

# Traitor Tracing in Multi-sender Setting

## (TMCFE: Traceable Multi-client Functional Encryption)

Xuan Thanh Do<sup>1</sup>, Dang Truong Mac<sup>2</sup>, Ky Nguyen<sup>3</sup> (✉), Duong Hieu Phan<sup>4</sup>, and Quoc-Huy Vu<sup>5</sup>

<sup>1</sup> Institute of Cryptography Science and Technology, Vietnam  
[thanhkhtn@gmail.com](mailto:thanhkhtn@gmail.com)

<sup>2</sup> Vietnam National University, Hanoi, Vietnam  
[macdangtruong258@gmail.com](mailto:macdangtruong258@gmail.com)

<sup>3</sup> Sorbonne Université, CNRS, LIP6, Paris, France  
[ky.nguyen@lip6.fr](mailto:ky.nguyen@lip6.fr)

<sup>4</sup> LTCI, Telecom Paris, Institut Polytechnique de Paris, France  
[hieu.phan@telecom-paris.fr](mailto:hieu.phan@telecom-paris.fr)

<sup>5</sup> De Vinci Higher Education, De Vinci Research Center, Paris, France  
[quoc.huy.vu@ens.fr](mailto:quoc.huy.vu@ens.fr)

**Abstract.** Traitor tracing systems is a decades-old cryptographic primitive: a *single* sender responsible for encrypting all the data to *multiple legitimate receivers*, and when the output of decryption is leaked, one can trace to the source of the leakage. Recent works explore traceability of advanced encryption such as *Functional Encryption* (FE), where decryption gives *fine-grained* evaluation on the plaintext. However all still stays in the single-sender setting. In this paper we take the first step towards addressing the tracing problem in the general context with *multiple* senders, further allowing *fine-grained* decryption for joint computation over senders' independent data (i.e., the context of *multi-client* FE a.k.a MCFE). Our contributions are twofold:

- **Conceptually:** We propose the first tracing model in the context of multi-sender encryption, namely *Traceable Multi-Client Functional Encryption* (TMCFE), which enables tracing corrupted receivers from any pirate, who is allowed to extract secret information from both receivers and senders. Our model supports strong and naturally admissible decoders, removing artificial restrictions on the pirate decoder and thus addressing the shortcomings of existing traceable functional encryption schemes designed for the single-sender setting.
- **Technically:** We present a concrete lattice-based scheme TMCFE for computing inner products that achieves post-quantum security under standard assumptions. To achieve our conceptual objective, we build upon the recently improved *strong* IND-CPA security for MCFE (Nguyen, Phan, Pointcheval ACNS'23, PKC'25). Our main technical contribution is a generic compiler that transforms a large class of MCFE schemes with weak admissibility into schemes with strong admissibility.

## 1 Introduction

*Fine-grained decryption.* Recent advances in modern cryptography have enabled decryption outcome to be defined in a fine-grained manner, notably through the advent of *functional encryption* [11]. Functional Encryption (FE) is an advanced encryption framework that enhances traditional public-key encryption by providing fine-grained control over the information that is revealed. The initial form of functional encryption requires the entire dataset to be encrypted simultaneously by a *single sender* (i.e., within the one-to-many paradigm). Interesting generalization includes Multi-Input and Multi-Client FE that were proposed by [14,20], enabling independent encryption by *multiple* senders. Particularly, Multi-Client FE (MCFE) can operate in the arguably more complicated framework of multi-sender, i.e., the many-to-many communication. In this setting, *multiple clients* act as senders, each encrypts *independently* their own data using their own *secret encryption key* to obtain partial ciphertexts. These partial ciphertexts can be jointly decrypted using a functional decryption key given by a trusted authority, when all partial ciphertexts share a same *tag*.

*Traitor Tracing.* Traitor tracing systems [13] are a cryptographic technique designed to identify malicious users who leak secret decryption keys in a broadcast environment. It is particularly relevant in scenarios such as digital content protection, where content providers distribute encrypted media to legitimate subscribers but need to prevent unauthorized redistribution. If any collection of users (which are called *receivers* in the context of network communication) attempts to create and sell a decoding box that can be used to decrypt the content. The tracing algorithm, given access to such pirate decoder, is guaranteed to identify at least one corrupt users, i.e., a member of those who contributed to the creation of the decoder. Over the past decades, an important research direction has extended traitor tracing to more advanced encryption paradigms. These include group encryption [26], attribute-based encryption [12], and, more recently, functional encryption [18,19,27]. Throughout more than three decades of development, traitor tracing techniques, however, have remained within the one-to-many setting, where a *single sender* is responsible for encrypting all the data.

*Traitor tracing in the multi-sender setting.* For many-to-many communication, traitor tracing comes up naturally in virtually any application to which multi-client functional encryption is applicable, especially in the era of cloud computing and big data, where privacy-preserving joint computation is essential. Despite a large body of works on both the foundational and applicability aspects of MCFE, as well as the development of traitor tracing schemes for public key encryption, the notion of accountability in MCFE has remained unexplored. Moreover, great care is required due to the fact that in MCFE each client has a *secret encryption key*, as highlighted in the seminal works of [14,20]. Notably, in all existing models of security for MCFE [14,30], when secret encryption keys are corrupted by an attacker, they give inherently more power to that attacker. Thus one

has to put some *admissible* conditions so that the aforementioned attacker *does not trivially break security* using only functional decryption keys (to decrypt for function evaluations). This leads to a more complex setting with two adversarial corruptions: *encryption-key corruption* and *decryption-key corruption*. Simply adapting existing security model and techniques from the one-to-many traitor tracing paradigm, *which deals only with corrupted decryption keys*, is insufficient for designing an MCFE scheme that incorporates tracing capabilities. Our work aims to address these challenges by establishing a rigorous framework for traitor tracing in MCFE and proposing a novel construction that ensures tracing mechanism in the many-to-many communication setting.

*Our Contributions.* Our main contributions are summarized as follows.

(1) *A new model for traitor tracing in the multi-sender setting.* We introduce the notion of traceable multi-client functional encryption (TMCFE) as a new model to build traitor tracing systems in the many-to-many communication setting. In particular, we propose a standard and natural security model for TMCFE, removing all restrictions from previous works in single-sender environment. We also prove that public tracing and IND-CPA security *cannot* be achieved simultaneously for certain non-trivial function classes ( Theorem 1). Our impossibility exploits the possibility of corruption of *both senders (with private-key encryption) and receivers* in the context of MCFE, thus does not arise in the single-client setting. This makes generalizing the single-client tracing security to the multi-client case non-trivial.

(2) *Enhancing tracing security for TMCFE from strong IND-CPA security.* We identify a close connection between the admissibility condition required<sup>6</sup> in the IND-CPA security definition for MCFE and our tracing security definition of TMCFE, allowing the removal of artificial restrictions in [18, 27]. To that end, we also construct a *generic* compiler that transforms MCFE for *inner-product with access control* under certain structure of decryption<sup>7</sup> from weak admissibility to strong admissibility [30], with little overhead.

(3) *Bounded Embedded Identities TMCFE for Inner-Product.* We finally present a construction of TMCFE for *inner-product with bounded embedded identities and fully collusion-resistance* [23], based on standard *Learning With Errors* assumptions. This demonstrates the efficiency and practicality of our tracing notion in practical scenarios, e.g., where MCFE allows a statistical measure computed on the aggregated data, without revealing any additional information and with capability to trace the source of any breach by dishonest participants.

<sup>6</sup> Admissibility conditions are required to prevent trivial attacks in IND-CPA for MCFE, when secret encryption key of clients are corrupted.

<sup>7</sup> All known constructions of MCFE from lattice assumptions satisfy these requirements.

*Concurrent work.* A concurrent and independent work by [19] investigates traitor tracing in the single-client setting, achieving single-sender and multi-target security, and also proposes an embedded-identity construction from pairings. Although both works build upon the spirit of [23], our work tackles the significantly more challenging multi-client setting, which introduces new technical difficulties absent in the single-client case.

## 2 Technical Overview

We start by briefly recalling the standard IND-CPA security model for MCFE [14, 20, 21]. In the security game, the adversary is given access to a number of oracles (**Initialise**, **KeyGen**, **Enc**, **Corrupt**) and a challenge oracle **LoR** to which they can make queries. The names of oracles are self-explanatory, in particular, the adversary can query to **Corrupt** to obtain the secret encryption keys, *separately* from the corruption of the decryption keys that are associated with functions, which can be obtained via **KeyGen**. We remark that as MCFE are secret-key primitives, there is an **Enc** oracle to which the adversary can ask for any encryption of its choice for each encryption slot. The ability to corrupt makes the adversary stronger and thus leads to a stronger notion of security, as reflected via the notion of *admissibility* conditions for adversary.

### 2.1 Definition of Traitor Tracing in the Multi-Client Setting

*Our tracing model: Multi-Sender, Multi-Target, No Restriction.* We emphasize that existing works on tracing in FE focus only on *single-sender* FE [18, 27, 32]. We will extend this to the *multi-sender* setting, henceforth called *traceable multi-client FE*, with *embedded identities*. A traitor tracing system for MCFE with  $n$  clients consists probabilistic poly-time algorithms – **Setup**, **KeyGen**, **Enc**, **Dec** and **Trace**, which expands with the new **Trace** into the standard syntax of MCFE. The **Setup** algorithm takes as input security parameter  $\lambda$ , the identity space  $\mathcal{ID}$ , and generates a master secret key  $\text{msk}$ , encryption keys  $(\text{ek}_i)_{i \in [n]}$  and a tracing key  $\text{tk}$ . The key generation algorithm **KeyGen**, on input an identity  $\text{id}$  and a functional vector  $\mathbf{y}$ , where  $\mathbf{y} = [\mathbf{y}_1 \parallel \dots \parallel \mathbf{y}_n]$ , outputs a decryption key  $\text{sk}_{\text{id}, \mathbf{y}}$ . Each client can use the encryption algorithm **Enc** to encrypt a vector message  $\mathbf{x}_i$  using encryption key  $\text{ek}_i$ . For *correctness*, given a list of ciphertexts of all  $n$  clients under the same tag, the decryption algorithm can decrypt using any one of the decryption key  $\text{sk}_{\text{id}, \mathbf{y}}$  to obtain the function evaluation on all client inputs  $\mathbf{x}_1 \parallel \dots \parallel \mathbf{x}_n$  with functional parameters  $\mathbf{y}$ . The tracing algorithm **Trace** takes tracing key  $\text{tk}$ , two adversarially chosen challenges  $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$ , a challenge tag and some success parameter  $y$ , and is given (black-box) oracle access to a pirate decoding algorithm  $D$ :  $\text{Trace}^D(\text{tk}, 1^y, \text{tag}, \mathbf{x}^{(0)}, \mathbf{x}^{(1)})$ . It outputs a set  $S \subseteq \mathcal{ID}$  of users signaling that the keys  $\text{sk}_j$  for  $j \in S$  were used to create the pirate decoder  $D$ . Two essential *security properties* include the *security against chosen-plaintext attacks* (IND-CPA) and the *tracing security*. The IND-CPA follows as that of the “vanilla” MCFE, where the adversary is not only allowed to

request decryption keys (corrupting multiple users) but also allowed to corrupt the clients to obtain their encryption keys (corrupting multiple senders). The *tracing security* of the system states that: (i) the tracing algorithm outputs *with high probability* at least one identity from the set of colluding users, and (ii) without false accusations of honest users, except with negligible probability, and (iii) no constraints whatsoever are put on the key queries  $\mathbf{y}_i$ . We emphasize that our (iii) make our notion stronger than previous single-sender notion. Indeed, from [27] that improves on [18], security model allows the adversary to request multiple decryption keys adaptively for a set of functional vectors  $\{\mathbf{y}_i\}_i$ , while still requiring that each queried  $\mathbf{y}_i$  satisfies  $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle \neq \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$  for the two chosen messages  $(\mathbf{x}^{(0)} \neq \mathbf{x}^{(1)})$ <sup>8</sup>.

Our observation: Relation to Strong Admissibility for IND-CPA. A generic approach to prove tracing security is to reduce it to the IND-CPA security of the encryption scheme. At the heart of the IND-CPA is the notion of *admissibility* condition for adversary, as with all security models for advanced encryption notion, where we have to exclude attacks that trivially use the nature of the function class to break *any efficient schemes*. An attack is *admissible* if it does require the specific details of the scheme to succeed. The evolution of how *admissible* an adversary can be with respect to the security notion is highlighted in the seminal work on MCFE [14] and follow-up works to enhance MCFE with a more fine-grained corruption model [5, 30]. A key aspect of these models involves how to appropriately constrain the adversary’s behavior when it corrupts the secret encryption keys of some clients  $i$ , given access to the **Corrupt** oracle. Over time, the admissibility conditions tied to such corruptions have become more expressive. We take the case of MCFE computing  $\sum_i \langle \mathbf{x}_i, \mathbf{y}_i \rangle$  from inputs  $\mathbf{x}_i$  of client  $i$ . In this paper we refer to the admissibility condition in [30] ( $\langle \mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)}, \mathbf{y}_i \rangle = 0$  for any corrupted client  $i$ ) as *strong admissibility*, meanwhile the admissibility condition in [14] and a lot of the follow-ups [1, 2, 5, 15, 16, 25] ( $\mathbf{x}_i^{(0)} = \mathbf{x}_i^{(1)}$  whenever the client  $i$  is corrupted) as *weak admissibility*. Under the *weak admissibility* condition when the adversary corrupts a client  $i$  thereby obtaining the secret encryption key  $\mathbf{ek}_i$ , IND-CPA security restricts the adversary to submit *identical* messages as challenge inputs for slot  $i$ . Then, if during the tracing security game that  $\mathbf{ek}_i$  is corrupted to construct the pirate decoder, and the tracing adversary outputs a pair of different challenge messages on  $i$ , there exists no reduction that can transform different challenge messages from tracing security into two identical messages for IND-CPA security<sup>9</sup>. However, this issue does *not* arise when employing the *strong admissibility* condition, since the adversary can still output two different messages as challenge. As a consequence, it turns out that

<sup>8</sup> This restriction condition was removed in the work of [19] in the same single-sender setting.

<sup>9</sup> We note that this argument is applicable if we want to *generically* base tracing security on IND-CPA security. Other approaches, for example, basing tracing security from some concrete hardness assumptions, might evade this barrier.

this strong (and optimal) admissibility notion studied in [30] is essential for a meaningful notion of traitor tracing FE in the multi-client setting.

*Our Characterization Result: Black-box but Private Tracing.* Another aspect of the tracing security is the *black-box* nature of the tracing algorithm, that is, the tracing algorithm ignores the implementation of the pirate decoder, only makes queries to the decoder and observes the outputs. This limits the information that is available to the tracing algorithm, and thus makes the tracing security more resilient. For the case of MCFE, it turns out that: *given function classes with some solvability properties (see Definition 6 for formal definition), public tracing with black-box access to the pirate decoder is not possible, assuming the MCFE scheme is IND-CPA secure under strong admissibility.* This result is proved in Theorem 1. Implications and discussion on the theorem are also given in Section 4.2.

## 2.2 Our Framework to TTMCFE: Black-box but Private Tracing

In Section 5, we generalize the framework in [23] to the case of bounded EITT MCFE with: the definition of *private linear broadcast encryption* to the multi-client functional encryption setting (PL-MCFE), and show how to use PL-MCFE to construct bounded EITT MCFE. Details of intermediate steps are given through the sequence: our LWE-based AB-MCFE ([17, App. F]) + our core compiler (Section 6)  $\rightarrow$  EI-PL-MCFE ([17, App. E])  $\rightarrow$  Indexed EITT MCFE ([17, App. D])  $\rightarrow$  bounded EITT MCFE ([17, App. C]). Notably, Section 6 presents our main compiler that achieves strong from weak admissible security for AB-MCFE computing  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$ . This compiler is applied to our main building block, LWE-based AB-MCFE, for which we highlight our key ideas below.

*Our main building block: constructing AB-MCFE ([17, App. F]).* One cornerstone in our construction is an AB-MCFE scheme for inner-product. Our AB-MCFE for inner-product needs to satisfy the following requirements:

- R.1** Support a specific access control where all clients have the same attribute, evaluated over the same policy function. This is enough for our application of building EI-PLMCFE in [17, App. E], as the attributes are chosen by the reduction.
- R.2** Adaptive corruption, which follows from the fact that the adversary in our tracing model can corrupt any client adaptively before outputting a pirate decoder.
- R.3** Security against an adversary who can ask polynomially-many number of challenge queries on arbitrary challenge plaintext pairs and arbitrary tag. In the IND-CPA security game of MCFE, this is captured as allowing repetition on both private-inputs (i.e., the plaintexts) and public-inputs (i.e., the tags) in the admissibility condition <sup>10</sup>.

<sup>10</sup> This is due to the fact that in our tracing definitions (Definition 3), the adversary has no restriction on what it can ask to the KeyGen and Enc oracle before outputting

Requirements (R.1) and (R.2) can be obtained using known techniques. Even though we do not know how to achieve requirement (R.3) in the most general case, our key idea is to use requirement (R.1) to reduce requirement (R.3) to a weaker one with private-inputs repetition. Our key novel idea is: repetition queries are only useful when the decryption key is decryptable (i.e., the *same* attribute of the ciphertext, following (R.1), satisfies the policy of the decryption key). Thus, we can consider the *same* attribute as a part of the tag for each encryption. The adversary is then forced to use the same attribute and the same tag for each *complete* challenge query. This effectively prevents the adversary from taking the advantages of mix-and-match attacks. Consequently, we only need to consider the weak security with private-inputs only repetition, where the adversary can ask many challenge queries for a single tag. In [17, App. F] we construct a MCFE  $\mathcal{E}^w$  for  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$  that is IND-CPA secure under the *weak admissibility*. In short, each client  $i$  in our weak MCFE scheme  $\mathcal{E}^w$ , encrypts their  $\text{ek}_i, \mathbf{x}_i, (\text{tag}, \text{ac-ct}_i)$  using the AB-IPFE encryption  $\text{Enc}^{\text{one-slot}}(\text{pp}_i, \star)$  from [24] by:

$$\begin{aligned} \mathcal{E}^w.\text{Enc}(\text{ek}_i := (\text{pp}_i^{\text{one-slot}}, \{k_{i,j}\}_j), \mathbf{x}_i, (\text{tag}, \text{ac-ct}_i)) &\rightarrow \text{ct}_{i,\text{tag}} \\ \text{where } \text{ct}_{i,\text{tag}} &:= \mathcal{E}^{\text{one-slot}}.\text{Enc}(\text{pp}_i^{\text{one-slot}}, \mathbf{w}_i, (\text{tag}, \text{ac-ct}_i)) , \\ \text{where } \mathbf{w}_i &:= [\mathbf{0} \parallel \dots \parallel \mathbf{0} \parallel \mathbf{x}_i \parallel \mathbf{0} \parallel \dots \parallel \mathbf{0}] + \mathbf{t}_{i,\text{tag}} , \\ \text{with keys: } \text{dk}_{\mathbf{y}, \text{ac-k}} &\leftarrow \mathcal{E}^w.\text{KeyGen}(\text{msk} := \text{msk}^{\text{one-slot}}, \mathbf{y}, \text{ac-k}) \\ &:= \mathcal{E}^{\text{one-slot}}.\text{KeyGen}(\text{msk}, \mathbf{y}, \text{ac-k}) , \end{aligned} \quad (1)$$

and  $\mathbf{t}_{i,\text{tag}} \leftarrow \sum_{j \neq i} (-1)^{j < i} \text{PRF}(k_{i,j}, \text{tag})$ , where PRF keys  $k_{i,j}$  of each  $i$ -th client is generated by a pseudorandom zero-sharing scheme [1]. The  $\text{ek}_i$  is private for each  $i$  so that the PRF keys are secret. The *correctness* can be verified whenever the attributes  $(\text{ac-ct}_i)_i$  satisfy the policy  $\text{ac-k}$  in a key  $((\mathbf{y}_i)_i, \text{ac-k})$ . The CPA-security is ensured by the CPA-security of the underlying single-client FE scheme  $\text{Enc}^{\text{one-slot}}(\text{pp}_i, \star)$  for inner products with access control, after applying the PRF security to the masks. Last but not least, a subtle observation is also needed to handle the challenge of different tag/attribute/message repetition in the challenge phase, which we also use ubiquitously in this work.

### 2.3 Our Generic Compiler: From Weak to Strong Admissibility for Inner Product with Fine-Grained Access Control

The main technical ingredient for our final concrete traceable MCFE is a MCFE scheme satisfying the strong admissibility condition for a concrete functionality:  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$  that *integrates access control into inner product compu-*

---

its pirate decoder, along with a pair of challenge plaintexts and a tag. We note that this is stronger than what is typically considered in the IND-CPA security for MCFE, where the adversary is not allowed to learn encryption of plaintexts under its challenge tag.

tation, (see Definition 2, Definition 1 presents  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}}$ ), as initiated in previous works [4, 28, 29]. Each client encrypts their individual vector  $\mathbf{x}_i$  of length  $N_i$  together with some attributes  $\text{ac-ct}_i$ . The functional key is associated with  $[\mathbf{y}_1 \| \dots \| \mathbf{y}_n]$  and some policy  $\text{ac-k}$ . The decryption algorithm outputs  $\sum_i \langle \mathbf{x}_i, \mathbf{y}_i \rangle$  if the policy  $\text{ac-k}$  is satisfied by the attributes  $(\text{ac-ct}_i)_{i=1}^n$ . We first recall the admissibility conditions for the concrete functions  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$  from [30], then describe our compiler.

- S.1** For all challenge vectors data-attribute  $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}, (\text{tag}, \text{ac-ct}_i))$  that is queried to **LoR**, for all function-policy  $((\mathbf{y}_i)_{i \in [n]}, \text{ac-k})$  to **KeyGen**, for all corrupted  $i$ , up to repetitions, it holds that:  $\langle \mathbf{x}_i^{(b)} - \mathbf{x}_i^{(1)}, \mathbf{y}_i \rangle = 0$ .
- S.2** For all challenge vectors data-attribute  $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}, (\text{tag}, \text{ac-ct}_i))$  that is queried to **LoR**, for all function-policy  $((\mathbf{y}_i)_{i \in [n]}, \text{ac-k})$  to **KeyGen** that can decrypt the challenge ciphertexts, *i.e.* the policy  $\text{ac-k}$  is satisfied by the attributes  $(\text{ac-ct}_i)_{i \in [n]}$ , summing over all *honest*  $i$ , up to repetitions of  $\mathbf{x}_i$  at each  $i^{11}$ , does *not* differ between challenge bit  $b = 0$  and  $b = 1$ .

When considering the *weak* admissibility condition from [14], condition **S.1** is replaced by

- W.1** For all challenge vectors data-attribute  $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}, (\text{tag}, \text{ac-ct}_i))$  that is queried to **LoR**, for all *corrupted*  $i$ , the two challenge components at  $i$ , up to repetitions, are equal:  $\mathbf{x}_i^{(b)} = \mathbf{x}_i^{(1)}$ .

The *strong* admissibility condition **S.1** is *less restrictive* than the *weak* admissibility condition **W.1**, thus leading to a more robust security notion that can cover *more attacks*. We use as an example in this overview the MCFE  $\mathcal{E}^w$  for  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$  from Section 2.2.  $\mathcal{E}^w$  is IND-CPA secure under the *weak admissibility* of (**W.1, S.2**), and our compiler leverages it to IND-CPA under (**S.1, S.2**).

We make use of two structural properties of  $\mathcal{E}^w$ : (i) *encryption and decryption can be done partially, before a global final linear combination* and (ii)  $\mathcal{E}^w$  has *two-step decryption* (a formal definition can be found in [1, Def. 3.1]). Roughly, we modify  $\mathcal{E}^w$  step-by-step as follows, new modifications are boxed:  $r_i$  is random

$$\begin{aligned}
& \mathcal{E}.\text{Enc}(\text{ek}_i, \mathbf{x}_i, (\text{tag}, \text{ac-ct}_i)) \rightarrow \text{ct}_{i, \text{tag}} := (\text{ct}_i^w, \text{ct}_i^{\text{als-ip}}) \\
& \text{where } \text{ct}_i^w \leftarrow \mathcal{E}^w.\text{Enc}(\text{ek}_i^w, \text{tag}, \boxed{\mathbf{x}_i - r_i \cdot \mathbf{v}_i}, \text{ac-ct}_i) , \\
& \text{and } \boxed{\text{ct}_i^{\text{als-ip}}} \leftarrow \mathcal{E}^{\text{als-ip}}.\text{Enc}(\text{pk}_i^{\text{als-ip}}, \boxed{r_i \cdot \mathbf{t}_i}) \\
& \text{with: } \mathcal{E}.\text{KeyGen}(\text{msk}, \mathbf{y}, \text{ac-k}) \rightarrow \text{dk}_{\mathbf{y}, \text{ac-k}} := (\text{dk}_{\mathbf{y}, \text{ac-k}}^w, (\text{dk}_i^{\text{als-ip}})_i) \\
& \text{where } \mathbf{y} := [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{Z}_q^{n \cdot N} \\
& \text{and } \text{dk}_{\mathbf{y}, \text{ac-k}}^w \leftarrow \mathcal{E}^w.\text{KeyGen}(\text{msk}^w, \mathbf{y}, \text{ac-k}) \\
& \text{and } \boxed{\text{dk}_i^{\text{als-ip}}} \leftarrow \mathcal{E}^{\text{als-ip}}.\text{KeyGen} \left( \text{msk}_i^{\text{als-ip}}, \left( \frac{\mathbf{v}_i[k]}{\mathbf{t}_i[k]} \right)_{k=1}^N \circ \mathbf{y}_i \right)
\end{aligned}$$

<sup>11</sup> This is a strong security guarantee that allows the adversary to reuse the tag at slots  $i$ , which is considered standard in the context of MCFE [5, 29, 31].

where, as previously mentioned, the function is written  $\mathbf{y} = [\mathbf{y}_1 \parallel \dots \parallel \mathbf{y}_n]$ . The *correctness* follows that of the single-client IPFE scheme  $\mathcal{E}^{\text{als-ip}}$ , and the correctness of  $\mathcal{E}^w$  notably with the preserved structural properties. Theorem 2 proves IND-CPA under (S.1, S.2). We highlight the main points below. Let  $b \xleftarrow{\$} \{0, 1\}$  denote the challenge bit. At an intuition level, the challenge at  $i$ , where  $i$  is corrupted and  $\mathbf{x}_i^{(0)} \neq \mathbf{x}_i^{(1)}$  to **LoR**, contains  $\text{ct}_{i, \text{tag}}^{(b)} = (\text{ct}_i^{(w, b)}, \text{ct}_i^{\text{als-ip}})$  and  $\text{ct}_i^{(w, b)} \leftarrow \mathcal{E}^w.\text{Enc}(\text{ek}_i^w, \text{tag}, \boxed{\mathbf{x}_i^{(b)} - r_i \cdot \mathbf{v}_i}, \text{ac-ct}_i)$ . This ensures that the difference  $\Delta \mathbf{x} := \mathbf{x}_i^{(1)} - \mathbf{x}_i^{(b)}$ <sup>12</sup> is hidden in the randomness  $r_i \cdot \mathbf{v}_i$ . Under partial decryption, this leaks  $\langle \mathbf{x}_i^{(1)}, \mathbf{y}_i \rangle - r_i \cdot \langle \mathbf{v}_i, \mathbf{y}_i \rangle - \underbrace{\langle \Delta \mathbf{x}_i, \mathbf{y}_i \rangle}_{=0 \text{ thanks to S.1}}$  making the quantity independent

of  $b$ . The new challenges are  $\mathbf{x}_i^{(b)} - r_i \cdot \mathbf{v}_i$  and identical, thus one can reduce to IND-CPA under W.1. This step is formalized using the two-step decryption property under IND-CPA of  $\mathcal{E}^w$ . For the above to work, we need to ensure that the IPFE layer *leaks nothing but the inner product evaluation*, which is solely needed for error-correcting the two-step decryption. This leads to an exigence of *simulation-based* security for the IPFE scheme  $\mathcal{E}^{\text{als-ip}}$ , else an indistinguishability-based *cannot* guarantee security of  $\boxed{r_i \cdot \mathbf{t}_i}$ , conditioned on the value  $\langle r_i \cdot \mathbf{t}_i, (\frac{\mathbf{v}_i[k]}{\mathbf{t}_i[k]})_{k=1}^N \circ \mathbf{y}_i \rangle$  where the adversary knows  $\boxed{(\frac{\mathbf{v}_i[k]}{\mathbf{t}_i[k]})_{k=1}^N \circ \mathbf{y}_i}$  due to *corruption*. The simulator of  $\mathcal{E}^{\text{als-ip}}$  is used to simulate  $\text{Sim}^{\text{als-ip}}.\text{Enc}(\text{pk}_i^{\text{sim-als}}, \boxed{r_i \cdot \mathbf{t}_i})$ , knowing the error-correcting values expected at the *corrupted client*  $i$  is  $\langle r_i \cdot \mathbf{t}_i, (\frac{\mathbf{v}_i[k]}{\mathbf{t}_i[k]})_{k=1}^N \circ \mathbf{y}_i \rangle$  while leaking nothing else. Last but not least, when the functional keys does not decrypt the challenge ciphertexts, the admissibility condition S.2 over the sum of honest  $i$  is irrelevant, corrupted  $i$  is already treated above, and the simulation security of  $\mathcal{E}^{\text{als-ip}}$  also extends to this case. In the end, the details of our transformation are given in Section 6. By combining with our MCFE  $\mathcal{E}^w$  from LWE, the simulation secure IPFE scheme  $\mathcal{E}^{\text{als-ip}}$  from LWE in [6, 7], the LWE-based PRFs from [8, 9], we arrive at the first LWE-based MCFE for the class of inner products with access control that satisfies the *strong* admissibility condition. Remark 2 discusses our improvements from this compiler over [30].

### 3 Preliminaries

*Notation.* We denote the security parameter by  $\lambda$ . A polynomial time algorithm  $A$  runs in time polynomial in the (implicit) security parameter  $\lambda$ . We denote “probabilistic polynomial time” by PPT. A function  $f(\lambda)$  is negligible in  $\lambda$  if it is  $\mathcal{O}(\lambda^{-c})$  for every  $c \in \mathbb{N}$ . We write  $f = \text{negl}(\lambda)$  for short. Similarly, we write  $f = \text{poly}(\lambda)$  if  $f(\lambda)$  is a polynomial with variable  $\lambda$ . If  $D$  is a probability distribution,  $x \leftarrow D$  means that  $x$  is sampled from  $D$  and if  $S$  is a set,  $x \leftarrow S$  means that  $x$  is sampled uniformly and independently at random from  $S$ . We also write  $|S|$  for the cardinality of set  $S$ . We denote with  $[n]$  the set  $\{1, \dots, n\}$  for  $n \in \mathbb{N}$ . We use bold font for (row) vectors, and capitals for matrices. The

<sup>12</sup> This is  $\mathbf{0}$  when  $b = 1$ , possibly non-zero otherwise.

formal definitions of all cryptographic primitives considered in this work are given in [17, App. A]. For the sake of readability, we will recall them in the main body of the paper when necessary.

**Definition 1 (Inner-Product Functionality).** *Let  $\lambda \in \mathbb{N}$ . Let  $B = B(\lambda)$ ,  $n = n(\lambda)$ ,  $N_1(\lambda), \dots, N_n(\lambda)$  be polynomials and  $\mathcal{D}_i = [-B; B]^{N_i}$  for  $i \in [n]$ . We denote by  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} = \{f_{\mathbf{y}_1, \dots, \mathbf{y}_n} : \mathbf{y}_1 \in \mathcal{D}_1, \dots, \mathbf{y}_n \in \mathcal{D}_n\}$  the family of functions where  $f_{\mathbf{y}_1, \dots, \mathbf{y}_n} : \mathcal{D}_1 \times \dots \times \mathcal{D}_n \rightarrow \mathbb{Z}_p$  is defined as  $f_{\mathbf{y}_1, \dots, \mathbf{y}_n}(\mathbf{x}_1, \dots, \mathbf{x}_n) := \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$ . We also use  $\mathbf{x}$  without a subscript to denote the concatenation of  $n$  slots, that is,  $\mathbf{x} := \mathbf{x}_1 \parallel \dots \parallel \mathbf{x}_n$ .*

**Definition 2 (Attribute-based Functionality).** *The function class  $\mathcal{F} \times \text{AC-K}$  for functional evaluation with fine-grained access control is described as:*  
*(i) The function class  $\mathcal{F} \times \text{AC-K}$  contains  $(F_\lambda, \text{ac-k})$  having public attributes  $(\text{AC-Ct}_i)_{i \in [n]}$ . (ii) The function  $F_\lambda$  has domain  $\mathcal{D}_{\lambda,1} \times \dots \times \mathcal{D}_{\lambda,n}$  and range  $\mathcal{R}_\lambda$ . (iii) The public attributes of each  $i$  come from  $\text{Tag} \times \text{AC-Ct}_i$  for some set  $\text{AC-Ct}_i$ . (iv) The access control is defined via a relation  $\text{Rel} : \text{AC-K} \times \text{AC-Ct}_1 \times \dots \times \text{AC-Ct}_n \rightarrow \{0, 1\}$ , for some set  $\text{AC-K}$ . Evaluating  $(F_\lambda, \text{ac-k})$  on  $((x_i, \text{ac-ct}_i)_{i=1}^n, \text{ac-k})$  gives  $F_\lambda(x_1, \dots, x_n)$  if  $\text{Rel}(\text{ac-k}, (\text{ac-ct}_i)_i) = 1$ , otherwise  $\perp$ .*

## 4 Definitions: MCFE Traitor Tracing with Embedded Identities

### 4.1 Embedded-Identity Traitor Tracing

We will now present the syntax and definitions for general traitor tracing with embedded identities. In this work, we only consider the *bounded* embedded identity setting, where we have an a priori bound  $n_{\text{bd}}$  that is fixed during setup, and security is guaranteed only if the adversary gets at most  $n_{\text{bd}}$  secret keys. We also give definition for an intermediate notion of *indexed* embedded identity. In the following, the grey-boxed text is considered for *bounded* embedded identity traitor tracing, and the boxed text is considered for *indexed* embedded identity traitor tracing.

**Definition 3 ((Bounded Indexed) Embedded Identities Traceable Multi-Client Functional Encryption).** *A bounded indexed keys, embedded-identities tracing scheme  $\mathcal{T}$  for a function class  $\mathcal{F}$  consists of five algorithms (Setup, KeyGen, Enc, Dec, Trace) that are defined below:*

$(\text{msk}, \text{pp}, (\text{ek}_i)_{i \in [n]}, \text{tk}) \leftarrow \text{Setup}(1^\lambda, 1^n, 1^\kappa, \text{}n_{\text{bd}}\text{}, \text{}n_{\text{indx}}\text{})$ : *Given as input the security parameter  $\lambda$ , the number of clients  $n$ , the identity space parameter  $\kappa$ , and bound on number of key queries  $n_{\text{bd}}$  index space  $n_{\text{indx}}$ , output a master secret key  $\text{msk}$ , public parameters  $\text{pp}$ ,  $n$  encryption keys  $(\text{ek}_i)_{i \in [n]}$  and a tracing key  $\text{tk}$ .*

$\text{dk}_{\text{id}, \text{}j\text{}, F_\lambda}} \leftarrow \text{KeyGen}(\text{msk}, \text{id}, \text{}j\text{}, F_\lambda)$ : *Given the master secret key  $\text{msk}$ , an identity  $\text{id} \in \{0, 1\}^\kappa$ , an index  $j \in [n_{\text{indx}}]$ , and a function description  $F_\lambda \in \mathcal{F}$ , output a decryption key  $\text{dk}_{\text{id}, \text{}j\text{}, F_\lambda}}$ .*

- **Initialise**( $1^\lambda$ ) :  $b \xleftarrow{\$} \{0, 1\}$ ;  $(\text{msk}, (\text{ek}_i)_{i \in [n]}) \leftarrow \text{Setup}(1^\lambda)$ ;  $\mathcal{Q} := \emptyset$ ,  $\mathcal{C} := \emptyset$ ,  $\mathcal{H} := [n]$ ; Return  $\text{pk}$ .
- **LoR**( $i, x_i^{(0)}, x_i^{(1)}, (\text{tag}^*, \text{ac-ct}_i^{(\text{chal})})$ ) :  $\text{Enc}(\text{ek}_i, x_i^{(b)}, (\text{tag}^*, \text{ac-ct}_i^{(\text{chal})})) \rightarrow \text{ct}_{\text{tag}^*, i}^{(b)}$ ; Return  $\text{ct}_{\text{tag}^*, i}^{(b)}$ .
- **Enc**( $i, x_i, (\text{tag}, \text{ac-ct}_i)$ ) : Return  $\text{Enc}(\text{ek}_i, x_i, (\text{tag}, \text{ac-ct}_i))$ .
- **Corrupt**( $i$ ) :  $\mathcal{C} := \mathcal{C} \cup \{i\}$ ;  $\mathcal{H} := \mathcal{H} \setminus \{i\}$ ; Return  $\text{ek}_i$ .
- **Finalise**( $b'$ ) : If  $\mathcal{A}$  is NOT admissible w.r.t  $(\mathcal{C}, \mathcal{Q}, \mathcal{H})$ , return  $b' \xleftarrow{\$} 0$ ; Else return  $(b' \stackrel{?}{=} b)$ .
- **KeyGen**( $F, \text{ac-k}$ ) :  $\mathcal{Q} := \mathcal{Q} \cup \{(F, \text{ac-k})\}$ ;  $\text{dk}_{F, \text{ac-k}} \leftarrow \text{KeyGen}(\text{msk}, F, \text{ac-k})$ ; Return  $\text{dk}_{F, \text{ac-k}}$ .

**Fig. 1.** The security games  $\text{Expr}_{\mathcal{E}, \mathcal{F}, \mathcal{A}}^{\text{mc-ind-cpa}}(1^\lambda)$  for Definition 4.

$\text{ct}_{\text{tag}, i} \leftarrow \text{Enc}(\text{pp}, \text{ek}_i, \text{tag}, \mathbf{x}_i)$ : Given as inputs the public parameters  $\text{pp}$ , an encryption key  $\text{ek}_i$ , a message  $\mathbf{x}_i \in \mathcal{D}_{\lambda, i}$ , and a tag  $\text{tag}$ , output a ciphertext  $\text{ct}_{\text{tag}, i}$ .

$z \leftarrow \text{Dec}(\text{pp}, \text{dk}_{F_\lambda}, \mathbf{c})$ : Given the public parameters  $\text{pp}$ , a decryption key  $\text{dk}_{F_\lambda}$  and a vector of ciphertexts  $\mathbf{c} := (\text{ct}_{\text{tag}, i})_i$  of length  $n$ , output an element  $z \in \mathcal{R}_\lambda \cup \{\perp\}$ .

$T \leftarrow \text{Trace}^D(\text{tk}, 1^y, \text{tag}, \mathbf{x}^{(0)}, \mathbf{x}^{(1)})$ : Given oracle access to a program  $D$ , the tracing key  $\text{tk}$ , parameters  $y$ , a tag  $\text{tag}$  and two messages  $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$ , where  $\mathbf{x}^{(b)}$  is the concatenation of  $n$  encryption slots, that is,  $\mathbf{x}^{(b)} := \mathbf{x}_1^{(b)} \parallel \dots \parallel \mathbf{x}_n^{(b)}$ , output a set  $T$  of identities where  $T \subseteq \{0, 1\}^\kappa$ .  
We call the tracing as public or private depending on whether  $\text{tk}$  is equal to  $\text{pp}$  or it is kept secret.

The correctness requirement and the IND-CPA security definition is identical to the one in Definition 4.

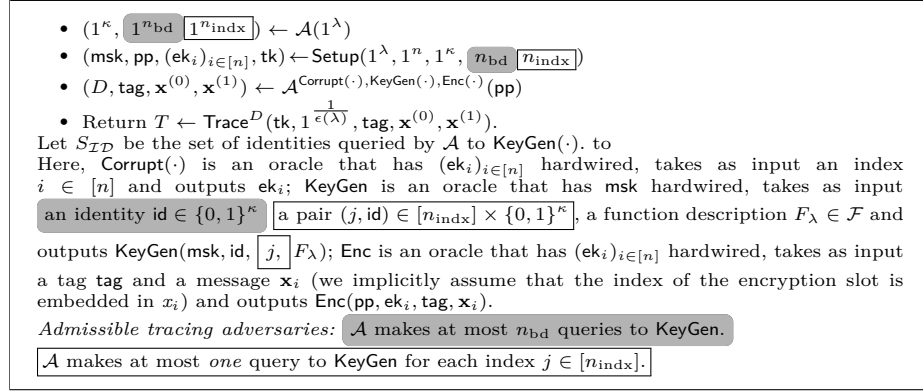
**Definition 4 (Strong IND-security for MCFE).** An MCFE scheme  $\mathcal{E} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$  for the function class  $\mathcal{F} = \{F_\lambda\}_{\lambda \in \mathbb{N}}$  is  $\text{xx}$ -secure if for all PPT adversaries  $\mathcal{A}$ , and for all sufficiently large  $\lambda \in \mathbb{N}$ , the following probability is negligible

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

The game  $\text{Expr}_{\mathcal{E}, \mathcal{F}, \mathcal{A}}^{\text{mc-ind-cpa}}(1^\lambda)$  is depicted in Figure 1. Strong admissibility is overviewed in (S.1, S.2) of Section 2 and formal definitions are in our full version [17, Def. 8]. The probability is taken over the random coins of  $\mathcal{A}$  and the algorithms.

We state here the tracing security.

**Definition 5 (Security of Tracing).** For any non-negligible function  $\epsilon(\cdot)$ , polynomial  $p(\cdot)$ , and all PPT adversary  $\mathcal{A}$ , consider the experiment  $\text{Expt}_{\mathcal{T}, \mathcal{A}, \epsilon}^{\text{TT}}(1^\lambda)$  defined in Fig. 2. We define the following events and corresponding probabilities (which are a function of  $\lambda$ , parameterized by  $\mathcal{A}, \epsilon$ ):

Fig. 2. Experiment  $\text{Expt}^{\text{TT}}$ .

- **GoodDec:**  $\Pr \left[ \begin{array}{l} D((\text{ct}_{\text{tag}, i})_{i \in [n]}) = b \mid \\ b \leftarrow_{\$} \{0, 1\}, \text{ct}_{\text{tag}, i} \leftarrow \text{Enc}(\text{pp}, \text{ek}_i, \text{tag}, \mathbf{x}_i^{(b)}) \end{array} \right] \geq \frac{1}{2} + \epsilon(\lambda).$   
 $\Pr\text{-GoodDec}_{\mathcal{A}, \epsilon}(\lambda) = \Pr[\text{GoodDec}].$
- **CorrectTr:**  $T \neq \emptyset \wedge T \subseteq S_{\text{TD}}.$   
 $\Pr\text{-CorrectTr}_{\mathcal{A}, \epsilon}(\lambda) = \Pr[\text{CorrectTr}].$
- **FalseTr:**  $T \not\subseteq S_{\text{TD}}.$   
 $\Pr\text{-FalseTr}_{\mathcal{A}, \epsilon}(\lambda) = \Pr[\text{FalseTr}].$

A traitor tracing scheme  $\mathcal{T}$  is said to be secure if for every PPT admissible adversary  $\mathcal{A}$ , polynomial  $q(\cdot)$  and non-negligible function  $\epsilon(\cdot)$ , there exist negligible functions  $\text{negl}_1(\cdot), \text{negl}_2(\cdot)$  such that for all  $\lambda \in \mathbb{N}$  and  $\epsilon(\lambda) > \frac{1}{q(\lambda)}$ , it holds that  $\Pr\text{-FalseTr}_{\mathcal{A}, \epsilon}(\lambda) \leq \text{negl}_1(\lambda)$  and  $\Pr\text{-CorrectTr}_{\mathcal{A}, \epsilon}(\lambda) \geq \Pr\text{-GoodDec}_{\mathcal{A}, \epsilon}(\lambda) - \text{negl}_2(\lambda)$ .

We note that in the tracing security game of Figure 2, the adversary can extract separately secret information from both senders (via **Corrupt** to obtain secret encryption keys) and receivers (via **KeyGen** to obtain functional keys).

*Remark 1 (On the case of decoders that use only corrupted encryption keys).* We remark that the tracing security definition does *not* require the traitor tracing algorithm to identify corrupted encryption keys. Fortunately, the IND-CPA security of the scheme ensures that, in the extreme case where the adversary corrupts only encryption keys, a pirate decoder *cannot* be good. Consequently, if the adversary is able to produce a good decoder, it must have corrupted at least one decryption key. In the general case where the adversary corrupts both encryption and decryption keys, the tracing algorithm can be used to identify all corrupted decryption keys, leaving the decoder – with solely corrupted encryption keys embedded – ineffective.

## 4.2 Impossibility

In the following we refer to queries of functions that do *not* differ on the challenge messages  $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$  as *non-differentiating* queries. We require some structural properties of the function class, given *non-differentiating* functions and the number of clients  $n \geq 2$ , in the following sense:

**Definition 6 (Slot-Differentiating Solvability).** *Let  $\lambda \in N$  and  $\mathcal{F}$  be a function class parameterized by  $\lambda$  containing  $F_\lambda : \mathcal{D}_1 \times \dots \times \mathcal{D}_n \rightarrow \mathcal{R}_\lambda$ , where  $n = \text{poly}(\lambda) \geq 2$ . We say that  $\mathcal{F}$  is slot-differentiating solvable if there exists  $F_\lambda \in \mathcal{F}$ , and fixed inputs  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)} \in \mathcal{D}_1 \times \dots \times \mathcal{D}_n$ , such that  $F(\mathbf{x}^{(0)}) = F(\mathbf{x}^{(1)})$  and there exists a PPT algorithm that given  $(F, \mathbf{x}^{(0)}, \mathbf{x}^{(1)})$  outputs a set of indices  $\emptyset \neq \mathcal{S} \subseteq [n]$  and  $(\tilde{\mathbf{x}}_i^{(0)}, \tilde{\mathbf{x}}_i^{(1)})_{i \in \mathcal{S}}$  satisfying the following  $F(\Delta_{\mathcal{S}}(k, (\tilde{\mathbf{x}}_i^{(0)})_{i \in \mathcal{S}}, (\mathbf{x}_i^{(0)})_{i \in [n] \setminus \mathcal{S}})_{i=1}^n) \neq F(\Delta_{\mathcal{S}}(k, (\tilde{\mathbf{x}}_i^{(1)})_{i \in \mathcal{S}}, (\mathbf{x}_i^{(1)})_{i \in [n] \setminus \mathcal{S}})_{i=1}^n)$ , where for  $b \in \{0, 1\}$*

$$\forall k \in [n] : \Delta_{\mathcal{S}}\left(k, (\tilde{\mathbf{x}}_i^{(b)})_{i \in \mathcal{S}}, (\mathbf{x}_i^{(b)})_{i \in [n] \setminus \mathcal{S}}\right) := \begin{cases} \tilde{\mathbf{x}}_k^{(b)} & \text{if } k \in \mathcal{S} \\ \mathbf{x}_k^{(b)} & \text{if } k \in [n] \setminus \mathcal{S} \end{cases}.$$

For the particular case of inner products  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}}$  that we consider in this work,  $\mathcal{A}$  can even find  $(F, \mathcal{S}, (\tilde{\mathbf{x}}_i^{(0)}, \tilde{\mathbf{x}}_i^{(1)})_{i \in \mathcal{S}})$  on-the-fly by solving a linear system, for which we use inherently the condition that  $n \geq 2$  as the number of clients/slots. In particular, we show an impossibility that there *cannot* exist a IND-CPA secure traitor tracing MCFE (following Definition 5), under strong admissibility, with a *public* tracing algorithm, for both bounded and indexed embedded identity traitor tracing.

**Theorem 1.** *Let  $\mathcal{T}$  be a traitor tracing MCFE scheme, for at least 2 clients and for a slot-differentiating solvable function class  $\mathcal{F}$ , so that  $\mathcal{T}$  is correct<sup>13</sup> with public black-box tracing. If  $\mathcal{T}$  is traceable (as per Definition 5), then  $\mathcal{T}$  is not IND-CPA secure as per Definition 4.*

As an immediate corollary of Theorem 1, we have to restrict the tracing algorithm to be *private* in order to have a secure traitor tracing MCFE scheme. Furthermore, two other implications of our theorem are worth noting: (1) Our IND-CPA adversary in the proof does *not* need to corrupt *secret* encryption keys for the clients. This allows us to extend the theorem even to a setting where no encryption keys can be corrupted by the tracing adversary. (2) The tracer succeeds even when running with *black-box* access to the decoder, which also covers *white-box* public tracing.

## 5 Framework for Embedded-Identity Traitor Tracing MCFE

Next, we give an overview of the framework that constructs bounded embedded-identity traitor tracing MCFE. Previous works of [22, 23] provides a framework

<sup>13</sup> We do not require correctness to hold with probability 1, *i.e.* perfect, over the random coins of algorithms. Overwhelming probability for correctness suffices.

to build *embedded-identity traitor tracing* (EITT) in the setting of *single-client*, that is there is only one sender that encrypts the message, facing *multiple users* that can decrypt on keys embedded with identities. A constructive framework for our *embedded-identity traitor tracing MCFE* (EITT-MCFE) will be the first to tackle the *multi-client* context, and face the following challenges:

- We choose to trace *privately* and preferably in a *black-box* manner (this is due to our impossibility Theorem 1).
- The complexity of *private-key with corruption* of multi-clients.
- The *fine-grained separate corruptions* of clients and functional keys under the necessity of *strong admissibility*.

*Our generalization of [23] to Traitor Tracing MCFE.* We generalize the framework of [23] to the setting of *multi-client functional encryption*, with traitor tracing capabilities facing *multi-sender* that can ask *bounded* number of functional keys. The final primitive is the *bounded embedded-identity traitor tracing* (EITT-MCFE) (Definition 3), for the class computing inner products  $\sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$  between  $n$  clients. In particular, we develop on both definitional and constructive aspects for the technique of *private linear broadcast tracing with embedded identities* [10, 23]. While our construction conceptually follows the framework of [23], the multi-client setting introduces further technical obstacles. Consequently, extending the framework of [23] to our setting requires several novel ideas:

1. First, we proceed via a new intermediate notion of *indexed embedded identities for MCFE* (correspondingly **Boxes** in Definition 3) where decryption keys are issued for any identity  $\text{id}$  along with an additional index  $i$ . The security notions have to incorporate the nature of MCFE (corruption of encryption keys, as well as IND-CPA security under strong admissibility). The main motivation behind this *indexed* version is that we provide, in our *multi-client* setting, a new *generic transformation* that turns an indexed EITT-MCFE scheme into a (bounded) EITT-MCFE (correspondingly **Grey boxes** in Definition 3). The details of this “indexed-to-bounded” transformation are in [17, App. C].
2. The above transformation reduces to the task of constructing indexed EITT for MCFE. To that end, we introduce an intermediate notion of *embedded-identity private linear multi-client functional encryption* (EI-PLMCFE, [17, App. D.1]). As for all primitives we extend to the multi-client setting, the security notions for EI-PLMCFE all imbue the fine-grained corruption of multiple clients and functional keys, while the strong-admissibly IND-CPA is reflected in the *message-hiding* of EI-PLMCFE. Eventually, we give a new *generic transformation* from EI-PLMCFE to indexed EITT-MCFE ([17, App. D.1]). Our notion of EI-PLMCFE generalizes the notion of *embedded-identity private linear broadcast encryption*, originally proposed by Boneh, Sahai, and Waters [10].
3. Finally, while deferring details to [17, App. E], we show *how to build generically* EI-PLMCFE by leveraging *Mixed Functional Encryption* (Mixed FE) (as introduced in [22]) and MCFE for inner products with access control

(AB-IPMCFE) with *strong* admissibility conditions. Both of them can be constructed based on standard LWE assumptions. We rely on [22] for Mixed FE. The latter building block of AB-IPMCFE ([17, App. F] under weak admissibility, then applying our core compiler from Section 6) is another new technical concrete construction.

## 6 From Weak Admissibility to Strong Admissibility for MCFE for Inner Products with Access Control

We present our core technical result: a transformation that turns an MCFE that is provably secure under the *weak* admissibility condition (**W.1** in Section 2, and [17, Def. 7]) into an MCFE that is provably secure under the *strong* admissibility condition (**S.1** in Section 2, and [17, Def. 8]), for the function class  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$ . Let  $\mathcal{E}^w = (\text{Setup}^w, \text{KeyGen}^w, \text{Enc}^w, \text{Dec}^w)$  be an MCFE that is provably secure under the *weak* admissibility condition for the function class  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$  ( $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}}$  is defined in Definition 1). We remark that the function class captures computation of sums of inner products of vectors, where the vectors are indexed by clients  $i \in [n]$ , together with an access control via AC-K. We require the following properties for  $\mathcal{E}^w$ :

1. The functional key of  $\mathcal{E}^w$  can be decomposed  $\text{dk}_{\mathbf{y}, \text{ac-k}}^w = (\widetilde{\text{dk}}_i^w)_i$  for  $i \in [n]$ .
2.  $\mathcal{E}^w$  satisfies a form of decryption that:

$$\mathcal{E}^w.\text{Dec}(\text{dk}_{\mathbf{y}, \text{ac-k}}^w, (\text{ct}_i)_{i=1}^n) = \text{decode} \left( \sum_{i=1}^n \text{PartialDec}(\widetilde{\text{dk}}_i^w, \widetilde{\text{ct}}_i^w) \right)$$

where  $\text{PartialDec}(\widetilde{\text{dk}}_i^w, \widetilde{\text{ct}}_i^w) = \text{encode}(\langle \widetilde{\mathbf{x}}_i, \widetilde{\mathbf{y}}_i \rangle + \text{share}_i)$ ,  $\text{encode}$  is homomorphic under  $+$  and  $\text{decode}(\text{encode}(\cdot))$  is identity only over a polynomially large interval, and  $\text{share}_i$  is an  $n$ -out-of- $n$  share of 0, or maybe some value  $T$  fixed and used only for decryption (*e.g.* the case of [25]).

The usage of  $\text{encode}$  and  $\text{decode}$  resembles the *two-step decryption* requirement in the transformation from FE to MIFE in [3], later treated in the multi-user setting in [2]. We denote by  $\mathcal{E}^{\text{als-ip}} = (\text{Setup}^{\text{als-ip}}, \text{KeyGen}^{\text{als-ip}}, \text{Enc}^{\text{als-ip}}, \text{Dec}^{\text{als-ip}})$  a generic (single-client) public key IPFE scheme, *i.e.*, for  $\mathcal{F}_{N_1, \dots, N_1}^{\text{ip}}$ . We transform  $\mathcal{E}^w$  into another MCFE  $\mathcal{E} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$  for the same function class  $\mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}$  as in Figure 3.

*Remark 2 (Comparison with [30]).* Our transformation differs from [30, Sect. 5.1] significantly, by making generic and explicit conditions that are needed to leverage a MCFE from weak admissibility to strong admissibility. Notably, the structural conditions are identified (such as the above two properties 1 and 2), along with a modular usage of a generic (single-client) public key IPFE scheme. From a conceptual point, our generic transformation thus has the advantage of potential applications in other context. From a constructive point of view, factoring the *ad hoc* and implicit ideas in [30, Sect. 5.1] allows us to treat the more general

case of computing sums of inner products of vectors, whereas [30, Sect. 5.1] only handles computation of individual scalar inner products.

### 6.1 Security

In the following Theorem 2 we prove the *adaptively one-challenge (with private-input repetitions) IND-security* of the transformed MCFE  $\mathcal{E}$ , under *dynamic* corruption and *selective* functions  $(\mathbf{y}_i)_i$  with *adaptive/selective* access control  $\text{ac-k}$ , assuming the *adaptively one-challenge (with private-input repetitions) IND-security* of  $\mathcal{E}^w$ , under the same constraints, and the *adaptive SIM-security* of  $\mathcal{E}^{\text{als-ip}}$ .

**Theorem 2.** *Suppose that  $\mathcal{E}^w$  and  $\mathcal{E}^{\text{als-ip}}$  decrypt to values in the same group, written additively. The MCFE scheme  $\mathcal{E} = (\text{Setup}, \text{DKeyGen}, \text{Enc}, \text{Dec})$  is one-challenge (with private-input repetitions) IND-CPA secure, under dynamic corruption and semi-adaptive queries (adaptive challenges with selective key queries for functions  $(\mathbf{y}_i)_i$  with adaptive access control  $\text{ac-k}$  ([17, Def. 9])), if  $\mathcal{E}^w$  is one-challenge (with private-input repetitions) IND-secure under the same constraint and  $\mathcal{E}^{\text{als-ip}}$  is SIM-secure.*

*Remark 3.* We note that for SIM-security of  $\mathcal{E}^{\text{als-ip}}$ , we can instantiate using a SIM-secure (variant of) LWE-based IPFE by Agrawal, Libert, and Stehlé [7], that is proved one-challenge SIM-secure in [6]. Moreover, the transformation produces a MCFE that is secure under the *strong* admissibility condition, resilient to *dynamic* corruption, *adaptive* challenges with *selective* functions  $(\mathbf{y}_i)_i$ , and *adaptive* access control  $\text{ac-k}$ . The *selective function* constraint is due to the usage of the simulator of  $\mathcal{E}^{\text{als-ip}}$ , where the simulator needs to know the functions in advance for successful simulation.

*Proof Of Theorem 2.* The advantage of an adversary  $\mathcal{A}$  in a game  $\mathbf{G}_i$  is denoted by  $\text{Adv}(\mathbf{G}_i) := |\Pr[\mathbf{G}_i = 1] - 1/2|$  where the probability is taken over the random choices of  $\mathcal{A}$  and coins of  $\mathbf{G}_i$ .

**Game  $\mathbf{G}_0$ :** This is the *one-challenge (with repetitions)* security game, with *strong* admissibility. The private-input repetitions at each position  $i \in [n]$  are indexed by  $\text{rep} \in [J_i]$  where  $J_i$  is the maximum repetitions queried for position  $i$ . We note that for different  $i$ , the bound  $J_i$  can be different. The challenge ciphertext encrypts subvectors  $\mathbf{x}_i^{(b, \text{rep})} \in \mathbb{Z}_q^N$ . For simplicity, we add a constraint that the challenge tag  $\text{tag}$  is *not* queried to **Enc**. This incurs a multiplicative loss factor in advantage up to an inverse of polynomial in  $\lambda$ , where we can reduce to the normal **1chal** by guessing the challenge  $\text{tag}$  among the tags for encryption, and responding all of its **Enc** queries  $(i, \mathbf{x}_i, (\text{tag}, \text{ac-ct}_i))$  by  $\text{LoR}(i, \mathbf{x}_i, \mathbf{x}_i, (\text{tag}, \text{ac-ct}_i))$ .

**Game  $\mathbf{G}_1$ :** If  $\Delta \mathbf{x}_i = \mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)} = 0$ , i.e.,  $\mathbf{x}_i^{(0)} = \mathbf{x}_i^{(1)}$ , our simulator encrypts  $\mathbf{x}_i^{(0)}$  using  $\text{LoR}^w$  of  $\mathcal{E}^w$  to simulate the partial challenge ciphertexts:

- If later  $i$  is corrupted, the weak admissibility condition (condition **W.1** in overview Section 2.3) of  $\mathcal{E}^w$  is already satisfied by  $\mathbf{x}_i^{(0)} = \mathbf{x}_i^{(1)}$ , and this gives us a reduction to the IND-security of  $\mathcal{E}^w$ .

Setup( $1^\lambda$ ): Run

$$\begin{cases} \mathcal{E}^w.\text{Setup}(1^\lambda) = \text{msk}^w, (\text{ek}_i^w)_{i=1}^n \\ \mathcal{E}^{\text{als-ip}}.\text{Setup}^{\text{als-ip}}(1^\lambda, 1^{N_i}) = (\text{msk}_i, \text{pk}_i) \text{ for each } i \end{cases}$$

Sample  $\mathbf{v}_i \xleftarrow{\$} \mathbb{Z}_q^{N_i}$ ,  $\mathbf{t}_i \xleftarrow{\$} (\mathbb{Z}_q^*)^{N_i}$  and output the encryption keys as well as the secret keys as follows:

$$\begin{cases} \text{ek}_i = (\text{pk}_i^{\text{als-ip}}, \text{ek}_i^w, \mathbf{t}_i, \mathbf{v}_i) \\ \text{msk} = ((\text{msk}_i^{\text{als-ip}})_{i=1}^n, \text{msk}^w, (\mathbf{v}_i[k]/\mathbf{t}_i[k])_{k=1}^{N_i})_{i=1}^n \end{cases}$$

KeyGen( $\text{msk}$ ,  $(\mathbf{y}_i)_{i=1}^n$ ,  $\text{ac-k}$ ): The parameters are  $\mathbf{y}_i \in \mathbb{Z}_q^{N_i}$  defining a function to compute inner products with respect to  $\mathbf{y} = [\mathbf{y}_1 \parallel \dots \parallel \mathbf{y}_n]$ . Parse the master secret key

$$\text{msk} = ((\text{msk}_i^{\text{als-ip}})_{i=1}^n, \text{msk}^w, (\mathbf{v}_i[k]/\mathbf{t}_i[k])_{k=1}^{N_i})_{i=1}^n$$

and compute

$$\text{dk}_{\mathbf{y}, \text{ac-k}}^w \leftarrow \mathcal{E}^w.\text{KeyGen}(\text{msk}^w, (\mathbf{y}_i)_{i=1}^n, \text{ac-k})$$

then for each  $i$ , generate  $\text{dk}_i^{\text{als-ip}} \leftarrow \mathcal{E}^{\text{als-ip}}.\text{KeyGen}(\text{msk}_i^{\text{als-ip}}, (\frac{\mathbf{v}_i[k] \cdot \mathbf{y}_i[k]}{\mathbf{t}_i[k]})_{k=1}^{N_i})$ . Output  $\text{dk}_{\mathbf{y}, \text{ac-k}} = (\text{dk}_{\mathbf{y}, \text{ac-k}}^w, (\text{dk}_i^{\text{als-ip}})_i)$ .

Enc( $\text{ek}_i$ ,  $\text{tag}$ ,  $\mathbf{x}_i \in \mathbb{Z}_q^{N_i}$ ,  $\text{ac-ct}_i$ ): Given a vector  $\mathbf{x}_i \in \mathbb{Z}_q^{N_i}$ , and public attributes  $\text{ac-ct}_i$ , parse the encryption key  $\text{ek}_i = (\text{pk}_i^{\text{als-ip}}, \text{ek}_i^w, \mathbf{t}_i, \mathbf{v}_i)$ . Then, sample  $r_i \xleftarrow{\$} \mathbb{Z}_q$  and compute

$$\text{ct}_i^w \leftarrow \mathcal{E}^w.\text{Enc}(\text{ek}_i^w, \text{tag}, \mathbf{x}_i - r_i \cdot \mathbf{v}_i, \text{ac-ct}_i)$$

and  $\text{ct}_i^{\text{als-ip}} \leftarrow \mathcal{E}^{\text{als-ip}}.\text{Enc}(\text{pk}_i^{\text{als-ip}}, r_i \cdot \mathbf{t}_i)$ . Output the ciphertext  $\text{ct}_i = (\text{ct}_i^w, \text{ct}_i^{\text{als-ip}})$ .

Dec( $\text{dk}_{\mathbf{y}, \text{ac-k}}$ ,  $(\text{ct}_i)_{i=1}^n$ ): Parse  $\text{dk}_{\mathbf{y}, \text{ac-k}} = (\text{dk}_{\mathbf{y}, \text{ac-k}}^w, (\text{dk}_i^{\text{als-ip}})_i)$  and  $\text{ct}_i = (\text{ct}_i^w, \text{ct}_i^{\text{als-ip}})$ . Compute

$$\begin{aligned} \text{out}^w &= \sum_{i=1}^n \text{PartialDec}(\text{dk}_{\mathbf{y}, \text{ac-k}}^w, \text{ct}_i^w) = \sum_{i=1}^n \text{encode}(\langle \mathbf{x}_i - r_i \mathbf{v}_i, \mathbf{y}_i \rangle + \text{share}_i) \\ &= \text{encode} \left( \sum_{i=1}^n \langle \mathbf{x}_i - r_i \mathbf{v}_i, \mathbf{y}_i \rangle \right) \end{aligned}$$

We also compute

$$\begin{aligned} \text{out}^{\text{als-ip}} &= \sum_{i=1}^n \text{encode}(\mathcal{E}^{\text{als-ip}}.\text{Dec}(\text{dk}_i^{\text{als-ip}}, \text{ct}_i^{\text{als-ip}})) = \sum_{i=1}^n \text{encode}(\langle r_i \mathbf{v}_i, \mathbf{y}_i \rangle) \\ &= \text{encode} \left( \sum_{i=1}^n \langle r_i \mathbf{v}_i, \mathbf{y}_i \rangle \right). \end{aligned}$$

In the end we compute (the operation “ $\text{out}^w + \text{out}^{\text{als-ip}}$ ” is well-defined because  $\mathcal{E}^w$  and  $\mathcal{E}^{\text{als-ip}}$  decrypt to values in the same group, written additively):

$$\text{decode}(\text{out}^w + \text{out}^{\text{als-ip}}) = \text{decode} \left( \text{encode} \left( \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle \right) \right) = \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$$

due to properties of the function class that  $\sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$  lies in a polynomially large interval.

**Fig. 3.** The transformation of an MCFE with weak admissibility into an MCFE with strong admissibility.

- Else,  $i$  stays honest, the fact that the admissibility condition of  $\mathcal{E}$  is also satisfied over the sum of  $i \in \mathcal{H}^{14}$  and  $\mathbf{x}_i^{(0)} = \mathbf{x}_i^{(1)}$  imply that the admissibility condition **W.1** as in Section 2 of  $\mathcal{E}^w$  is satisfied, which also gives us a reduction to the IND-security of  $\mathcal{E}^w$ .

The change is indistinguishably bounded by  $\text{Adv}_{\mathcal{E}^w, \mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}}^{\text{mc-pos-ind-1chal-cpa}}(1^\lambda)$ .

**Game  $G_2$ :** We treat the case  $\Delta \mathbf{x} \neq \mathbf{0}$  by programming the master secret  $(\mathbf{t}_i, \mathbf{v}_i)$  and switching from  $b \xleftarrow{\$} \{0, 1\}$  to 0. When receiving  $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)})$  and  $\mathbf{x}_i^{(0)} \neq \mathbf{x}_i^{(1)}$ , the simulator implicitly replaces  $\mathbf{v}_i$  in the ciphertext challenge query to its oracle  $\text{LoR}^w$  of  $\mathcal{E}^w$  by  $\tilde{\mathbf{v}}_i := \mathbf{v}_i - \frac{1}{r_i} \Delta \mathbf{x}_i$  where  $\Delta \mathbf{x}_i := \mathbf{x}_i^{(0)} - \mathbf{x}_i^{(1)}$  is known at the time of simulating ciphertext challenge queries. Moreover,  $r_i$  is chosen at the time of encryption to query  $\text{LoR}^w$  of  $\mathcal{E}^w$ . Since  $\mathbf{v}_i \xleftarrow{\$} \mathbb{Z}_q^{N_i}$  and  $r_i \xleftarrow{\$} \mathbb{Z}_q$ , the newly programmed  $\tilde{\mathbf{v}}_i$  has the same distribution as that of  $\mathbf{v}_i$ . Concretely, the pair of challenge queries to the  $\text{LoR}^w$  of  $\mathcal{E}^w$  is now

$$(\mathbf{x}_i^{(0)} - r_i \mathbf{v}_i, \mathbf{x}_i^{(1)} - r_i \tilde{\mathbf{v}}_i) = (\mathbf{x}_i^{(0)} - r_i \mathbf{v}_i, \mathbf{x}_i^{(0)} - r_i \mathbf{v}_i) \quad (2)$$

and the returned  $\mathcal{E}^w$  ciphertext does *not* depend on  $b$  anymore. Finally, using the simulator of  $\mathcal{E}^{\text{als-ip}}$  to simulate the parts  $\text{ct}_i^{(\text{als-ip}, \text{chal})}$  of the challenge ciphertexts, for encrypting  $\mathcal{E}^{\text{sim-als}}.\text{Enc}(\text{pk}_i^{\text{sim-als}}, r_i \cdot \mathbf{t}_i)$  using their simulated key  $\text{pk}_i^{\text{sim-als}}$ . The challenges to return the adversary are  $(\text{ct}_i^{(w, \text{chal})}, \text{ct}_i^{(\text{als-ip}, \text{chal})})$ .

We consider case by case:

- Necessarily the changes on  $\mathbf{v}_i$  into  $\tilde{\mathbf{v}}_i$  stay indistinguishable under the one-challenge SIM-security of  $\mathcal{E}^{\text{als-ip}}$ , using the simulated  $\text{Sim}^{\text{als-ip}}.\text{Enc}(\text{pk}_i^{\text{sim-als}}, r_i \cdot \mathbf{t}_i)$ , knowing the evaluations  $\text{encode}(\langle r_i \mathbf{v}_i, \mathbf{y}_i \rangle)$  due to the *selective* function constraint on the adversary.
- For the keys that ac-k is satisfied  $\text{Rel}(\text{ac-k}, (\text{ac-ct}_i)_i) = 1$ :
  - If  $i$  is corrupted, the weak admissibility for corrupted  $i$  is ensured by (2), i.e., two challenges messages are identical.
  - In any cases, the strong admissibility condition ([17, Def. 8], condition **S.1** in overview Section 2.3) of  $\mathcal{E}$  ensures  $\langle \Delta \mathbf{x}_i, \mathbf{y}_i \rangle = 0$  and correctness is ensured by

$$\mathcal{E}^w.\text{Dec} = \text{decode} \left( \sum_{i=1}^n \text{PartialDec}(\widetilde{\text{dk}}_i^w, \widetilde{\text{ct}}_i^w) \right).$$

We recall that  $\text{PartialDec}(\text{dk}_i^w, \text{ct}_i^w) = \text{encode}(\langle \mathbf{x}_i^{(0)}, \mathbf{y}_i \rangle + \text{share}_i)$  and the share  $\text{share}_i$  of 0 will absorb  $\langle \Delta \mathbf{x}_i, \mathbf{y}_i \rangle$ .

- For the keys that ac-k is *not* satisfied  $\text{Rel}(\text{ac-k}, (\text{ac-ct}_i)_i) = 0$ . It is potentially that  $\langle \Delta \mathbf{x}_i, \mathbf{y}_i \rangle \neq \mathbf{0}$ . Thanks to the *private-input* repetitions<sup>15</sup>, there exists one *unique set* of attributes  $\text{ac-ct}_i$  that corresponds to  $\Delta \mathbf{x}_i \neq \mathbf{0}$ , and moreover corresponds to the unique  $\text{ct}_i^{(w, \text{chal})}$ . This allows us to rely on the *private-input* repetitions security of  $\mathcal{E}^w$ .

<sup>14</sup> We recall that this honest sum condition is common for both weak and strong admissibility, condition **S.2** in overview Section 2.3.

<sup>15</sup> In our full version [17, Def.8], for any  $i \in [n]$ , there exists a unique query of the form  $(x_i^{(0)}, x_i^{(1)}, (\text{tag}, *))$ .

The difference in advantages is  $\text{Adv}_{\mathcal{E}^w, \mathcal{F}_{N_1, \dots, N_n}^{\text{ip}} \times \text{AC-K}}^{\text{mc-pos-ind-1chal-cpa}}(1^\lambda) + \text{Adv}_{\mathcal{E}^{\text{als-ip}}, \mathcal{F}_{N_1, \dots, N_1}^{\text{ip}}}^{\text{sim-1chal}}(1^\lambda)$ .

At the end of  $G_2$ , all ciphertext are independent from  $b$  and  $\text{Adv}(G_2) = 0$ .  $\square$

**Acknowledgments.** We thank David Pointcheval for fruitful discussions on early stages of this work. Xuan Thanh Do would like to thank Vietnam Institute for Advanced Study in Mathematics (VIASM) for providing a fruitful research environment and working condition. Part of this work was conducted while he was visiting the Advanced Cryptography Research Lab (ACR Lab). Xuan Thanh Do was supported by the 03/2025/KCM project (KCM/21-30 research program). Parts of this work was done while Ky Nguyen was at DIENS, ENS-PSL, Paris, France. Ky Nguyen was supported in part by the French ANR Project ANR-19-CE39-0011 PRESTO. Duong Hieu Phan was supported in part by the France 2030 ANR Project ANR-22-PECY-003 Secure-Compute. Quoc-Huy Vu was supported in part by the French ANR project CTQC (ANR-25-CE48-7926-01 CTQC).

## References

1. Abdalla, M., Benhamouda, F., Gay, R.: From single-input to multi-client inner-product functional encryption. In: ASIACRYPT 2019, Part III [5](#), [7](#), [8](#)
2. Abdalla, M., Benhamouda, F., Kohlweiss, M., Waldner, H.: Decentralizing inner-product functional encryption. In: PKC 2019, Part II [5](#), [15](#)
3. Abdalla, M., Catalano, D., Fiore, D., Gay, R., Ursu, B.: Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In: CRYPTO 2018, Part I [15](#)
4. Abdalla, M., Catalano, D., Gay, R., Ursu, B.: Inner-product functional encryption with fine-grained access control. In: ASIACRYPT 2020, Part III [8](#)
5. Agrawal, S., Goyal, R., Tomida, J.: Multi-party functional encryption. In: TCC 2021, Part II [5](#), [8](#)
6. Agrawal, S., Libert, B., Maitra, M., Titu, R.: Adaptive simulation security for inner product functional encryption. In: PKC 2020, Part I [9](#), [16](#)
7. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: CRYPTO 2016, Part III [9](#), [16](#)
8. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: EUROCRYPT 2012 [9](#)
9. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: TCC 2016-A, Part I [9](#)
10. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: EUROCRYPT 2006 [14](#)
11. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: TCC 2011 [2](#)
12. Chen, Y., Vaikuntanathan, V., Waters, B., Wee, H., Wichs, D.: Traitor-tracing from LWE made simple and attribute-based. In: TCC 2018, Part II [2](#)
13. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: CRYPTO'94 [2](#)
14. Chotard, J., Dufour Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Decentralized multi-client functional encryption for inner product. In: ASIACRYPT 2018, Part II [2](#), [4](#), [5](#), [8](#)

15. Chotard, J., Dufour-Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Dynamic decentralized functional encryption. In: CRYPTO 2020, Part I 5
16. Chotard, J., Dufour Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with repetition for inner product. Cryptology ePrint Archive, Report 2018/1021, <https://eprint.iacr.org/2018/1021> 5
17. Do, X.T., Mac, D.T., Nguyen, K., Phan, D.H., Vu, Q.H.: Traitor tracing in multi-sender setting (TMCFE: Traceable multi-client functional encryption). Cryptology ePrint Archive, Paper 2025/364 (2025), <https://eprint.iacr.org/2025/364> 6, 7, 10, 11, 14, 15, 16, 18
18. Do, X.T., Phan, D.H., Pointcheval, D.: Traceable inner product functional encryption. In: CT-RSA 2020 2, 3, 4, 5
19. Dutta, S., Pal, T., Singh, A.K., Mukhopadhyay, S.: Fully collusion resistant traceable identity-based inner product functional encryption. CiC (1) 2, 4, 5
20. Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: EUROCRYPT 2014 2, 4
21. Gordon, S.D., Katz, J., Liu, F.H., Shi, E., Zhou, H.S.: Multi-input functional encryption. Cryptology ePrint Archive, Report 2013/774, <https://eprint.iacr.org/2013/774> 4
22. Goyal, R., Koppula, V., Waters, B.: Collusion resistant traitor tracing from learning with errors. In: 50th ACM STOC 13, 14, 15
23. Goyal, R., Koppula, V., Waters, B.: New approaches to traitor tracing with embedded identities. In: TCC 2019, Part II 3, 4, 6, 13, 14
24. Lai, Q., Liu, F.H., Wang, Z.: New lattice two-stage sampling technique and its applications to functional encryption - stronger security and smaller ciphertexts. In: EUROCRYPT 2021, Part I 7
25. Libert, B., Titiu, R.: Multi-client functional encryption for linear functions in the standard model from LWE. In: ASIACRYPT 2019, Part III 5, 15
26. Libert, B., Yung, M., Joye, M., Peters, T.: Traceable group encryption. In: PKC 2014 2
27. Luo, F., Al-Kuwari, S., Wang, H., Han, W.: Generic construction of trace-and-revoke inner product functional encryption. In: ESORICS 2022, Part I 2, 3, 4, 5
28. Nguyen, K., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with fine-grained access control. In: ASIACRYPT 2022, Part I 8
29. Nguyen, K., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with public inputs and strong security. In: PKC 2025, Part III 8
30. Nguyen, K., Phan, D.H., Pointcheval, D.: Optimal security notion for decentralized multi-client functional encryption. In: ACNS 2023, Part II 2, 3, 5, 6, 8, 9, 15, 16
31. Nguyen, K., Pointcheval, D., Schädlich, R.: Dynamic decentralized functional encryption: Generic constructions with strong security. In: *Public-Key Cryptography - IACR PKC 2025*. Springer (2025) 8
32. Zheng, Q., Zhao, J.: Trace-and-revoke quadratic functional encryption. In: *Information Security*. Springer Nature Switzerland (2023) 4