f \subseteq [n-1] \times [k]$ satisfying the following two conditions.

- **Correctness.** For all $(\mathbf{x}_0, \mathbf{x}_{\geq 1}) \in \{0,1\}^{\{0\}\times[k]} \times \{0,1\}^{[n-1]\times[k]}$, if $f(\mathbf{x}_0, \mathbf{x}_{\geq 1}) = 0$ then there exists $\mathbf{v} \in V$ such that $J_{\mathbf{v}} = J_f$ and $\mathbf{x}_{\geq 1} \in X_{\mathbf{v}}$.
- **Security.** For all $\mathbf{x}_0 \in \{0,1\}^{\{0\}\times[k]}$ and $\mathbf{v} \in V$, if $J_{\mathbf{v}} = J_f$ then the function f is constant on the set $\{(\mathbf{x}_0, \mathbf{x}_{\geq 1}) : \mathbf{x}_{\geq 1} \in X_{\mathbf{v}}\}$.

We may consider the following examples of succinctly enumerable classes $\mathcal{F}$.

- **Constant Locality.** Let $k, n = \mathrm{poly}(\lambda)$ and $d = O(1)$ be some fixed upper bound such that each function $f \in \mathcal{F}$ depends on at most d out of the $k(n-1)$ bits in $\mathbf{x}_{\geq 1}$. For this class of functions, we define

$$V = \left\{\mathbf{v} \in \{0,1,\star\}^{[n-1]\times[k]} : |J_{\mathbf{v}}| = d\right\}$$

and, for $f \in \mathcal{F}$, we can pick any set $J_f \subseteq [n-1] \times [k]$ of size d which includes the (at most d) bits that the policy depends on. Correctness and security follow immediately from the fact that the inputs in $([n-1] \times [k]) \setminus J_f$ are irrelevant for the output of f. Therefore, it suffices to observe that

$$|V| \leq \binom{kn}{d} \cdot 2^d = \mathrm{poly}(\lambda) \ .$$

Note that this class includes all NC^0 policies which is one of the classes considered in [PS24]. However, we do *not* limit the bit dependency for $\mathbf{x}_0$ here, hence this class also includes policies that cannot be computed by an NC^0 circuit.

- **Polynomial Accepting Sets.** Let $k, n = \mathrm{poly}(\lambda)$ and $V \subseteq \{0,1\}^{[n-1]\times k}$ a set of polynomial size satisfying the following implication for each $(\mathbf{x}_0, \mathbf{x}_{\geq 1}) \in \{0,1\}^{\{0\}\times[k]} \times \{0,1\}^{[n-1]\times[k]}$ and $f \in \mathcal{F}$: if $f(\mathbf{x}_0, \mathbf{x}_{\geq 1}) = 0$ then $\mathbf{x}_{\geq 1} \in V$. When choosing $J_f = [n-1] \times [k]$ for each $f \in \mathcal{F}$, then correctness and security are immediate as the definition of V does not involve any $\star$-coordinates, and $|V| = \mathrm{poly}(\lambda)$ by definition.

  While this function class may seem rather limited, we observe that we do not limit the total number of accepting inputs but only the number of accepting $\mathbf{x}_{\geq 1}$ *per* $\mathbf{x}_0$. This is more general than the classes of $O(1)$-threshold and $(kn - O(1))$-threshold policies considered in [PS24] whose total number of accepting inputs is polynomial.

- **Logarithmic-Length Inputs.** Let k, n such that $kn = O(\log \lambda)$, i.e., the total length of the input is logarithmic in λ. For this parameter choice, every class of functions is succinctly enumerable as we can define $V = \{0,1\}^{[n-1]\times[k]}$ and $J_f = [n-1] \times [k]$ for each $f \in \mathcal{F}$. Correctness and security are immediate as the definition of V does not involve any $\star$-coordinates, and $|V| = 2^{k(n-1)} = \mathrm{poly}(\lambda)$ due to the small input length.

  This includes all NC^1 policies with logarithmic-length inputs as considered in [PS24] but equally applies to all short-input bounded-depth circuits as we show in Section 4.3.

Complete Policies. Let $k, n \in \mathbb{N}$. We define $c \colon \{0,1\}^{[0;n-1]\times\{0\}} \to \{0,1\}$ as

$$c((x_{i,0})_{i\in[0;n-1]}) = \bigwedge_{i\in[0;n-1]} x_{i,0} \ .$$

Given a policy $f \colon \{0,1\}^{[0;n-1]\times[k]} \to \{0,1\}$, we call $(c \wedge f) \colon \{0,1\}^{[0;n-1]\times[0;k]} \to \{0,1\}$ the *complete* policy defined as

$$(c \wedge f)\big((x_{i,j})_{i,j}\big) = c\big((x_{i,0})_{i\in[0;n-1]}\big) \wedge f\big((x_{i,j})_{(i,j)\in[0;n-1]\times[k]}\big) \ .$$

To deal with complete policies, we adapt our previous notations as follows:

- For a vector $\mathbf{v} \in \{0,1\star\}^{[k]}$, we write $\overline{\mathbf{v}} = (1, \mathbf{v}) \in \{0,1\}^{[0;k]}$.
- For $\mathbf{v} \in V$, we extend the notation of $J_{\mathbf{v}} = \{(i,j) \in [n-1] \times [k] : \mathbf{v}_i[j] \neq \star\}$ to

  $$\overline{J}_{\mathbf{v}} = J_{\mathbf{v}} \cup (\{0\} \times [k]) \cup ([0; n-1] \times \{0\}) \subseteq ([0; n-1] \times [0; k]) \ .$$

  Intuitively, for a given vector $\mathbf{x}_0 \in \{0,1\}^{\{0\}\times[k]}$, the set $\overline{J}_{\mathbf{v}}$ contains the indices of the non-$\star$ coordinates for the extension $(\overline{\mathbf{x}}_0, \overline{\mathbf{v}}_1, \ldots, \overline{\mathbf{v}}_{n-1}) \in \overline{V}_{\mathbf{x}_0}$ of a vector $(\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in V$.
- Similarly, for a function f with corresponding set J_f (as per Definition 21), we extend the notation of $J_f \subseteq [n-1] \times [k]$ to

  $$\overline{J}_f = J_f \cup (\{0\} \times [k]) \cup ([0; n-1] \times \{0\}) \subseteq ([0; n-1] \times [0; k]) \ .$$

4.1 Construction from Slotted IPFE

Let $\mathcal{F}$ be a succinctly enumerable class of functions $f\colon \{0,1\}^{[0;n-1]\times[k]} \to \{0,1\}$. We present our construction of MC-ABE for the predicate $P_{\mathcal{F}}\colon \mathcal{X}^n \times \mathcal{Y} \to \{0,1\}$, where $\mathcal{X} = \{0,1\}^{[k]}$, $\mathcal{Y} = \mathcal{F}$ and $P_{\mathcal{F}}((\mathbf{x}_0,\dots,\mathbf{x}_{n-1}), f) = f(\mathbf{x}_0,\dots,\mathbf{x}_{n-1})$. In this section, we give the construction using a slotted IPFE as building block. This allows basing security on MDDH_k on pairings. In Section 4.3, we give a modified version of the construction that makes use of evasive IPFE, thus providing an instantiation from (public-coin) evasive LWE.

Construction 22 (MC-ABE from Slotted IPFE)**.** Let $\mathcal{F} \subseteq \mathsf{NC}^1$ be a succinctly enumerable class of functions. Our construction uses the following ingredients:

- an identity-based slotted IPFE scheme $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ based on a pairing group $\mathbb{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}, g_1, g_2, g_\mathsf{t}, e, p)$ with identity space $([0;n-1]\times[0;k]) \cup \{\mathsf{pad}\}$,
- an identity-based encryption scheme $\mathsf{idFE} = (\mathsf{idSetup}, \mathsf{idEnc}, \mathsf{idKeyGen}, \mathsf{idDec})$ with identity space $\mathcal{L} \times \{0,1,\star\}^{[k]}$ for $\mathcal{L} = \{0,1\}^{\mathrm{poly}(\lambda)}$ and message space being the ciphertext space of iFE, and
- an (exact) linear garbling scheme $(\mathsf{Gen}, \mathsf{Share}, \mathsf{Rec}, \mathsf{Sim})$ for NC^1.

The MC-ABE scheme ABE for the predicate $P_{\mathcal{F}}$ with message space $\mathcal{M} = \mathbb{G}_\mathsf{t}$ and label space $\mathcal{L}$:

$\mathsf{Setup}(1^\lambda)$ takes as input the security parameter 1^λ and generates

$$\begin{aligned}\mathsf{iMPK}, \mathsf{iMSK} &\leftarrow \mathsf{iSetup}(1^\lambda, [2\kappa], \{2\kappa+1\})\\ \left\{(\mathsf{idMPK}_i, \mathsf{idMSK}_i) \leftarrow \mathsf{idSetup}(1^\lambda)\right\}_{i\in[n-1]}&.\end{aligned}$$

Then it outputs $(\mathsf{MPK}, \mathsf{MSK}, \{\mathsf{SK}_i\}_{i\in[n-1]})$ as follows:

$$\mathsf{MPK} = \left(\mathsf{iMPK}, \{\mathsf{idMPK}_i\}_{i\in[n-1]}\right),\ \ \mathsf{MSK} = \mathsf{iMSK},\ \ \{\mathsf{SK}_i = \mathsf{idMSK}_i\}_{i\in[n-1]}.$$

$\mathsf{Enc}(\mathsf{MPK}, L, \mathbf{x}_0, [\![\mu]\!]_\mathsf{t})$ takes $\mathsf{MPK} = (\mathsf{iMPK}, \{\mathsf{idMPK}_i\}_{i\in[n-1]})$, a label $L \in \mathcal{L}$, an attribute $\mathbf{x}_0 \in \{0,1\}^k$ and a message $[\![\mu]\!]_\mathsf{t} \in \mathbb{G}_\mathsf{t}$ as input. The algorithm picks $[\![\sigma]\!]_1 \xleftarrow{\$} \mathbb{G}_1$ and $[\![\mathbf{r}^\mathbf{v}]\!]_1 \xleftarrow{\$} \mathbb{G}_1^\kappa$ for each $\mathbf{v} \in V$, sets $\mathbf{v}_0 = \mathbf{x}_0$ and

$$\begin{aligned}\left\{\mathbf{p}^\mathbf{v}_\mathsf{pad} = (\mathbf{r}^\mathbf{v}, \sigma, \mathbf{0}_{\kappa-1})\right\}^{\mathbf{v}\in V}&\\ \left\{\mathbf{p}^\mathbf{v}_{i,j} = (0^{\bar{\mathbf{v}}_i[j]\cdot\kappa}, \mathbf{r}^\mathbf{v}, 0^{(1-\bar{\mathbf{v}}_i[j])\cdot\kappa})\right\}^{\mathbf{v}=(\mathbf{v}_1,\dots,\mathbf{v}_{n-1})\in V}_{(i,j)\in\bar{J}_\mathbf{v}}&,\end{aligned}$$

computes $[\![d]\!]_\mathsf{t} = [\![\mu+\sigma]\!]_\mathsf{t}$ and generates

$$\begin{aligned}\left\{\mathsf{iCT}^\mathbf{v}_\mathsf{pad} \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, \mathsf{pad}, [\![\mathbf{p}^\mathbf{v}_\mathsf{pad}]\!]_1)\right\}^{\mathbf{v}\in V}&\\ \left\{\mathsf{iCT}^\mathbf{v}_{i,j} \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, (i,j), [\![\mathbf{p}^\mathbf{v}_{i,j}]\!]_1)\right\}^{\mathbf{v}\in V}_{(i,j)\in\bar{J}_\mathbf{v}}&\\ \left\{\mathsf{idCT}^\mathbf{v}_{i,j} \leftarrow \mathsf{idEnc}(\mathsf{idMPK}_i, (L,\mathbf{v}_i), \mathsf{iCT}^\mathbf{v}_{i,j})\right\}^{\mathbf{v}=(\mathbf{v}_1,\dots,\mathbf{v}_{n-1})\in V}_{(i,j)\in\bar{J}_\mathbf{v}, i>0}&.\end{aligned}$$

Finally, the algorithm outputs the ciphertext

$$\mathsf{CT}_L = \left([\![d]\!]_\mathsf{t}, \{\mathsf{iCT}^\mathbf{v}_\mathsf{pad}\}^{\mathbf{v}\in V}, \{\mathsf{iCT}^\mathbf{v}_{0,j}\}^{\mathbf{v}\in V}_{(0,j)\in\bar{J}_\mathbf{v}}, \{\mathsf{idCT}^\mathbf{v}_{i,j}\}^{\mathbf{v}\in V}_{(i,j)\in\bar{J}_\mathbf{v}, i>0}\right).$$

$\mathsf{AKeyGen}(\mathsf{SK}_i, L, \mathbf{x}_i)$ takes as input $\mathsf{SK}_i = \mathsf{idMSK}_i$ for some $i \in [n-1]$, a label $L \in \mathcal{L}$ and an attribute $\mathbf{x}_i \in \{0,1\}^k$, defines

$$V_{\mathbf{x}_i} = \left\{\mathbf{v}_i : (\mathbf{v}_1,\dots,\mathbf{v}_{n-1}) \in V, \mathbf{x}_i \in X_{\mathbf{v}_i}\right\},$$

computes

$$\left\{\mathsf{idDK}^{\mathbf{v}_i}_i \leftarrow \mathsf{idKeyGen}(\mathsf{idMSK}_i, (L,\mathbf{v}_i))\right\}^{\mathbf{v}_i\in V_{\mathbf{x}_i}}$$

and outputs $\mathsf{DK}_{L,i} = \{\mathsf{idDK}^{\mathbf{v}_i}_i\}^{\mathbf{v}_i\in V_{\mathbf{x}_i}}$.

$\mathsf{PKeyGen}(\mathsf{MSK}, f)$ takes as input $\mathsf{MSK} = \mathsf{iMSK}$ and a policy $f \in \mathcal{F}$. We let $c \wedge f$ be the complete policy viewed as a function $\{0,1\}^{\overline{J}_f} \to \{0,1\}$.[9] The algorithm samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^\kappa$, generates shares $\{\mathbf{s}_{i,j}^\beta\}_{(i,j)\in\overline{J}_f}^{\beta\in\{0,1\}} \leftarrow \mathsf{Share}(c \wedge f, \mathbf{s})$, defines

$$\mathbf{q}_{\mathsf{pad}} = (\mathbf{s}, 1, \mathbf{0}_{\kappa-1}, 0) \,, \qquad \left\{\mathbf{q}_{i,j} = (\mathbf{s}_{i,j}^0, \mathbf{s}_{i,j}^1, 0)\right\}_{(i,j)\in\overline{J}_{\mathbf{v}}} \,,$$

and computes

$$\begin{aligned} \mathsf{iDK}_{\mathsf{pad}} &\leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, \mathsf{pad}, [\![\mathbf{q}_{\mathsf{pad}}]\!]_2) \\ \left\{\mathsf{iDK}_{i,j} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, (i,j), [\![\mathbf{q}_{i,j}]\!]_2)\right\}_{(i,j)\in\overline{J}_f} \,. \end{aligned}$$

Finally, it outputs $\mathsf{DK}_f = (\mathsf{iDK}_{\mathsf{pad}}, \{\mathsf{iDK}_{i,j}\}_{(i,j)\in\overline{J}_f})$.

$\mathsf{Dec}(\mathsf{DK}_f, \{\mathsf{DK}_{L,i}\}_{i\in[n-1]}, \mathsf{CT}_L)$ takes as input a decryption key DK_f for a policy $f \in \mathcal{F}$, a set of decryption keys $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$ created with respect to attributes $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1} \in \{0,1\}^k$ and a label $L \in \mathcal{L}$, and a ciphertext CT_L created with respect to an attribute $\mathbf{x}_0 \in \{0,1\}^k$ and the same label L. If $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = 1$, return $\perp$. Otherwise, parse DK_f, $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$ and CT_L as above. The algorithm picks $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in V$ such that $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in X_{\mathbf{v}}$ and $J_{\mathbf{v}} = J_f$. Then it recovers d_{pad} and $\mathbf{p}_{f,\mathbf{x}} = (\mathbf{p}_{f,\mathbf{x}}[i,j])_{(i,j)\in\overline{J}_{\mathbf{v}}}$ by decrypting

$$\begin{aligned} \left\{\mathsf{iCT}_{i,j}^{\mathbf{v}} \leftarrow \mathsf{idDec}(\mathsf{idDK}_i^{\mathbf{v}_i}, \mathsf{idCT}_{i,j}^{\mathbf{v}})\right\}_{(i,j)\in\overline{J}_{\mathbf{v}}, i>0} \\ \left\{[\![\mathbf{p}_{f,\mathbf{x}}[i,j]]\!]_{\mathsf{t}} \leftarrow \mathsf{iDec}(\mathsf{iDK}_{i,j}, \mathsf{iCT}_{i,j}^{\mathbf{v}})\right\}_{(i,j)\in\overline{J}_{\mathbf{v}}} \\ [\![d_{\mathsf{pad}}]\!]_{\mathsf{t}} \leftarrow \mathsf{iDec}(\mathsf{iDK}_{\mathsf{pad}}, \mathsf{iCT}_{\mathsf{pad}}^{\mathbf{v}}) \,. \end{aligned}$$

Furthermore, it computes $\mathbf{d}_{f,\mathbf{x}} = (\mathbf{d}_{f,\mathbf{x}}[i,j])_{(i,j)\in\overline{J}_{\mathbf{v}}} \leftarrow \mathsf{Rec}(c \wedge f, \mathbf{x})$, where $(c \wedge f)$ is again viewed as a function with domain $\overline{J}_{\mathbf{v}}$. Finally, the algorithm outputs $[\![\mu']\!]_{\mathsf{t}} = [\![d]\!]_{\mathsf{t}} + [\![\mathbf{p}_{f,\mathbf{x}}]\!]_{\mathsf{t}} \cdot \mathbf{d}_{f,\mathbf{x}}^\top - [\![d_{\mathsf{pad}}]\!]_{\mathsf{t}}$.

Correctness. Consider $f \in \mathcal{F}$ and $(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) \in \{0,1\}^{[0;n-1]\times[k]}$ such that $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = 0$. The correctness property of succinctly enumerable function classes guarantees that there exists $\mathbf{v} \in V$ such that $J_{\mathbf{v}} = J_f$ and $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in X_{\mathbf{v}}$. This implies that all idFE ciphertexts $\{\mathsf{idCT}_{i,j}^{\mathbf{v}}\}_{(i,j)\in\overline{J}_{\mathbf{v}}, i>0}$ and secret keys $\{\mathsf{idDK}_{i,j}^{\mathbf{v}}\}_{(i,j)\in\overline{J}_{\mathbf{v}}, i>0}$ used in the first step of the decryption algorithm are contained in CT_L and $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$, respectively. From the correctness of idFE and iFE it follows that $d_{\mathsf{pad}} = \sigma + \mathbf{s} \cdot \mathbf{r}^\top$ and $\mathbf{p}_{f,\mathbf{x}}[i,j] = \mathbf{s}_{i,j}^{\mathbf{x}_i[j]} \cdot \mathbf{r}^\top$ for each $(i,j) \in \overline{J}_{\mathbf{v}}$. Thus, we have

$$\mathbf{p}_{f,\mathbf{x}} \cdot \mathbf{d}_{f,\mathbf{x}}^\top = (\mathbf{s}_{i,j}^{\mathbf{x}_i[j]})_{(i,j)\in\overline{J}_{\mathbf{v}}} \cdot (\mathbf{I}_m \otimes \mathbf{r}^\top) \cdot \mathbf{d}_{f,\mathbf{x}}^\top = \mathbf{s} \cdot \mathbf{r}^\top \,,$$

where the last equality follows from the correctness of the linear garbling scheme (Lemma 8). Plugging everything together gives

$$[\![\mu']\!]_{\mathsf{t}} = [\![d]\!]_{\mathsf{t}} + [\![\mathbf{p}_{f,\mathbf{x}}]\!]_{\mathsf{t}} \cdot \mathbf{d}_{f,\mathbf{x}}^\top - [\![d_{\mathsf{pad}}]\!]_{\mathsf{t}} = [\![(\mu + \sigma) + \mathbf{s} \cdot \mathbf{r}^\top - (\sigma + \mathbf{s} \cdot \mathbf{r}^\top)]\!]_{\mathsf{t}} = [\![\mu]\!]_{\mathsf{t}} \,.$$

Security. We prove selective security.

Proposition 23. *If the* MDDH_k *assumption holds in* $\mathbb{G}_2$, iFE *is slot-mode correct and function-hiding and* idFE *is secure, then Construction 22 is* sel*-secure.*

When instantiating iFE with the identity-based version of slotted IPFE from [LL20] and idFE with the IBE from [HJP18], we obtain the following theorem.

Theorem 24. *Assuming* MDDH_k *on pairings, there exists a* sel*-secure MC-ABE for each succinctly enumerable policy class* $\mathcal{F} \subseteq \mathsf{NC}^1$.

[9] Note that this is well-defined since the definition of succinct enumerability implies that inputs in $([0; n-1] \times [0; k]) \setminus \overline{J}_f$ are irrelevant for the output of f.

4.2 Proof of Security

Proof of Proposition 23. Let $\mathsf{QEnc}(L, \mathbf{x}_0, [\![\mu^0]\!]_t, [\![\mu^1]\!]_t)$ be the (single) encryption query. We let $q = |V|$ and parse $V = \{\mathbf{v}_\ell\}_{\ell\in[q]}$, where $\mathbf{v}_\ell = (\mathbf{v}_{\ell,1}, \ldots, \mathbf{v}_{\ell,n-1})$ for each $\ell \in [q]$. For notational convenience, we also set $\mathbf{v}_{\ell,0} = \mathbf{x}_0$. For the reply to the encryption query, the challenger samples random elements $[\![\sigma]\!]_1 \xleftarrow{\$} \mathbb{G}_1$ and $[\![\mathbf{r}^{\mathbf{v}_\ell}]\!]_1 \xleftarrow{\$} \mathbb{G}_1^\kappa$ for each $\ell \in [q]$, sets

$$\left\{\mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{pad}} = (\mathbf{r}^{\mathbf{v}_\ell}, \sigma, \mathbf{0}_{\kappa-1})\right\}_{\ell\in[q]}, \quad \left\{\mathbf{p}^{\mathbf{v}_\ell}_{i,j} = (0^{\bar{\mathbf{v}}_{\ell,i}[j]\cdot\kappa}, \mathbf{r}^{\mathbf{v}_\ell}, 0^{(1-\bar{\mathbf{v}}_{\ell,i}[j])\cdot\kappa})\right\}_{\ell\in[q],(i,j)\in\bar{J}_{\mathbf{v}_\ell}},$$

computes $[\![d]\!]_t = [\![\mu + \sigma]\!]_t$ and generates

$$\begin{aligned}
\left\{\mathsf{iCT}^{\mathbf{v}_\ell}_{\mathsf{pad}} \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, \mathsf{pad}, [\![\mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{pad}}]\!]_1)\right\}_{\ell\in[q]} \\
\left\{\mathsf{iCT}^{\mathbf{v}_\ell}_{i,j} \leftarrow \mathsf{iSlotEnc}(\mathsf{iMPK}, (i,j), [\![\mathbf{p}^{\mathbf{v}_\ell}_{i,j}]\!]_1)\right\}_{\ell\in[q],(i,j)\in\bar{J}_{\mathbf{v}_\ell}} \\
\left\{\mathsf{idCT}^{\mathbf{v}_{\ell,i}}_{i,j} \leftarrow \mathsf{idEnc}(\mathsf{idMPK}_i, (L, \mathbf{v}_{\ell,i}), \mathsf{iCT}^{\mathbf{v}_\ell}_{i,j})\right\}_{\ell\in[q],(i,j)\in\bar{J}_{\mathbf{v}_\ell}, i>0} .
\end{aligned}$$

Then it sends the following challenge ciphertext CT_L to $\mathcal{A}$:

$$\mathsf{CT}_L = \left([\![d]\!]_t, \{\mathsf{iCT}^{\mathbf{v}_\ell}_{\mathsf{pad}}\}_{\ell\in[q]}, \{\mathsf{iCT}^{\mathbf{v}_\ell}_{0,j}\}_{\ell\in[q],(0,j)\in\bar{J}_{\mathbf{v}_\ell}}, \{\mathsf{idCT}^{\mathbf{v}_{\ell,i}}_{i,j}\}_{\ell\in[q],(i,j)\in\bar{J}_{\mathbf{v}_\ell}, i>0}\right) .$$

We consider a sequence of hybrid games G^b_ℓ, for $\ell \in [0;q]$ and $b \in \{0,1\}$, where G^b_ℓ is the same as $\mathbf{Exp}^{\mathsf{mc\text{-}abe}\text{-}b}_{\mathsf{ABE},\mathcal{A}}$ except that, for each $\ell' \in [q]$, the challenger computes

$$\mathbf{p}^{\mathbf{v}_{\ell'}}_{\mathsf{pad}} = \begin{cases} (\mathbf{r}^{\mathbf{v}_{\ell'}}, \boxed{0}, \mathbf{0}_{\kappa-1}) & \text{if } \ell' \le \ell \\ (\mathbf{r}^{\mathbf{v}_{\ell'}}, \sigma, \mathbf{0}_{\kappa-1}) & \text{if } \ell' > \ell \end{cases}$$

We note that $\mathsf{G}^b_0 = \mathbf{Exp}^{\mathsf{mc\text{-}abe}\text{-}b}_{\mathsf{ABE},\mathcal{A}}$ and $\mathsf{G}^0_q \equiv \mathsf{G}^1_q$. For the latter, we observe that in the hybrids G^0_q and G^1_q the random mask $[\![\sigma]\!]_t \xleftarrow{\$} \mathbb{G}_t$ appears only in the term $[\![d]\!]_t := [\![\sigma + \mu^b]\!]_t$, which implies that the distributions $[\![\sigma+\mu^0]\!]_t$ and $[\![\sigma+\mu^1]\!]_t$ are identical. Below we prove the following claim for all $\ell \in [q]$ and $b \in \{0,1\}$.

Claim 25. *We have* $\mathsf{G}^b_{\ell-1} \approx_c \mathsf{G}^b_\ell$ *under the slot-mode correctness and function-hiding security of* iFE*, the security of* idFE *and the* MDDH_k *assumption in* $\mathbb{G}_2$.

This concludes the proof of the proposition. □

We next prove the claim.

Proof of Claim 25. We consider a sequence of hybrids $\widehat{\mathsf{G}}^0_0, \ldots, \widehat{\mathsf{G}}^0_7, \widehat{\mathsf{G}}^1_7, \ldots, \widehat{\mathsf{G}}^1_0$, where $\widehat{\mathsf{G}}^0_0 = \mathsf{G}^b_{\ell-1}$ and $\widehat{\mathsf{G}}^1_0 = \mathsf{G}^b_\ell$. We argue $\widehat{\mathsf{G}}^{b'}_{\alpha-1} \approx_c \widehat{\mathsf{G}}^{b'}_\alpha$, for $\alpha \in [7]$ and $b' \in \{0,1\}$, as well as $\mathsf{G}^0_7 \equiv \mathsf{G}^1_7$. Then the claim follows via a hybrid argument. Modifications between consecutive games are highlighted using boxes.

Game $\widehat{\mathsf{G}}^{b'}_0$**:** This is $\mathsf{G}^b_{\ell-1+b'}$.

Game $\widehat{\mathsf{G}}^{b'}_1$**:** This game is the same as $\widehat{\mathsf{G}}^{b'}_0$, except that the challenger computes

$$\begin{aligned}
\mathsf{iCT}^{\mathbf{v}_\ell}_{\mathsf{pad}} &\leftarrow \boxed{\mathsf{iEnc}(\mathsf{iMSK}, \mathsf{pad}, [\![\widetilde{\mathbf{p}}^{\mathbf{v}_\ell}_{\mathsf{pad}}]\!]_1)} \\
\left\{\mathsf{iCT}^{\mathbf{v}_\ell}_{i,j} \leftarrow \boxed{\mathsf{iEnc}(\mathsf{iMSK}, (i,j), [\![\widetilde{\mathbf{p}}^{\mathbf{v}_\ell}_{i,j}]\!]_1)}\right\}_{(i,j)\in\bar{J}_{\mathbf{v}_\ell}} ,
\end{aligned}$$

where $\sigma^0 = \sigma$, $\sigma^1 = 0$ and the encrypted vectors have an additional 0-coordinate taking the private input of Enc into account:

$$\begin{aligned}
\widetilde{\mathbf{p}}^{\mathbf{v}_\ell}_{\mathsf{pad}} &= (\mathbf{r}^{\mathbf{v}_\ell}, \sigma^{b'}, \mathbf{0}_{\kappa-1}, \boxed{0}) \\
\widetilde{\mathbf{p}}^{\mathbf{v}_\ell}_{i,j} &= (0^{\bar{\mathbf{v}}_{\ell,i}[j]\cdot\kappa}, \mathbf{r}^{\mathbf{v}_\ell}, 0^{(1-\bar{\mathbf{v}}_{\ell,i}[j])\cdot\kappa}, \boxed{0}) .
\end{aligned}$$

We have $\widehat{\mathsf{G}}^{b'}_0 \equiv \widehat{\mathsf{G}}^{b'}_1$ which follows from the slot-mode correctness of iFE.

Game $\widehat{\mathsf{G}}_2^{b'}$: We define the set $K_\ell \subseteq [0;n-1]\times[0;k]$ containing the indices of all ciphertexts $\mathsf{iCT}_{i,j}^{\mathbf{v}_\ell}$ known to the adversary:

$$K_\ell = \{(i,j)\in \overline{J}_{\mathbf{v}_\ell} : [\, i\in\mathcal{C}\cup\{0\}\,] \vee [\,\exists (i,L,\mathbf{x}_i)\in \mathcal{Q}_{\mathsf{akey}} \text{ s.t. } \mathbf{x}_i \in X_{\mathbf{v}_{\ell,i}}\,]\} \ .$$

This game is the same as $\widehat{\mathsf{G}}_1^{b'}$ except that the challenger sets $\widetilde{\mathbf{p}}_{i,j}^{\mathbf{v}_\ell} = (\mathbf{0}_{2\kappa}, 0)$ if $(i,j)\notin K_\ell$. We have $\widehat{\mathsf{G}}_1^{b'} \approx_c \widehat{\mathsf{G}}_2^{b'}$ from the security of idFE.

Game $\widehat{\mathsf{G}}_3^{b'}$: Next we deal with the ciphertexts $\mathsf{iCT}_{i,j}^{\mathbf{v}_\ell}$ such that $(i,j)\in K_\ell$. To reply to the encryption query, the challenger now sets

$$\widetilde{\mathbf{p}}_{\mathsf{pad}}^{\mathbf{v}_\ell} = (\mathbf{0}_\kappa, 0, \mathbf{0}_{\kappa-1}, 1)$$

$$\widetilde{\mathbf{p}}_{i,j}^{\mathbf{v}_\ell} = \begin{cases} (\mathbf{0}_\kappa, \mathbf{0}_\kappa, 1) & \text{if } (i,j)\in K_\ell \\ (\mathbf{0}_\kappa, \mathbf{0}_\kappa, 0) & \text{if } (i,j)\notin K_\ell \ . \end{cases}$$

When receiving a query $\mathsf{QPKeyGen}(f)$, the challenger samples a fresh secret $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^\kappa$, generates shares $\{\mathbf{s}_{i,j}^\beta\}_{(i,j)\in\overline{J}_f}^{\beta\in\{0,1\}} \leftarrow \mathsf{Share}(c\wedge f, \mathbf{s})$ and defines

$$\mathbf{q}_{\mathsf{pad}} = (\mathbf{s}, 1, \mathbf{0}_{\kappa-1}, \mathbf{s}\cdot(\mathbf{r}^{\mathbf{v}_\ell})^\top + \sigma^{b'})$$

$$\mathbf{q}_{i,j} = \begin{cases} (\mathbf{s}_{i,j}^0, \mathbf{s}_{i,j}^1, \mathbf{s}_{i,j}^{\overline{\mathbf{v}}_{\ell,i}[j]}\cdot(\mathbf{r}^{\mathbf{v}_\ell})^\top) & \text{if } (i,j)\in \overline{J}_f\cap K_\ell \\ (\mathbf{s}_{i,j}^0, \mathbf{s}_{i,j}^1, 0) & \text{if } (i,j)\in \overline{J}_f\setminus K_\ell \ , \end{cases}$$

where $\mathbf{r}^{\mathbf{v}_\ell} \xleftarrow{\$} \mathbb{Z}_p^\kappa$ is sampled at the outset of the game and used in all queries to $\mathsf{QPKeyGen}$. As the public parts of $\mathbf{q}_{\mathsf{pad}}$, $\{\mathbf{q}_{i,j}\}_{i,j}$ and the inner products $\widetilde{\mathbf{p}}_{\mathsf{pad}}^{\mathbf{v}_\ell}\mathbf{q}_{\mathsf{pad}}^\top$, $\{\widetilde{\mathbf{p}}_{i,j}^{\mathbf{v}_\ell}\mathbf{q}_{i,j}^\top\}_{i,j}$ do not change, it follows $\widehat{\mathsf{G}}_2^{b'} \approx_c \widehat{\mathsf{G}}_3^{b'}$ from the function-hiding security of iFE.

Game $\widehat{\mathsf{G}}_4^{b'}$: This game is the same as $\widehat{\mathsf{G}}_3^{b'}$ except that, upon receiving a query $\mathsf{QPKeyGen}(f)$, the challenger samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^\kappa$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{\kappa(m-1)}$, defines

$$\widehat{s} = \mathbf{s}\cdot(\mathbf{r}^{\mathbf{v}_\ell})^\top \in \mathbb{Z}_p \ , \qquad \widehat{\mathbf{t}} = \mathbf{t}\cdot(\mathbf{I}_{m-1}\otimes(\mathbf{r}^{\mathbf{v}_\ell})^\top) \in \mathbb{Z}_p^{m-1} \ ,$$

and computes the vectors $\mathbf{q}_{\mathsf{pad}}$ and $\{\mathbf{q}_{i,j}\}_{i,j}$ as follows:

$$\{\mathbf{s}_{i,j}^\beta\}_{(i,j)\in\overline{J}_f}^{\beta\in\{0,1\}} \leftarrow \mathsf{Share}(c\wedge f, \mathbf{s}; \mathbf{t}) \ , \quad \{\widehat{s}_{i,j}^\beta\}_{(i,j)\in\overline{J}_f}^{\beta\in\{0,1\}} \leftarrow \mathsf{Share}(c\wedge f, \widehat{s}; \widehat{\mathbf{t}}) \ ,$$

$$\mathbf{q}_{\mathsf{pad}} = (\mathbf{s}, 1, \mathbf{0}_{\kappa-1}, \widehat{s} + \sigma^{b'}) \ ,$$

$$\mathbf{q}_{i,j} = \begin{cases} (\mathbf{s}_{i,j}^0, \mathbf{s}_{i,j}^1, \widehat{s}_{i,j}^{\overline{\mathbf{v}}_{\ell,i}[j]}) & \text{if } (i,j)\in \overline{J}_f\cap K_\ell \\ (\mathbf{s}_{i,j}^0, \mathbf{s}_{i,j}^1, 0) & \text{if } (i,j)\in \overline{J}_f\setminus K_\ell \ . \end{cases}$$

We have $\widehat{\mathsf{G}}_3^{b'} \equiv \widehat{\mathsf{G}}_4^{b'}$ as the modifications are purely syntactical.

Game $\widehat{\mathsf{G}}_5^{b'}$: This game is the same as $\widehat{\mathsf{G}}_4^{b'}$ except that, upon receiving a query $\mathsf{QPKeyGen}(f)$, the challenger samples $\widehat{s} \xleftarrow{\$} \mathbb{Z}_p$ and $\widehat{\mathbf{t}} \xleftarrow{\$} \mathbb{Z}_p^{m-1}$ at random. We have that $\widehat{\mathsf{G}}_4^{b'} \approx_c \widehat{\mathsf{G}}_5^{b'}$ under the MDDH_κ assumption in $\mathbb{G}_2$. For the simulation of the games, we recall the fact that the Share algorithm of the garbling scheme can also take $([\![\widehat{s}]\!]_2, [\![\widehat{\mathbf{t}}]\!]_2)$ encoded in $\mathbb{G}_2$, run the sharing algorithm in the exponent and output $\{[\![\widehat{s}_{i,j}^\beta]\!]_2\}_{(i,j)\in\overline{J}_f}^{\beta\in\{0,1\}}$.

Game $\widehat{\mathsf{G}}_6^{b'}$: This game is the same as $\widehat{\mathsf{G}}_5^{b'}$ except that, upon receiving a query $\mathsf{QPKeyGen}(f)$, the challenger computes

$$\{\widehat{s}_{i,j}^\beta\}_{(i,j)\in\overline{J}_f}^{\beta\in\{0,1\}} \leftarrow \mathsf{Sim}(c\wedge f; \widehat{\mathbf{t}}) \ .$$

It follows from the adversary's admissibility condition and the security property of succinctly enumerable function classes that the adversary does not hold a subset of $\{\widehat{s}_{i,j}^\beta\}_{i,j}$ which enables the reconstruction of $\widehat{s}$. Therefore, we can conclude from the security of the garbling scheme that $\widehat{\mathsf{G}}_5^{b'} \approx_s \widehat{\mathsf{G}}_6^{b'}$.

Game $\widehat{\mathsf{G}}_7^{b'}$: This game is the same as $\widehat{\mathsf{G}}_6^{b'}$ except that, when receiving a query $\mathsf{QPKeyGen}(f)$, the challenger sets

$$\mathbf{q}_{\mathsf{pad}} = (\mathbf{s}, 1, \mathbf{0}_{\kappa-1}, \widehat{s} + \sigma^{b'}) .$$

We note that the random element $\widehat{s} \xleftarrow{\$} \mathbb{Z}_p$ appears only in $\mathbf{q}_{\mathsf{pad}}$, so it perfectly masks $\sigma^{b'}$ and we obtain that $\widehat{\mathsf{G}}_6^{b'} \equiv \widehat{\mathsf{G}}_7^{b'}$. Furthermore, we can observe that the challenger's output in G_7^{β} is independent of the bit b' which implies that $\widehat{\mathsf{G}}_7^0 \equiv \widehat{\mathsf{G}}_7^1$.

□

4.3 Construction from Evasive IPFE

We present our construction of MC-ABE based on evasive IPFE.

Construction 26 (MC-ABE from Evasive IPFE)**.** Let $d, k, n \in \mathbb{N}$ and $\mathcal{F} \subseteq \mathcal{F}_{kn,d}^{\mathsf{circ}}$ be a succinctly enumerable class of functions. Our construction uses the following ingredients:

- an evasive identity-based IPFE scheme $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ with identity space $([0; n-1] \times [0; k]) \cup \{0\}$,
- an identity-based encryption scheme $\mathsf{idFE} = (\mathsf{idSetup}, \mathsf{idEnc}, \mathsf{idKeyGen}, \mathsf{idDec})$ with identity space $\mathcal{L} \times \{0, 1, \star\}^{[k]}$ for $\mathcal{L} = \{0,1\}^{\mathrm{poly}(\lambda)}$ and message space being the ciphertext space of iFE, and
- a noisy linear garbling scheme $\mathsf{NLG} = (\mathsf{gSetup}, \mathsf{gGenF}, \mathsf{gEval})$ supporting $\mathcal{F}_{kn,d}^{\mathsf{circ}}$.

The MC-ABE scheme ABE for the predicate $P_{\mathcal{F}}$ with message space $\mathcal{M} = \{0,1\}$ and label space $\mathcal{L}$ works as follows:

$\mathsf{Setup}(1^\lambda)$ takes as input the security parameter 1^λ and generates

$$(\mathsf{iMPK}, \mathsf{iMSK}) \leftarrow \mathsf{iSetup}(1^\lambda, 1^{2n_r}) , \quad \left\{(\mathsf{idMPK}_i, \mathsf{idMSK}_i) \leftarrow \mathsf{idSetup}(1^\lambda)\right\}_{i\in[n-1]} , \quad \mathsf{gPP} \leftarrow \mathsf{gSetup}(1^\lambda) .$$

Then it outputs $(\mathsf{MPK}, \mathsf{MSK}, \{\mathsf{SK}_i\}_{i\in[n-1]})$ as follows:

$$\mathsf{MPK} = \left(\mathsf{gPP}, \mathsf{iMPK}, \{\mathsf{idMPK}_i\}_{i\in[n-1]}\right) , \qquad \{\mathsf{SK}_i = \mathsf{idMSK}_i\}_{i\in[n-1]} , \qquad \mathsf{MSK} = \mathsf{iMSK} .$$

$\mathsf{Enc}(\mathsf{MPK}, L, \mathbf{x}_0, \mu)$ takes $\mathsf{MPK} = (\mathsf{iMPK}, \{\mathsf{idMPK}_i\}_{i\in[n-1]})$, a label $L \in \mathcal{L}$, an attribute $\mathbf{x}_0 \in \{0,1\}^k$ and a message $\mu \in \{0,1\}$ as input. The algorithm picks a sufficiently large message encoding scalar $r_{\mathsf{msg}} \xleftarrow{\$} \mathbb{Z}_p \setminus [-2 \cdot 2^{-\varepsilon} p; 2 \cdot 2^{-\varepsilon} p]$ and secret vectors $\mathbf{r}^{\mathbf{v}} \xleftarrow{\$} \mathbb{Z}_p^{n_r}$ for each $\mathbf{v} \in V$, sets $\mathbf{v}_0 = \mathbf{x}_0$ and

$$\left\{\mathbf{p}_{\mathsf{msg}}^{\mathbf{v}} = (\mathbf{r}^{\mathbf{v}}, \mu r_{\mathsf{msg}}, \mathbf{0}_{n_r-1})\right\}^{\mathbf{v}\in V}, \quad \left\{\mathbf{p}_{i,j}^{\mathbf{v}} = (\mathbf{r}^{\mathbf{v}}, \overline{\mathbf{v}}_i[j] \cdot \mathbf{r}^{\mathbf{v}})\right\}_{(i,j)\in\overline{J}_{\mathbf{v}}}^{\mathbf{v}=(\mathbf{v}_1,\ldots,\mathbf{v}_{n-1})\in V} .$$

Then it generates iFE and idFE ciphertexts

$$\begin{aligned}
&\left\{\mathsf{iCT}_{\mathsf{msg}}^{\mathbf{v}} \leftarrow \mathsf{iEnc}(\mathsf{iMPK}, 0, \mathbf{p}_{\mathsf{msg}}^{\mathbf{v}})\right\}^{\mathbf{v}\in V} , \\
&\left\{\mathsf{iCT}_{i,j}^{\mathbf{v}} \leftarrow \mathsf{iEnc}(\mathsf{iMPK}, (i,j), \mathbf{p}_{i,j}^{\mathbf{v}})\right\}_{(i,j)\in\overline{J}_{\mathbf{v}}}^{\mathbf{v}\in V} , \\
&\left\{\mathsf{idCT}_{i,j}^{\mathbf{v}} \leftarrow \mathsf{idEnc}(\mathsf{idMPK}_i, (L, \mathbf{v}_i), \mathsf{iCT}_{i,j}^{\mathbf{v}})\right\}_{(i,j)\in\overline{J}_{\mathbf{v}}, i>0}^{\mathbf{v}=(\mathbf{v}_1,\ldots,\mathbf{v}_{n-1})\in V} ,
\end{aligned}$$

and outputs the ciphertext

$$\mathsf{CT}_L = \left(\{\mathsf{iCT}_{\mathsf{msg}}^{\mathbf{v}}\}^{\mathbf{v}\in V}, \{\mathsf{iCT}_{0,j}^{\mathbf{v}}\}_{(0,j)\in\overline{J}_{\mathbf{v}}}^{\mathbf{v}\in V}, \{\mathsf{idCT}_{i,j}^{\mathbf{v}}\}_{(i,j)\in\overline{J}_{\mathbf{v}}, i>0}^{\mathbf{v}\in V}\right) .$$

$\mathsf{AKeyGen}(\mathsf{SK}_i, L, \mathbf{x}_i)$ takes as input $\mathsf{SK}_i = \mathsf{idMSK}_i$ for some $i \in [n-1]$, a label $L \in \mathcal{L}$ and an attribute $\mathbf{x}_i \in \{0,1\}^k$, computes

$$\left\{\mathsf{idDK}_i^{\mathbf{v}_i} \leftarrow \mathsf{idKeyGen}(\mathsf{idMSK}_i, (L, \mathbf{v}_i))\right\}^{\mathbf{v}_i \in V_{\mathbf{x}_i}}$$

and outputs $\mathsf{DK}_{L,i} = \{\mathsf{idDK}_i^{\mathbf{v}_i}\}^{\mathbf{v}_i\in V_{\mathbf{x}_i}}$.

$\mathsf{PKeyGen}(\mathsf{MSK}, f)$ takes as input $\mathsf{MSK} = \mathsf{iMSK}$ and a policy $f \in \mathcal{F}$, generates a noisy linear garbling of $c \wedge f$ by

$$\left(1^{n_f}, \mathbf{w}_{\mathsf{out}}, \{\mathbf{W}^0_{i,j}, \mathbf{W}^{\times}_{i,j}\}_{(i,j)\in\bar{J}_f}, \mathbf{T}, \mathbf{R}\right) \leftarrow \mathsf{gGenF}(\mathsf{gPP}, c \wedge f) \ ,$$

where $c \wedge f$ denotes the complete policy viewed as a function $\{0,1\}^{\bar{J}_f} \to \{0,1\}$, and samples a random public matrix $\mathbf{B} \xleftarrow{s} \mathbb{Z}_p^{n_r \times n_f}$. Then the algorithm computes

$$\begin{aligned} \mathbf{q}_{\mathsf{msg}} &= \left((\mathbf{B}\mathbf{w}^{\top}_{\mathsf{out}})^{\top}, 1, \mathbf{0}_{n_r-1}\right) , \quad \left\{\mathbf{Q}_{i,j} = \left((\mathbf{B}\mathbf{W}^0_{i,j})^{\top}, (\mathbf{B}\mathbf{W}^{\times}_{i,j})^{\top}\right)\right\}_{(i,j)\in\bar{J}_f} , \\ \mathbf{Q}_{\mathsf{t}} &= \left((\mathbf{B}\mathbf{T})^{\top}, \mathbf{0}_{\star\times n_r}\right) , \end{aligned}$$

and generates iFE decryption keys

$$\begin{aligned} \mathsf{iDK}_{\mathsf{msg}} &\leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, 0, \mathbf{q}_{\mathsf{msg}}) , \\ \mathsf{iDK}_{\mathsf{t}} &\leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, 0, \mathbf{Q}_{\mathsf{t}}) , \\ \left\{\mathsf{iDK}_{i,j} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}, (i,j), \mathbf{Q}_{i,j})\right\}_{(i,j)\in\bar{J}_f} . \end{aligned}$$

Finally, it outputs

$$\mathsf{DK}_f = \left(\mathsf{iDK}_{\mathsf{msg}}, \mathsf{iDK}_{\mathsf{t}}, \{\mathsf{iDK}_{i,j}\}_{(i,j)\in\bar{J}_f}\right) .$$

$\mathsf{Dec}(\mathsf{DK}_f, \{\mathsf{DK}_{L,i}\}_{i\in[n-1]}, \mathsf{CT}_L)$ takes as input a decryption key DK_f for a policy $f \in \mathcal{F}$, a set of decryption keys $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$ created with respect to attributes $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1} \in \{0,1\}^k$ and a label $L \in \mathcal{L}$, and a ciphertext CT_L created with respect to an attribute $\mathbf{x}_0 \in \{0,1\}^k$ and the same label L. If $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = 1$, return $\perp$. Otherwise, parse DK_f, $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$ and CT_L as above. The algorithm picks $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \in V$ such that $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in X_{\mathbf{v}}$ and $J_{\mathbf{v}} = J_f$. Subsequently, it recovers w_{msg}, $\mathbf{t}$ and $\{\mathbf{w}_{i,j}\}_{(i,j)\in\bar{J}_{\mathbf{v}}}$ by decrypting

$$\begin{aligned} w_{\mathsf{msg}} &\leftarrow \mathsf{iDec}(\mathsf{iDK}_{\mathsf{msg}}, \mathsf{iCT}^{\mathbf{v}}_{\mathsf{msg}}), & \left\{\mathsf{iCT}^{\mathbf{v}}_{i,j} \leftarrow \mathsf{idDec}(\mathsf{idDK}^{\mathbf{v}_i}_i, \mathsf{idCT}^{\mathbf{v}}_{i,j})\right\}_{(i,j)\in\bar{J}_{\mathbf{v}}, i>0}, \\ \mathbf{t} &\leftarrow \mathsf{iDec}(\mathsf{iDK}_{\mathsf{t}}, \mathsf{iCT}^{\mathbf{v}}_{\mathsf{msg}}), & \left\{\mathbf{w}_{i,j} \leftarrow \mathsf{iDec}(\mathsf{iDK}_{i,j}, \mathsf{iCT}^{\mathbf{v}}_{i,j})\right\}_{(i,j)\in\bar{J}_{\mathbf{v}}} . \end{aligned}$$

It evaluates the garbling

$$w_{\mathsf{out}} \leftarrow \mathsf{gEval}(\mathsf{gPP}, f, \mathbf{R}, \{\mathbf{x}_i[j]\}_{i,j\in\bar{J}_{\mathbf{v}}}, \{\mathbf{w}_{i,j}\}_{i,j\in\bar{J}_{\mathbf{v}}}, \mathbf{t}) ,$$

and outputs the decryption result

$$\mu' \leftarrow \begin{cases} 0 & \text{if } w_{\mathsf{msg}} - w_{\mathsf{out}} \in [-2^{-\varepsilon}p; 2^{-\varepsilon}p] , \\ 1 & \text{otherwise} . \end{cases}$$

Correctness. The construction is correct for a suitable choice of parameters.

Proposition 27. *Construction 26 is correct if*

- idFE *is correct,*
- NLG *is* $(B_{\mathsf{in}}, B_{\mathsf{out}})$*-correct for* $B_{\mathsf{out}} \le 2^{-\varepsilon}p$*, and*
- iFE *is* B_{ipfe}*-correct for some* $B_{\mathsf{ipfe}} \le B_{\mathsf{in}}$.

Proof. Let $f \in \mathcal{F}$ and $(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) \in \{0,1\}^{[0;n-1]\times[k]}$ such that $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = 0$. The correctness property of succinctly enumerable function classes guarantees the existence of a vector $\mathbf{v} \in V$ such that $J_{\mathbf{v}} = J_f$ and $(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in X_{\mathbf{v}}$. This implies that all idFE ciphertexts $\{\mathsf{idCT}^{\mathbf{v}}_{i,j}\}_{(i,j)\in\bar{J}_{\mathbf{v}}, i>0}$ and secret keys $\{\mathsf{idDK}_{i,j}\}_{(i,j)\in\bar{J}_{\mathbf{v}}, i>0}$

used in the first step of decryption are contained in CT_L and $\{\mathsf{DK}_{L,i}\}_{i\in[n-1]}$, respectively. From the correctness of idFE and iFE, it follows that

$$\begin{aligned} w_{\mathsf{msg}} &= \mathbf{p}^{\mathbf{v}}_{\mathsf{msg}}\mathbf{q}^\top_{\mathsf{msg}} + e_{\mathsf{out}} = \mathbf{r}^{\mathbf{v}}\mathbf{B}\mathbf{w}^\top_{\mathsf{out}} + \mu r_{\mathsf{msg}} + e_{\mathsf{out}} \,, \\ \mathbf{t} &= \mathbf{p}^{\mathbf{v}}_{\mathsf{msg}}\mathbf{Q}^\top_{\mathsf{t}} + \mathbf{e}_{\mathsf{t}} = \mathbf{r}^{\mathbf{v}}\mathbf{B}\mathbf{T} + \mathbf{e}_{\mathsf{t}} \,, \\ \mathbf{w}_{i,j} &= \mathbf{p}^{\mathbf{v}}_{i,j}\mathbf{Q}^\top_{i,j} + \mathbf{e}_{i,j} = \mathbf{r}^{\mathbf{v}}\mathbf{B}(\mathbf{W}^0_{i,j} + \overline{\mathbf{v}}_i[j]\mathbf{W}^\times_{i,j}) + \mathbf{e}_{i,j} \quad \text{for each } (i,j) \in \overline{J}_{\mathbf{v}} \,, \end{aligned}$$

where $\mathbf{r}^{\mathbf{v}}\mathbf{B}$ randomizes the garbling, and the errors from iFE decryption are bounded by $B_{\mathsf{ipfe}} \le B_{\mathsf{in}}$. By the $(B_{\mathsf{in}}, B_{\mathsf{out}})$-correctness of garbling, we have

$$w_{\mathsf{out}} - \mathbf{r}^{\mathbf{v}}\mathbf{B}\mathbf{w}^\top_{\mathsf{out}} + e_{\mathsf{out}} \in [-B_{\mathsf{out}}; B_{\mathsf{out}}] \subseteq [-2^{-\varepsilon}p; 2^{-\varepsilon}p] \,.$$

If $\mu = 0$, it always holds that $w_{\mathsf{msg}} - w_{\mathsf{out}} \in [-2^{-\varepsilon}p; 2^{-\varepsilon}p]$, which is never the case if $\mu = 1$ by the choice of r_{msg}. This concludes that Construction 26 is correct. □

Security. We prove very selective security.

Proposition 28. *Construction 26 is* stat*-secure if*

- iFE *is restricted-*σ_{ipfe}*-secure (as per Definition 15),*
- NLG *is* B_{short}*-short (as per Definition 9) and* GenNoise*-secure (as per Definition 10) such that the output of* GenNoise *is in* $[-B_{\mathsf{nlg}}; B_{\mathsf{nlg}}]$ *and* $2^{\varepsilon+6}B_{\mathsf{nlg}} \le \sigma_{\mathsf{ipfe}}$*, and*
- $\mathsf{LWE}_{n_{\mathsf{r}},\mathrm{poly}(\lambda),p,\sigma_{\mathsf{r}}}$ *(as per Definition 7) holds for some* $\sigma_{\mathsf{r}} \le 2^{-\varepsilon-6}/(\sqrt{\varepsilon}\cdot B_{\mathsf{short}})$.

When instantiating iFE and NLG with the evasive IFPE and noisy linear garbling from [HLL24], and idFE with the IBE from [ABB10], we obtain the following theorem.

Theorem 29. *Assuming the hardness of LWE with subexponential modulus-to-noise ratio and (public-coin) evasive LWE with public-coin samplers (in the variant of [HLL24]), there exists a very selectively secure MC-ABE for each succinctly enumerable policy classes* $\mathcal{F} \subseteq \mathcal{F}^{\mathsf{circ}}_{kn,d}$.

4.4 Proof of Security

Proof of Proposition 28. We consider a sequence of hybrid games $\mathsf{G}^0_0, \dots, \mathsf{G}^0_3, \mathsf{G}^1_3, \dots, \mathsf{G}^1_0$, where $\mathsf{G}^b_0 = \mathbf{Exp}^{\text{mc-abe-}b}_{\mathsf{ABE},\mathcal{A}}$ for $b \in \{0,1\}$. We argue $\mathsf{G}^b_{\alpha-1} \approx_c \mathsf{G}^b_\alpha$, for $\alpha \in [3]$, and $\mathsf{G}^0_3 \equiv \mathsf{G}^1_3$. Then the claim follows via a hybrid argument.

Game G^b_0**:** This is $\mathbf{Exp}^{\text{mc-abe-}b}_{\mathsf{ABE},\mathcal{A}}$. We let $q_v = |V|$ and parse $V = \{\mathbf{v}_\ell\}_{\ell\in[q_v]}$, where $\mathbf{v}_\ell = (\mathbf{v}_{\ell,1}, \dots, \mathbf{v}_{\ell,n-1})$ for each $\ell \in [q_v]$. For notational convenience, we also set $\mathbf{v}_{\ell,0} = \mathbf{x}_0$. For the reply to the challenge encryption query $\mathsf{QEnc}(L, \mathbf{x}_0, \mu^0, \mu^1)$, the challenger samples a sufficiently large message encoding scalar $r_{\mathsf{msg}} \xleftarrow{\$} \mathbb{Z}_p \setminus [-2\cdot 2^{-\varepsilon}p; 2\cdot 2^{-\varepsilon}p]$ and secret vectors $\mathbf{r}^{\mathbf{v}_\ell} \xleftarrow{\$} \mathbb{Z}^{n_r}_p$ for each $\ell \in [q_v]$. Then it sets

$$\left\{\mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{msg}} = (\mathbf{r}^{\mathbf{v}_\ell}, \mu^b r_{\mathsf{msg}}, \mathbf{0}_{n_r-1})\right\}_{\ell\in[q_v]} , \qquad \left\{\mathbf{p}^{\mathbf{v}_\ell}_{i,j} = (\mathbf{r}^{\mathbf{v}_\ell}, \overline{\mathbf{v}}_{\ell,i}[j]\cdot\mathbf{r}^{\mathbf{v}_\ell})\right\}_{\ell\in[q_v],(i,j)\in\overline{J}_{\mathbf{v}_\ell}} ,$$

and generates

$$\begin{aligned} &\left\{\mathsf{iCT}^{\mathbf{v}_\ell}_{\mathsf{msg}} \leftarrow \mathsf{iEnc}(\mathsf{iMPK}, (0,\mathbf{v}_\ell), \mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{msg}})\right\}_{\ell\in[q_v]} , \\ &\left\{\mathsf{iCT}^{\mathbf{v}_\ell}_{i,j} \leftarrow \mathsf{iEnc}(\mathsf{iMPK}, (i,j,\mathbf{v}_\ell), \mathbf{p}^{\mathbf{v}_\ell}_{i,j})\right\}_{\ell\in[q_v],(i,j)\in\overline{J}_{\mathbf{v}_\ell}} , \\ &\left\{\mathsf{idCT}^{\mathbf{v}_\ell}_{i,j} \leftarrow \mathsf{idEnc}(\mathsf{idMPK}_i, (L,\mathbf{v}_{\ell,i}), \mathsf{iCT}^{\mathbf{v}_\ell}_{i,j})\right\}_{\ell\in[q_v],(i,j)\in\overline{J}_{\mathbf{v}_\ell},i>0} . \end{aligned}$$

Finally, it sends the following challenge ciphertext CT_L to $\mathcal{A}$:

$$\mathsf{CT}_L = \left(\{\mathsf{iCT}^{\mathbf{v}_\ell}_{\mathsf{msg}}\}_{\ell\in[q_v]}, \{\mathsf{iCT}^{\mathbf{v}_\ell}_{0,j}\}_{\ell\in[q_v],(0,j)\in\overline{J}_{\mathbf{v}_\ell}}, \{\mathsf{idCT}^{\mathbf{v}_{\ell,i}}_{i,j}\}_{\ell\in[q_v],(i,j)\in\overline{J}_{\mathbf{v}_\ell},i>0}\right).$$

Game G_1^b**:** We recall the definition of the set $K_\ell \subseteq [0; n-1] \times [0; k]$ containing the indices of all ciphertexts $\mathsf{iCT}_{i,j}^{\mathbf{v}_\ell}$ known to the adversary:

$$K_\ell = \left\{(i,j) \in \overline{J}_{\mathbf{v}_\ell} : \left[i \in \mathcal{C} \cup \{0\}\right] \vee \left[\exists (i, L, \mathbf{x}_i) \in \mathcal{Q}_{\mathsf{akey}} \text{ s.t. } \mathbf{x}_i \in X_{\mathbf{v}_{\ell,i}}\right]\right\} \ .$$

This game is the same as G_0^b except that, for all $\ell \in [q_v]$ and $(i,j) \in \overline{J}_{\mathbf{v}_\ell}$, the challenger sets $\mathsf{iCT}_{i,j}^{\mathbf{v}_\ell} = \bot$ if $(i,j) \notin K_\ell$. We have $\mathsf{G}_0^b \approx_c \mathsf{G}_1^b$ from the security of idFE.

Game G_2^b**:** This game is the same as G_1^b except that the challenger samples the message encoding scalar $r_{\mathsf{msg}} \xleftarrow{\$} \mathbb{Z}_p$. Clearly $\mathsf{G}_1^b \approx_s \mathsf{G}_2^b$.

Game G_3^b**:** This game is the same as G_2^b except that the challenger replaces vectors inside iFE ciphertexts by random. Clearly, $\mathsf{G}_3^0 \equiv \mathsf{G}_3^1$. The rest of the proof is devoted to showing $\mathsf{G}_2^b \approx_c \mathsf{G}_3^b$.

We want to rely on the restricted-σ_{ipfe}-security of iFE. To do so, we consider the following evasive IPFE sampler $\mathcal{S}^b = (\mathcal{S}_{\mathsf{V}}, \mathcal{S}_{\mathsf{U}}^b)$ for an efficient and admissible adversary $\mathcal{A}$.

- $\mathcal{S}_{\mathsf{V}}(r_{\mathsf{pub}})$ parses $r_{\mathsf{pub}} = (r_{\mathcal{A}}, r_{\mathsf{gSetup}}, r_{\mathsf{gGenF}})$ and runs $\mathcal{A}(1^\lambda; r_{\mathcal{A}})$ to obtain the challenge $(L, \mathbf{x}_0, \mu^0, \mu^1)$, a set of attribute key queries $\mathcal{Q}_{\mathsf{akey}} = \{(i, L, \mathbf{x}_i)\}$ and a set of policy key queries $\mathcal{Q}_{\mathsf{pkey}} = \{f^{\ell'}\}^{\ell' \in [q_f]}$.

  The algorithm sets up the noisy linear garbling by running

  $$\mathsf{gPP} \leftarrow \mathsf{gSetup}(1^\lambda; r_{\mathsf{gSetup}}) \ ,$$

  and deterministically picks a suitable n_r. It runs

  $$\left(1^{n_{f^{\ell'}}}, \mathbf{w}_{\mathsf{out}}^{\ell'}, \{\mathbf{W}_{i,j}^{\ell',0}, \mathbf{W}_{i,j}^{\ell',\times}\}_{(i,j) \in \overline{J}_f}, \mathbf{T}^{\ell'}, \mathbf{R}^{\ell'}\right) \leftarrow \mathsf{gGenF}(\mathsf{gPP}, c \wedge f^{\ell'})$$

  and samples $\mathbf{B}^{\ell'} \xleftarrow{\$} \mathbb{Z}_p^{n_r \times n_{f^{\ell'}}}$ for all $\ell' \in [q_f]$. Suppose $\mathbf{W}_{i,j}^{\ell',0} \in \mathbb{Z}_p^{n_{f^{\ell'}} \times m_{i,j}^{\ell'}}$ and $\mathbf{T}^{\ell'} \in \mathbb{Z}_p^{n_{f^{\ell'}} \times m_{\mathsf{t}}^{\ell'}}$. Then the algorithm sets

  $$N = 2n_r \ , \qquad \mathsf{ID} = \bigcup_{\ell \in [q_v]} \overline{J}_{\mathbf{v}_\ell} \cup \{0\} \ ,$$

  $$(\mathsf{i}_{\mathsf{id}}, I_{\mathsf{id}}) = \begin{cases} \left([q_v], \ |\mathsf{i}_{\mathsf{id}}|\right) & \text{if } \mathsf{id} = 0 \\ \left(\{\ell \in [q_v] : \mathsf{id} \in \overline{J}_{\mathbf{v}_\ell}\}, \ |\mathsf{i}_{\mathsf{id}}|\right) & \text{if } \mathsf{id} \in \mathsf{ID} \setminus \{0\} \ , \end{cases}$$

  $$(\mathsf{j}_{\mathsf{id}}, J_{\mathsf{id}}) = \begin{cases} \left([q_f], \ \sum_{\ell' \in \mathsf{j}_{\mathsf{id}}} (m_{\mathsf{t}}^{\ell'} + 1)\right) & \text{if } \mathsf{id} = 0 \\ \left(\{\ell' \in [q_f] : \mathsf{id} \in \overline{J}_{f^{\ell'}}\}, \ \sum_{\ell' \in \mathsf{j}_{\mathsf{id}}} m_{\mathsf{id}}^{\ell'}\right) & \text{if } \mathsf{id} \in \mathsf{ID} \setminus \{0\} \ , \end{cases}$$

  $$\mathbf{V}_0 = \begin{bmatrix} \vdots & \\ — \ \mathbf{q}_{\mathsf{msg}}^{\ell'} \ — \\ — \ \mathbf{Q}_{\mathsf{t}}^{\ell'} \ — \\ \vdots & \end{bmatrix}_{\ell' \in \mathsf{j}_0} \quad \text{where} \quad \begin{aligned} \mathbf{q}_{\mathsf{msg}}^{\ell'} &= \left((\mathbf{B}^{\ell'} (\mathbf{w}_{\mathsf{out}}^{\ell'})^\top)^\top, 1, \mathbf{0}_{n_r - 1}\right) , \\ \mathbf{Q}_{\mathsf{t}}^{\ell'} &= \left((\mathbf{B}^{\ell'} \mathbf{T}^{\ell'})^\top, \mathbf{0}_{\star \times n_r}\right) , \end{aligned}$$

  $$\mathbf{V}_{\mathsf{id}} = \begin{bmatrix} \vdots & \\ — \ \mathbf{Q}_{\mathsf{id}}^{\ell'} \ — \\ \vdots & \end{bmatrix}_{\ell' \in \mathsf{j}_{\mathsf{id}}} \quad \text{where} \quad \begin{aligned} &\mathsf{id} \in \mathsf{ID} \setminus \{0\} \ , \\ &\mathbf{Q}_{\mathsf{id}}^{\ell'} = \left((\mathbf{B}^{\ell'} \mathbf{W}_{i,j}^{\ell',0})^\top, (\mathbf{B}^{\ell'} \mathbf{W}_{i,j}^{\ell',\times})^\top\right) , \end{aligned}$$

  and outputs $(1^N, \mathsf{ID}, \{1^{I_{\mathsf{id}}}, 1^{J_{\mathsf{id}}}, \mathbf{V}_{\mathsf{id}}\}_{\mathsf{id} \in \mathsf{ID}})$.

- $\mathcal{S}^b_{\mathsf{U}}(r_{\mathsf{pub}}; r_{\mathsf{pri}})$ starts by running the deterministic subroutine $\mathcal{S}^b_{\mathsf{A0}}(r_{\mathsf{pub}})$ which does the following. First, it reruns $\mathcal{A}(1^\lambda; r_{\mathcal{A}})$ to obtain the challenge query $(L, \mathbf{x}_0, \mu^0, \mu^1)$. Then the algorithm then picks matrices $\{\mathbf{A}_{\mathsf{id},0,\ell}\}_{\mathsf{id}\in\mathsf{ID},\ell\in\mathsf{i}_{\mathsf{id}}}$ such that

$$\mathbf{d}_0\mathbf{A}_{0,0,\ell} = \mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{msg}} = (\mathbf{r}^{\mathbf{v}_\ell}, \mu r_{\mathsf{msg}}, \mathbf{0}_{n_r-1})$$
$$\mathbf{d}_{i,j}\mathbf{A}_{(i,j),0,\ell} = \mathbf{p}^{\mathbf{v}_\ell}_{i,j} = (\mathbf{r}^{\mathbf{v}_\ell}, \bar{\mathbf{v}}_{\ell,i}[j]\cdot\mathbf{r}^{\mathbf{v}_\ell})$$

where $\mathbf{d}_0 = (r_{\mathsf{msg}}, \mathbf{r}^{\mathbf{v}_1}, \ldots, \mathbf{r}^{\mathbf{v}_{q_v}})$ and $n' = n_r q_v + 1$, and outputs the tuple $(1^{n'}, \{\mathbf{A}_{\mathsf{id},0,\ell}\}_{\mathsf{id}\in\mathsf{ID},\ell\in\mathsf{i}_{\mathsf{id}}})$. After completing $\mathcal{S}^b_{\mathsf{A0}}$, the algorithm $\mathcal{S}^b_{\mathsf{U}}$ sample $\mathbf{d}_0 \xleftarrow{\$} \mathbb{Z}_p^{n'}$ using r_{pri} and outputs

$$\mathbf{U}_{\mathsf{id}} = \begin{bmatrix} \vdots \\ \text{—} \quad \mathbf{d}_0\mathbf{A}_{\mathsf{id},0,\ell} \quad \text{—} \\ \vdots \end{bmatrix}_{\ell\in\mathsf{i}_{\mathsf{id}}} \qquad \text{for all } \mathsf{id}\in\mathsf{ID}\,.$$

We observe that G^b_2 and G^b_3 correspond to $\mathrm{IPFEsec}^{\mathcal{S}^b}_{\mathsf{pre}}$ and $\mathrm{IPFEsec}^{\mathcal{S}^b}_{\mathsf{post}}$, respectively. It therefore follows $\mathsf{G}^b_2 \approx_c \mathsf{G}^b_3$ from the restricted-σ_{ipfe}-security of iFE and the following claim for $b\in\{0,1\}$.

Claim 30. *$\mathcal{S}^b$ has pseudorandom noisy inner products, i.e., $\mathrm{IPFEsec}^{\mathcal{S}^b}_{\mathsf{pre}}$ holds.*

This completes the proof of the proposition. □

Proof of Claim 30. We consider a sequence of hybrids $\widehat{\mathsf{G}}_0, \ldots, \widehat{\mathsf{G}}_5$, where $\widehat{\mathsf{G}}_0$ and $\widehat{\mathsf{G}}_5$ are the left and right distribution of $\mathrm{IPFEsec}^{\mathcal{S}^b}_{\mathsf{pre}}$, respectively. We argue $\widehat{\mathsf{G}}_{\alpha-1} \approx_c \widehat{\mathsf{G}}_\alpha$, for $\alpha\in[5]$. Then the claim follows via a hybrid argument. Modifications between consecutive games are highlighted using boxes.

Game $\widehat{\mathsf{G}}_0$: This is the L.H.S. distribution of $\mathrm{IPFEsec}^{\mathcal{S}^b}_{\mathsf{pre}}$ (Definition 15). We recall that $r_{\mathsf{pub}} = (r_{\mathcal{A}}, r_{\mathsf{gSetup}}, r_{\mathsf{gGenF}})$ and the inner products are of the form

$$\begin{aligned}
w^{\ell,\ell'}_{\mathsf{msg}} &= \mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{msg}}(\mathbf{q}^{\ell'}_{\mathsf{msg}})^\top + e^{\ell,\ell'}_{\mathsf{out}} \\
&= \mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'}(\mathbf{w}^{\ell'}_{\mathsf{out}})^\top + \mu r_{\mathsf{msg}} + e^{\ell,\ell'}_{\mathsf{out}}\,, && \text{for } \ell\in\mathsf{i}_0, \ell'\in\mathsf{j}_0 \\
\mathbf{t}^{\ell,\ell'} &= \mathbf{p}^{\mathbf{v}_\ell}_{\mathsf{msg}}(\mathbf{Q}^{\ell'}_{\mathsf{t}})^\top + \mathbf{e}^{\ell,\ell'}_{\mathsf{t}} \\
&= \mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'}\mathbf{T}^{\ell'} + \mathbf{e}^{\ell,\ell'}_{\mathsf{t}}\,, && \text{for } \ell\in\mathsf{i}_0, \ell'\in\mathsf{j}_0 \\
\mathbf{w}^{\ell,\ell'}_{i,j} &= \mathbf{p}^{\mathbf{v}_\ell}_{i,j}(\mathbf{Q}^{\ell'}_{i,j})^\top + \mathbf{e}^{\ell,\ell'}_{i,j} \\
&= \mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'}(\mathbf{W}^{\ell',0}_{i,j} + \bar{\mathbf{v}}_{\ell,i}[j]\mathbf{W}^{\ell',\times}_{i,j}) + \mathbf{e}^{\ell,\ell'}_{i,j}\,, && \text{for } \ell\in\mathsf{i}_{i,j}, \ell'\in\mathsf{j}_{i,j}
\end{aligned}$$

where $(i,j)\in\mathsf{ID}\setminus\{0\}$ and the entries of $\mathbf{e}$'s are independent $\mathcal{D}_{\mathbb{Z},\sigma_{\mathsf{ipfe}}}$.

Game $\widehat{\mathsf{G}}_1$: This is the same as $\widehat{\mathsf{G}}_0$ except that there are additional small noises attached to $\{\mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'}\}_{\ell\in[q_v]}$. For all $\ell\in[q_v]$, $\ell'\in[q_f]$, sample $\mathbf{e}^{\ell,\ell'}_{\mathsf{r}} \leftarrow \mathcal{D}^{n_{f^{\ell'}}}_{\mathbb{Z},\sigma_{\mathsf{r}},\le\sigma_{\mathsf{r}}\sqrt{\varepsilon}}$ and set the inner products to

$$\begin{aligned}
w^{\ell,\ell'}_{\mathsf{msg}} &= (\mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'} + \boxed{\mathbf{e}^{\ell,\ell'}_{\mathsf{r}}})(\mathbf{w}^{\ell'}_{\mathsf{out}})^\top + \mu r_{\mathsf{msg}} + e^{\ell,\ell'}_{\mathsf{out}}\,, \\
\mathbf{t}^{\ell,\ell'} &= (\mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'} + \boxed{\mathbf{e}^{\ell,\ell'}_{\mathsf{r}}})\mathbf{T}^{\ell'} + \mathbf{e}^{\ell,\ell'}_{\mathsf{t}}\,, \\
\mathbf{w}^{\ell,\ell'}_{i,j} &= (\mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'} + \boxed{\mathbf{e}^{\ell,\ell'}_{\mathsf{r}}})(\mathbf{W}^{\ell',0}_{i,j} + \bar{\mathbf{v}}_{\ell,i}[j]\mathbf{W}^{\ell',\times}_{i,j}) + \mathbf{e}^{\ell,\ell'}_{i,j}\,.
\end{aligned}$$

The introduced errors are bounded by

$$B_{\mathsf{short}}\cdot\sigma_{\mathsf{r}}\sqrt{\varepsilon} \le 2^{-\varepsilon-6}\sigma_{\mathsf{ipfe}}\,,$$

so they are flooded (Lemma 6) by $\{e^{\ell,\ell'}_{\mathsf{out}}, \mathbf{e}^{\ell,\ell'}_{\mathsf{t}}, \mathbf{e}^{\ell,\ell'}_{i,j}\}^{\ell,\ell'}_{i,j}$ which implies that $\widehat{\mathsf{G}}_0 \approx_s \widehat{\mathsf{G}}_1$.

Game $\widehat{\mathsf{G}}_2$**:** This is the same as $\widehat{\mathsf{G}}_1$ except that the $\mathbf{e}_{\mathsf{r}}^{\ell,\ell'}$'s are no longer truncated, i.e., $\mathbf{e}_{\mathsf{r},\ell} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_{\mathsf{r}}}^{n_{f^{\ell'}}}$. We have $\widehat{\mathsf{G}}_1 \approx_s \widehat{\mathsf{G}}_2$ by Lemma 5.

Game $\widehat{\mathsf{G}}_3$**:** This is the same as $\widehat{\mathsf{G}}_2$ except that the garblings use uniform randomness. For all $\ell \in [q_v]$, $\ell' \in [q_f]$, sample $\mathbf{s}^{\ell,\ell'} \xleftarrow{\$} \mathbb{Z}_p^{n_{f^{\ell'}}}$ and set the inner products to

$$\begin{aligned} w_{\mathsf{msg}}^{\ell,\ell'} &= \mathbf{s}^{\ell,\ell'}(\mathbf{w}_{\mathsf{out}}^{\ell'})^\top + \mu r_{\mathsf{msg}} + e_{\mathsf{out}}^{\ell,\ell'} , \\ \mathbf{t}^{\ell,\ell'} &= \mathbf{s}^{\ell,\ell'}\mathbf{T}^{\ell'} + \mathbf{e}_{\mathsf{t}}^{\ell,\ell'} , \\ \mathbf{w}_{i,j}^{\ell,\ell'} &= \mathbf{s}^{\ell,\ell'}(\mathbf{W}_{i,j}^{\ell',0} + \overline{\mathbf{v}}_{\ell,i}[j]\mathbf{W}_{i,j}^{\ell',\times}) + \mathbf{e}_{i,j}^{\ell,\ell'} , \end{aligned}$$

By $[q_v]$ hybrids of applying $\mathsf{LWE}_{n_{\mathsf{r}},\mathrm{poly}(\lambda),p,\sigma_{\mathsf{r}}}$ over $\{\mathbf{v}_\ell\}_{\ell\in[q_v]}$ with public matrix $(\mathbf{B}^1,\dots,\mathbf{B}^{q_f})$, we have

$$\{\mathbf{r}^{\mathbf{v}_\ell}\mathbf{B}^{\ell'} + \mathbf{e}_{\mathsf{r}}^{\ell,\ell'}\}^{\ell,\ell'} \approx_c \{\mathbf{s}^{\mathbf{v}_\ell}\}^{\ell,\ell'} ,$$

which implies $\widehat{\mathsf{G}}_2 \approx_s \widehat{\mathsf{G}}_3$.

Game $\widehat{\mathsf{G}}_4$**:** This is the same as $\widehat{\mathsf{G}}_3$ except that there are additional noises generated by $\mathsf{GenNoise}$ attached to the garblings. For all $\ell \in [q_v]$, $\ell' \in [q_f]$, run

$$\begin{pmatrix} \overline{e}_{\mathsf{out}}^{\ell,\ell'}, \overline{\mathbf{e}}_{\mathsf{t}}^{\ell,\ell'} \\ (\overline{\mathbf{e}}_{i,j}^{\ell,\ell'})_{(i,j)\in\overline{J}_{f^{\ell'}}} \end{pmatrix} \leftarrow \mathsf{GenNoise}\begin{pmatrix} \mathsf{gPP}, f^{\ell'}, \mathbf{w}_{\mathsf{out}}^{\ell'}, \{\mathbf{W}_{i,j}^{\ell',0}, \mathbf{W}_{i,j}^{\ell',\times}\}_{(i,j)\in\overline{J}_{f^{\ell'}}} \\ \mathbf{T}^{\ell'}, \mathbf{R}^{\ell'}, (\mathbf{v}_{\ell,i}[j])_{(i,j)\in\overline{J}_{f^{\ell'}}} \end{pmatrix} ,$$

where $\mathsf{GenNoise}$ can interpret $\star$-coordinates of $\mathbf{v}^\ell$ as 0 since these coordinates are irrelevant for $f^{\ell'}$. Then set the inner products to

$$\begin{aligned} w_{\mathsf{msg}}^{\ell,\ell'} &= \mathbf{s}^{\ell,\ell'}(\mathbf{w}_{\mathsf{out}}^{\ell'})^\top + \overline{e}_{\mathsf{out}}^{\ell,\ell'} + \mu r_{\mathsf{msg}} + e_{\mathsf{out}}^{\ell,\ell'} , \\ \mathbf{t}^{\ell,\ell'} &= \mathbf{s}^{\ell,\ell'}\mathbf{T}^{\ell'} + \overline{\mathbf{e}}_{\mathsf{t}}^{\ell,\ell'} + \mathbf{e}_{\mathsf{t}}^{\ell,\ell'} , \\ \mathbf{w}_{i,j}^{\ell,\ell'} &= \mathbf{s}^{\ell,\ell'}(\mathbf{W}_{i,j}^{\ell',0} + \overline{\mathbf{e}}_{i,j}^{\ell,\ell'} + \overline{\mathbf{v}}_{\ell,i}[j]\mathbf{W}_{i,j}^{\ell',\times}) + \mathbf{e}_{i,j}^{\ell,\ell'} . \end{aligned}$$

The introduced $\overline{e},\overline{\mathbf{e}}$'s are bounded by $B_{\mathsf{NLG}} \le 2^{-\varepsilon-6}\sigma_{\mathsf{ipfe}}$, so they are flooded (Lemma 6) by their counterparts $e,\mathbf{e}$ which implies $\widehat{\mathsf{G}}_3 \approx_s \widehat{\mathsf{G}}_4$.

Game $\widehat{\mathsf{G}}_5$**:** This is the same as $\widehat{\mathsf{G}}_4$ except that all inner products are replaced by random, i.e., the right distribution of $\mathrm{IPFEsec}_{\mathsf{pre}}^{\mathcal{S}^b}$. We note that in $\widehat{\mathsf{G}}_4$ all inner products are well-randomized garblings (by $\mathbf{s},\overline{e},\overline{\mathbf{e}}$) plus independent noises ($e,\mathbf{e}$'s). Therefore, by applying $\mathsf{GenNoise}$-security of NLG in a sequence of hybrids over all $(\ell,\ell') \in [q_v]\times[q_f]$, we conclude that $\widehat{\mathsf{G}}_4 \approx_c \widehat{\mathsf{G}}_5$.

□

5 MC-ABE for Conjunction Predicates

In this section, we present our construction of secret-key MC-ABE for conjunctions of a predicate P.

Definition 31 (Conjunctions of Predicates)**.** Let $\{P_\lambda\}_{\lambda\in\mathbb{N}}$, $\{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be sequences of predicates, ciphertext attribute spaces and key attribute spaces, respectively. For $n\in\mathbb{N}$, we denote $\{P_{\lambda,n}\}_{\lambda\in\mathbb{N}}$ the sequence of conjunction predicates, where $P_{\lambda,n}\colon (\mathcal{X}_\lambda\cup\{\star\})^{[0;n-1]} \times \mathcal{Y}_\lambda^{[0;n-1]} \to \{0,1\}$ is defined as

$$P_{\lambda,n}\big((x_0,\dots,x_{n-1}),(y_0,\dots,y_{n-1})\big) = P_\lambda^\star(x_0,y_0)\wedge\cdots\wedge P_\lambda^\star(x_{n-1},y_{n-1}) ,$$

where $P_\lambda^\star(x,y) = P_\lambda(x,y)$ if $x\in\mathcal{X}_\lambda$ and $P_\lambda^\star(\star,y) = 0$ for each $y\in\mathcal{Y}_\lambda$.

5.1 Construction

Construction 32 (MC-ABE for Conjunction Predicates)**.** Our construction uses the following ingredients:

- an AB-IPFE scheme $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iDec})$ for a predicate $P^\star\colon (\mathcal{X} \cup \{\star\}) \times \mathcal{Y} \to \{0,1\}$ which provides

  either $\boxed{\text{function-hiding security based on a pairing group } \mathbb{G}}$,

  or $\boxed{\text{standard security against bounded collusions}}$,

- a family of PRFs $\{\mathsf{PRF}_K\}_{K\in\mathcal{K}}$ where $\mathsf{PRF}_K\colon (\mathcal{L}\times[\kappa]) \to \mathbb{Z}_p$ for each key $K \in \mathcal{K}$, and
- a hash function $\boxed{H\colon \mathbb{G}_\mathsf{t} \to \mathcal{M}}$ $\boxed{H\colon \mathbb{Z}_p \to \mathcal{M}}$ for $\mathcal{M} = \{0,1\}^{\mathrm{poly}(\lambda)}$.

If iFE provides function-hiding security, we let κ be the parameter of the MDDH_κ assumption which we want to rely on during the security proof. If iFE provides standard security against Q key queries, then we choose the parameters κ and τ according to Lemma 20. For notational convenience, we assume that iFE is a secret-key AB-IPFE scheme. But this is clearly without loss of generality; if iFE is a public-key scheme, we may simply concatenate iMPK and iMSK and pass this concatenation to both iEnc and iKeyGen.

The MC-ABE scheme ABE for conjunctions of P with message space $\mathcal{M}$ and label space $\mathcal{L}$ works as follows:

$\mathsf{Setup}(1^\lambda)$ takes as input 1^λ and samples $K_i \xleftarrow{\$} \mathcal{K}$ for $i \in [n-1]$, $i < j$. Then the algorithm runs $\mathsf{iMSK}_i \leftarrow \mathsf{iSetup}(1^\lambda, \boxed{1^{\kappa+2}}\ \boxed{1^{\kappa+1}})$ for $i \in [0; n-1]$ and outputs $\mathsf{SK}_0 = (\mathsf{iMSK}_0, \{K_i\}_{i\in[n-1]})$, $\{\mathsf{SK}_i = (\mathsf{iMSK}_i, K_i)\}_{i\in[n-1]}$ and $\mathsf{MSK} = \{\mathsf{iMSK}_i\}_{i\in[0;n-1]}$.

$\mathsf{Enc}(\mathsf{SK}_0, L, x_0, \boldsymbol{\mu})$ takes SK_0, a label $L \in \mathcal{L}$, an attribute $x_0 \in \mathcal{X}$ and a message $\boldsymbol{\mu} \in \mathcal{M}$ as input and sets $\mathbf{t}_{L,0} \in \mathbb{Z}_p^\kappa$ component-wise as

$$\mathbf{t}_{L,0}[\nu] = -\sum_{i\in[n-1]} \mathsf{PRF}_{K_i}(L,\nu) \quad \text{for } \nu \in [\kappa] \ ,$$

samples $r \xleftarrow{\$} \mathbb{Z}_p$, runs

$\boxed{\begin{array}{l}\mathbf{c} = \boldsymbol{\mu} \oplus H([\![r]\!]_\mathsf{t}) \\ \mathsf{iCT}_{L,0} \leftarrow \mathsf{iEnc}\big(\mathsf{iMSK}_0, x_0, [\![(\mathbf{t}_{L,0}, r, 0)]\!]_1\big)\end{array}}$ $\boxed{\begin{array}{l}\mathbf{c} = \boldsymbol{\mu} \oplus H(r) \\ \mathsf{iCT}_{L,0} \leftarrow \mathsf{iEnc}\big(\mathsf{iMSK}_0, x_0, (\mathbf{t}_{L,0}, r)\big)\end{array}}$

and outputs the ciphertext $\mathsf{CT}_L = (\mathsf{iCT}_{L,0}, \mathbf{c})$.

$\mathsf{AKeyGen}(\mathsf{SK}_i, L, x_i)$ takes as input SK_i for some $i \in [n-1]$, a label $L \in \mathcal{L}$ and an attribute $x_i \in \mathcal{X}$, and defines $\mathbf{t}_{L,i} \in \mathbb{Z}_p^\kappa$ component-wise as $\mathbf{t}_{L,i}[\nu] = \mathsf{PRF}_{K_i}(L,\nu)$ for $\nu \in [\kappa]$. Then the algorithm generates an iFE ciphertext

$\boxed{\mathsf{iCT}_{L,i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, [\![(\mathbf{t}_{L,i}, 0, 0)]\!]_1)}$ $\boxed{\mathsf{iCT}_{L,i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, (\mathbf{t}_{L,i}, 0))}$,

and outputs $\mathsf{DK}_{L,i} = \mathsf{iCT}_{L,i}$.

$\mathsf{PKeyGen}(\mathsf{MSK}, \mathbf{y})$ takes as input MSK and attributes $\mathbf{y} \in \mathcal{Y}^{[0;n-1]}$, and samples

$\boxed{\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^\kappa}$ $\boxed{\Delta \leftarrow \mathsf{SamplerSet}(\kappa, Q, \tau)\ ,\quad \mathbf{e}_\Delta = \textstyle\sum_{\nu\in\Delta} \mathbf{e}_\nu}$,

where $\mathbf{e}_\nu$ denotes the ν-th unit vector in $\mathbb{Z}_p^\kappa$. Then the algorithm generates

$\boxed{\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, \mathbf{y}[i], [\![(\mathbf{s}, 1, 0)]\!]_2)}$ $\boxed{\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, \mathbf{y}[i], (\mathbf{e}_\Delta, 1))}$,

for all $i \in [0; n-1]$, and outputs $\mathsf{DK}_\mathbf{y} = \{\mathsf{iDK}_i\}_{i\in[0;n-1]}$.

$\mathsf{Dec}(\mathsf{DK}_\mathbf{y}, \{\mathsf{DK}_{L,i}\}_{i\in[n-1]}, \mathsf{CT}_L)$ takes as input a policy decryption key $\mathsf{DK}_\mathbf{y} = \{\mathsf{iDK}_i\}_{i\in[0;n-1]}$, a set of attribute decryption keys $\{\mathsf{DK}_{L,i} = \mathsf{iCT}_{L,i}\}_{i\in[n-1]}$ and a ciphertext $\mathsf{CT}_L = (\mathsf{iCT}_{L,0}, \mathbf{c})$. For each $i \in [0; n-1]$, the algorithm runs $\theta_i \leftarrow \mathsf{iDec}(\mathsf{iDK}_i, \mathsf{iCT}_{L,i})$ and immediately outputs $\perp$ if $\theta_i = \perp$, Otherwise, it computes $\theta = \sum_{i\in[0;n-1]} \theta_i$ and outputs $\boldsymbol{\mu} = \mathbf{c} \oplus H(\theta)$.

Correctness. This readily follows from the correctness of iFE and the fact that $\sum_{i\in[0;n-1]} \mathbf{t}_{L,i} = \mathbf{0}$ for each $L \in \mathcal{L}$.

Security. The security naturally depends on the security of the employed AB-IPFE scheme.

Proposition 33. *If* iFE *provides function-hiding* sel*-security and the* MDDH_κ *assumption holds in* $\mathbb{G}_2$*, then the resulting MC-ABE scheme* ABE *in Construction 32 is* sel*-secure in the random oracle model. If* iFE *is* stat*-secure against Q key queries, then so is* ABE *relying additionally on random oracles.*

The restriction to stat security if iFE is not function-hiding comes from the fact that we cannot hardcode decryption results in the keys during the security proof but only in the ciphertexts. This essentially forces us to flip the usual proof technique, leading to selective key queries instead of selective encryption queries. Technically, this would allow for adaptive encryption queries if the employed iFE scheme did. But the only LWE-based instantiation of iFE in the literature that we are aware of [LLW21] requires selective encryption queries. Therefore, we prove security only in the static setting. The full proof of Proposition 33 can be found in Sections 5.2 and 5.3.

When instantiating iFE in Construction 32 with either the AB-IPFE implied by [ATY23] or the AB-IPFE of [LLW21], we obtain the following theorem.

Theorem 34.
- *Assuming MDDH on pairings, there exists a selectively secure MC-ABE for conjunctions of ABPs in the random oracle model.*
- *Assuming the hardness of LWE with subexponential modulus-to-noise ratio, there exists a bounded-collusion statically secure MC-ABE for conjunctions of bounded-depth circuits in the random oracle model.*

5.2 Proof of Security – Function-Hiding Case

The proof makes use of the following technical lemma.

Lemma 35. *Let $q = q(\lambda)$ be a polynomial, S a finite set of superpolynomial size and $H\colon S \to \{0,1\}^{\mathrm{poly}(\lambda)}$ a hash function modelled as a random oracle. Then, for $s, r_1, \dots, r_q \xleftarrow{\$} S$ and $\mathbf{y}_1, \dots, \mathbf{y}_q \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$, the following distributions are computationally indistinguishable:*

$$\{(s_j, H(r_j))\}_{j\in[q]} \approx_c \{(s_j, \mathbf{y}_j)\}_{j\in[q]} \ ,$$

where $s_j := s + r_j$ for all $i \in [q]$.

Proof of Lemma 35. Let $\mathcal{A}$ be a PPT adversary running in time $t = t(\lambda)$. By the superpolynomial order of S, we have $r_j \neq r_{j'}$ for all distinct $j, j' \in [q]$, $j \neq j'$, with overwhelming probability. Therefore, by the definition of a random oracle, the hash values $\{H(r_j)\}_{i\in[q]}$ are all independent and, for a fixed r_j, $H(r_j)$ can be distinguished from a random element $\mathbf{y}_j \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$ if and only if r_j is known. We hence need to bound the probability that the adversary learns one of the values in $\{r_1, \dots, r_q\}$. For a single query to the random oracle, the probability to have asked an r_j is bounded by $q/|S|$. This allows the adversary to exclude up to q^2 points using the constraints $\{s + r_j = s_j\}_{j\in[q]}$. Thus, after t queries, the probability to have queried the random oracle on one of the r_j is at most $qt/(|S| - q^2 t)$ which is negligible in λ. □

Proof of Proposition 33 – Function-Hiding Case. Let q_L denote the number of different labels being input to a QEnc query and let $\mathcal{Q}_L = \{L_1, \dots, L_{q_L}\}$ be the set of these labels. The proposition is proven via a sequence of hybrid games $\mathsf{G}_0^0, \dots, \mathsf{G}_{q_L}^0, \mathsf{G}_{q_L}^1, \dots, \mathsf{G}_0^1$, where $\mathsf{G}_0^b \approx_c \mathbf{Exp}^{\text{mc-abe-}b}_{\mathsf{ABE},\mathcal{A}}$ for $b \in \{0,1\}$. Modifications between consecutive games are highlighted using boxes. In G_0^b, the challenger behaves as follows. Initially, sample $\widehat{\mathbf{t}}_{L,i} \xleftarrow{\$} \mathbb{Z}_p^k$ for each $L \in \mathcal{Q}_L$ and $i \in [n-1] \setminus \mathcal{C}$.

- For each encryption query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$, define $\mathbf{t}_{L,0} \in \mathbb{Z}_p^\kappa$ component-wise as

$$\mathbf{t}_{L,0}[\nu] = -\sum_{i\in\mathcal{C}} \mathsf{PRF}_{K_i}(L,\nu) - \sum_{i\in[n-1]\setminus\mathcal{C}} \widehat{\mathbf{t}}_{L,i}[\nu] \quad \text{for } \nu \in [\kappa] \ ,$$

  sample $r \xleftarrow{\$} \mathbb{Z}_p$, run $\mathsf{iCT}_{L,0} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, [\![(\mathbf{t}_{L,0}, r, 0)]\!]_1)$ and return $\mathsf{CT}_L = (\mathsf{iCT}_{L,0}, \mathbf{c} = \boldsymbol{\mu}^b \oplus H([\![r]\!]_t))$.

- Upon receiving an attribute key query $\mathsf{QAKeyGen}(i, L, x_i)$, define $\mathbf{t}_{L,i} \in \mathbb{Z}_p^\kappa$ component-wise for $\nu \in [\kappa]$ as

$$\mathbf{t}_{L,i}[\nu] = \begin{cases} \widehat{\mathbf{t}}_{L,i}[\nu] & \text{if } L \in \mathcal{Q}_L \text{ and } i \in [n-1] \setminus \mathcal{C} \,, \\ \mathsf{PRF}_{K_i}(L, \nu) & \text{otherwise} \,, \end{cases}$$

  run $\mathsf{iCT}_{L,i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, [\![(\mathbf{t}_{L,i}, 0, 0)]\!]_1)$ and return $\mathsf{DK}_{L,i} = \mathsf{iCT}_{L,i}$.

- Upon receiving a policy key query $\mathsf{QPKeyGen}(\mathbf{y})$, sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^\kappa$, generate $\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, \mathbf{y}[i], [\![(\mathbf{s}, 1, 0)]\!]_2)$ for $i \in [0; n-1]$ and return $\mathsf{DK}_\mathbf{y} = \{\mathsf{iDK}_i\}_{i \in [0;n-1]}$.

It is not hard to see that $\mathbf{Exp}^{\mathsf{mc\text{-}abe}\text{-}b}_{\mathsf{ABE},\mathcal{A}} \approx_c \mathsf{G}_0^b$ by the security of PRF. The game G_ℓ^b, for $\ell \in [q_L]$ and $b \in \{0,1\}$, is the same as G_0^b except that for encryption queries $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $L \in \{L_1, \ldots, L_\ell\}$, the challenger returns $\mathsf{CT}_L = (\mathsf{iCT}_{L,0}, \mathbf{c} = \boldsymbol{\mu}^0 \oplus H([\![r]\!]_\mathsf{t}))$. Below we proof the following claim for all $\ell \in [q_L]$ and $b \in \{0,1\}$

Claim 36. *If* iFE *is function-hiding and* MDDH_κ *holds in* $\mathbb{G}_2$*, then* $\mathsf{G}_{\ell-1}^b \approx_c \mathsf{G}_\ell^b$.

Furthermore, we observe that $\mathsf{G}_{q_L}^0 \equiv \mathsf{G}_{q_L}^1$ as the challenger's output is independent of the bit b in these games. This concludes the proof of the proposition. □

Proof of Claim 36. We start with an analysis of the admissibility condition. For each policy query $\mathsf{QPKeyGen}(\mathbf{y})$, we can distinguish the following two cases:

1. all queries $\mathsf{QEnc}(L_\ell, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ satisfy $\boldsymbol{\mu}^0 = \boldsymbol{\mu}^1$ or $P^\star(x_0, \mathbf{y}[0]) \neq 0$, or
2. there exists at least one query $\mathsf{QEnc}(L_\ell, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$ and $P^\star(x_0, \mathbf{y}[0]) = 0$.

In case 2, the admissibility condition implies that there exists at least one index $i_\mathbf{y} \in ([n-1] \setminus \mathcal{C})$ such that it holds $P^\star(x, \mathbf{y}[i_\mathbf{y}]) \neq 0$ for all queries of the form $\mathsf{QAKeyGen}(i_\mathbf{y}, L_\ell, x)$.

The proof considers a sequence of hybrids $\widehat{\mathsf{G}}_0^0, \ldots, \widehat{\mathsf{G}}_5^0, \widehat{\mathsf{G}}_5^1, \ldots, \widehat{\mathsf{G}}_0^1$, where $\widehat{\mathsf{G}}_0^0 = \mathsf{G}_{\ell-1}^b$ and $\widehat{\mathsf{G}}_0^1 = \mathsf{G}_\ell^b$. We argue $\widehat{\mathsf{G}}_{\alpha-1}^{b'} \approx_c \widehat{\mathsf{G}}_\alpha^{b'}$, for $\alpha \in [5]$ and $b' \in \{0,1\}$, as well as $\mathsf{G}_5^0 \equiv \mathsf{G}_5^1$. Then the claim follows via a hybrid argument.

Game $\widehat{\mathsf{G}}_0^{b'}$: This is $\mathsf{G}_{\ell-1+b'}^b$.

Game $\widehat{\mathsf{G}}_1^{b'}$: This is the same as $\widehat{\mathsf{G}}_0^{b'}$ except that we hardcode $\{\mathbf{t}_{L_\ell,i}\}_{i \in [0;n-1]}$ in the policy decryption keys in the following way:

- for the response to an encryption query $\mathsf{QEnc}(L_\ell, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$, the challenger computes

$$\mathsf{iCT}_{L_\ell,0} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, [\![(\mathbf{0}, r, 1)]\!]_1) \,,$$

- for the response to an attribute key query $\mathsf{QAKeyGen}(i, L_\ell, x_i)$, the challenger computes

$$\mathsf{iCT}_{L_\ell,i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, [\![(\mathbf{0}, 0, 1)]\!]_1) \,,$$

- for the response to a policy key query $\mathsf{QPKeyGen}(\mathbf{y})$, the challenger computes,

$$\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, \mathbf{y}[i], [\![(\mathbf{s}, 1, \mathbf{s} \cdot \mathbf{t}_{L_\ell,i}^\top)]\!]_2) \qquad \text{for } i \in [0; n-1] \,.$$

  It is not hard to see that the inner products between vectors encoded in iFE ciphertexts and decryption key do not change. Thus, we have $\widehat{\mathsf{G}}_0^{b'} \approx_c \widehat{\mathsf{G}}_1^{b'}$ under the function-hiding security of iFE.

Game $\widehat{\mathsf{G}}_2^{b'}$: This is the same as $\widehat{\mathsf{G}}_1^{b'}$ except that, upon receiving a policy key query $\mathsf{QPKeyGen}(\mathbf{y})$ of type 2, the challenger computes

$$\mathsf{iDK}_{i_\mathbf{y}} \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_{i_\mathbf{y}}, \mathbf{y}[i_\mathbf{y}], [\![(\mathbf{s}, 1, \mathbf{0})]\!]_2)$$

We have $\widehat{\mathsf{G}}_0^{b'} \approx_c \widehat{\mathsf{G}}_1^{b'}$ under the security of iFE since $P^\star(x, \mathbf{y}[i_\mathbf{y}]) \neq 0$ for all attribute key queries of the form $\mathsf{QAKeyGen}(i_\mathbf{y}, L_\ell, x)$.

Game $\widehat{\mathsf{G}}_3^{b'}$: This is the same as $\widehat{\mathsf{G}}_2^{b'}$ except that, upon receiving a policy key query $\mathsf{QPKeyGen}(\mathbf{y})$ of type 2, the challenger samples $s_\ell \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\mathsf{iDK}_0 \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_0, \mathbf{y}[0], [\![(\mathbf{s}, 1, \mathbf{s}\cdot\mathbf{t}_{L_\ell,0}^\top + \boxed{s_\ell})]\!]_2) \ .$$

We have $\widehat{\mathsf{G}}_2^{b'} \approx_c \widehat{\mathsf{G}}_3^{b'}$ since $\mathbf{t}_{L_\ell,0}$ contains the term $\widehat{\mathbf{t}}_{L_\ell,i_\mathbf{y}} \xleftarrow{\$} \mathbb{Z}_p$, and the MDDH_κ assumption in $\mathbb{G}_2$ implies that $[\![\mathbf{s}\cdot\widehat{\mathbf{t}}_{\ell,0}^\top]\!]_2 \approx_c [\![\mathbf{s}\cdot\widehat{\mathbf{t}}_{\ell,0}^\top + s_\ell]\!]_2$.

Game $\widehat{\mathsf{G}}_4^{b'}$: This is the same as $\widehat{\mathsf{G}}_3^{b'}$ except that we move the random scalars s_ℓ into the ciphertexts:

- for the response to an encryption query $\mathsf{QEnc}(L_\ell, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ satisfying $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$, the challenger computes

$$\mathsf{iCT}_{L_\ell,0} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, [\![(\mathbf{0}, r + \boxed{s_\ell}, 1)]\!]_1) \ ,$$

- for the response to a policy key query $\mathsf{QPKeyGen}(\mathbf{y})$, the challenger computes

$$\mathsf{iDK}_0 \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_0, \mathbf{y}[0], [\![(\mathbf{s}, 1, \mathbf{s}\cdot\mathbf{t}_{L_\ell,0}^\top + \boxed{\cancel{s_\ell}})]\!]_2) \ .$$

We have $\widehat{\mathsf{G}}_3^{b'} \approx_c \widehat{\mathsf{G}}_4^{b'}$ under the security of iFE. To see this, we distinguish two cases for any pair of queries $\mathsf{QPKeyGen}(\mathbf{y})$ and $\mathsf{QEnc}(L_\ell, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$. Let $\{\mathsf{iDK}_i\}_{i\in[0;n-1]}$ and $\mathsf{iCT}_{L_\ell,0}$ denote the challenger's responses. If $P^\star(x_0, \mathbf{y}[0]) \neq 0$, then the adversary learns nothing at all about the vectors encoded in $\mathsf{iCT}_{L_\ell,0}$ and iDK_0. If $P^\star(x_0, \mathbf{y}[0]) = 0$, then the adversary can decrypt this combination of ciphertext and decryption key. However, $\mathsf{QPKeyGen}(\mathbf{y})$ is a type 2 query in this case, so the inner products of the vectors encoded in $\mathsf{iCT}_{L_\ell,0}$ and iDK_0 do not change from $\widehat{\mathsf{G}}_3^{b'}$ to $\widehat{\mathsf{G}}_4^{b'}$.

Game $\widehat{\mathsf{G}}_5^{b'}$: This game is the same as G_4^b except that, for each query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$, the challenger samples a random element $\mathbf{h} \xleftarrow{\$} \mathcal{M}$ and computes $\mathbf{c} = \boldsymbol{\mu}^b \oplus \boxed{\mathbf{h}}$. To argue $\mathsf{G}_4^b \approx_c \mathsf{G}_5^b$, we let $\mathsf{QEnc}(L_\ell, x_0^j, \boldsymbol{\mu}^{j,0}, \boldsymbol{\mu}^{j,1})$ denote the j-th encryption query for the label L_ℓ such that $\boldsymbol{\mu}^{j,0} \neq \boldsymbol{\mu}^{j,1}$, and q_ℓ refers to the number of such queries. For the reply to the j-th query, the challenger samples $r^j \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\mathsf{iCT}_{L_\ell,0} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, [\![(\mathbf{0}, r^j + s_\ell, 1)]\!]_1) \ .$$

For each query $\mathsf{QPKeyGen}(\mathbf{y})$, the adversary learns either nothing about r^j if $P^\star(x_0^j, \mathbf{y}[0]) \neq 0$, or it learns $r^j + s_\ell$ if $P^\star(x_0^j, \mathbf{y}[0]) = 0$. Thus, the adversary's view is captured by $\{(r^j + s_\ell, H(r^j))\}_{j\in[q_\ell]}$. Then $\mathsf{G}_4^b \approx_c \mathsf{G}_5^b$ follows from an application of Lemma 35 for $s = s_\ell$ and $q = q_\ell$.

Finally, we observe that $\mathsf{G}_5^0 \equiv \mathsf{G}_5^1$ since for each query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$, the random element $\mathbf{h} \xleftarrow{\$} \mathcal{M}$ perfectly hides $\boldsymbol{\mu}^b$ in $\mathbf{c} = \boldsymbol{\mu}^b \oplus \mathbf{h}$. This concludes the proof.

□

5.3 Proof of Security – Non-Function-Hiding Case

Proof of Proposition 33 – Non-Function-Hiding Case. The proposition is proven via the following sequence of hybrid games. Modifications between consecutive games are highlighted using boxes.

Game G_0^b for $b \in \{0,1\}$: This is the experiment $\mathbf{Exp}^{\mathsf{mc\text{-}abe\text{-}}b}_{\mathsf{ABE},\mathcal{A}}(1^\lambda)$. In particular, the challenger behaves as follows.

- Upon receiving an encryption query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$, define $\mathbf{t}_{L,0} \in \mathbb{Z}_p^\kappa$ component-wise as

$$\mathbf{t}_{L,0}[\nu] = -\sum_{i\in[n-1]} \mathsf{PRF}_{K_i}(L, \nu) \quad \text{for } \nu \in [\kappa] \ ,$$

sample $r \xleftarrow{\$} \mathbb{Z}_p$, run $\mathsf{iCT}_{L,0} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_0, x_0, (\mathbf{t}_{L,0}, r))$ and return $\mathsf{CT}_L = (\mathsf{iCT}_{L,0}, \mathbf{c} = \boldsymbol{\mu}^b \oplus H(r))$.

- Upon receiving an attribute key query $\mathsf{QAKeyGen}(i, L, x_i)$, set $\mathbf{t}_{L,i} \in \mathbb{Z}_p^\kappa$ component-wise as $\mathbf{t}_{L,i}[v] = \mathsf{PRF}_{K_i}(L, v)$ for $v \in [\kappa]$, run $\mathsf{iCT}_{L,i} \leftarrow \mathsf{iEnc}(\mathsf{iMSK}_i, x_i, (\mathbf{t}_{L,i}, 0))$ and return $\mathsf{DK}_{L,i} = \mathsf{iCT}_{L,i}$.
- Let q_k denote the number of queries to $\mathsf{QPKeyGen}$. Upon receiving the ℓ-th policy key query $\mathsf{QPKeyGen}(\mathbf{y}^\ell)$ for $\ell \in [q_k]$, sample a subset $\Delta_\ell \leftarrow \mathsf{SamplerSet}(\kappa, Q, \tau)$ and define the vector $\mathbf{e}_{\Delta_\ell} = \sum_{v \in \Delta_\ell} \mathbf{e}_v$. Then generate $\mathsf{iDK}_i^\ell \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, \mathbf{y}^\ell[i], (\mathbf{e}_{\Delta_\ell}, 1))$ for all $i \in [0; n-1]$ and return $\mathsf{DK}_{\mathbf{y}}^\ell = \{\mathsf{iDK}_i^\ell\}_{i \in [0; n-1]}$.

Similar to the function-hiding case, a policy key query $\mathsf{QPKeyGen}(\mathbf{y}^\ell)$ can be of two types w.r.t. a label $L \in \mathcal{L}$.

1. all queries $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ satisfy $\boldsymbol{\mu}^0 = \boldsymbol{\mu}^1$ or $P^\star(x_0, \mathbf{y}^\ell[0]) \neq 0$, or
2. there exists at least one query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$ and $P^\star(x_0, \mathbf{y}^\ell[0]) = 0$.

In case 2, the admissibility condition implies that there exists at least one index $i_\ell \in ([n-1] \setminus \mathcal{C})$ such that $P^\star(x, \mathbf{y}^\ell[i_\ell]) \neq 0$ for all queries of the form $\mathsf{QAKeyGen}(i_\ell, L, x)$.

Game G_1^b for $b \in \{0,1\}$: This game is the same as game G_0^b except that the challenger picks the subsets $\Delta_\ell \leftarrow \mathsf{SamplerSet}(\kappa, Q, \tau)$ for $\ell \in [q_k]$ in one shot at the outset of the game. For every $\ell \in [q_k]$, the challenger chooses an index

$$\delta_\ell \in \Delta'_\ell := \Delta_\ell \setminus \Big(\bigcup_{\ell' \in [q_k] \setminus \{k\}} \Delta_{\ell'} \Big)$$

if $\Delta'_\ell \neq \varnothing$, and aborts otherwise. The admissibility condition requires that $q_k \leq Q$. Therefore, it follows from an application of Lemma 20 that $\Delta'_\ell \neq \varnothing$ for all $\ell \in [q_k]$ with overwhelming probability which implies that $\mathsf{G}_1^b \approx_s \mathsf{G}_2^b$.

Game G_2^b for $b \in \{0,1\}$: This game is the same as G_1^b except that, upon receiving a query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$, the challenger computes

$$\mathsf{iCT}_{L,0} \leftarrow \mathsf{iEnc}\Big(\mathsf{iMSK}_0, x_0, \Big(\mathbf{t}_{L,0} + \boxed{r \cdot \sum_{\ell \in [q_k]} \mathbf{e}_{\delta_\ell}}, \boxed{0}\Big)\Big) \ .$$

We note that all inner products of vectors embedded in ciphertexts and keys of the 0-th iFE instance do not change from G_1^b to G_2^b. Thus, we have $\mathsf{G}_1^b \approx_c \mathsf{G}_2^b$ under the security of iFE.

Game G_3^b for $b \in \{0,1\}$: This game is the same a G_2^b except that, upon receiving a query $\mathsf{QAKeyGen}(L, x_i)$, the challenger computes $\mathbf{t}_{L,i}$ component-wise for each $v \in [\kappa]$ as

$$\mathbf{t}_{L,i}[v] = \begin{cases} \boxed{0} & \text{if} \quad \begin{array}{l} \bullet\ v = \delta_\ell \text{ for some } \ell \in [q_k], \\ \bullet\ \mathsf{QPKeyGen}(\mathbf{y}^\ell) \text{ is a type 2 query w.r.t. } L, \\ \bullet\ i = i_\ell, \end{array} \\ \mathsf{PRF}_{K_i}(L, v) & \text{otherwise.} \end{cases}$$

We argue that the adversary cannot notice such a modification in $\mathbf{t}_{L,i}$. First, as $\mathsf{QPKeyGen}(\mathbf{y}^\ell)$ is a type 2 query w.r.t. L, the iFE decryption key iDK_i^ℓ cannot be used to decrypt the ciphertext $\mathsf{iCT}_{\ell,i}$ encoding $\mathbf{t}_{L,i}$. Second, since $v = \delta_\ell$, all other decryption keys $\mathsf{iDK}_i^{\ell'}$, for $\ell' \neq \ell$, encode a vector $\mathbf{e}_{\Delta_{\ell'}}$ such that $\mathbf{e}_{\Delta_{\ell'}}[v] = 0$ by construction. So the inner product does not change and we have $\mathsf{G}_2^b \approx \mathsf{G}_3^b$ under the security of iFE.

Game G_4^b for $b \in \{0,1\}$: Let $\mathcal{Q}_L = \{L_1, \ldots, L_{q_L}\}$ be the set of all labels that appear in an encryption query and let $q_L = |\mathcal{Q}_L|$. This game is the same as G_3^b except that the challenger initially samples a scalar $s_L \xleftarrow{\$} \mathbb{Z}_p$ for each $L \in \mathcal{Q}_L$. Upon receiving a query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$, it now computes

$$\mathsf{iCT}_{L,0} \leftarrow \mathsf{iEnc}\Big(\mathsf{iMSK}_0, x_0, \Big(\mathbf{t}_{L,0} + r \cdot \sum_{\ell \in [q_k]} \mathbf{e}_{\delta_\ell} + \boxed{s_L \cdot \sum_{\substack{\ell \in [q_k], \\ P^\star(x_0, \mathbf{y}^\ell[0]) = 0}} \mathbf{e}_{\delta_\ell}}, 0\Big)\Big) \ .$$

We have $\mathsf{G}_3^b \approx_c \mathsf{G}_4^b$ which can be seen as follows. Let $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ and $\ell \in [q_k]$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$ and $P^\star(x_0, \mathbf{y}^\ell[0]) = 0$, i.e. $\mathsf{QPKeyGen}(\mathbf{y}^\ell)$ is a type 2 query w.r.t. the label L. After the changes from the previous game, we can observe that the value $\mathsf{PRF}_{K_{i_\ell}}(L, \delta_\ell)$ now only appears inside $\mathbf{t}_{L,0}$. Furthermore, the slot $i_\ell \in [n-1] \setminus \mathcal{C}$ is honest, so the PRF key $K_{0,i}$ is not known to the adversary. This implies that $\{\mathsf{PRF}_{K_{i_\ell}}(L, \delta_\ell)\} \approx_c \{\mathsf{PRF}_{K_{i_\ell}}(L, \delta_\ell) + s_L\}$ by the security of PRF, where the first distribution corresponds to G_3^b and the second distribution corresponds to G_4^b, followed by the same post-processing.

Game $\mathsf{G}_{4,\alpha}^b$ for $\alpha \in [0; q_L]$ and $b \in \{0,1\}$: This game is the same as game G_4^b except that, when receiving a query $\mathsf{QEnc}(L_{\alpha'}, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\alpha' \in [\alpha]$ and $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$, the challenger samples a random element $\mathbf{h} \xleftarrow{\$} \mathcal{M}$ and computes $\mathbf{c} = \boldsymbol{\mu}^b \oplus \mathbf{h}$. Clearly, we have $\mathsf{G}_{4,0}^b = \mathsf{G}_4^b$ as $[0] = \varnothing$. Furthermore, we can observe that $\mathsf{G}_{4,q_L}^0 \equiv \mathsf{G}_{4,q_L}^1$ since for each query $\mathsf{QEnc}(L, x_0, \boldsymbol{\mu}^0, \boldsymbol{\mu}^1)$ such that $\boldsymbol{\mu}^0 \neq \boldsymbol{\mu}^1$, the random element $\mathbf{h} \xleftarrow{\$} \mathcal{M}$ perfectly hides $\boldsymbol{\mu}^b$ in $\mathbf{c} = \boldsymbol{\mu}^b \oplus \mathbf{h}$. It remains to argue that $\mathsf{G}_{4,\alpha-1}^b \approx_c \mathsf{G}_{4,\alpha}^b$ for all $\alpha \in [q_L]$. Let $\mathsf{QEnc}(L_\alpha, x_0^j, \boldsymbol{\mu}^{j,0}, \boldsymbol{\mu}^{j,1})$ denote the j-th encryption query for the label L_α such that $\boldsymbol{\mu}^{j,0} \neq \boldsymbol{\mu}^{j,1}$, and q_α refers to the number of such queries. For each $\ell \in [q_k]$, the adversary learns either nothing about r^j if $P^\star(x_0^j, \mathbf{y}^\ell[0]) \neq 0$, or it learns $r^j + s_{L_\alpha}$ if $P^\star(x_0^j, \mathbf{y}^\ell[0]) = 0$. Thus, the adversary's view is captured by $\{(r^j + s_{L_\alpha}, H(r^j))\}_{j \in [q_\alpha]}$. Then an application of Lemma 35 for $s = s_{L_\alpha}$ and $q = q_\alpha$ implies that $\mathsf{G}_{4,\alpha-1}^b \approx_c \mathsf{G}_{4,\alpha}^b$ which concludes the proof.

□

Acknowledgments

This work was supported in part by the France 2030 ANR Project ANR-22-PECY-003 SecureCompute.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, Berlin, Heidelberg, August 2010.

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

[ABG19] Michel Abdalla, Fabrice Benhamouda, and Romain Gay. From single-input to multi-client inner-product functional encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 552–582. Springer, Cham, December 2019.

[ABKW19] Michel Abdalla, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. Decentralizing inner-product functional encryption. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 128–157. Springer, Cham, April 2019.

[ACF+18] Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 597–627. Springer, Cham, August 2018.

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

[Agr17] Shweta Agrawal. Stronger security for reusable garbled circuits, general definitions and attacks. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 3–35. Springer, Cham, August 2017.

[AGRW17] Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 601–626. Springer, Cham, April / May 2017.

[AGT21a] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-input quadratic functional encryption from pairings. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 208–238, Virtual Event, August 2021. Springer, Cham.

[AGT21b] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-party functional encryption. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 224–255. Springer, Cham, November 2021.

[AGT22] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-input quadratic functional encryption: Stronger security, broader functionality. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 711–740. Springer, Cham, November 2022.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Berlin, Heidelberg, August 2015.

[AKY24] Shweta Agrawal, Simran Kumari, and Shota Yamada. Pseudorandom multi-input functional encryption and applications. Cryptology ePrint Archive, Paper 2024/1720, 2024.

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

[ARYY23] Shweta Agrawal, Mélissa Rossi, Anshu Yadav, and Shota Yamada. Constant input attribute based (and predicate) encryption from evasive and tensor LWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 532–564. Springer, Cham, August 2023.

[ATY23] Shweta Agrawal, Junichi Tomida, and Anshu Yadav. Attribute-based multi-input FE (and more) for attribute-weighted sums. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 464–497. Springer, Cham, August 2023.

[AY20] Shweta Agrawal and Shota Yamada. CP-ABE for circuits (and more) in the symmetric key setting. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 117–148. Springer, Cham, November 2020.

[AYY22] Shweta Agrawal, Anshu Yadav, and Shota Yamada. Multi-input attribute based encryption and predicate encryption. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 590–621. Springer, Cham, August 2022.

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

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, August 2001.

[BJK+18] Zvika Brakerski, Aayush Jain, Ilan Komargodski, Alain Passelègue, and Daniel Wichs. Non-trivial witness encryption and null-iO from standard assumptions. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 425–441. Springer, Cham, September 2018.

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

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer, Berlin, Heidelberg, February 2007.

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

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

[CLW25] Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. Lattice-based multi-authority/client attribute-based encryption for circuits. *IACR Communications in Cryptology*, 1(4), 2025.

[DOT18] Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 245–277. Springer, Cham, March 2018.

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

[FFMV23] Danilo Francati, Daniele Friolo, Giulio Malavolta, and Daniele Venturi. Multi-key and multi-input predicate encryption from learning with errors. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 573–604. Springer, Cham, April 2023.

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

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

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

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[HJP18] Dennis Hofheinz, Dingding Jia, and Jiaxin Pan. Identity-based encryption tightly secure under chosen-ciphertext attacks. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 190–220. Springer, Cham, December 2018.

[HLL24] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. A general framework for lattice-based ABE using evasive inner-product functional encryption. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 433–464. Springer, Cham, May 2024.

[JR17] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. *Journal of Cryptology*, 30(4):1116–1156, October 2017.

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

[LL20] Huijia Lin and Ji Luo. Compact adaptively secure ABE from k-Lin: Beyond NC^1 and towards NL. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 247–277. Springer, Cham, May 2020.

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

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

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In Irit Dinur, editor, *57th FOCS*, pages 11–20. IEEE Computer Society Press, October 2016.

[LW11] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, Berlin, Heidelberg, May 2011.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012.

[NPP22] Ky Nguyen, Duong Hieu Phan, and David Pointcheval. Multi-client functional encryption with fine-grained access control. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 95–125. Springer, Cham, December 2022.

[NPP24] Ky Nguyen, Duong Hieu Phan, and David Pointcheval. Multi-client functional encryption with public inputs and strong security. Cryptology ePrint Archive, Report 2024/740, 2024.

[PS24] David Pointcheval and Robert Schädlich. Multi-client attribute-based and predicate encryption from standard assumptions. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part III*, volume 15366 of *LNCS*, pages 31–64. Springer, Cham, December 2024.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[SBC+07] Elaine Shi, John Bethencourt, Hubert T.-H. Chan, Dawn Xiaodong Song, and Adrian Perrig. Multi-dimensional range query over encrypted data. In *2007 IEEE Symposium on Security and Privacy*, pages 350–364. IEEE Computer Society Press, May 2007.

[Sch25] Robert Schädlich. Multi-client attribute-based and predicate encryption, revisited. In Tibor Jager and Jiaxin Pan, editors, *Public-Key Cryptography - PKC 2025*, Cham, 2025. Springer Nature Switzerland.

[SV23] Elaine Shi and Nikhil Vanjani. Multi-client inner product encryption: Function-hiding instantiations without random oracles. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 622–651. Springer, Cham, May 2023.

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

[Tom19] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 459–488. Springer, Cham, December 2019.

[TÜ24] Erkan Tairi and Akin Ünal. Lower bounds for lattice-based compact functional encryption. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 249–279. Springer, Cham, May 2024.

[Üna20] Akin Ünal. Impossibility results for lattice-based functional encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 169–199. Springer, Cham, May 2020.

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

[Wee17] Hoeteck Wee. Attribute-hiding predicate encryption in bilinear groups, revisited. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 206–233. Springer, Cham, November 2017.

A AB-MIFE for Inner Products Against Bounded Collusions

A.1 Definitions for MIFE with Corruptions

We recall the definition of Multi-Input Functional Encryption (MIFE). Let $\{\mathcal{U}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{V}_\lambda\}_{\lambda\in\mathbb{N}}$ be sequences of domain and range spaces, where $\mathcal{U}_\lambda = \mathcal{U}_{\lambda,\mathsf{pub}} \times \mathcal{U}_{\lambda,\mathsf{pri}}$ consists of a public and a private component. We consider a function class $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ where each $\mathcal{F}_\lambda$ contains functions $f\colon \mathcal{U}_\lambda^n \to \mathcal{V}_\lambda$ and $n = \mathrm{poly}(\lambda)$.

Definition 37 (Syntax of MIFE)**.** A MIFE scheme for the functionality $\mathcal{F}$ consists of four efficient algorithms:

$\mathsf{Setup}(1^\lambda) \to (\{\mathsf{SK}_i\}_{i\in[n]}, \mathsf{MSK})$**:** On input the security parameter 1^λ, this algorithm outputs a set of secret keys $\{\mathsf{SK}_i\}_{i\in[n]}$ and a master secret key MSK.

$\mathsf{Enc}(\mathsf{SK}_i, u_{i,\mathsf{pub}}, u_{i,\mathsf{pri}}) \to \mathsf{CT}_i$**:** On input a secret key SK_i, a public input $u_{i,\mathsf{pub}} \in \mathcal{U}_{\lambda,\mathsf{pub}}$ and a private input $u_{i,\mathsf{pri}} \in \mathcal{U}_{\lambda,\mathsf{pri}}$, this algorithm outputs a ciphertext CT_i. We assume that CT implicitly contains u_{pub}.

$\mathsf{KeyGen}(\mathsf{MSK}, f) \to \mathsf{DK}$**:** On input the master secret key $\mathsf{MSK}$and a function $f \in \mathcal{F}_\lambda$, this algorithm outputs a decryption key DK. We assume that DK implicitly contains f.

$\mathsf{Dec}(\mathsf{DK}, \{\mathsf{CT}_i\}_{i\in[n]}) \to v$**:** On input a decryption key DK and a set of ciphertexts $\{\mathsf{CT}_i\}_{i\in[n]}$, this algorithm outputs a value $v \in \mathcal{V}_\lambda$.

Note that, in contrast to our definition of MC-ABE, we use slot indices $i \in [n]$ instead of $[0; n-1]$ because this choice seems more common and natural in the general MIFE setting.

Correctness. A MIFE scheme for a functionality $\mathcal{F}$ is correct if for all $\lambda \in \mathbb{N}$, $u_1, \ldots, u_n \in \mathcal{U}_\lambda$ and $f \in \mathcal{F}_\lambda$, it holds that

$$\Pr\left[v = f(u) \;\middle|\; \begin{array}{l} (\{\mathsf{SK}_i\}_{i\in[n]}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \forall i \in [n]\colon\ \mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{SK}_i, u_i) \\ \mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f) \\ v := \mathsf{Dec}(\mathsf{DK}, \{\mathsf{CT}_i\}_{i\in[n]}) \end{array} \right] = 1 \ ,$$

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

Security. We define IND-CPA security with static corruptions.

Definition 38 (Security Against Bounded Collusions)**.** Let $\mathsf{xxx} \in \{\mathsf{adp}, \mathsf{stat}\}$. A MIFE scheme FE for a functionality $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is secure against Q key queries if $\mathbf{Exp}^{\mathsf{mife\text{-}0}}_{\mathsf{FE},\mathcal{A}}(1^\lambda) \approx_c \mathbf{Exp}^{\mathsf{mife\text{-}1}}_{\mathsf{FE},\mathcal{A}}(1^\lambda)$ for all PPT adversaries $\mathcal{A}$, where $\mathbf{Exp}^{\mathsf{mife}\text{-}b}_{\mathsf{FE},\mathcal{A}}$ for $b \in \{0,1\}$ proceeds as follows.

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ and receive from it a set $\mathcal{C} \subseteq [n]$ of corrupted slots. Run $(\{\mathsf{SK}_i\}_{i\in[n]}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and send $\{\mathsf{SK}_i\}_{i\in\mathcal{C}}$ to $\mathcal{A}$. Initialize a counter $m = 0$.
- **Query.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ has two options.
  - $\mathsf{QEnc}(i, u_i^0, u_i^1)$: upon $\mathcal{A}$ submitting an index $i \in [n]$ and a pair of messages $u_i^0, u_i^1 \in \mathcal{U}_\lambda$, run $\mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{SK}_i, u_i^b)$ and return CT_i to $\mathcal{A}$.
  - $\mathsf{QKeyGen}(f)$: upon $\mathcal{A}$ submitting a function $f \in \mathcal{F}_\lambda$, run $\mathsf{DK} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f)$ and return DK to $\mathcal{A}$. Set $m := m + 1$.
- **Guess.** The adversary outputs a bit $b' \in \{0,1\}$. The outcome of the experiment is b' if $\mathcal{A}$ satisfies the following admissibility condition. Otherwise, the outcome is set to 0. The adversary $\mathcal{A}$ is admissible if

- $m \le Q$,
- for $i \in \mathcal{C}$, each query $\mathsf{QEnc}(i, u_i^0, u_i^1)$ satisfies $u_i^0 = u_i^1$,
- for $i \in [n] \setminus \mathcal{C}$, each query $\mathsf{QEnc}(i, u_i^0, u_i^1)$ satisfies $u_{i,\mathsf{pub}}^0 = u_{i,\mathsf{pub}}^1$,
- for all sequences $(x_1^0, \ldots, x_n^0, x_1^1, \ldots, x_n^1, f)$ such that
  - for all $i \in \mathcal{C}$, we have $u_i^0 = u_i^1$,
  - for all $i \in [n] \setminus \mathcal{C}$, $\mathsf{QEnc}(i, u_i^0, u_i^1)$ is queried, and
  - $\mathsf{QKeyGen}(f)$ is queried,

  it holds $f(u_1^0, \ldots, u_n^0) = f(u_1^1, \ldots, u_n^1)$, and
- if xxx = stat, then $\mathcal{A}$ submits all queries to QEnc and QKeyGen upfront in one shot.

If we do not explicitly specify Q, then the adversary can submit any number of queries to QKeyGen and we say that iFE is secure against *unbounded* collusions.

Function Class. We define the attribute-based inner-product functionality (AB-IP) for *conjunctions* of a predicate family.

Definition 39 (MIFE for AB-IP)**.** Let $d, \ell, \ell' \in \mathbb{N}$ and $\{p_\lambda\}_{\lambda\in\mathbb{N}}$, $\{P_\lambda\}_{\lambda\in\mathbb{N}}$, $\{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be sequences of prime numbers, predicates, ciphertext attribute spaces and key attribute spaces, respectively. For each $\lambda \in \mathbb{N}$, we let $\mathcal{U}_{\lambda,\mathsf{pub}} = \mathcal{X}_\lambda \cup \{\star\}$, $\mathcal{U}_{\lambda,\mathsf{pri}} = \mathbb{Z}_{p_\lambda}^\ell$ and $\mathcal{V}_\lambda = \mathbb{Z}_{p_\lambda} \cup \{\perp\}$. The AB-IP functionality is a family of functions $\{\mathcal{F}_\lambda = (\mathcal{Y}_\lambda \times \mathbb{Z}_{p_\lambda}^\ell)^n\}_{\lambda\in\mathbb{N}}$, where $f = ((y_1, \mathbf{v}_1), \ldots, (y_n, \mathbf{v}_n)) \in \mathcal{F}_\lambda$ represents the function $f : \mathcal{U}_\lambda^n \to \mathcal{V}_\lambda$ defined as

$$f\big((x_1, \mathbf{u}_1), \ldots, (x_n, \mathbf{u}_n)\big) = \begin{cases} \sum_{i\in[n]} \mathbf{u}_i \mathbf{v}_i^\top & \text{if } P^\star(x_i, y_i) = 0 \text{ for all } i \in [n]\,, \\ \perp & \text{otherwise}\,, \end{cases}$$

where $P_\lambda^\star(x, y) = P_\lambda(x, y)$ if $x \in \mathcal{X}_\lambda$ and $P_\lambda^\star(\star, y) = 0$ for each $y \in \mathcal{Y}_\lambda$.

Definition 40 (Legitimate Keys)**.** We say that a MIFE scheme for AB-IP is secure against legitimate key queries if it is secure against all legitimate adversaries. An adversary $\mathcal{A}$ is called legitimate if it is admissible (as per Definition 38) and satisfies the following additional condition. Let q_k denote the number of queries to QKeyGen and, for $i \in [n]$, let $q_{c,i}$ denote the number of queries of the form $\mathsf{QEnc}(i, \star, \star)$. Let

$$\big(\mathcal{C}, \{(i, u_i^{j,0}, u_i^{j,1})\}_{i\in[n], j\in[q_{c,i}]}, \{f^k\}_{k\in[q_k]}\big)$$

be the query of the adversary. Parse $u_i^{j,b} = (x_i^j, \mathbf{u}_i^{j,b})$ and $f^k = \{(y_i^k, \mathbf{v}_i^k)\}_{i\in[n]}$. We say that $\mathcal{A}$ is legitimate if for all $k \in [q_k]$ and $i \in [n] \setminus \mathcal{C}$, there exists $j_i^k \in [q_{c,i}]$ such that $P^\star(x_i^{j_i^k}, y_i^k) = 0$.

A.2 Construction of AB-MIFE Against Legitimate Keys

Construction 41 (AB-MIFE Against Bounded Collusions)**.** For a fixed collusion bound Q, we choose the parameters κ and τ according to Lemma 20. The construction employs an AB-IPFE scheme $\mathsf{iFE} = (\mathsf{iSetup}, \mathsf{iEnc}, \mathsf{iKeyGen}, \mathsf{iDec})$ for a predicate $P : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. The details of the MIFE scheme FE for AB-IP go as follows:

$\mathsf{Setup}(1^\lambda)$ takes as input the security parameter 1^λ, samples $\mathbf{t}_1, \ldots, \mathbf{t}_{n-1} \xleftarrow{\$} \mathbb{Z}_p^\kappa$, sets $\mathbf{t}_n = -\sum_{i\in[n-1]} \mathbf{t}_i$ and runs $(\mathsf{iMPK}_i, \mathsf{iMSK}_i) \leftarrow \mathsf{iSetup}(1^\lambda, 1^{\kappa+\ell})$ for $i \in [n]$. Then it outputs secret keys $\{\mathsf{SK}_i = (\mathsf{iMPK}_i, \mathbf{t}_i)\}_{i\in[n]}$ and the master secret key $\mathsf{MSK} = \{\mathsf{iMSK}_i\}_{i\in[n]}$.

$\mathsf{Enc}(\mathsf{SK}_i, x_i, \mathbf{u}_i)$ takes as input SK_i, an attribute $x_i \in \{0, 1\}^{\ell'}$ and a message vector $\mathbf{u}_i \in \mathbb{Z}_p^\ell$. The algorithm runs $\mathsf{iCT}_i \leftarrow \mathsf{iEnc}(\mathsf{iMPK}_i, x_i, (\mathbf{u}_i, \mathbf{t}_i))$ and outputs the ciphertext $\mathsf{CT}_i = \mathsf{iCT}_i$.

$\mathsf{KeyGen}(\mathsf{MSK}, \{(y_i, \mathbf{v}_i)\}_{i\in[n]})$ takes as input MSK and n policy-message pairs $(y_i, \mathbf{v}_i) \in \mathcal{Y} \times \mathbb{Z}_p^\ell$, samples a subset $\Delta \leftarrow \mathsf{SamplerSet}(\kappa, Q, \tau)$ of $[\kappa]$ and defines the vector $\mathbf{e}_\Delta = \sum_{k\in\Delta} \mathbf{e}_k$, where $\mathbf{e}_k$ denotes the k-th unit vector in $\mathbb{Z}_p^\kappa$. Then the algorithm generates $\mathsf{iDK}_i \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, y_i, (\mathbf{v}_i, \mathbf{e}_\Delta))$ for $i \in [n]$ and outputs $\mathsf{DK} = \{\mathsf{iDK}_i\}_{i\in[n]}$.

$\mathsf{Dec}(\mathsf{DK}, \{\mathsf{CT}_i\}_{i\in[\ell]})$ takes as input a decryption key $\mathsf{DK} = \{\mathsf{iDK}_i\}_{i\in[\ell]}$ and ciphertexts $\{\mathsf{CT}_i = \mathsf{iCT}_i\}_{i\in[n]}$. For each $i \in [n]$, the algorithm runs $d_i \leftarrow \mathsf{iDec}(\mathsf{iDK}_i, \mathsf{iCT}_i)$ and immediately outputs $\perp$ if $d_i = \perp$. Otherwise, it outputs $d = \sum_{i\in[n]} d_i$.

Correctness. Correctness readily follows from the correctness of iFE and the fact that $\sum_{i\in[n]} \mathbf{t}_i = \mathbf{0}$.

Security. We first prove security against legitimate keys.

Proposition 42. *If* iFE *is* stat*-secure against Q key queries, then the AB-MIFE scheme* FE *in Construction 41 is* stat*-secure against Q legitimate key queries.*

Proof of Proposition 42. The proof considers the following sequence of hybrids.

Game G_0^b for $b \in \{0,1\}$: This is the same as the experiment $\mathbf{Exp}^{\mathsf{mife}\text{-}b}_{\mathsf{FE},\mathcal{A}}(1^\lambda)$.

Game G_1^b for $b \in \{0,1\}$: This game is the same as G_0^b except that the challenger picks random subsets $\Delta_k \leftarrow \mathsf{SamplerSet}(\kappa, Q, \tau)$ for $k \in [q_k]$ upfront. Note that the admissibility condition requires $q_k \le Q$. Upon receiving the k-th key query $\mathsf{QKeyGen}(\{(y_i^k, \mathbf{v}_i^k)\}_{i\in[n]})$, the challenger returns $\mathsf{DK}_k = \{\mathsf{iDK}_i^k\}_{i\in[n]}$ computed as follows:

$$\mathsf{iDK}_i^k \leftarrow \mathsf{iKeyGen}(\mathsf{iMSK}_i, y_i^k, (\mathbf{v}_i^k, \mathbf{e}_{\Delta_k})) .$$

We have $\mathsf{G}_0^b \equiv \mathsf{G}_1^b$ as the view of the adversary does not change.

Game G_2^b for $b \in \{0,1\}$: This game is the same as G_1^b except that, for every $k \in [q_k]$, the challenger chooses an index $\delta_k \in \Delta_k' := \Delta_k \setminus (\bigcup_{k'\in[q_k]\setminus\{k\}} \Delta_{k'})$ if $\Delta_k' \neq \varnothing$, and aborts otherwise. From Lemma 20, it follows that $\Delta_k' \neq \varnothing$ for all $k \in [q_k]$ with overwhelming probability, which implies that $\mathsf{G}_1^b \approx_s \mathsf{G}_2^b$.

Following the notation of Definition 40, we denote the j-th encryption query for the i-th slot by $\mathsf{QEnc}(i, (x_i^j, \mathbf{u}_i^{j,0}), (x_i^j, \mathbf{u}_i^{j,1}))$ and the k-th key generation query by $\mathsf{QKeyGen}(\{(y_i^k, \mathbf{v}_i^k)\}_{i\in[n]})$. For $i \in [n]$, $j \in [q_{c,i}]$ and $k \in [q_k]$, we let

$$h_i^{j,k,b} = (\mathbf{u}_i^{j,b} - \mathbf{u}_i^{j,0}) \cdot (\mathbf{v}_i^k)^\top$$

and $\mathcal{Q}_i^k = \{j \in [q_{c,i}] : y_i^k(x_i^j) = 0\}$. We note that the adversary's admissibility condition implies the following properties:

1. For $i \in [n]$ and $k \in [q_k]$, the value $h_i^{j,k,b}$ is constant for all $j \in \mathcal{Q}_i^k$. Therefore, we can omit the index j in this case and simply write $h_i^{k,b} = h_i^{j,k,b}$ for $j \in \mathcal{Q}_i^k$.
2. For all $i \in \mathcal{C}$ and $k \in [q_k]$, we have $h_i^{k,b} = 0$.
3. For all $k \in [q_k]$, we have $\sum_{i\in[n]} h_i^{k,b} = 0$.

Game G_3^b for $b \in \{0,1\}$: This game is the same as the previous game G_2^b except that when receiving an encryption query $\mathsf{QEnc}(i, (x_i^j, \mathbf{u}_i^{j,0}), (x_i^j, \mathbf{u}_i^{j,1}))$, the challenger computes and returns

$$\mathsf{iCT}_i^j \leftarrow \mathsf{Enc}\big(\mathsf{MSK}, i, x_i^j, (\mathbf{u}_i^{j,0}, \mathbf{t}_i - \textstyle\sum_{k\in[q_k]} h_i^{k,b} \cdot \mathbf{e}_{\delta_k})\big)$$

as opposed to $\mathsf{iCT}_i^j \leftarrow \mathsf{Enc}(\mathsf{MSK}, i, x_i^j, (\mathbf{u}_i^{j,b}, \mathbf{t}_i))$ in G_2^b. To argue indistinguishability, we consider three cases for $i \in [n]$, $j \in [q_{c,i}]$ and $k \in [q_k]$.

- If $i \in [n] \setminus \mathcal{C}$ and $P^\star(x_i^j, y_i^k) \neq 0$, then $\mathsf{iDec}(\mathsf{DK}_i^k, \mathsf{iCT}_i^j) = \perp$ and nothing is leaked about the vector encoded in iCT_i^j by the security of iFE.
- If $i \in [n] \setminus \mathcal{C}$ and $P^\star(x_i^j, y_i^k) = 0$, then property 1 implies that the iFE decryption result $\mathsf{iDec}(\mathsf{DK}_i^k, \mathsf{iCT}_i^j)$ is the same in G_2^b and G_3^b, so we can again rely on the security of iFE to conclude that the ciphertext iCT_i^j in the two games cannot be distinguished.
- If $i \in \mathcal{C}$, then we have $h_i^{k,b} = 0$ from property 2, so nothing changes.

In the end, we have $\mathsf{G}_2^b \approx_c \mathsf{G}_3^b$.

Game G_4^b for $b \in \{0,1\}$: This game is the same as game G_3^b except that, when receiving an encryption query $\mathsf{QEnc}(i, (x_i^j, \mathbf{u}_i^{j,0}), (x_i^j, \mathbf{u}_i^{j,1}))$, the challenger computes and returns

$$\mathsf{iCT}_i^j \leftarrow \mathsf{Enc}\big(\mathsf{MSK}, i, x_i^j, (\mathbf{u}_i^{j,0}, \mathbf{t}_i)\big) .$$

We note that property 3 implies that the distributions $\{\mathbf{t}_i\}_{i\in[\ell]}$ and $\{\mathbf{t}_i - \sum_{k\in[q_k]} h_i^{k,b} \cdot \mathbf{e}_{\delta_\kappa}\}_{i\in[\ell]}$ are identical, so $\mathsf{G}_3^b \equiv \mathsf{G}_4^b$. Moreover, we can observe that the challenger's responses to oracle queries in G_4^b are independent of the bit b. So we have $\mathsf{G}_4^0 \equiv \mathsf{G}_4^1$. This concludes the proof.

□

A.3 Upgrade to Any Keys

A recent work [ATY23] presents a generic compiler to remove the restriction to legitimate keys from the security model of MIFE for AB-IP.[10] The compiler uses a *public-key* ciphertext-policy ABE (CP-ABE) for the policy class supported by the AB-IP scheme (bounded-depth circuits in our case). To our knowledge, there is currently no CP-ABE for this policy class in the literature whose security can be based on the hardness of LWE. Fortunately, the work of [AY20] proposes a CP-ABE for bounded-depth circuits in the *secret-key* setting. As MIFE is a secret-key primitive anyway, this fact does not matter for us, and we observe that the compiler of [ATY23] can also be instantiated from secret-key ABE.

When instantiating (1) the AB-IPFE scheme iFE in Construction 41 with the AB-IPFE of [LLW21], and (2) the CP-ABE scheme in the conversion to any keys with the CP-ABE of [AY20], we obtain the following theorem.

Theorem 43. *Assuming the hardness of* LWE*, there exists a MIFE scheme for the AB-IP functionality for bounded-depth circuits that is secure against any Q key queries, i.e., bounded collusions.*

[10] In fact, their compiler covers the so-called *attribute-based attribute-weighted sums* functionality which is a generalization of AB-IP.