Multi-Client Inner-Product Functional Encryption in the Random-Oracle Model

Michel Abdalla, Florian Bourse, Hugo Marival, David Pointcheval, Azam Soleimanian, Hendrik Waldner

▶ **To cite this version:**

Michel Abdalla, Florian Bourse, Hugo Marival, David Pointcheval, Azam Soleimanian, et al.. Multi-Client Inner-Product Functional Encryption in the Random-Oracle Model. SCN 2020 - 12th International Conference Security and Cryptography for Networks., Sep 2020, Amalfi / Virtual, Italy. pp.525-545, 10.1007/978-3-030-57990-6_26 . hal-02948657

HAL Id: hal-02948657
https://inria.hal.science/hal-02948657

Submitted on 5 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Client Inner-Product Functional Encryption in the Random-Oracle Model

Michel Abdalla[1,2], Florian Bourse[1,2], Hugo Marival[1,2],
David Pointcheval[1,2], Azam Soleimanian[1,2], and Hendrik Waldner[3]

[1] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[2] INRIA, Paris, France
[3] University of Edinburgh, Edinburgh, UK

Abstract. Multi-client functional encryption (MCFE) is an extension of functional encryption (FE) in which the decryption procedure involves ciphertexts from multiple parties. It is particularly useful in the context of data outsourcing and cloud computing where the data may come from different sources and where some data centers or servers may need to perform different types of computation on this data. In order to protect the privacy of the encrypted data, the server, in possession of a functional decryption key, should only be able to compute the final result in the clear, but no other information regarding the encrypted data. In this paper, we consider MCFE schemes supporting encryption labels, which allow the encryptor to limit the amount of possible mix-and-match that can take place during the decryption. This is achieved by only allowing the decryption of ciphertexts that were generated with respect to the same label. This flexible form of FE was already investigated by Abdalla et al. [Asiacrypt 2019] and Chotard et al. [Asiacrypt 2018]. The former provided a general construction based on different standard assumptions, but its ciphertext size grows quadratically with the number of clients. The latter gave a MCFE based on Decisional Diffie-Hellman (DDH) assumption which requires a small inner-product space. In this work, we overcome the deficiency of these works by presenting three constructions with linear-sized ciphertexts based on the Matrix-DDH (MDDH), Decisional Composite Residuosity (DCR) and Learning with Errors (LWE) assumptions in the random-oracle model. We also implement our constructions to evaluate their concrete efficiency.

Keywords: Functional encryption, multi-client, inner-product functionality, random oracle.

1 Introduction

Functional encryption (FE) [BSW11,O'N10] is an encryption scheme that goes beyond all-or-nothing decryption, allowing users in possession of a secret functional decryption key to learn a specific function of the encrypted message, and nothing else. More formally, in an FE scheme for a class of functions F, a ciphertext encrypting a message x can be used in conjunction with a functional decryption key dk_f, derived for a function f from F, in order to compute $f(x)$ while no more information about x should be leaked. Due to its generality, FE encompasses many existing notions, such as identity-based encryption [BF01,Coc01,Wat05] and attribute-based encryption [GPSW06,OSW07,Wat11]. Now, general purpose FE is seen as a holy grail for modern cryptography. Several works have made progress towards this goal [GGH+13,Wat15,BCP14], but no constructions are known from standard assumptions. Since general-purpose FE still remains far from reality, different lines of work focused on building FE for specialized classes of functions, such as predicate encryption or inner-product FE.

Inner-product FE (IPFE) is a special case of FE [ABDP15] in which the encrypted messages are vectors $\boldsymbol{x}$, and the functional decryption keys $\mathsf{dk}_{\boldsymbol{y}}$, are associated with vectors $\boldsymbol{y}$ of the same dimension, and the decryption yields the inner-product between those two vectors (i.e., $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$). It was first considered by [ABDP15] as the first efficient encryption scheme going beyond all-or-nothing decryption. The class of functions defined is simple enough to allow for practical instantiation, as it is only linear, but still allows for many applications. In particular, it allows for any bounded depth computation by properly increasing the size of the inputs [ALS16,AR17].

Multi-client FE (MCFE), introduced in [GGG+14][1], is a natural extension of FE where data comes from different sources/clients that may not trust each other and can be independently and adaptively corrupted by the adversary. The special case of Multi-input FE (MIFE) [ACF+18,AGRW17] corresponds to the setting where the clients are honest but curious, and each coordinate of a vector can be encrypted separately before being combined during the decryption procedure. The main challenge to overcome when designing MCFE is that the different parts of the ciphertext have to be crafted without sharing any randomness, as opposed to what happens in all the existing constructions for single-input IPFE (or simply IPFE).

MCFE with labels, introduced in [GGG+14] and recast in the context of the inner-product functionality by [CDG+18a], allows for more control over the data during encryption. In an MCFE scheme with labels, ciphertexts strictly depend on labels. When combining ciphertexts during decryption, data associated with different labels cannot be mixed to give a valid decryption or useful information. Thus, the data from different sources can only be combined if they have the same label. The construction suggested in [CDG+18a] for the inner-product functionality is based on the Decisional Diffie-Hellman (DDH) assumption, and one of its drawbacks is that the decryption algorithm needs to compute the discrete logarithm of a group element, which means it can only support a small range of values for the inner-products, thus limiting its possible applications. Abdalla et al. [ABG19] proposed a general conversion from single-input to MCFE. In their scheme, each client i encrypts its message x_i as the vector $(0||\dots||0||x_i||0||\dots||0) + \boldsymbol{t}_{i,\ell}$ where $\boldsymbol{t}_{i,\ell}$ is generated by a PRF with shared keys such that $\sum_{i=1}^{n} \boldsymbol{t}_{i,\ell} = 0$, with n the number of clients. From there, by applying a layer of single-input IPFE they get a labeled MCFE. This explains why the size of the ciphertext in their scheme is quadratic w.r.t the number of the slots/clients. Note that their scheme also needs a master secret key of size $O(n^2)$ which is the number of keys $k_{i,j}$ shared between clients i and j.

1.1 Challenges and Contributions

This paper aims at constructing efficient labeled MCFE schemes based on different assumptions. Our contributions can be summarized as follows:

Efficient decryption and shorter ciphertext. We present two constructions: one based on the Decisional Composite Residuosity (DCR) assumption and the other one based on the Learning with Errors (LWE) assumption. These constructions can cope with the drawbacks in the constructions of [CDG+18a] and [ABG19], i.e. they do not require a discrete-logarithm computation in the decryption algorithm while the size of the ciphertext is smaller (w.r.t the number of clients). The security proof for our constructions based on DCR and LWE can be more challenging than MCFE based on DDH. This difficulty comes from the fact that MCFE is in the symmetric key setting and the hybrid argument for many challenges can be complicated. More precisely, one needs to show that in the current hybrid game, given the information regarding the master key that is leaked through all other queries (encryption, functional keys, and random-oracle (RO) queries) the master key still has enough entropy to hide the chosen bit in the challenge ciphertext. This is easier to prove in DDH-based MCFE schemes since the master secret key is uniformly distributed over $\mathbb{Z}_q$ and the ciphertexts are defined in a group with the same order. This common modulus not only helps to interpret the leaked information more straightforwardly, but also to prove that the chosen bit can be perfectly hidden. However, for our DCR-based MCFE, this is not the case, and one needs to check how the leaked information can change the lattice to which the master secret key belongs (since the master key is distributed over the lattice $\mathbb{Z}^n$) and how it can affect the challenge which is a value modulo N. By relying on a theorem from lattice-based cryptography and setting the parameters similarly to the single-input IPFE [ALS16], and also by a proper simulation of random-oracle queries one can guarantee that the information leaked through the encryption queries is still tolerable, and that the security

[1] However, the authors have named it as multi-input functional encryption.

proof works. Slightly in detail, the proper simulation of random-oracle queries let us to unify the leakage from all other ciphertexts. This unified information is the same as the leaked information from the public-key in [ALS16]. Then, we can use the same strategy of [ALS16] to show that challenge ciphertext hides the chosen bit statistically w.r.t a selective-security notion. But the good point is that all other steps which are based on the computational-assumption DCR are adaptively secure. Thus, we only need to lift the security back to adaptive in our statistical argument, which is possible by a proper choice of parameters.
All is left is to discuss the simulation of RO queries such that it can unify and properly interpret the leakage from all other ciphertexts. Here, we use random self-reducibility of DCR assumption which let us build polynomially many random samples of DCR from one given sample. Then RO queries can be replaced with these random samples. The common point about all these samples is that they are indistinguishable from elements in the class of N residues in $\mathbb{Z}_{N^2}$ and so they all have the same structure $z_\ell^N \mod N^2$. Having this N common among all the RO queries is what we needed as a tool to unify the leakage from ciphertexts. More precisely, now the leakage from all other ciphertexts can be interpreted independently of ℓ as $\boldsymbol{s} \mod \lambda$ (where $\boldsymbol{s}$ is the secret-key, $\mathcal{H}(\ell)^{\boldsymbol{s}}$ is appeared in the ciphertexts, and λ is such that $z_\ell^{N\lambda} = 1 \mod N^2$).

For our LWE-based MCFE (which can bee seen as the main contribution), it is more challenging since the leaked information through the encryption queries cannot be simulated during the security proof, because of the noise terms introduced by the LWE assumption. We overcome this challenge by using noise flooding techniques, and avoid the inefficiency drawback by rounding the ciphertext down to a smaller space. This way, the noise vanishes during this rounding operation. The remaining leakage concerns a part of the master key that is uniformly random, and can be easily simulated. More precisely, in our LWE-based construction, ciphertext includes a multiplication term $\mathbf{Z}_i \cdot \mathcal{H}(\ell)$ where $\mathbf{Z}_i = (\boldsymbol{s}_i, \boldsymbol{t}_i)$ comes from the master key and $\mathcal{H}(\ell)$ is a hash function modeled as RO. This has to be a RO on $\mathbb{Z}_q$ leading us to replace it with LWE samples (which give randomness over $\mathbb{Z}_q$) i.e., $\mathcal{H}(\ell) = (\boldsymbol{a}_\ell, \mathbf{S}\boldsymbol{a}_\ell + \boldsymbol{e}_\ell)$. The term $t_i \cdot \boldsymbol{e}_\ell$ is what can dramatically leak information about $\mathbf{Z}_i$. In the proof of Agrawal et al. [ALS16] for IPFE, the term $\mathbf{S}\boldsymbol{a}$ can be placed in the ciphertext directly since the client knows the secret $\mathbf{S}$. But for our labeled MCFE this is not the case and the term $\boldsymbol{e}_\ell$ has to be there which leads to the leakage $\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell$. Thus, we map the ciphertext from $\mathbb{Z}_q$ to a small space $\mathbb{Z}_{q_0}$ such that the term $\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell$ is small enough to be neglected after this change. The term $\boldsymbol{t}_i \cdot \mathbf{S}\boldsymbol{a}_\ell$ would be hidden through the term $\boldsymbol{s}_i \cdot \boldsymbol{a}_\ell$ where $\boldsymbol{s}_i$ is uniform[2]. These two strategies give us the guarantee that no information about $\boldsymbol{t}_i$ is leaked through encryption queries. We then show that given the other sources of information that the adversary may access (functional keys and corruption queries), the master secret key $\boldsymbol{t}_i$ still has enough entropy to be used in a left-over hash lemma argument and statistically hides the message-challenge w.r.t a selective-security notion. Then similar to our discussion for DCR-based MCFE, one can simply lift the security to the adaptive case by a proper choice of parameters.

Now we discuss a bit about the simulation of RO queries relying on the computational-assumption LWE. A curious reader may already have noticed that unlike DCR-based MCFE where we use random self-reducibility of the DCR assumption, we may not be able to do the same here. Fortunately, the definition of the LWE problem already provides polynomially many samples for the same secret $\mathbf{S}$ as $(\boldsymbol{a}_\ell, \mathbf{S}\boldsymbol{a}_\ell + \boldsymbol{e}_\ell)$ where $\mathbf{S}$ is a vector. We simply extend it to the case where $\mathbf{S}$ is a matrix. Note that the requirement for a matrix-secret instead of vector-secret comes from the security proof, since having $\mathbf{S}$ as a matrix gives $\boldsymbol{t}_i$ as a vector (note that in the ciphertext we have $\mathbf{Z}_i \cdot \mathcal{H}(\ell) = \boldsymbol{s}_i \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot (\mathbf{S}\boldsymbol{a}_\ell + \boldsymbol{e}_\ell)$ where $\mathbf{Z}_i = (\boldsymbol{s}_i, \boldsymbol{t}_i)$ is the secret-key). Then having $\boldsymbol{t}_i$ as a vector provides enough entropy in the term $(x^1 - x^0) \cdot (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)^T$ which will be used in a left-over-hash-lemma argument to conclude that the challenge ciphertext is statistically independent of chosen bit.

[2] Note that we have $\mathbf{Z}_i \cdot \mathcal{H}(\ell) = \boldsymbol{s}_i \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot (\mathbf{S}\boldsymbol{a}_\ell + \boldsymbol{e}_\ell)$

Various assumptions. Following the constructions proposed by Chotard et al. [CDG+18a], we present a generalization of their scheme, relying on the Matrix-DDH (MDDH) assumption[3]. Our Labeled MCFE based on DCR assumption is the first labeled MCFE scheme based on this assumption and with linear ciphertext size. Our labeled MCFE based on LWE is the most efficient MCFE scheme based on this assumption compared to [ABG19,LT19], albeit in the RO model.

Decentralization (with linear size of the secret key). Abdalla et al. [ABG19,ABKW19] presented a compiler converting a MCFE scheme to a decentralized MCFE (DMCFE) requiring a square size of the secret key (w.r.t the number of clients)[4]. Chotard et al. [CDG+18a] also extended their DDH-based MCFE scheme to its decentralized counterpart using pairings and with linear key size. Here we present a generalization of their scheme to get DMCFE with linear key size and without pairings. In our proof, the security assumption/requirement associated with underlying building blocks is also weaker than [CDG+18a]. This part is discussed in Appendix D.

Implementation. We also have implemented our constructions showing that for applications with large message spaces our DCR-based MCFE scheme is quite reliable while for small message space our LWE-based MCFE scheme is more efficient. This gives enough flexibility to choose the scheme that better fits the application. Apart from the size of the message space, other parameters are chosen so that the schemes can support different applications.

1.2 Related Work

Here, we mainly discuss the three mentioned works [ABG19,CDG+18a,LT19] which are directly relevant to our contributions. The main security notions used in these papers are one-security and pos+-security. In one-security, the adversary can ask for many labels but for each label it can issue only one complete ciphertext. In pos+-security, the adversary can ask for many ciphertexts per label.[5]

In [CDG+18a], instead of proving pos+-security, the authors first prove one-security for their construction and then apply a compiler similar to [ACF+18,AGRW17] to lift the security to pos+. As in [ACF+18,AGRW17], this compiler is actually a single-input IPFE layer. We also use this technique in this paper. The security in [CDG+18a] relies on the DDH assumption in the RO model. The ciphertext in their scheme has the form $\mathsf{ct}_{i,\ell} = g^{x_i} \cdot \mathcal{H}(\ell)^{s_i}$. The main challenge in the proof is to bound the leakage from the ciphertexts (as we are in the symmetric key setting with many ciphertexts to be handled directly). The idea is to change the RO queries in an indistinguishable way such that all the encryption queries, except for the challenge, have the same form (i.e., $\mathcal{H}(\ell) = g^{\boldsymbol{u}_\ell}$ where $\boldsymbol{u}_\ell = r_\ell \cdot \boldsymbol{a}$, $\boldsymbol{a} = (1 \quad a)^T$, $r_\ell, a \xleftarrow{R} \mathbb{Z}_p$) leading to the same leakage $\boldsymbol{s}_i \cdot \boldsymbol{a}$ from all other encryption queries. This leakage, along with the leakage from the functional secret keys and corrupted individual encryption keys, would change the distribution of the master secret key such that the multiplication $\boldsymbol{s}_i \cdot \boldsymbol{u}_{\ell^*}$ (where $\boldsymbol{u}_{\ell^*} = u_1\boldsymbol{a} + u_2\boldsymbol{a}^\perp$, $u_1 \xleftarrow{R} \mathbb{Z}_p$, $u_2 \xleftarrow{R} \mathbb{Z}_p^*$) perfectly hides the chosen bit in the challenge. More precisely, the secret key is computed as $\boldsymbol{s}_i + \boldsymbol{a}^\perp\gamma(x_i^1 - x_i^0)$ where $\gamma = -1/\boldsymbol{u}_{\ell^*} \cdot \boldsymbol{a}^T$ and $\boldsymbol{s}_i \xleftarrow{R} \mathbb{Z}_p^2$. The ciphertext is of linear size while one needs to compute a discrete-logarithm during the decryption.

In [ABG19], as we mentioned at the beginning of this section, each client builds a value $\boldsymbol{t}_{i,\ell}$ such that $\sum \boldsymbol{t}_{i,\ell} = 0$. For the security proof, they simply change the values of $\boldsymbol{t}_{i,\ell}$ among the slots such that each $\boldsymbol{t}_{i,\ell}$ is replaced with a random value except one of them associated with an honest slot, called i^*, which takes care of the relation $\sum \boldsymbol{t}_{i,\ell} = 0$. Then, one-security would be

[3] which is a generalization of the DDH assumption including many other assumptions such as k-LIN and 2-Scasc [EHK+13], as special cases.

[4] One can decrease the size of the key to $O(n)$ by relying on RO and CDH assumption (similar to the DSum construction in [CDG+18b]). But this will add a new computational assumption which is not appropriate in many applications

[5] Note that these security notions are respectively called *without repetition* and *with repetition* in [CDG+18a,CDG+18b] . Here we are following the terminologies of [ABG19].

Scheme	$\lvert\mathsf{sk}_i\rvert$	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{ct}\rvert$	q	σ (msk)	model
[ABG19]+ [ALS16]	$O(n\kappa)$	$O(n_0(n_0+n)\log q)$	$n^2 \log q$	$\mathrm{poly}(n_0)$	$\mathrm{poly}(n_0)$	SM
[LT19]	$O(\kappa^5)$	$O(\kappa^{13})$	$O(n\kappa^7 \log q)$	2^{κ^2}	$O(2^{\kappa^2})$	SM
our scheme	$n_0 + m_0$	$n_0 + m_0$	$n \log q_0$	$\Omega({n_0}^{\omega(1)} q_0 B)$	$\omega(1)$	ROM

Table 1: Comparison for LWE-based MCFE schemes

reduced to the PRF property.[6] Despite relying on standard assumptions, their scheme needs $O(n^2)$ secret key and the ciphertext-size is $O(n^2)$.

In [LT19], similarly to [CDG+18a], each ciphertext $\mathsf{ct}_{i,\ell} = \mathbf{G}_0^T \cdot x_i + \mathbf{A}(\ell)^T \cdot \boldsymbol{s}_i + \mathsf{noise}$ has a product term $\mathbf{A}^T(\ell) \cdot \boldsymbol{s}_i$ which hides the chosen bit in the challenge. Unlike our constructions, the matrix $\mathbf{A}(\ell)$ is built from some public matrices and the label ℓ, rather than a RO, using an idea from [LST18] to derive $\mathbf{A}(\ell)$ from some public matrices using the Gentry-Sahai-Waters (GSW) fully homomorphic encryption scheme [GSW13]. That is, $\mathbf{A}(\ell)$ is the product of GSW ciphertexts dictated by a special hash applied to ℓ. The security proof relies on the fact that, with noticeable probability, $\mathbf{A}(\ell)$ is a GSW encryption of 1. From there, it can be indistinguishably changed to the GSW-encryption of 0 in all other encryption queries, except for the challenge. Finally, an argument similar to [CDG+18a] (through the lossy form of matrix $\mathbf{A}$) is used to conclude the proof.

Table 1 compares our LWE-based MCFE scheme with the schemes of [ABG19] and [LT19]. We have considered the instantiation of [ABG19] based on the LWE-based IPFE scheme of [ALS16]. In this table, κ and n_0 are security parameters where n_0 is the size of the secret. In our scheme, $m_0 > \Omega(\log q)$ for selective security and $m_0 > \Omega(\log q + 4n \cdot \log P)$ for the adaptive case where n is the number of slots and P defines the bound of the message-space. And we also have $q_0 = \mathrm{poly}(n_0)$ and B is a constant as the bound of the error-space. And σ stands for the standard-deviation used in the generation of msk. So as one can conclude from this table, for the client i, the size of its secret-key sk_i and also the size of public-parameters pp in [ABG19], depend on the number of clients n, while in [LT19] and in our scheme they are constant (w.r.t n). Still one can argue that for the scheme of [LT19], the size of sk_i and pp is much larger comparing with our scheme. Note that the security parameter for their scheme is κ and for our scheme is n_0 which means that our scheme has linear-size of sk_i and pp w.r.t to the security parameter. While in [LT19] they are polynomials respectively of degree 5 and 13 (w.r.t the security parameter). In [ABG19], the size of public-parameters also depends on n_0 which is the security-parameter for the underlying LWE scheme [ALS16]. In our scheme the only public-parameter is the hash function (modeled as random oracle) and it is a vector of size $n_0 + m_0$. While [ALS16] has some matrices as the public parameters leading to a size of degree 2 polynomial for $\lvert\mathsf{pp}\rvert$ (w.r.t the security parameter), while in our scheme it is linear. About size of the ciphertext ct, in [ABG19], it has square-size w.r.t the number of clients and in [LT19] has 7-degree-size w.r.t the security parameter. While in our scheme it is linear w.r.t to n and logarithmic w.r.t the security parameter.

Putting together, this table shows that having constant or linear size of sk_i, pp or ct w.r.t n can be challenging and leads to a polynomial-size of large degree w.r.t other parameters. We avoid this inefficiency by relying on the RO assumption.

2 Preliminaries

Notation. We use $[n]$ to denote the set $\{1, \ldots, n\}$. We write $\boldsymbol{x}$ for vectors and x_i for the i-th element. In this paper, κ stands for the security parameter. The function $\mathrm{poly}(\cdot)$ shows an arbitrary polynomial function. The computational indistinguishability of two distributions G_0 and G_1, is denoted by $\mathsf{G}_0 \cong \mathsf{G}_1$. The function $\mathrm{negl}(\cdot)$ denotes the negligible function. In this

[6] In their construction, they apply the compiler, for going from one to pos$^+$, which gives pos$^+$ directly.

paper all the algorithms are Probabilistic Polynomial Time (p.p.t.) with respect to the length of the input. For security parameter κ and additional parameters n, we denote the winning probability of an adversary $\mathcal{A}$ in a game or experiment G as $\mathsf{Win}^{\mathsf{G}}_{\mathcal{A}}(\kappa, n)$. The probability is taken over the random coins of G and $\mathcal{A}$. We define the distinguishing advantage between games G_0 and G_1 of an adversary $\mathcal{A}$ in the following way: $\mathsf{Adv}^{\mathsf{G}}_{\mathcal{A}}(\kappa, n) = |\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa, n)|$.

2.1 Multi-Client Functional Encryption

A labeled MCFE scheme is formally defined as follows, which is an adaptation of the MIFE definition [GGG+14] with labels.

Definition 1. (Multi-Client Functional Encryption) Let $\mathcal{F} = \{\mathcal{F}_\rho\}_\rho$ be a family (indexed by ρ) of sets $\mathcal{F}_\rho$ of functions $f\colon \mathcal{X}_{\rho,1} \times \cdots \times \mathcal{X}_{\rho,n_\rho} \to \mathcal{Y}_\rho$.[7] Let $\mathsf{Labels} = \{0,1\}^*$ or $\{\perp\}$ be a set of labels. A multi-client functional encryption scheme (MCFE) for the function family $\mathcal{F}$ and the label set Labels is a tuple of five algorithms $\mathsf{MCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\kappa, 1^n)$**:** Takes as input a security parameter κ and the number of parties n, and generates public parameters pp. The public parameters implicitly define an index ρ corresponding to a set $\mathcal{F}_\rho$ of n-ary functions (i.e., $n = n_\rho$).

$\mathsf{KeyGen}(\mathsf{pp})$**:** Takes as input the public parameters pp and outputs n secret keys $\{\mathsf{sk}_i\}_{i\in[n]}$ and a master secret key msk.

$\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, f)$**:** Takes as input the public parameters pp, the master secret key msk and a function $f \in \mathcal{F}_\rho$, and outputs a functional decryption key sk_f.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, \ell)$**:** Takes as input the public parameters pp, a secret key sk_i, a message $x_i \in \mathcal{X}_{\rho,i}$ to encrypt, a label $\ell \in \mathsf{Labels}$, and outputs ciphertext $\mathsf{ct}_{i,\ell}$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}_{1,\ell}, \ldots, \mathsf{ct}_{n,\ell})$**:** Takes as input the public parameters pp, a functional key sk_f and n ciphertexts under the same label ℓ and outputs a value $y \in \mathcal{Y}_\rho$.

A scheme MCFE is correct, if for all $\kappa, n \in \mathbb{N}$, $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa, 1^n)$, $f \in \mathcal{F}_\rho$, $\ell \in \mathsf{Labels}$, $x_i \in \mathcal{X}_{\rho,i}$, when $(\{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$ and $\mathsf{sk}_f \leftarrow \mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, f)$, we have

$$\Pr\left[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_1, x_1, \ell), \ldots, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_n, x_n, \ell)) = f(x_1, \ldots, x_n)\right] = 1.$$

Please note that each slot i in a MCFE scheme has a different secret key sk_i, which can be individually corrupted. In addition, one also needs to consider corruptions to handle possible collusions between different parties. In the following, we formally define the security notion of a MCFE scheme.

Definition 2. (Security of MCFE) Let MCFE be an MCFE scheme and Labels a label set. For $\beta \in \{0,1\}$, we define the experiment $\mathrm{IND}^{\mathsf{MCFE}}_\beta$ in Fig. 1, where the oracles are defined as:

Corruption oracle $\mathsf{QCor}(i)$**:** Outputs the encryption key sk_i of slot i. We denote by $\mathcal{CS}$ the set of corrupted slots at the end of the experiment.

Left-Right oracle $\mathsf{QLeftRight}(i, x^0_i, x^1_i, \ell)$**:** Outputs $\mathsf{ct}_{i,\ell} = \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x^\beta_i, \ell)$ on a query (i, x^0_i, x^1_i, ℓ). We denote by $Q_{i,\ell}$ the number of queries of the form $\mathsf{QLeftRight}(i, \cdot, \cdot, \ell)$.

Encryption oracle $\mathsf{QEnc}(i, x_i, \ell)$**:** Outputs $\mathsf{ct}_{i,\ell} = \mathsf{Enc}(\mathsf{sk}_i, x_i, \ell)$ on a query (i, x_i, ℓ).

Key derivation oracle $\mathsf{QKeyD}(f)$**:** Outputs $\mathsf{dk}_f = \mathsf{KeyGen}(\mathsf{msk}, f)$.

and where *Condition (*)* holds if all the following conditions hold:

- If $i \in \mathcal{CS}$ (i.e., slot i is corrupted): for any query $\mathsf{QLeftRight}(i, x^0_i, x^1_i, \ell)$, $x^0_i = x^1_i$.

[7] All the functions inside the same set $\mathcal{F}_\rho$ have the same domain and the same range.

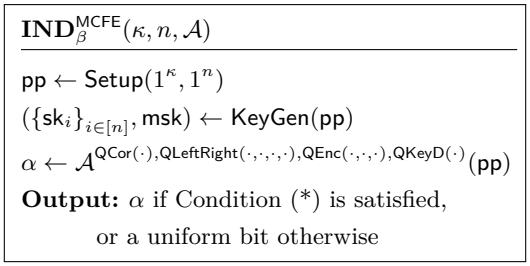

Fig. 1: Security games for MCFE

– For any label $\ell \in \mathsf{Labels}$, for any family of queries $\{\mathsf{QLeftRight}(i, x_i^0, x_i^1, \ell)$ or $\mathsf{QEnc}(i, x_i, \ell)\}_{i\in[n]\setminus\mathcal{CS}}$, for any family of inputs $\{x_i \in \mathcal{X}\}_{i\in\mathcal{CS}}$, for any query $\mathsf{QKeyD}(f)$, we define $x_i^0 = x_i^1 = x_i$ for any slot $i \in \mathcal{CS}$ and any slot queried to $\mathsf{QEnc}(i, x_i, \ell)$, we require that: $f(\boldsymbol{x}^0) = f(\boldsymbol{x}^1)$ where $\boldsymbol{x^b} = (x_1^b, \ldots, x_n^b)$ for $b \in \{0, 1\}$.
We insist that, if one index $i \notin \mathcal{CS}$ is not queried for the label ℓ, there is no restriction.

The weaker versions of the security are defined as xx-yy-zz-$\mathrm{IND}_\beta^{\mathsf{MCFE}}$ (xx, yy, zz may be empty when we do not have the corresponding restriction), where,

– When xx = sta: the adversary should output the set $\mathcal{CS}$ at the beginning of the game, and it does not have access to the oracle QCor after that.
– When yy = one: for any slot $i \in [n]$ and $\ell \in \mathsf{Labels}$, $Q_{i,\ell} \in \{0, 1\}$, and if $Q_{i,\ell} = 1$, then for any slot $j \in [n] \setminus \mathcal{CS}$, $Q_{j,\ell} = 1$. In other words, for any label, either the adversary makes no left-right query or makes exactly one left-right query for each $i \in [n] \setminus \mathcal{CS}$.
– When yy = pos^+: for any slot $i \in [n]$ and $\ell \in \mathsf{Labels}$, if $Q_{i,\ell} > 0$, then for any slot $j \in [n]\setminus\mathcal{CS}$, $Q_{j,\ell} > 0$. In other words, for any label, either the adversary makes no left-right encryption query or makes at least one left-right encryption query for each slot $i \in [n] \setminus \mathcal{CS}$.
– When zz = sel: the adversary should output the challenges at the beginning of the game, and it does not have access to the oracle $\mathsf{QLeftRight}$ after that. This case is referred as the selective security.

We define the advantage of an adversary $\mathcal{A}$ in the following way:

$$\mathsf{Adv}^{\text{xx-yy-zz-IND}}_{\mathsf{MCFE},\mathcal{A}}(\kappa, n) = \big|\Pr[\text{xx-yy-zz-IND}_0^{\mathsf{MCFE}}(\kappa, n, \mathcal{A}) = 1] - \Pr[\text{xx-yy-zz-IND}_1^{\mathsf{MCFE}}(\kappa, n, \mathcal{A}) = 1]\big|.$$

A multi-client functional encryption scheme MCFE is xx-yy-zz-IND secure, if for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function negl such that: $\mathsf{Adv}^{\text{xx-yy-zz-IND}}_{\mathsf{MCFE},\mathcal{A}}(\kappa, n) \leq \mathrm{negl}(\kappa)$.

We omit n when it is clear from the context. We also often omit $\mathcal{A}$ from the parameter of experiments or games when it is clear from context.

Definition 3. (1-label Security) Let MCFE be an MCFE scheme, $\mathcal{F} = \{\mathcal{F}_\rho\}_\rho$ a function family indexed by ρ and Labels a label set. For xx, yy, zz defined as Theorem 2, and $\beta \in \{0, 1\}$, we define the experiment xx-yy-zz-1-$\mathrm{label}_\beta^{\mathsf{MCFE}}$ exactly as in Fig. 1, where the oracles are defined as for Theorem 2, except:

Left-Right oracle $\mathsf{QLeftRight}(i, x_i^0, x_i^1, \ell)$**:** Outputs $\mathsf{ct}_{i,\ell} = \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i^\beta, \ell)$ on a query (i, x_i^0, x_i^1, ℓ). This oracle can be queried at most on one label. Further queries with distinct labels will be ignored.

Encryption oracle $\mathsf{QEnc}(i, x_i, \ell)$ Outputs $\mathsf{ct}_{i,\ell} = \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, \ell)$. If this oracle is queried on the same label that is queried to $\mathsf{QLeftRight}$, the game ends and returns 0.

Condition (*) is defined as for Theorem 2. We define the advantage of an $\mathcal{A}$ as follows:

$$\mathsf{Adv}^{\text{xx-yy-zz-IND-1-label}}_{\mathsf{MCFE},\mathcal{A}}(\kappa,n) = |\Pr[\text{xx-yy-zz-IND-1-label}^{\mathsf{MCFE}}_0(\kappa,n,\mathcal{A})=1] - \Pr[\text{xx-yy-zz-IND-1-label}^{\mathsf{MCFE}}_1(\kappa,n,\mathcal{A})=1]|.$$

Lemma 4. (From one to many labels [ABG19]) *Let* MCFE *be a scheme that is* xx-yy-zz*-IND-1-label secure. Then it is also secure against p.p.t. adversaries that query* QLeftRight *on many distinct labels (*xx-yy-zz*-IND security). Namely, for any p.p.t. adversary $\mathcal{A}$, there exists a p.p.t. adversary $\mathcal{B}$ such that:* $\mathsf{Adv}^{\text{xx-yy-zz-IND}}_{\mathsf{MCFE},\mathcal{A}}(\kappa,n) \leq q_{\mathsf{Enc}} \cdot \mathsf{Adv}^{\text{xx-yy-zz-IND-1-label}}_{\mathsf{MCFE},\mathcal{B}}(\kappa,n)$,
By q_{Enc} we denote the number of distinct labels queried by $\mathcal{A}$ to QLeftRight.

2.2 Inner-Product Functionality

We describe the functionalities supported by the constructions in this paper, by considering the index ρ of $\mathcal{F}$ in more detail.

The index of the family is defined as $\rho = (\mathcal{R}, n, m, X, Y)$ where $\mathcal{R}$ is either $\mathbb{Z}$ or $\mathbb{Z}_L$ for some integer L, and n, m, X, Y are positive integers. If X, Y are omitted, then $X = Y = L$ is used (i.e., no constraint). This defines $\mathcal{F}_\rho = \{f_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n} : (\mathcal{R}^m)^n \to \mathcal{R}\}$ where $f_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{i=1}^n \langle \boldsymbol{x}_i, \boldsymbol{y}_i\rangle = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$, the vectors satisfy the following bounds: $\|\boldsymbol{x}_i\|_\infty < X, \|\boldsymbol{y}_i\|_\infty < Y$ for $i \in [n]$, and $\boldsymbol{x} \in \mathcal{R}^{mn}$ and $\boldsymbol{y} \in \mathcal{R}^{mn}$ are the vectors corresponding to the concatenation of the n vectors $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ and $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n$ respectively.

We note that since this work focuses on labeled MCFE schemes for the IP functionality, the setup algorithm of all our constructions implicitly takes this functionality as an input.

3 Constructions

In this section, we present our MCFE constructions for the inner-product functionality based on the MDDH, DCR and LWR assumptions. Intuitively, we extend single-input IPFE techniques to their counterpart MCFE schemes by considering each slot as an independent client such that the clients can share the required randomness through the random oracle. While the IPFE constructions are based on a combination of the randomness and the public-key, we replace it with a combination of random oracle and the master key in our MCFE schemes. The use of random oracles for generating randomness also explains why we ended up with one-IND security (which can be easily extended to pos$^+$-security via an existing compiler [CDG+18b]). Regarding the security proof, we present in Section 3.5 a general proof sketch covering the main proof idea of all three constructions, despite some differences in the actual proof details.

3.1 MCFE based on the MDDH Assumption

In this section, we present a MCFE scheme supporting labels, based on the MDDH assumption. One can see this construction as an extension of single-input IPFE scheme where the term h_i^r is replaced with $\mathcal{H}(\ell)^{\mathbf{S}_i}$ (the value h_i is the public-key of IPFE scheme) and the value $\mathcal{H}(\ell)$ generates the required randomness. The MDDH assumption was initially introduced in [EHK+13]. We recap it here:

Definition 5 (Matrix Distribution [EHK+13]). Let $\ell, k \in \mathbb{N}$ with $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs (in polynomial time and with overwhelming probability) matrices in $\mathbb{Z}_p^{\ell\times k}$ of full rank k. We define $\mathcal{D}_k = \mathcal{D}_{k+1,k}$.

Definition 6 ($\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption [EHK+13]). Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. We define the advantage of an adversary $\mathcal{A}$ for the $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption in the following way:

$$\mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{D}_{\ell,k},\mathcal{A}}(\kappa) := |\Pr[\mathcal{A}(1^\kappa,\mathcal{G},[\mathbf{A}],[\mathbf{A}\boldsymbol{w}])=1] - \Pr[\mathcal{A}(1^\kappa,\mathcal{G},[\mathbf{A}],[\boldsymbol{u}])=1]|,$$

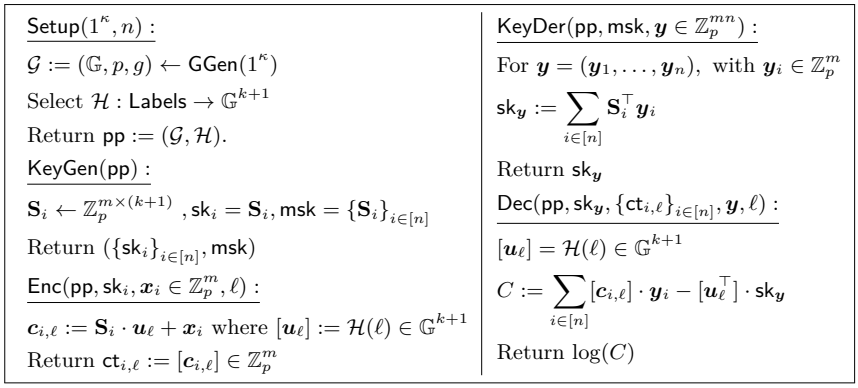

Fig. 2: MCFE based on the MDDH assumption. C is computed by group operations

where $\mathcal{G} = (\mathbb{G}, g, p) \leftarrow \mathsf{GGen}(1^\kappa), \mathbf{A} \leftarrow \mathcal{D}_{\ell,k}, \boldsymbol{w} \leftarrow \mathbb{Z}_p^k, \boldsymbol{u} \leftarrow \mathbb{Z}_p^\ell$. We say that the $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption ($\mathcal{D}_{\ell,k}$-MDDH) holds in group $\mathbb{G}$, if for all p.p.t. adversaries $\mathcal{A}$, there exists a negligible function negl such that: $\mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{D}_{\ell,k},\mathcal{A}}(\kappa) \leq \mathrm{negl}(\kappa)$.

Our MDDH-based MCFE construction is given in Fig. 2.

Theorem 7. *Assume that the $\mathcal{D}_k$-MDDH assumption holds, then the MCFE scheme described in Fig. 2 is* one*-IND-secure in the random oracle model.*

The correctness and the security of this construction are given in Appendix A.

3.2 MCFE based on the DCR Assumption

In this section we present a MCFE scheme based on the DCR assumption in the random-oracle model. As we mentioned, the main benefit of this construction is that one can retrieve the final result without computing the discrete-logarithm value. The following notations are used in this section. $\mathcal{D}_{\mathbb{Z}^k,\sigma}$ stands for the Gaussian distribution over $\mathbb{Z}^k$ with the standard deviation σ and the mean 0 (this notation is also used in the next section). $\mathbb{Z}_N$ is the additive group of integers modulo N and $\mathbb{Z}_N^*$ denotes the multiplicative group of integers modulo N. That is, including all $a \in \mathbb{Z}_N$ such that $\gcd(a, N) = 1$ where $\gcd(b, c)$ is the greatest common divisor of b and c. Let $N = pq$ be a safe modulus, meaning that p and q are large safe primes in the form of $p = 2p' + 1$ and $q = 2q' + 1$, where $p', q' > 2^\kappa$. In this paper $\mathsf{SP}(\kappa)$ is the algorithm producing safe-primes p, q as above. It is believed that for a given N as above it is hard to find p, q.

The single-input functional encryption scheme based on the Paillier cryptosystem has been proposed by Agrawal et al. [ALS16]. Their IPFE scheme is presented in Appendix B.1. In their construction, the encryption algorithm includes 2 main parts: $\mathsf{ct}_0 = g^r$ where $r \xleftarrow{R} \{1, \ldots, [\frac{N}{4}]\}$ and $\mathsf{ct}_i = (1 + N)^{x_i} \cdot h_i^r$ for $i = 1, \ldots, n$ where $h_i = g^{s_i}$ is the public key. The term $h_i^r = g^{rs_i}$ can be replaced with $\mathcal{H}(\ell)^{s_i}$ which removes the need for sharing a random r among the clients, since the random oracle $\mathcal{H}(\cdot)$ is publicly known. This explains the intuition for our MCFE scheme represented in Fig. 3. Regarding the security proof, the indistinguishable changes in RO-queries lead to an indistinguishable change in the (sub)lattice the master key belongs to (Note that the master secret key is chosen from lattice $\mathbb{Z}^n$ by a Gaussian distribution $\mathcal{D}_\sigma$). From there, a theorem from lattice-based cryptography and similar parameter setting to the single-input IPFE [ALS16] can guarantee the new distribution of the master secret key (along side the proper change in the RO-query associated with the challenge) is well enough for the security proof to work.

Definition 8 (Decisional Composite Residuosity (DCR) Assumption). Let $N = pq$ for two safe-primes p and q. We define the advantage of an adversary $\mathcal{A}$ for the DCR assumption

$\underline{\mathsf{Setup}(1^\kappa, n)}$:

– run $\mathsf{SP}(\kappa)$ to get (p, q) and compute $N = pq$.

– Let $\mathcal{H} : \mathsf{Labels} \to \mathbb{Z}^*_{N^2}$ be a full-domain hash function.

– set $X < \sqrt{N/2n}$

Return $\mathsf{pp} = (N, \mathcal{H}, X)$

$\underline{\mathsf{KeyGen}(\mathsf{pp})}$:

– sample $\boldsymbol{s} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}$ where

$\sigma > \sqrt{\kappa} \cdot N^{5/2}$ for the selective security

$\sigma > \sqrt{\kappa + 2n \cdot \log(2X)} \cdot N^{5/2}$ for the adaptive security.

Return $\mathsf{msk} = \boldsymbol{s}$ and $\mathsf{sk}_i = s_i$.

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, i, \ell)}$:

To encrypt a message $\boldsymbol{x} \in \mathbb{Z}^n$ with $|x_i| \leq X$:

– compute: $\mathsf{ct}_i = (1+N)^{x_i}.\mathcal{H}(\ell)^{s_i} \mod N^2$.

Return $\mathsf{ct} = \{\mathsf{ct}_i\}_i$

$\underline{\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \boldsymbol{y})}$:

For vector $\boldsymbol{y} \in \mathbb{Z}^n$

with $|y_i| \leq Y < \sqrt{N/2n}$:

– compute $\mathsf{sk}_y = \Sigma_i y_i.s_i$

Return sk_y

$\underline{\mathsf{Dec}(\mathsf{pp}, \boldsymbol{y}, \mathsf{sk}, \{\mathsf{ct}_{i,\ell}\}_{i\in[n]}, l)}$:

compute $C = \prod_i \mathsf{ct}_{i,\ell}^{y_i} \cdot \mathcal{H}(\ell)^{-\mathsf{sk}}$

Return $\dfrac{C - 1 \bmod N^2}{N}$

Fig. 3: MCFE based on the DCR assumption

in the following way:

$$\mathsf{Adv}^{\mathsf{DCR}}_{N,\mathcal{A}}(\kappa) := |\Pr[\mathcal{A}(1^\kappa, z^N \bmod N^2) = 1] - \Pr[\mathcal{A}(1^\kappa, z) = 1]|, \quad \text{where } z \leftarrow \mathbb{Z}^*_{N^2}.$$

We say that the DCR Assumption holds, if for all p.p.t. adversaries $\mathcal{A}$, there exists a negligible function negl such that: $\mathsf{Adv}^{\mathsf{DCR}}_{N,\mathcal{A}}(\kappa) \leq \mathrm{negl}(\kappa)$.

Theorem 9. *Assume that the DCR assumption holds, then the MCFE scheme described in Fig. 3 is* one*-IND-secure in the random-oracle model.*

The proof of the correctness and security can be found in Appendix B.2.

3.3 MCFE based on LWE Assumption

In this section, the notation $\lfloor a \rfloor$ denotes the largest integer number smaller than a.

Learning With Errors. The problem of Learning with Errors (LWE) was introduced in a seminal work of Regev [Reg05]. The idea for LWE problem is to provide a system of linear equations such that each equation is associated with an error term. Regev showed that in this case the number of equations does not really matter and it is hard to find any information about the secret. This problem is formally defined as follows.

Definition 10 (Decisional LWE assumption). Let q, α be functions of parameter n_0. The Learning with Error ($\mathsf{LWE}_{q,\alpha}$) problem is to distinguish two following distributions given access to polynomially many samples for a fixed vector $\boldsymbol{s} \xleftarrow{R} \mathbb{Z}_q^{n_0}$,

$$\mathcal{D} = \{(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e) \ : \ \boldsymbol{a} \xleftarrow{R} \mathbb{Z}_q^{n_0},\ e \xleftarrow{R} \mathcal{D}_{\mathbb{Z},\alpha q}\},\ \mathcal{D}' = \{(\boldsymbol{a}, u) \ : \ \boldsymbol{a} \xleftarrow{R} \mathbb{Z}_q^{n_0},\ u \xleftarrow{R} \mathbb{Z}_q\}$$

Concretely, for any adversary $\mathcal{A}$ there exists a negligible function negl such that:

$$\mathsf{Adv}^{\mathsf{LWE}}_{\mathcal{A}}(n_0) = |\Pr[\mathcal{A}^{\mathcal{D}(\boldsymbol{s},\cdot)}(\alpha, q, n_0) = 1] - \Pr[\mathcal{A}^{\mathcal{D}'(\cdot)}(\alpha, q, n_0) = 1]| \leq \mathrm{negl}(n_0)$$

where the oracles $\mathcal{D}(\boldsymbol{s}, \cdot)$ and $\mathcal{D}'(\cdot)$ output samples respectively from $\mathcal{D}$ (with a fixed secret $\boldsymbol{s}$) and $\mathcal{D}'$.

3.4 Our MCFE Construction Based on LWE

In this section we propose a MCFE construction based on the LWE problem as an extension of single-input FE presented by Agrawal et al. [ALS16] (see Appendix C.1). Although the intuition for our construction is similar to the previous constructions, we highlight here the differences that are the use of a rounding-map, and the part of the secret key that is uniform. In [ALS16] the mheLWE assumption is used to simulate all the queries in a correct way, as the inputs of the assumption are enough for this purpose. After applying this assumption (on one ciphertext) a product between parts of the master secret key and a uniformly random vector appears in the ciphertext. If the first factor of this multiplication has enough min-entropy, conditioned on the information available to the adversary, applying the leftover hash lemma guarantees that this product seems uniform, which concludes the proof. Now all is left to prove is that the part of the master secret key that is involved has enough min-entropy conditioned on what the adversary can see. Since in [ALS16], we are in the public-key setting, all the information (regarding the master key) the adversary can extract from all other honestly generated ciphertexts is the same as what it gets from the public-key. Thus, the leakage of all the honestly generated ciphertexts can be precisely quantified, and simulated using only the information contained in the public parameters. In this work, we need to change to the symmetric-key setting (as is the case in MCFE), so it is not as straightforward to quantify the leakage from all the ciphertext queries, and the information required to simulate the ciphertexts during the proof cannot be hidden in the public parameters. And in fact, in our case this leakage is really noticeable, especially since the ciphertexts are generated by different parties and each ciphertexts can leak information about different parts of the master secret key. Leveraging the use of a random oracle, we argue that the leakage coming from all the ciphertext queries can be deduced from leakage about some secret matrix, together with some noise term, under the LWE assumption. The leakage about the secret matrix is completely hidden by the uniform secret key $\boldsymbol{s}_i$, whereas the rounding-map completely removes the noise term $\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell$ when the parameters are carefully selected.

In our construction we are using a rounding-map which is formally defined as follows.

Definition 11 (Rounding-map from $\mathbb{Z}_q$ to $\mathbb{Z}_{q_0}$). For $q \geq q_0 \geq 2$, a rounding map $\lfloor . \rceil_{q_0} : \mathbb{Z}_q \longrightarrow \mathbb{Z}_{q_0}$ is defined as $\lfloor x \rceil_{q_0} = \lfloor (q_0/q) \cdot \bar{x} \rceil$ where $\bar{x} = x \mod q$ and $\lfloor \cdot \rceil$ is a classical rounding function over integers[8]. This notation can be extended component-wise to vectors and matrices over $\mathbb{Z}_q$.

Our MCFE scheme based on the LWE assumption is given in Fig. 4. The setting of the parameters is discussed separately in Appendix C.3.

Theorem 12. *The presented MCFE scheme in Fig. 4, is an one-IND-secure MCFE scheme under the LWE assumption and in the random-oracle model.*

The proof of the correctness and security can be found in Appendix C.2.

3.5 Security Analysis

Proof (Proof Overview). To prove the security of our constructions under the different assumptions, we consider the case where $\mathcal{A}$ only queries $\mathsf{QLeftRight}$ on one label $\ell^\star$, and never queries QEnc on $\ell^\star$. In more detail, we show that: $\mathsf{Adv}^{\text{one-1-label}}_{\mathsf{MCFE},\mathcal{A}'}(\kappa, n) \leq \mathrm{negl}(\kappa)$, where $\mathsf{Adv}^{\text{one-1-label}}_{\mathsf{MCFE},\mathcal{A}}(\kappa, n)$ is defined as described in Theorem 3. Then we use Theorem 4 to obtain the theorem.

For the proof of the 1-label security we proceed via a hybrid argument, using the games described in Fig. 5. The game G_0 corresponds to $\text{one-1-label}^{\mathsf{MCFE}}_0(\kappa, n, \mathcal{A})$ and the game G_7 to $\text{one-1-label}^{\mathsf{MCFE}}_1(\kappa, n, \mathcal{A})$. This yields: $\mathsf{Adv}^{\text{one-1-label}}_{\mathsf{MCFE},\mathcal{A}}(\kappa, n) = |\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_7}_{\mathcal{A}}(\kappa, n)|$.

[8] i.e., $\lfloor a \rceil$ is $\lfloor a \rfloor$ if $a \leq \lfloor a \rfloor + 1/2$ and it is $(\lfloor a \rfloor + 1)$ if $a > \lfloor a \rfloor + 1/2$.

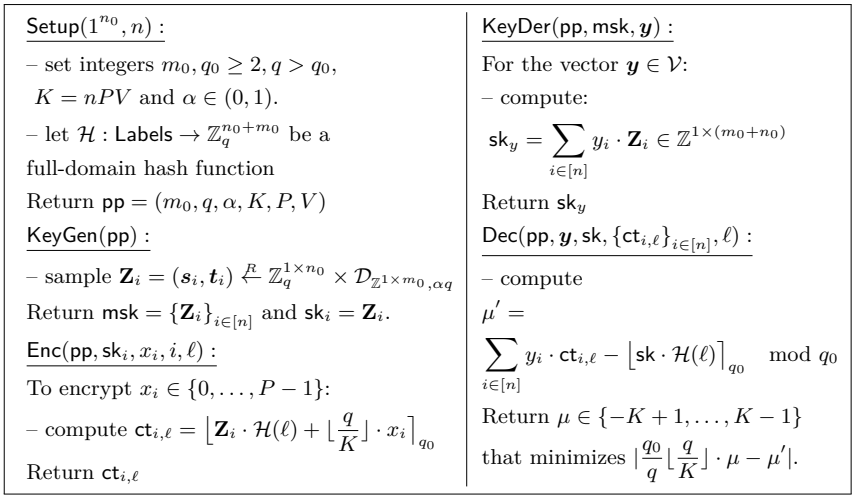

Fig. 4: MCFE based on the LWE assumption.

Intuitively, we change the random-oracle queries for $\ell \neq \ell^*$ and $\ell = \ell^*$ in a somehow orthogonal way. Meaning that, the proper change for $\ell \neq \ell^*$, changes the distribution of the master key (indistinguishable in the adversary's view) such that the multiplication of this master key and the new value for RO-query associated with ℓ^* can perfectly (for MDDH scheme) or statistically (for DCR and LWE schemes) hide the message in the challenge.

Game G_1: In game G_1, we replace the hash function $\mathcal{H}$, that is evaluated in every random-oracle query ℓ, with a random function RF. The random function has different outputs corresponding to the different schemes: The random function outputs an element $z \leftarrow \mathbb{Z}_p^{k+1}$ in the case of the MDDH scheme, an element $z \leftarrow \mathbb{Z}_{N^2}^*$ in the case of the DCR scheme and a couple $(\boldsymbol{a}, \boldsymbol{u})$ with $\boldsymbol{a} \leftarrow \mathbb{Z}_q^{n_0}$ and $\boldsymbol{u} \leftarrow \mathbb{Z}_q^{m_0}$ in the case of the LWE scheme. This results in a perfect transition from G_0 to G_1. This results in: $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_0}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(\kappa, n)| = 0$.

Game G_2: In game G_2, we answer the random-oracle queries for the label $\ell \neq \ell^*$ with an element that is indistinguishable from a random element, by relying on the corresponding computational assumption. We describe the random-oracle outputs under the label ℓ in more detail:

MDDH: we output a vector z such that z is contained in the span of $\mathbf{A}$, i.e. $z = \mathbf{A}\boldsymbol{y}$ with a random vector $\boldsymbol{y} \leftarrow \mathbb{Z}_p^k$.

DCR: we output an element $z^N \mod N^2$, with a random element $z \leftarrow \mathbb{Z}_{N^2}^*$.

LWE: we output a tuple $(\boldsymbol{a}, \mathbf{S} \cdot \boldsymbol{a} + \boldsymbol{e})$, with $\mathbf{S} \xleftarrow{R} \mathbb{Z}^{m_0 \times n_0}$, $\boldsymbol{a} \xleftarrow{R} \mathbb{Z}_q^{n_0}$, $e \xleftarrow{R} \mathcal{D}_{\mathbb{Z}^{m_0}, \alpha q}$. (we note that before proceeding to the next game for LWE scheme we need some extra games where we remove $\boldsymbol{t}_i \cdot \boldsymbol{e}$ and $\boldsymbol{t}_i \cdot \mathbf{S}$ from all ciphertexts queries through the property of the rounding-map and the uniform distribution of $\boldsymbol{s}_i$).

This results in: $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n)| \leq \mathrm{negl}(\kappa)$. where $\mathrm{negl}(\kappa)$ depends on the advantage of the attacker to the underlying assumption.

Here we note that the current modifications also change the distribution of the master key in the adversary's view (in an indistinguishable way).

MDDH the master secret key for MDDH scheme is distributed as $\mathbf{S} + \gamma(\boldsymbol{x}_1^{\ell^*} - \boldsymbol{x}_0^{\ell^*}) \cdot (\boldsymbol{a}^{\perp})^T$ for some $\gamma \in \mathbb{Z}_q$.

DCR the master secret key for DCR scheme is distributed as $\boldsymbol{s} + \lambda(\boldsymbol{x}_1^{\ell^*} - \boldsymbol{x}_0^{\ell^*}) \cdot \mu$ for some $\mu \in \mathbb{Z}^n$. Where $\lambda = 2p'q'$ is the order of elements $z^N \mod N^2$.

LWE the master secret key $\boldsymbol{t}$ for LWE scheme is distributed as $\boldsymbol{t} + (\boldsymbol{x}_1^{\ell^*} - \boldsymbol{x}_0^{\ell^*}) \cdot \mu$ for some $\mu \in \mathbb{Z}^n$ (here for the sake of simplicity, many details are missing).

Game G_3: In game G_3, we answer random-oracle queries for the label $\ell^\star$ as follows:

MDDH: we rely on the fact that $\mathbf{A}$ has rank k and find a vector $\boldsymbol{a}^\perp \leftarrow \mathbb{Z}_p^{k+1}$ such that $(\boldsymbol{a}^\perp)^\top \mathbf{A} = \mathbf{0}$ (this means $(\mathbf{A}, \boldsymbol{a}^\perp)$ is a base for $\mathbb{Z}^{k+1}$). Then we set $\mathsf{RF}(\ell^*) = \mathbf{A} \cdot \mathsf{RF}'(\ell^*) + \boldsymbol{a}^\perp \cdot \mathsf{RF}''(\ell^*)$ such that $\mathsf{RF}''(\ell) \neq 0$ (which is satisfies except with negligible probability negl), for random functions RF' and RF''.

DCR: we rely on an isomorphism ε from $\mathbb{Z}_N \times \mathbb{Z}_N^*$ to $\mathbb{Z}_{N^2}^*$ to write the random element $\mathsf{RF}(\ell^*) = z \bmod N^2$ in its corresponding representation $\varepsilon^{-1}(z) = (1+N)^a \cdot b^N \bmod N^2$ for $a, b \in \mathbb{Z}_N^*$ (which is satisfied expect with negligible probability negl).

LWE: we set $\mathsf{RF}(\ell^*) = \mathbf{S} \cdot \boldsymbol{a} + \boldsymbol{e} + \mathsf{RF}'(\ell^*)$ where RF' is a random function (again here there is an extra game which remove the term $\boldsymbol{t}_i \cdot \boldsymbol{e}$ from the ciphertext-challenge).

This results in: $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(\kappa, n)| \leq \mathrm{negl}(\kappa)$.

Game G_4 : In game G_4, we change the answers for left-or-right oracle queries under $\ell^\star$ from encryptions of x_i^0 to encryptions of x_i^1. for the MDDH we manage to show this change is perfectly-indistinguishable, while for the DCR and LWE schemes it needs a statistical argument to justify the transition from game G_3 to game G_4. It follows that: $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_4}(\kappa, n)| = f(\kappa)$. where for MDDH, $f(\kappa) = 0$ and for DCR and LWE schemes $f(\kappa) = 2^{-\kappa}$. In fact, we prove that a multiplication (which has already appeared in the ciphertext-challenge) of the master secret key (in its new representation) and the new values $\mathsf{RF}(\ell^*)$ can perfectly (for MDDH) or statistically (for DCR and LWE) hide the message in the challenge.

Games $\mathsf{G}_5, \ldots, \mathsf{G}_8$ One can define these games as the backward-counterparts of games G_3 to G_0 while hidden bit associated with the challenge is $b = 1$.

Putting everything together, we obtain the theorem. □

Game	$\mathsf{ct}_{i,\ell^\star}$	$\boldsymbol{u}_\ell$	justification/remark
G_0	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^0, \ell^\star)$	$\mathcal{H}(\ell)$	
G_1	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^0, \ell^\star)$	$\boxed{\mathsf{RF}(\ell)}$	Replace the hash function with a random function
G_2	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^0, \ell^\star)$	$\mathsf{RF}(\ell), \ell = \ell^\star$ $\boxed{z,\ \ell \neq \ell^\star}$	Simulate the hash function for $\ell \neq \ell^\star$ using z which is indistinguishable from a random element if the underlying hardness assumption (MDDH, DCR, LWE) holds
G_3	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^0, \ell^\star)$	$\boxed{\overline{\mathsf{RF}}(\ell), \ell = \ell^\star}$ $z,\ \ell \neq \ell^\star$	Simulate the hash function for $\ell = \ell^\star$ using a different representation of $\mathsf{RF}(\ell^*)$ corresponding to the underlying assumption
G_4	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boxed{\boldsymbol{x}_i^1}, \ell^\star)$	$\overline{\mathsf{RF}}(\ell), \ell = \ell^\star$ $z',\ \ell \neq \ell^\star$	Change from left to right encryption

Fig. 5: Overview of the games to prove the security of the MCFE schemes.

4 Implementation

To show the efficiency of our schemes, we provide three implementations of schemes described on Figs. 2 to 4. In this table encryption time is per slot. Before describing the choices made during implementation, we show the timings for these implementations on Fig. 6.

Before heading into details relative to each implementation, let us review the choices common to the three implementations.

Instantiating the random oracle. We chose to replace the random oracle by the SHA-256 hash function, thus we were able to take advantage of the OpenSSL library, that provides efficient and well spread implementation of SHA-256. As the size of the random oracle were different to the output size of SHA-256, we used it multiple times, changing the input each time by incrementing a counter that was concatenated with the label.

Operation	mpk Generation	msk Generation	sk_y Derivation	Encryption	Decryption
DDH	0.038843 s	0.028417 s	negligible	0.000439 s	m μs
DCR	0.201445 s	1.576873 s	negligible	0.280378 s	0.313167 s
LWE	n/a	0.017957 s	0.048872 s	0.001207 s	0.000989 s

Fig. 6: Timings of the concrete implementations, encrypting vectors of dimension 100. The code was run on a laptop running an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz. m is the discrete-logarithm value to be retrieved (the inner-product value).

Parameter	Message space	Ciphertext size	Secret key size
DDH	bounded by computation	512 bits	512 bits
DCR	4096 bits	9192 bits	55152 bits
LWE	20 bits	32 bits	704000 bits

Fig. 7: Capacity of the implementations and memory cost.

Choice of the message space. We tested our code with vectors of dimension 100, computing the sum of the first 100 squares. We wanted to keep the message space as small as $2^{20} = 1048576$, in order for the LWE ciphertexts to be held by 32 bits integers, then the message space was kept the same for the DCR implementation for fair comparison. It is worth noting that the DCR implementation could have encrypted vectors with coordinates up to 4000 bits large without being any slower, since the complexity only depends on the dimensions of the vectors, and the only bound on the message space is that it has to stay smaller than the RSA number N. On the other hand, the DDH implementation is limited by the computation of a discrete logarithm regardless of the parameter choice, and the LWE implementation can hardly increase the message space without having to pump the parameters Indeed, the modulus is tied only to message space and not so to security as in the case of DCR, so we don't have this spare space in the message space. We wanted to keep the ciphertexts small enough so that we can rely on fast hardware optimizations on arithmetic operations, using bigger message spaces would require to use large number libraries, but is definitely doable.

Discussion. The timings we have are very reasonable, and can be brought down quite a lot for any given application. We tried to push the parameters so that our implementations can be trusted as proofs of concept without knowing what applications will come in the future, but for given specific requirements in terms of security and efficiency, there is a lot of room for improvement. We also tried to give a flexible implementation that can be used to estimate the timings for different parameters easily. This also leaves room for optimization once the parameters are chosen for a particular application. If we are to compare the different schemes, it looks like the scheme based on LWE is much more efficient than the scheme based on DCR. One has to be careful when making such comparisons. Indeed, the DCR scheme supports very big messages, because the modulo N has to be set very large for security reasons. In comparison, the efficiency of the LWE scheme would degrade with the size of the messages to encrypt, so for applications with large messages, the DCR implementation might actually become much faster.

Details of implementation for each of mentioned schemes are separately discussed in Appendix E.

Acknowledgments.

This work was supported in part by the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 339563 – CryptoCloud), the European Community's Horizon 2020 Project FENTEC (Grant Agreement no. 780108), and the French FUI ANBLIC Project.

References

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

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, Heidelberg, 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, Heidelberg, 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, Heidelberg, August 2018.

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, Heidelberg, April / May 2017.

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

AR17. Shweta Agrawal and Alon Rosen. Functional encryption for bounded collusions, revisited. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 173–205. Springer, Heidelberg, November 2017.

BCP03. Emmanuel Bresson, Dario Catalano, and David Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 37–54. Springer, Heidelberg, November / December 2003.

BCP14. Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, Heidelberg, February 2014.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

BPR12. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

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

CDG+18b. Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Multi-client functional encryption with repetition for inner product. Cryptology ePrint Archive, Report 2018/1021, 2018. https://eprint.iacr.org/2018/1021.

Coc01. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, Heidelberg, December 2001.

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

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

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.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

LST18. Benoît Libert, Damien Stehlé, and Radu Titiu. Adaptively secure distributed PRFs from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 391–421. Springer, Heidelberg, November 2018.

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

MW17. Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 455–485. Springer, Heidelberg, August 2017.

O'N10. Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/2010/556.

OSW07. Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 195–203. ACM Press, October 2007.

Pai99. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Heidelberg, May 1999.

PR06. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 145–166. Springer, Heidelberg, March 2006.

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.

Wat05. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Heidelberg, May 2005.

Wat11. Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Heidelberg, March 2011.

Wat15. Brent Waters. A punctured programming approach to adaptively secure functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 678–697. Springer, Heidelberg, August 2015.

A MCFE based on the MDDH Assumption

Here we discuss the correctness and the security of our MDDH-based MCFE construction (Fig. 2).

Correctness. To prove the correctness of our construction, we consider the output of the decryption procedure for a correctly generated encryptions of the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{Z}_p$ under the same label $\ell \in \mathsf{Labels}$ using a correctly generated functional key $\mathsf{sk}_{\boldsymbol{y}}$ with $\boldsymbol{y} := (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \in \mathbb{Z}_p^{mn}$:

$$\begin{aligned} C &= \sum_{i\in[n]} [\boldsymbol{c}_{i,\ell}] \cdot \boldsymbol{y}_i - [\boldsymbol{u}_\ell^\top] \cdot \mathsf{sk}_{\boldsymbol{y}} \\ &= \sum_{i\in[n]} [\mathbf{S}_i \cdot \boldsymbol{u}_\ell + \boldsymbol{x}_i] \cdot \boldsymbol{y}_i - [\boldsymbol{u}_\ell^\top] \cdot \sum_{i\in[n]} \mathbf{S}_i^\top \boldsymbol{y}_i \\ &= \sum_{i\in[n]} [\langle \mathbf{S}_i \cdot \boldsymbol{u}_\ell, \boldsymbol{y}_i \rangle + \langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle] - \sum_{i\in[n]} [\langle \mathbf{S}_i \cdot \boldsymbol{u}_\ell, \boldsymbol{y}_i \rangle] \\ &= \sum_{i\in[n]} [\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle] = [\langle \boldsymbol{x}, \boldsymbol{y} \rangle] \end{aligned}$$

Since the decryption procedure outputs $\log(C)$, correctness directly follows.

After showing the correctness of our scheme, we are also proving its security.

Theorem 13. *Assume that the $\mathcal{D}_k$-MDDH assumption holds, then the MCFE scheme described in Fig. 2 is* one*-IND-secure in the random-oracle model. Namely, for any p.p.t. adversary $\mathcal{A}$, there exist a p.p.t. adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\text{one-IND}}_{\mathsf{MCFE},\mathcal{A}}(\kappa,n) \leq q_{\mathsf{Enc}}\left(4\cdot\mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{B}}(\kappa)+\frac{2}{p-1}+\frac{2}{p}\right),$$

where q_{Enc} denotes the number of distinct labels queried to QLeftRight*.*

Proof. To prove this statement, we consider the case where $\mathcal{A}$ only queries QLeftRight on one label $\ell^\star$, and never queries QEnc on $\ell^\star$. We build a p.p.t. adversary $\mathcal{B}$ such that: $\mathsf{Adv}^{\text{one-1-label}}_{\mathsf{MCFE},\mathcal{A}}(\kappa,n) \leq 4\cdot\mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{B}}(\kappa)+\frac{2}{p-1}+\frac{2}{p}$, where $\mathsf{Adv}^{\text{one-1-label}}_{\mathsf{MCFE},\mathcal{A}}(\kappa,n)$ is defined as described in Theorem 3. Then we use Theorem 4 to obtain the theorem.

For the proof of the 1-label security we proceed via a hybrid argument, using the games described in Fig. 8. The game G_0 corresponds to one-IND$_0^{\mathsf{MCFE}}(\kappa,n,\mathcal{A})$ and the game G_4 to one-IND$_1^{\mathsf{MCFE}}(\kappa,n,\mathcal{A})$. This yields:

$$\mathsf{Adv}^{\text{one-1-label}}_{\mathsf{MCFE},\mathcal{A}}(\kappa,n) = |\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa,n) - \mathsf{Win}^{\mathsf{G}_7}_{\mathcal{A}}(\kappa,n)|.$$

Game G_1: In game G_1, we replace the hash function $\mathcal{H}$, that is evaluated in every random-oracle query ℓ, with a truly random function RF. This results in a perfect transition from G_0 to G_1. Namely, in Theorem 15, we show that:

$$|\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa,n) - \mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa,n)| = 0.$$

Game G_2: In game G_2, we replace the random function RF, that is evaluated in every random-oracle query ℓ, with an element in the span of a matrix $\mathbf{A}$, sampled from a matrix distribution $\mathcal{D}_k$. To generate the final element in the span, we multiply $\mathbf{A}$ with a random element in $\mathbb{Z}_p^k$, sampled using the random function RF′. The transition from G_1 to G_2 is justified by the Multi-MDDH assumption. Namely, in Theorem 16, we exhibit a p.p.t. adversary $\mathcal{B}_0$ such that:

$$|\mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa,n) - \mathsf{Win}^{\mathsf{G}_2}_{\mathcal{A}}(\kappa,n)| \leq \mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{B}_0}(\kappa) + \frac{1}{p-1}.$$

Game G_3: In game G_3, we answer a random-oracle query for the label $\ell^\star$ with an element that is generated as a linear combination of $\mathbf{A}$ and $\boldsymbol{a}^\perp$, with $\boldsymbol{a}^\perp \leftarrow \mathbb{Z}_p^{k+1}$ such that $(\boldsymbol{a}^\perp)^\top\mathbf{A} = \mathbf{0}$. For every other random-oracle query $\ell \neq \ell^\star$, the output is still an element in the span of $\mathbf{A}$. The transition between G_2 and G_3 is justified by the MDDH assumption. Namely, in Theorem 17, we exhibit a p.p.t. adversary $\mathcal{B}_1$ such that:

$$|\mathsf{Win}^{\mathsf{G}_2}_{\mathcal{A}}(\kappa,n) - \mathsf{Win}^{\mathsf{G}_3}_{\mathcal{A}}(\kappa,n)| \leq \mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{B}_1}(\kappa) + \frac{1}{p}.$$

Game G_4 : In game G_4, we change the answers for left-or-right oracle queries under $\ell^\star$ from encryptions of $x_i^{0,\ell^\star}$ to encryptions of $x_i^{1,\ell^\star}$. We rely on complexity leveraging and a statistical argument to justify the transition from game G_3 to game G_4. Namely, in Theorem 18, we show that:

$$|\mathsf{Win}^{\mathsf{G}_3}_{\mathcal{A}}(\kappa,n) - \mathsf{Win}^{\mathsf{G}_4}_{\mathcal{A}}(\kappa,n)| = 0.$$

Game G_5: In game G_5, we answer a random-oracle query for the label $\ell^\star$ in the same way as for every other label $\ell \neq \ell^\star$, i.e. with an element in the span of $\mathbf{A}$. The transition from game G_4 to G_5 is symmetric to the transition from G_2 to G_3, justified by the MDDH assumption. Namely, it can be proven as in Theorem 17 that there exists a p.p.t. adversary $\mathcal{B}_2$ such that:

$$|\mathsf{Win}^{\mathsf{G}_4}_{\mathcal{A}}(\kappa,n) - \mathsf{Win}^{\mathsf{G}_5}_{\mathcal{A}}(\kappa,n)| \leq \mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{B}_2}(\kappa) + \frac{1}{p}.$$

We defer to the proof of Theorem 17 for further details.

Game G_6: In game G_6, we answer every random-oracle query ℓ with the evaluation of a random function $\mathsf{RF}(\ell)$ instead of an element in the span of $\mathbf{A}$. The transition from game G_5 to G_6 is symmetric to the transition from G_1 to G_2, justified by the Multi-MDDH assumption. Namely, it can be proven as in Theorem 16 that there exists a p.p.t. adversary $\mathcal{B}_3$ such that:

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_5}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_6}(\kappa, n)| \leq \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{MDDH}}(\kappa) + \frac{1}{p-1}.$$

We defer to the proof of Theorem 16 for further details.

Game G_7: This game is one-$\mathsf{IND}_1^{\mathsf{MCFE}}(\kappa, n, \mathcal{A})$. The transition from G_6 to G_7 is symmetric to the transition from G_0 to G_1. Namely, it can be proven as in Theorem 15 that:

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_6}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_7}(\kappa, n)| = 0.$$

We defer to the proof of Theorem 15 for further details.

Putting everything together, we obtain the theorem. □

Game	$\mathsf{ct}_{i,\ell^\star}$	$\boldsymbol{u}_\ell$	justification/remark
G_0	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{0,\ell^\star}, \ell^\star)$	$\mathcal{H}(\ell)$	
G_1	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{0,\ell^\star}, \ell^\star)$	$\boxed{\mathsf{RF}(\ell), \text{ with } \mathsf{RF}(\ell) \in \mathbb{Z}_p^{k+1}}$	Replace the hash function with a random function
G_2	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{0,\ell^\star}, \ell^\star)$	$\boxed{\begin{array}{l}\mathbf{A} \leftarrow \mathcal{D}_k \\ \mathbf{A} \cdot \mathsf{RF}'(\ell), \text{ with } \mathsf{RF}'(\ell) \in \mathbb{Z}_p^k\end{array}}$	Simulate the hash function using the span of $\mathbf{A}$ (Multi-MDDH)
G_3	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{0,\ell^\star}, \ell^\star)$	$\begin{array}{l}\boxed{\begin{array}{l}\mathbf{A} \leftarrow \mathcal{D}_k, \boldsymbol{a}^\perp \leftarrow \mathbb{Z}_p^{k+1} \setminus \{\mathbf{0}\} \\ \text{s.t. } (\boldsymbol{a}^\perp)^\top \mathbf{A} = \mathbf{0} \\ \mathbf{A} \cdot \mathsf{RF}'(\ell) + \boldsymbol{a}^\perp \cdot \mathsf{RF}''(\ell), \text{ if } \ell = \ell^\star\end{array}} \\ \mathbf{A} \cdot \mathsf{RF}'(\ell), \text{ if } \ell \neq \ell^\star \\ \text{with } \mathsf{RF}'(\ell) \in \mathbb{Z}_p^k \text{ and } \mathsf{RF}''(\ell) \in \mathbb{Z}_p^*\end{array}$	For $\ell = \ell^\star$ simulate using a random element from the span of $\mathbf{A}$ and $\boldsymbol{a}^\perp$
G_4	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boxed{\boldsymbol{x}_i^{1,\ell^\star}}, \ell^\star)$	$\begin{array}{l}\mathbf{A} \leftarrow \mathcal{D}_k, \boldsymbol{a}^\perp \leftarrow \mathbb{Z}_p^{k+1} \setminus \{\mathbf{0}\}, \\ \text{s.t. } (\boldsymbol{a}^\perp)^\top \mathbf{A} = \mathbf{0} \\ \mathbf{A} \cdot \mathsf{RF}'(\ell) + \boldsymbol{a}^\perp \cdot \mathsf{RF}''(\ell), \text{ if } \ell = \ell^\star \\ \mathbf{A} \cdot \mathsf{RF}'(\ell), \text{ if } \ell \neq \ell^\star \\ \text{with } \mathsf{RF}'(\ell) \in \mathbb{Z}_p^k \text{ and } \mathsf{RF}''(\ell) \in \mathbb{Z}_p^*\end{array}$	Change from left to right encryption
G_5	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{1,\ell^\star}, \ell^\star)$	$\boxed{\begin{array}{l}\mathbf{A} \leftarrow \mathcal{D}_k \\ \mathbf{A} \cdot \mathsf{RF}'(\ell), \text{ with } \mathsf{RF}'(\ell) \in \mathbb{Z}_p^k\end{array}}$	Simulate the hash function using the span of $\mathbf{A}$ (Multi-MDDH)
G_6	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{1,\ell^\star}, \ell^\star)$	$\boxed{\mathsf{RF}(\ell), \text{ with } \mathsf{RF}(\ell) \in \mathbb{Z}_p^{k+1}}$	Replace the hash function with a random function
G_7	$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i^{1,\ell^\star}, \ell^\star)$	$\boxed{\mathcal{H}(\ell)}$	Replace the random function with a hash function

Fig. 8: Overview of the games to prove the security of the MCFE scheme based on the MDDH assumption.

Theorem 14 (Random self-reducibility of MDDH [EHK+13]). *For any p.p.t. adversary $\mathcal{A}$, there exist a p.p.t. adversary $\mathcal{B}$ such that*

$$|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], ([\mathbf{A}\boldsymbol{w}_i])_{i\in[n]}) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], ([\boldsymbol{v}_i])_{i\in[n]}) = 1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{MDDH}}(\kappa) + \frac{1}{p-1},$$

with $\mathbf{A} \leftarrow \mathcal{D}_k, \boldsymbol{w}_i \leftarrow \mathbb{Z}_p^k$ *and* $\boldsymbol{v}_i \leftarrow \mathbb{Z}_p^{k+1}$ *for all* $i \in [n]$.

Lemma 15 (Transition from G_0 to G_1). *For any p.p.t. adversary $\mathcal{A}$, it holds that*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_0}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(\kappa, n)| = 0.$$

Proof. This is a perfect simulation of the random-oracle $\mathcal{H}$ using a random function $\mathsf{RF}(\ell) \in \mathbb{Z}_p^{k+1}$, which gives us $|\mathsf{Win}_{\mathsf{Adv},\mathcal{A}}^{\mathsf{G}_0}(\kappa, n) - \mathsf{Win}_{\mathsf{Adv},\mathcal{A}}^{\mathsf{G}_1}(\kappa, n)| = 0$. □

Lemma 16 (Transition from G_1 to G_2). *For any p.p.t. adversary $\mathcal{A}$, there exists a p.p.t. adversary $\mathcal{B}$ such that*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{MDDH}}(\kappa) + \frac{1}{p-1}.$$

Proof. We replace the random function $\mathsf{RF}(\ell) \in \mathbb{Z}_p^{k+1}$ in the random oracle with a truly random element in the span of $\mathbf{A}$, where matrix $\mathbf{A}$ is sampled from the Gaussian distribution $\mathcal{D}_k$ and multiplied with a random element generated by $\mathsf{RF}'(\ell) \in \mathbb{Z}_p^k$. This directly mirrors the random self-reducibility of MDDH assumption as described in Theorem 14, which yields $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{MDDH}}(\kappa) + \frac{1}{p-1}$ as a bound. □

Lemma 17 (Transition from G_2 to G_3). *For any p.p.t. adversary $\mathcal{A}$, there exists a p.p.t. adversary $\mathcal{B}'$ such that*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(\kappa, n)| \leq \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{MDDH}}(\kappa) + \frac{1}{p}.$$

Proof. We change the output of the random oracle for the query $\ell^\star$ from an element output in the span of $\mathbf{A}$ to a linear combination of the matrices $\mathbf{A}$ and the vector $\boldsymbol{a}^\perp$. In more detail, we generate a random vector in $\mathbb{Z}_p^{k+1}$ by sampling $\boldsymbol{u}_1 \leftarrow \mathbb{Z}_p^k$ and $\boldsymbol{u}_2 \leftarrow \mathbb{Z}_p^*$ and computing $\mathbf{A} \cdot \boldsymbol{u}_1 + \boldsymbol{a}^\perp \cdot \boldsymbol{u}_2$. Due to the way in which $\boldsymbol{a}^\perp$ is constructed and the we can span the whole space $\mathbb{Z}_p^{k+1}$ using $\mathbf{A}$ and $\boldsymbol{a}^\perp$. This sampling is justified by the MDDH assumption, it changes the view of the adversary by statistical distance of $\frac{1}{p}$, which yields to the above mentioned bound. □

Lemma 18 (Transition from G_3 to G_4). *For any p.p.t. adversary $\mathcal{A}$, it holds that*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_4}(\kappa, n)| = 0.$$

Proof. We proceed in two different steps for this part of the proof:

1. We apply a complexity leveraging argument to change the games G_3 and G_4 from the adaptive security case into the selective security case. The resulting games are denoted with $\mathsf{G}_3^\star$ and $\mathsf{G}_4^\star$.
2. We use a statistical argument to prove the transition from $\mathsf{G}_3^\star$ to $\mathsf{G}_4^\star$.

1. Let $\mathcal{A}_t$ be an adversary in the adaptive secure games G_t and $\mathcal{B}_t^\star$ an adversary in the corresponding selectively secure games $\mathsf{G}_t^\star$, for $t = 3, 4$.

We transform the adversary $\mathcal{A}_t$ into a selective adversary $\mathcal{B}_t^\star$, such that:

$$\mathsf{Adv}_{\mathcal{A}_t}^{\mathsf{G}_t}(\kappa, n) \leq 2^{-n} \cdot (2X)^{-2mn} \cdot \mathsf{Adv}_{\mathcal{B}_t^\star}^{\mathsf{G}_t^\star}(\kappa, n), \text{ for } t = 3, 4.$$

We describe the simulation of the adaptive security by the adversary $\mathcal{B}_t^\star$ to $\mathcal{A}_t$, for $t = 3, 4$, when $\mathcal{B}_t^\star$ interacts with the corresponding selective security experiment.

2. After adversary $\mathcal{B}_t^\star$ made its guesses $\boldsymbol{z}_i = (\boldsymbol{x}_i^{0,\ell^\star}, \boldsymbol{x}_i^{1,\ell^\star})$ for the set label $\ell^\star$ and all $i \in [n]$. It simulates $\mathcal{A}_t$'s experiment using its own selective experiment. When $\mathcal{B}_t^\star$ receives a challenge query

from $\mathcal{A}_t$, it checks if the guess was successful. If it was, it continues simulating $\mathcal{A}_t$'s experiment, otherwise, it returns 0. When the guess is successful, $\mathcal{B}_t^\star$ perfectly simulates $\mathcal{A}_t$'s view.

To show that the two distributions (with $\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp \rangle \neq 0$):

$$\{\mathbf{S}_i\}_{i\in[n],\boldsymbol{z}_i} \text{ and } \{\mathbf{S}_i - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})(\boldsymbol{a}^\perp)^\top\}_{i\in[n],\boldsymbol{z}_i}$$

are indistinguishable, we show the simulation of $\mathcal{B}_t^\star$ for the different queries:

Corruption oracle $\mathsf{QCor}(i)$: If slot i gets corrupted (and the simulation happened successfully), it holds that $\boldsymbol{x}_i^{0,\ell^\star} = \boldsymbol{x}_i^{1,\ell^\star}$ under label $\ell^\star$. This results in $\mathbf{S}_i - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})(\boldsymbol{a}^\perp)^\top = \mathbf{S}_i - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle} \cdot \mathbf{0} \cdot (\boldsymbol{a}^\perp)^\top = \mathbf{S}_i$.

Key oracle $\mathsf{QKeyD}(\boldsymbol{y})$: The key generation procedure will output:

$$\begin{aligned}\mathsf{sk}_{\boldsymbol{y}} &= \sum_{i\in[n]} \mathbf{S}_i^\top \cdot \boldsymbol{y}_i - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{a}^\perp)(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})^\top \cdot \boldsymbol{y}_i \\ &= \sum_{i\in[n]} \mathbf{S}_i^\top \cdot \boldsymbol{y}_i - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{a}^\perp)\underbrace{(\langle \boldsymbol{x}_i^{0,\ell^\star}, \boldsymbol{y}_i\rangle - \langle \boldsymbol{x}_i^{1,\ell^\star}, \boldsymbol{y}_i\rangle)}_{=0},\end{aligned}$$

which is equal to a functional key generated using $\{\mathbf{S}_i\}_{i\in[n],\boldsymbol{z}_i}$.

Left-or-Right query $\mathsf{QLeftRight}(i, \boldsymbol{x}_i^0, \boldsymbol{x}_i^1, \ell^\star)$: The term $-\frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})(\boldsymbol{a}^\perp)^\top$ also appears in the encryption queries under label $\ell^\star$.

The ciphertext under label $\ell^\star$ has the structure

$$\begin{aligned}&[\mathbf{S}_i\boldsymbol{u}_{\ell^\star}] - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})(\boldsymbol{a}^\perp)^\top[\boldsymbol{u}_{\ell^\star}] + [\boldsymbol{x}_i^{0,\ell^\star}] \\ =&[\mathbf{S}_i\boldsymbol{u}_{\ell^\star}] + [-\underbrace{\frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle} \cdot \langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}_{=1}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star}) + \boldsymbol{x}_i^{0,\ell^\star}] \\ =&[\mathbf{S}_i\boldsymbol{u}_{\ell^\star}] + [\boldsymbol{x}_i^{1,\ell^\star}].\end{aligned}$$

Encryption query $\mathsf{QEnc}(i, \boldsymbol{x}_i, \ell)$: If an encryption query gets asked for a label $\ell \neq \ell^\star$, with $\boldsymbol{u}_\ell = \mathbf{A} \cdot \mathsf{RF}'(\ell)$, then the ciphertext has the following structure:

$$\begin{aligned}&[\mathbf{S}_i\boldsymbol{u}_\ell] - \frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})(\boldsymbol{a}^\perp)^\top[\mathbf{A} \cdot \mathsf{RF}'(\ell)] + [\boldsymbol{x}_i] \\ =&[\mathbf{S}_i\boldsymbol{u}_\ell] + [-\frac{1}{\langle \boldsymbol{u}_{\ell^\star}, \boldsymbol{a}^\perp\rangle}(\boldsymbol{x}_i^{0,\ell^\star} - \boldsymbol{x}_i^{1,\ell^\star})\underbrace{(\boldsymbol{a}^\perp)^\top\mathbf{A}}_{=0} \cdot \mathsf{RF}'(\ell)] + [\boldsymbol{x}_i] \\ =&[\mathbf{S}_i\boldsymbol{u}_\ell] + [\boldsymbol{x}_i]\end{aligned}$$

□

B MCFE Based on DCR Assumption

B.1 A Review on Single-Input FE based on DCR

In this section, we recall the single-input functional encryption scheme based on Paillier, proposed by Agrawal et al. [ALS16]. Their construction is presented in Fig. 9 which is mainly based on the idea of [BCP03]. Bresson et al. [BCP03] present a public key cryptosystem with two trapdoors: one is λ which needs the knowledge of the factorization of N, and the second trapdoor is the secret key which makes the decryption possible without knowing λ. The security of this scheme is based on a variant of the DDH assumption over $\mathbb{Z}_{N^2}^*$. Agrawal et al. extended this

idea to cyclic subgroups of $2N$ residues modulo N^2 to design a secure functional encryption system based on the DCR assumption. They showed that the decryption algorithm based on the second trapdoor can be adopted to the FE setting by having functional secret keys (instead of the secret key in public-key setting [BCP03]).

The use of cyclic group of $2N$ residues modulo N^2 (instead of the quadratic residues group in [BCP03]) and the DCR assumption makes it possible to ensure that secret keys do not leak sensitive information in FE case [ALS16].

Remark 19 (A note on the space and distribution of master secret key). The master secret key is sampled from $\mathbb{Z}$ through a Gaussian sampler. This can guarantee the correctness and the security as well. More precisely, everyone holding the secret key and ciphertext should be able to compute the value $C \mod N^2$ and it means that sk has to be given over $\mathbb{Z}$ or modulo any multiple of λ (due to the fact that the order of g is λ and $\mathsf{ct}_0^{-\mathsf{sk}} = \mathsf{ct}_0^{\mathsf{sk} \mod \lambda} \mod N^2$). Having sk modulo λ can leak the value of λ through different secret key queries and it means that anyone can directly decrypt the ciphertext to get the plain message x in a similar way to [Pai99] or [BCP03] based on the first trapdoor. Thus, the value sk cannot be given modulo $k \cdot \lambda$ for an arbitrary $k \in \mathbb{Z}$. Moreover, during the security proof, it is required that the master secret key is defined over the same set that sk is defined (since the distribution of the master secret key in the view of the adversary is conditioned on the secret key values). Putting it all together, the master secret key $\boldsymbol{s}$ should be sampled from the set $\mathbb{Z}$ and Gaussian distribution is a good candidate for this sampling. In fact, based on its density function if the standard deviation is noticeably larger than N, then it seems like uniform distribution modulo N and this fact is used in the security proof.

$\underline{\mathsf{Setup}(1^\kappa, n)}$:

– run $\mathsf{SP}(\kappa)$ to get safe-primes p, q.

– compute $N = pq$

– sample $g' \xleftarrow{R} \mathbb{Z}^*_{N^2}$

– compute $g = g'^{2N} \mod N^2$.

Return $\mathsf{pp} = (N, g)$

$\underline{\mathsf{KeyGen}(\mathsf{pp})}$:

– sample $\boldsymbol{s} \leftarrow D_{\mathbb{Z}^n,\sigma}$ where $D_{\mathbb{Z}^n,\sigma}$ is the Gaussian distribution of standard deviation $\sigma > \sqrt{\kappa} \cdot N^{5/2}$.

– compute $h_i = g^{s_i} \mod N^2$.

Return $\begin{cases} \mathsf{mpk} = \{h_i\}_{i\in[n]} \\ \mathsf{msk} = \boldsymbol{s} \end{cases}$

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{mpk}, \boldsymbol{x})}$:

For vector $\boldsymbol{x} \in \mathbb{Z}^n$ with $||\boldsymbol{x}||_\infty \leq X < \sqrt{N/n}$:

– sample $r \xleftarrow{R} \{0, \ldots, [\frac{N}{4}]\}$.

– compute $\begin{cases} \mathsf{ct}_0 = g^r \mod N^2, \\ \mathsf{ct}_i = (1+N)^{x_i} \cdot h_i^r \mod N^2 \end{cases}$

Return $\mathsf{ct} = (\mathsf{ct}_0, \{\mathsf{ct}_i\}_i)$

$\underline{\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \boldsymbol{y})}$:

For vector $\boldsymbol{y} \in \mathbb{Z}^n$ with $||\boldsymbol{y}||_\infty \leq Y < \sqrt{N/n}$:

– compute $\mathsf{sk}_y = \Sigma_i y_i.s_i$ over $\mathbb{Z}$

Return sk_y

$\underline{\mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \boldsymbol{y}, \mathsf{sk}, \mathsf{ct})}$:

– compute $C = \prod_i \mathsf{ct}_i^{y_i} \cdot \mathsf{ct}_0^{-\mathsf{sk}}$

Return $\dfrac{C - 1 \mod N^2}{N}$

Fig. 9: Single-input FE based on the DCR assumption [ALS16]

B.2 Correctness and Security of our DCR-based MCFE

Here we recap the definition of Carmichael function and some theorems which would be used in the security proof of DCR-based MCFE scheme. In the following definition $\mathrm{lcm}(b, c)$ stands for the least common multiple of b and c.

Definition 20 (Carmichael function). The Carmichael function is defined as :

$$\lambda(n) = \begin{cases} \text{lcm}(\lambda(p_1^{a_1}), \ldots, \lambda(p_k^{a_k})) & n = p_1^{a_1} \cdot p_2^{a_2} \cdot \ldots \cdot p_k^{a_k} \\ p^{a-1}(p-1) & n = p^a, p \neq 2 \text{ or } a < 2 \\ p^{a-1}(p-1)/2 & n = p^a, p = 2, a > 2 \end{cases}$$

All through the paper we denote $\lambda(N)$ as λ. The following lemma is due to the Carmichael theorem.

Theorem 21 (Carmichael Theorem). *Assume that $N = pq$ is a safe-prime modulus, then for any $w \in \mathbb{Z}^*_{N^2}$:*

$$\begin{cases} w^{\lambda} = 1 \ mod \ N \\ w^{N\lambda} = 1 \ mod \ N^2 \end{cases}$$

The following lemma is introducing an isomorphism between $\mathbb{Z}_N \times \mathbb{Z}^*_N$ and $\mathbb{Z}^*_{N^2}$.

Lemma 22 ([Pai99]). *The function $\varepsilon : \mathbb{Z}_N \times \mathbb{Z}^*_N \longrightarrow \mathbb{Z}^*_{N^2}$, defined as $\varepsilon(a, b) = (1+N)^a \cdot b^n$, is a bijective map.*

Now, we are ready to discuss the correctness and security of our DCR-based MCFE scheme (Fig. 3).

Correctness. We show that our construction in Fig. 3 is correct. Note that $(1 + N)$ is of order N and the following statement is satisfied.

$$\forall\ a \in \mathbb{Z}\ :\ (1+N)^a = (1+aN) \bmod N^2.$$

Now by the obtained value from decryption algorithm, we have:

$$C = \prod_i \mathsf{ct}_i^{y_i}.\mathcal{H}(\ell)^{-\mathsf{sk}} = (1+N)^{\Sigma_{i=1}^n x_i.y_i \mod N} \mod N^2 \Rightarrow$$

$$C = (1 + (\Sigma_{i=1}^n x_i \cdot y_i \mod N) \cdot N) \bmod N^2 \Rightarrow$$

$$\frac{C - 1 \bmod N^2}{N} = \Sigma_{i=1}^n x_i.y_i \mod N$$

Security Analysis. In this section we analysis the security of our construction. Our security proof is based on a combination of proof techniques of [CDG+18a] and [ALS16]. In the security proof we aim for adaptive security[9], we have a sequence of games such that any two adjacent games can be proved indistinguishable based on a computational assumption or a statistical argument.

Sketch of the proof. Here we give a simple but not accurate intuition of the proof. Many details are missing due to the sake of simplicity.

We start with the real game conditioned the hidden bit is $b = 0$ (game G_0). Then we try to replace the RO-queries with $\mathsf{RF}'(\ell)^N$ through the DCR assumption where $\mathsf{RF}'(\ell)$ is a truly random function (games G_1 and G_2). From this point we start a hybrid argument over RO-queries, while all the RO-queries are answered by $\mathsf{RF}'(\ell)^N$, the RO-query associated with the challenge would be replaced with $\mathsf{RF}(\ell)$ (Game $\mathsf{G}_{2.q.2}$) which tanks to the isomorphism ε can be seen as $(1+N)^a \cdot b^N$. These two changes are somehow orthogonal meaning that through $\mathcal{H}(\ell) = \mathsf{RF}'(\ell)^N$ we can simultaneously change the distribution of the master secret key $(\boldsymbol{s} + \lambda(\boldsymbol{x}_1 - \boldsymbol{x}_0) \cdot \mu$ for some $\mu \in \mathbb{Z})$ such that this change is indistinguishable in the adversary's view. Then thanks to the new distribution of the master key and the change $\mathcal{H}(\ell_q) = (1+N)^{a_q} \cdot b_q^N$, one can see that a new term as $\boldsymbol{x}_b + a_q\lambda \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_0) \mod N$ would be appeared in the challenge-ciphertext.

[9] I.e., $zz = \emptyset$.

From there we just need to say that this term can statistically hide the bit b in the challenge (by a statistical argument in $\mathsf{G}_{2.q.4}$).

We remark that our statistical argument (game $\mathsf{G}_{2.q.4}$) is the only game in the sequence which we cannot directly prove its indistinguishability from its previous game, in an adaptive setting [10]. Though, the positive side is that since this restriction is happening only in the statistical argument, by a technique similar to the complexity leveraging, one can find the proper parameters to lift the security again to the adaptive, without losing any factor of security. That is why in the construction we have considered two cases for the parameters-setting. This will help the user to set the parameters based on its chosen security model.

Game	description	justification
G_0	$\mathsf{ct}_i = (1+N)^{x_{i0}} \cdot \mathcal{H}(\ell)^{s_i} \mod N^2 \qquad \mathcal{H}(\ell) \in \mathbb{Z}^*_{N^2}$	real game $b=0$
G_1	$\mathsf{ct}_i = (1+N)^{x_{i0}} \cdot \mathsf{RF}(\ell)^{s_i} \qquad \boxed{\mathcal{H}(\ell) = \mathsf{RF}(\ell) \in \mathbb{Z}^*_{N^2}}$	RO
G_2	$\mathsf{ct}_i = (1+N)^{x_{i0}} \cdot \mathsf{RF}'(\ell)^{Ns_i} \qquad \boxed{\mathcal{H}(\ell) = \mathsf{RF}'(\ell)^N \in \mathbb{Z}^*_{N^2}}$	DCR
$\mathsf{G}_2.q.1$	$\mathsf{ct}_i = \begin{cases} (1+N)^{x_{i0}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell \geq \ell_q \\ (1+N)^{x_{i1}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell < \ell_q \end{cases} \qquad \mathcal{H}(\ell) = \mathsf{RF}'(\ell)^N$	$\mathsf{G}_{2.q,1} = \mathsf{G}_1$
$\mathsf{G}_2.q.2$	$\mathsf{ct}_i = \begin{cases} (1+N)^{x_{i0}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell > \ell_q \\ (1+N)^{x_{i0}} \cdot \mathsf{RF}(\ell)^{s_i} & \ell = \ell_q \\ (1+N)^{x_{i1}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell < \ell_q \end{cases} \quad \mathcal{H}(\ell) = \begin{cases} \mathsf{RF}'(\ell)^N & \ell \neq \ell_q \\ \boxed{\mathsf{RF}(\ell)} & \ell = \ell_q \end{cases}$	DCR
$\mathsf{G}_2.q.3$	$\mathsf{ct}_i = \begin{cases} (1+N)^{x_{i0}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell > \ell_q \\ (1+N)^{x_{i0}+a_\ell s_i} b_\ell^{Ns_i} & \ell = \ell_q \\ (1+N)^{x_{i1}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell < \ell_q \end{cases} \quad \mathcal{H}(\ell) = \begin{cases} \mathsf{RF}'(\ell)^N & \ell \neq \ell_q \\ \boxed{(1+N)^{a_\ell} b_\ell^N} & \ell = \ell_q \end{cases}$	Theorem 22
$\mathsf{G}_2.q.4$	$\mathsf{ct}_i = \begin{cases} (1+N)^{x_{i0}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell > \ell_q \\ (1+N)^{\boxed{x_{i1}}+a_\ell s_i} b_\ell^{Ns_i} & \ell = \ell_q \\ (1+N)^{x_{i1}} \cdot \mathsf{RF}'(\ell)^{Ns_i} & \ell < \ell_q \end{cases} \quad \mathcal{H}(\ell) = \begin{cases} \mathsf{RF}'(\ell)^N & \ell \neq \ell_q \\ (1+N)^{a_\ell} b_\ell^N & \ell = \ell_q \end{cases}$	stat.argu. $\mathsf{G}_{2.q.4} \cong \mathsf{G}_{2,q+1.1}$ backward steps
G_3	$\mathsf{ct}_i = (1+N)^{x_{i1}} \cdot \mathsf{RF}'(\ell)^{Ns_i} \qquad \mathcal{H}(\ell) = \mathsf{RF}'(\ell)^N$	$\mathsf{G}_3 = \mathsf{G}_{2.q_{\mathsf{Enc}}+1,1}$
G_4	$\mathsf{ct}_i = (1+N)^{x_{i1}} \cdot \mathsf{RF}(\ell)^{s_i} \qquad \boxed{\mathcal{H}(\ell) = \mathsf{RF}(\ell) \in \mathbb{Z}^*_{N^2}}$	DCR
G_5	$\mathsf{ct}_i = (1+N)^{x_{i1}} \cdot \mathcal{H}(\ell)^{s_i} \qquad \boxed{\mathcal{H}(\ell) \in \mathbb{Z}^*_{N^2}}$	RO real game $b=1$

Fig. 10: Overview of the games for MCFE based on the DCR assumption

Theorem 23. *The presented MCFE scheme in Fig. 3, is one-IND-MCFE secure under the DCR assumption and in the random-oracle model. More precisely:*

$$\mathsf{Avd}^{\text{one-IND}} \leq (2q_{\mathsf{Enc}} + 2) \cdot \mathsf{Avd}^{\mathsf{DCR}} + 4q_{\mathsf{Enc}} \cdot \mathrm{negl}_1(\kappa) + q_{\mathsf{Enc}} \cdot 2^{-\kappa}$$

where q_{Enc} is the number of random-oracle queries which are used in some LR *encryption queries,* negl_1 *shows the advantage of an adversary in distinguishing $\mathbb{Z}_N$ from $\mathbb{Z}^*_N$ and the term $2^{-\kappa}$ is appeared due to the fact that in our Gaussian distribution $\sigma > \sqrt{\kappa} \cdot N^{5/2}$.*

[10] More precisely, the security notion here is selective per label (lSEL) where the adversary is restricted not to issue LR-challenges on a new label as far as it has not completed the challenges associated with the label in the progress. While it may ask secret-key queries or corruption-queries adaptively. This security notion make sense specially in the time-stamp applications that one can not come back to the previous time-labels. This is, in a predefined time-stamp all the ciphertexts should be provided otherwise any ciphertext would be discarded before going to the next time-stamp.

Proof. We define a sequence of games started from G_0 which is the real game when the challenger answers to LR queries through the chosen bit $b = 0$ and ended with G_5 which is the real game for the bit $b = 1$.Thus,

$$\mathsf{Adv}^{\text{one-IND}}_{\mathsf{MCFE},\mathcal{A}}(\kappa, n) = |\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_5}_{\mathcal{A}}(\kappa, n)|.$$

This sequence of games is shown in Fig. 10. In this table RF, RF', RF_a, RF_b are different random functions respectively from labels set Labels to $\mathbb{Z}^*_{N^2}, \mathbb{Z}^*_{N^2}, \mathbb{Z}^*_N, \mathbb{Z}^*_N$.

Game G_0: is the real game where the challenger answers to the queries $\mathsf{QLeftRight}(x_0, x_1, i, \ell)$ by $\mathsf{Enc}(x_0, i, \ell)$. Note that hash function is modeled as random oracle $\mathcal{H}$ onto $\mathbb{Z}^*_{N^2}$.

Game G_1: is similar to the game G_0, except that, each new RO-query is answered by a fresh truly random in $\mathbb{Z}^*_{N^2}$. That is, $\mathcal{H}(\ell) = \mathsf{RF}(\ell)$. All other queries are simulated by running the real algorithms (based on these current RO values). Clearly,

$$|\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa, n)| = 0.$$

Game G_2: is similar to the game G_1, except that, each RO-query is answered by $\mathcal{H}(\ell) = \mathsf{RF}'(\ell)^N \bmod N^2$. Theorem 25 proves that,

$$|\mathsf{Win}^{\mathsf{G}_2}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa, n)| \leq \mathsf{Adv}^{\mathsf{DCR}}_{\mathcal{B}}(\kappa) + q_{\mathsf{Enc}} \cdot \mathrm{negl}_1 .$$

An adversary attacking to the random self-reducibility of DCR can simply simulate the game for the attacker to the indistinguishability of G_2 and G_1. The random self-reducibility of DCR expresses that from a single sample $w \leftarrow \mathcal{D}$ (similarly, $w \leftarrow \mathcal{D}'$) given by the DCR challenger, one can build many random samples w' from the same distribution $\mathcal{D}$ (respectively, $\mathcal{D}'$).

Game G_3 : is similar to the game G_2, except that, queries $\mathsf{QLeftRight}(x_0, x_1, i, \ell)$ are answered by $\mathsf{Enc}(x_1, i, \ell)$. In Theorem 26, we show that these two games are indistinguishable by a hybrid argument on RO-queries, yielding that,

$$|\mathsf{Win}^{\mathsf{G}_3}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_2}_{\mathcal{A}}(\kappa, n)| \leq q_{\mathsf{Enc}} \cdot (2 \cdot \mathsf{Adv}^{\mathsf{DCR}}_{\mathcal{B}}(\kappa) + 2 \cdot \mathrm{negl}_1(\kappa) + 2^{-\kappa}).$$

We prove this claim through a sequence of hybrids on the RO-queries. As we already mentioned in the proof-sketch, by the previous changes on the RO-queries, we are ready to start our statistical argument here. We will show that the master secret key in the adversary's view belongs to a sublattice. While in the challenge, the message is added to the master secret key module N. From there, we use a theorem from the lattice-based cryptography saying that if the variance of Gaussian distribution is enough larger than N, then sampling from this sublattice module N is close to uniform distribution over $\mathbb{Z}_N$, which then can hide bit b.

Games $\mathsf{G}_4, \mathsf{G}_5$: These games are respectively the counterparts of $\mathsf{G}_2, \mathsf{G}_1, \mathsf{G}_0$ and their indistinguishability relies on a similar reasoning in the backward steps.

□

Lemma 24 (Random self-reducibility of DCR [Pai99]). *The DCR assumption is random self reducible. Concretely, for any $k \in \mathbb{N}$,*

$$\mathsf{Adv}^{\mathsf{DCR},k}_{\mathcal{A}'}(\kappa) \leq \mathsf{Adv}^{\mathsf{DCR}}_{\mathcal{A}}(\kappa) + k \cdot \mathrm{negl}_1$$

where $\mathsf{Adv}^{\mathsf{DCR},k}_{\mathcal{A}'}(\kappa)$, is the advantage of adversary receiving k random samples of DCR. I.e.,

$$\mathsf{Adv}^{\mathsf{DCR},k}_{\mathcal{A}'}(\kappa) = |\Pr[\mathcal{A}'(w' \xleftarrow{R} \mathcal{D}) = 1] - \Pr[\mathcal{A}'(w' \xleftarrow{R} \mathcal{D}') = 1]|$$

where $w' = \{w'_i\}_{i=1}^k$ and $\mathcal{D} = \{z \xleftarrow{R} \mathbb{Z}^*_{N^2}\}$ and $\mathcal{D}' = \{z^N \bmod N^2 \ : \ z \xleftarrow{R} \mathbb{Z}^*_{N^2}\}$.

Proof. Let $\mathcal{A}$ be the attacker to the DCR assumption. It simulates the game for adversary $\mathcal{A}'$ as follows:

- $\mathcal{A}$ receives a sample w from its challenger.
- It samples $\alpha_i, \beta_i \xleftarrow{R} \mathbb{Z}_N$ for $i = 1, \ldots, k$.
- It sets $w'_i = w^{\alpha_i} \cdot \beta_i^N \mod N^2$ and sends back w'_i to $\mathcal{A}'$.
- $\mathcal{A}$ outputs the bit b' given by $\mathcal{A}'$.

To show this simulation is correct, one should prove that if $w \leftarrow \mathcal{D}$ (similarly, $w \leftarrow \mathcal{D}'$), then each w'_i is uniformly sampled from $\mathcal{D}$ (respectively, $\mathcal{D}'$). If $w \leftarrow \mathcal{D}$, then by Theorem 22, we can set $w = (1+N)^a b^N \mod N^2$. Thus, $w_i = (1+N)^{a\alpha_i} \cdot (b^{\alpha_i}\beta_i)^N \mod N^2$. Since $a \in \mathbb{Z}_N$ is invertible expect with negligible probability negl_1, $a\alpha_i \mod N$ is uniform over $\mathbb{Z}_N$. Similarly, $b^{\alpha_i}\beta_i \mod N$ is uniform over $\mathbb{Z}_N^*$ except with negligible probability negl_1 (because b^{α_i} is invertible in $\mathbb{Z}_N$). Then, again by isomorphism ε, the value $w'_i = (1+N)^{a\alpha_i} \cdot (b^{\alpha_i}\beta_i)^N \mod N^2$ is uniform over $\mathbb{Z}_{N^2}^*$.

if $w \leftarrow \mathcal{D}'$, then there exists $\iota \in \mathbb{Z}_{N^2}^*$ such that $w = \iota^N$. Thus, $w'_i = (\iota^{\alpha_i}\beta_i)^N \mod N^2$. Since ι is invertible over $\mathbb{Z}_{N^2}$, then $\iota^{\alpha_i}\beta_i$ is uniform over $\mathbb{Z}_{N^2}^*$ except with negligible probability negl_1. Consequently, w'_i is uniformly sampled from $\mathcal{D}'$. □

Lemma 25 (Transition from G_1 to G_2). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(\kappa, n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DCR}}(\kappa) + q_{\mathsf{Enc}} \cdot \mathrm{negl}_1 .$$

Proof. Assume that $\mathcal{B}$ is the attacker to the random-self-reducibility of DCR problem (Theorem 24) and $\mathcal{A}$ is the adversary trying to distinguish between games G_1 and G_2.

When the adversary $\mathcal{A}$ issues RO-queries, the adversary $\mathcal{B}$ simply returns $\mathsf{RF}(\ell) = w'_\ell$ where w'_ℓ is a sample from tis challenger. All other queries are answered by running the real algorithms.

If w'_ℓ for $\ell = 1, \ldots, q_{\mathsf{Enc}}$ is sampled from the distribution $\mathcal{D} = \{z \xleftarrow{R} \mathbb{Z}_{N^2}^*\}$, then $\mathcal{B}$ is simulating the G_1, and if w'_ℓ is sampled from the distribution $\mathcal{D}' = \{z^N \bmod N^2 \ : \ z \xleftarrow{R} \mathbb{Z}_{N^2}^*\}$, then $\mathcal{B}$ is simulating G_2. Thus, $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(\kappa, n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DCR}, q_{\mathsf{Enc}}}(\kappa)$. The upper-bound $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DCR}}(\kappa) + q_{\mathsf{Enc}} \cdot \mathrm{negl}_1$ is due to Theorem 24. □

Lemma 26 (Transition from G_2 to G_3). *Two mentioned games G_2 and G_3 in Theorem 23 are indistinguishable. More precisely,*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(\kappa, n)| \leq q_{\mathsf{Enc}} \cdot (2\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DCR}}(\kappa) + 2\,\mathrm{negl}_1(\kappa) + 2^{-\kappa}),$$

where q_{Enc} and negl_1 are as explained in Theorem 23.

Proof. For each $q = 1, \ldots, q_{\mathsf{Enc}}$, four games $\mathsf{G}_{2.q.1}$, to $\mathsf{G}_{2.q.4}$ are defined such that $\mathsf{G}_2 \cong \mathsf{G}_{2.1.1}$, and for any q, $\mathsf{G}_{2.q.1} \cong \mathsf{G}_{2.q.2} \cong \mathsf{G}_{2.q.3} \cong \mathsf{G}_{2.q.4}$ and $\mathsf{G}_{2.q.4} \cong \mathsf{G}_{2.q+1.1}$ where $\mathsf{G}_{2.q_{\mathsf{Enc}}+1.1} \cong \mathsf{G}_3$.

Game $\mathsf{G}_{2.q.1}$: without loss of generality, we consider a partial order relation for RO-queries. For the ciphertext associated with labels less than ℓ_q, the LR queries are answered by bit $b = 1$ while the other LR queries are answered by $b = 0$. Clearly, $\mathsf{G}_2 = \mathsf{G}_{2.1.1}$.

Game $\mathsf{G}_{2.q.2}$: is similar to the previous game, except that, qth RO-query associated with label ℓ_q is answered by $\mathsf{RF}(\ell_q)$. By the DCR assumption,

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.2}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.1}}(\kappa, n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DCR}}(\kappa).$$

Game $\mathsf{G}_{2.q.3}$: is similar to the previous game, except that, qth RO-query associated with label ℓ_q is answered by $(1+N)^{a_q} \cdot b_q^N$ where $a_q = \mathsf{RF}_a(\ell_q), b_q = \mathsf{RF}_b(\ell_q)$. By the isomorphism ε in Theorem 22, we have:

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.2}}(\kappa, n)| \leq \mathrm{negl}_1(\kappa).$$

The term $\mathsf{negl}_1(\kappa)$ is appeared due to the fact that instead of $a_q \xleftarrow{R} \mathbb{Z}_N$, as it is in Theorem 22, we have $a_q \xleftarrow{R} \mathbb{Z}_N^*$. More precisely, $\mathsf{negl}_1(\kappa) \leq \frac{1}{\sqrt{N}}$ which is the advantage of the adversary in distinguishing $\mathbb{Z}_N$ from $\mathbb{Z}_N^*$.

Game $\mathsf{G}_{2.q.4}$: is similar to the game $\mathsf{G}_{2.q.3}$, except that, the encryption queries $\mathsf{QLeftRight}(x_0, x_1, i, \ell_q)$ for label ℓ_q corresponding to the qth RO-query is answered by $\mathsf{Enc}(x_1, i, \ell_q)$. Note that $\mathsf{G}_{2.q.4} = \mathsf{G}_{2.q+1,1}$ and $\mathsf{G}_{2,q_{\mathsf{Enc}}+1,1} = \mathsf{G}_3$. In Theorem 27, we prove that

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa}.$$

□

Note that if qth RO-query is not used by any encryption query, then the games $\mathsf{G}_{2.q.4}$ and $\mathsf{G}_{2.q.3}$ are identical. But for the case that it is used by an encryption query, we claim that $\mathsf{G}_{2.q.4} \cong \mathsf{G}_{2.q.3}$. This step is similar to the security proof technique of single-input FE scheme based of Paillier in [ALS16]. The difference is that the information leaked through different ciphertext in our MCFE scheme are the same as what the adversary gets in single-input FE through the public key[11]. The formal proof is as follows:

Lemma 27 (Transition from $\mathsf{G}_{2.q.3}$ to $\mathsf{G}_{2.q.4}$). *If $\sigma > \sqrt{\kappa + 2n \cdot \log(2X)} \cdot N^{5/2}$ and $X < \sqrt{N/2n}$, then for any adversary $\mathcal{A}$,*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa}.$$

Proof. Here at first we prove that the selective[12] versions of these two games are indistinguishable and then by a technique similar to complexity leveraging we extend the security to their adaptive versions[13]. Let $\mathsf{G}^*_{2.q.3}$ and $\mathsf{G}^*_{2.q.4}$ be the selective versions of $\mathsf{G}_{2.q.3}$ and $\mathsf{G}_{2.q.4}$, respectively. We show that,

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa}.$$

We define a new game $\mathsf{G}^b_{2.q.3}$, depending on a random bit $b \xleftarrow{R} \{0,1\}$ such that when $b = 0$ it is the same as $\mathsf{G}^*_{2.q.3}$ and when $b = 1$ is the same as $\mathsf{G}^*_{2.q.4}$. Thus, in the game $\mathsf{G}^b_{2.q.3}$, we have $\mathsf{QLeftRight}(x_0, x_1, i, \ell_q) = (1+N)^{x^q_{ib} + a_q s_i} \cdot b_q^{N s_i} = \mathsf{ct}_{iq}$ (where x^q_{ib} is the i-th entry of the message $\boldsymbol{x}^q_b$ associated with the challenge ℓ_q) and all other queries are answered similar to the game $\mathsf{G}^*_{2.q.3}$. We claim that ct_{iq} for $i = 1, \ldots, n$, statically hides $b \in \{0,1\}$. To prove this, we try to show that conditioned on all the leaked information, $X \cdot (\boldsymbol{x}^q_b + a_q \boldsymbol{s}) \mod N$ can statistically hide bit b where X is an invertible matrix modulo N and independent of bit b. This can complete the proof.

Let $\boldsymbol{x}^q_\beta = (x^q_{1,\beta}, \ldots, x^q_{n,\beta})$, $\beta \in \{0,1\}$ are the challenges associated with label ℓ_q and $\boldsymbol{x}^q = \frac{1}{g}(\boldsymbol{x}^q_1 - \boldsymbol{x}^q_0)$ where $g = \gcd(x^q_{1,1} - x^q_{1,0}, \ldots, x^q_{n,1} - x^q_{n,0})$. Without loss of generality, we assume the n_0 first entries of $\boldsymbol{x}^q$ are zero, and all remaining entries are non-zero. The matrix X is considered as $X = \begin{bmatrix} X_{top} \\ X_{bot} \end{bmatrix}$ where X_{top} and X_{bot} are as follows:

$$X_{top} = \left(\begin{array}{c|ccccc} I_{n_0} & & & & & \\ \hline & -x^q_{n_0+2} & x^q_{n_0+1} & & & \\ & & -x^q_{n_0+3} & x^q_{n_0+2} & & \\ & & & \ddots & \ddots & \\ & & & & x^q_n & x^q_{n-1} \end{array}\right), \qquad X_{bot} = (\boldsymbol{x}^q)^T$$

[11] Note that in MCFE, we are in the symmetric key setting and the security game is involved with many ciphertexts queries

[12] In fact, we can prove that their variants for *selective per label* are indistinguishable.

[13] We emphasize that the standard complexity leveraging argument over a computational assumption reduces the strength of the computational argument, whereas here it's only leveraging on the statistical argument, so it's not as harmful.

For this matrix, $\det(XX^T) = (\prod_{i=n_0+1}^{n-1} (x_i^q)^2) \cdot ||\boldsymbol{x}^q||^4$. Each $(x_i^q)^2$ is small and non-zero. Thus, the term $(\prod_{i=n_0+1}^{n-1} (x_i^q)^2)$ is non-zero modulo N otherwise it gives a factorization for N. Similarly, we can assume that $\gcd(||\boldsymbol{x}^q||, N) = 1$, otherwise it gives a non-trivial factor of N. Putting together, $\det(X)^2 \neq 0 \mod N$ which means X is invertible over $\mathbb{Z}_N$. Coming back to the main goal, we show that $X \cdot (\boldsymbol{x}_b^q + a_q \boldsymbol{s}) \mod N$ hides the bit b. In fact, what we would show is that $X_{top} \cdot (\boldsymbol{x}_b^q + a_q \boldsymbol{s}) \mod N$ is completely independent of b and $X_{bot} \cdot (\boldsymbol{x}_b^q + a_q \boldsymbol{s}) \mod N$ is close to uniform and therefore statistically hides b.

– Step 1: $X_{top} \cdot (\boldsymbol{x}_b^q + a_q \boldsymbol{s}) \mod N$ is completely independent of b:
This is satisfied due to the fact that $X_{top} \cdot (\boldsymbol{x}_0^q - \boldsymbol{x}_1^q) = 0$ over integers (one can check it through the construction of matrix X_{top}).
– Step 2: $X_{bot} \cdot (\boldsymbol{x}_b^q + a_q \boldsymbol{s}) \mod N$ is close to uniform: Which can be written as

$$\langle \boldsymbol{x}^q, \boldsymbol{x}_b^q \rangle + a_q \langle \boldsymbol{x}^q, \boldsymbol{s} \rangle \mod N. \tag{1}$$

Let $\boldsymbol{s}_0 = (s_1^0 \ldots, s_n^0)$ be a possible value for the master key. Now we try to find the distribution of $\boldsymbol{s}$ from the adversary's view. The adversary can get information about the master secret key through:

1. All ciphertexts associated with $l \neq \ell_q$: the leaked information about $\boldsymbol{s}_0$ comes from $\mathsf{RF}'(\ell)^{N\boldsymbol{s}_0} \mod N^2,\ \ell \neq \ell_q$. Note that the adversary also knows $\mathsf{RF}'(\ell)^N$ through the RO queries.
2. Secret key queries: the leaked information is essentially $\langle \boldsymbol{y}, \boldsymbol{s}_0 \rangle$ for all the key quires $\boldsymbol{y}$.
3. Corruption queries: It leaks the value s_i^0 for the corrupted slot i.

Thus, the distribution of master key in the adversary's view is

$$\{\boldsymbol{s}_0 + \boldsymbol{t} \ :\ \boldsymbol{t} \xleftarrow{R} \mathcal{D}_{\Lambda,\sigma,-\boldsymbol{s}_0}\}$$

where the lattice Λ is as follows[14]:

$$\Lambda = \{\boldsymbol{t} \ :\ \boldsymbol{t} = \lambda \cdot (\boldsymbol{x}_1^q - \boldsymbol{x}_0^q) \cdot \mu,\ \mu \in \mathbb{Z}\}$$

And that is because for $\boldsymbol{s} = \boldsymbol{s}_0 + \boldsymbol{t}$,

$$RF'(\ell)^{N\boldsymbol{s}} = RF'(\ell)^{N\boldsymbol{s}_0} \mod N^2 \iff s = s_0 \mod \lambda, \text{ for } \ell \neq \ell_q$$
$$\langle \boldsymbol{y}, \boldsymbol{s} \rangle = \langle \boldsymbol{y}, \boldsymbol{s}_0 \rangle \text{ over } \mathbb{Z} \text{ for all secret key queries } \boldsymbol{y}$$
$$s_i = s_i^0 \text{ for the corrupted slot } i.$$

Note that the first equality is satisfied due to the fact that $\mathsf{RF}'(\ell)$ is a random function onto $\mathbb{Z}_{N^2}^*$ and $\mathsf{RF}'(\ell)^N$ is of order λ. We have also relied on the Condition$(*)$ of the security definition (Theorem 2), in two last equalities (and in the construction of vector $\boldsymbol{t}$ and matrix X as well). Due to the norm bounds, $\langle \boldsymbol{x}_1 - \boldsymbol{x}_0, \boldsymbol{y} \rangle = 0 \mod N$ means that $\langle \boldsymbol{x}_1 - \boldsymbol{x}_0, \boldsymbol{y} \rangle = 0$ over $\mathbb{Z}$ which is used in the second equality.
We write the lattice Λ as $\Lambda = \lambda \cdot \mathbb{Z} \cdot \boldsymbol{x}^q$. Conditioned on the leaked information, the distribution $\langle \boldsymbol{x}^q, \boldsymbol{s} \rangle$ is:

$$\langle \boldsymbol{s}_0, \boldsymbol{x}^q \rangle + \mathcal{D}_{\lambda \cdot ||\boldsymbol{x}^q||^2 \cdot \mathbb{Z}, ||\boldsymbol{x}^q||\sigma, -c}$$

where $c = \langle \boldsymbol{s}_0, \boldsymbol{x}^q \rangle \in \mathbb{Z}$

Agrawal et al. showed that if $\sigma > \sqrt{\kappa} \cdot N^{5/2}$, then $\mathcal{D}_{\lambda \cdot ||\boldsymbol{x}^q||^2 \cdot \mathbb{Z}, ||\boldsymbol{x}^q||\sigma, -c}$ over $\Lambda_0 = \lambda \cdot ||\boldsymbol{x}^q||^2 \cdot \mathbb{Z}$ modulo the lattice $\Lambda_0' = \lambda \cdot ||\boldsymbol{x}^q||^2 \cdot (N\mathbb{Z})$ is within statistical distance $2^{-\kappa}$ from the uniform distribution over $\frac{\Lambda_0}{\Lambda_0'}$ [GPV08]. And since $\gcd(\lambda \cdot ||\boldsymbol{x}^q||^2, N) = 1$, then $\frac{\Lambda_0}{\Lambda_0'}$ is isomorphic to $\mathbb{Z}_N$. This means that $\langle \boldsymbol{x}^q, \boldsymbol{s} \rangle$ modulo N is within statistical distance $2^{-\kappa}$ from the uniform distribution over

[14] One may tend to consider the lattice Λ as a linear combination of (linearly independent) LR encryption queries. But it essentially leaks the same information as what we have already considered

$\mathbb{Z}_N$. Now by the Eq. (1), since $a_q \in \mathbb{Z}_N^*$ is invertible modulo N, the term $\langle \boldsymbol{x}^q, \boldsymbol{x}_b^q \rangle$ is statistically hidden. This completes the proof for $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa}$.

Then by applying a technique similar to complexity leveraging, we have:

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n) = (2X)^{2n} \cdot \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.3}}(\kappa, n), \quad \text{and} \quad \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.4}}(\kappa, n) = (2X)^{2n} \cdot \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.4}}(\kappa, n)$$

Thus,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n) = (2X)^{2n} \cdot (\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.3}}(\kappa, n))$$

Meaning that if $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa} \cdot (2X)^{-2n}$ then, $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa}$. Clearly, if in the last part of the proof one sets $\sigma > \sqrt{\kappa + 2n \cdot \log(2X)} \cdot N^{5/2}$ then the above reasoning result in $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.4}}(\kappa, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.q.3}}(\kappa, n)| \leq 2^{-\kappa}$ which proves the adaptive security. □

Our DCR-based MCFE scheme can simply be extended to pos$^+$-IND secure, IND secure or to the decentralized version through some existing general compilers. These extensions are given in Appendix F.1.

C MCFE based on the LWE Assumption

C.1 A Review on Single-Input FE based on LWE

Agrawal et al. [ALS16] proposed a single-input FE based on the LWE problem which is shown in Fig. 11. In this construction $\mathcal{P} = \{0, \ldots, P-1\}^n$ and $\mathcal{V} = \{0, \ldots, V-1\}^n$ are respectively the message and the key space associated with secret keys, for integers P, V. The inner product between message and key vectors belongs to $\{0, \ldots, K-1\}$ with $K = nPV$ and the prime modulus q is significantly larger than K. With this construction, inner-product is evaluated over $\mathbb{Z}$ and the decryption algorithm is completely efficient. Their single-input FE scheme is secure under a new hardness assumption named mheLWE. They also proved a reduction from LWE to mheLWE. Their construction is reminded through Fig. 11.

$\underline{\mathsf{Setup}(1^{n_0}, n)}$:
– set integers $m_0, q \geq 2$
$K = nPV$ and $\alpha \in (0, 1)$.
– sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{m_0 \times n_0}$
Return
$\mathsf{pp} = (\mathbf{A}, m_0, q, \alpha, K, P, V)$

$\underline{\mathsf{KeyGen}(\mathsf{pp})}$:
–sample $\mathbf{Z} \xleftarrow{R} \tau$ where
τ is a special distribution
over $\mathbb{Z}^{n \times m_0}$.
compute $U = \mathbf{Z} \cdot \mathbf{A}$.
Return $\mathsf{mpk} = U$ and $\mathsf{msk} = \mathbf{Z}$.

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{mpk}, \boldsymbol{x})}$:
To encrypt the vector $\boldsymbol{x} \in \mathcal{P}$:
– sample $\boldsymbol{s} \xleftarrow{R} \mathbb{Z}_q^{n_0}$, $\boldsymbol{e}_0 \xleftarrow{R} \mathcal{D}_{\mathbb{Z}, \alpha q}^{m_0}$ and $\boldsymbol{e}_1 \xleftarrow{R} \mathcal{D}_{\mathbb{Z}, \alpha q}^{n}$.
– compute $\begin{cases} \mathsf{ct}_0 = \mathbf{A}^T \cdot \boldsymbol{s} + \boldsymbol{e}_0 \in \mathbb{Z}_q^{m_0}, \\ \mathsf{ct}_1 = U \cdot s + e_1 + \left\lfloor \frac{q}{K} \right\rfloor \cdot \boldsymbol{x} \in \mathbb{Z}_q^n \end{cases}$
Return $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$

$\underline{\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \boldsymbol{y})}$:
To generate a secret key for the vector $\boldsymbol{y} \in \mathcal{V}$:
– compute $\mathsf{sk}_y = \boldsymbol{y}^T \cdot \mathbf{Z} \in \mathbb{Z}^{m_0}$
Return sk_y

$\underline{\mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \boldsymbol{y}, \mathsf{sk}, \mathsf{ct})}$:
– compute $\mu' = \langle \boldsymbol{y}, \mathsf{ct}_1 \rangle - \langle \mathsf{sk}, \mathsf{ct}_0 \rangle \mod q$
Return $\mu \in \{-K+1, \ldots, K-1\}$ that
minimizes $|\lfloor \frac{q}{K} \rfloor \cdot \mu - \mu'|$.

Fig. 11: Single-input FE based on LWE assumption [ALS16]

C.2 Correctness and Security of our LWE-based MCFE

Our security proof is using the following lemma which is applied in the security-reduction from LWE problem to LWR problem in [BPR12].

Lemma 28 (Extracted from Theorem 3.2 [BPR12]). *If $\mathcal{X}$ is a B-bounded distribution and $q \geq q_0 \cdot B \cdot {n_0}^{\omega(1)}$, then for any distribution over a fixed vector $\boldsymbol{s} \in \mathbb{Z}_q^{n_0}$, the statistical difference between two distributions $\{(\boldsymbol{a}, \lfloor \langle \boldsymbol{a}, \boldsymbol{s} \rangle \rceil_{q_0}) \ : \ \boldsymbol{a} \leftarrow \mathbb{Z}_q^{n_0}\}$ and $\{(\boldsymbol{a}, \lfloor \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e \rceil_{q_0}) \ : \ \boldsymbol{a} \leftarrow \mathbb{Z}_q^{n_0}, \ e \leftarrow \mathcal{X}\}$ is ${n_0}^{-\omega(1)}$.*

The above lemma shows that if the modulus q is chosen super-polynomially big, then the noise term in the LWE problem can be absorbed in the rounding.

Here we discuss the correctness and security of our LWE-based MCFE scheme.

Decryption Correctness. To show the correctness of the scheme, we first define $e_i = \mathsf{ct}_{i,\ell} - \frac{q_0}{q}(\mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot x_i)$ and $e_0 = \lfloor \mathsf{sk} \cdot \mathcal{H}(\ell) \rceil_{q_0} - \frac{q_0}{q} \mathsf{sk} \cdot \mathcal{H}(\ell)$, thus:

$$
\begin{aligned}
\mu' &= \sum_{i \in [n]} y_i \cdot \mathsf{ct}_{i,\ell} - \lfloor \mathsf{sk} \cdot \mathcal{H}(\ell) \rceil_{q_0} \\
&= \sum_{i \in [n]} y_i \left(\frac{q_0}{q} \left(\mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot x_i \right) + e_i) \right) - \left(\frac{q_0}{q} \mathsf{sk} \cdot \mathcal{H}(\ell) + e_0 \right) \\
&= \sum_{i \in [n]} \frac{q_0}{q} \left(y_i \mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot y_i x_i \right) + y_i e_i - \frac{q_0}{q} \mathcal{H}(\ell) \sum_{i \in [n]} y_i \cdot \mathbf{Z}_i - e_0 \\
&= \frac{q_0}{q} \lfloor \frac{q}{K} \rfloor \sum_{i \in [n]} x_i y_i + \sum_{i \in [n]} y_i e_i + e_0
\end{aligned}
$$

To guarantee the correctness, the relation $|\sum y_i e_i + e_0| < \lfloor \frac{q_0}{2K} \rfloor$ should be satisfied. Since, $|e_i| \leq \frac{1}{2}$ and $|e_0| \leq \frac{1}{2}$, $|\sum y_i e_i + e_0| \leq \frac{1}{2}(\sum y_i + 1) \leq \frac{1}{2}(nV + 1)$. Meaning that if $q_0 > K(nV + 1)$, then the scheme is correct.

Security Analysis. To simplify the proof and without loss of generality, we consider the case where $m = 1$, meaning that the input of each client is a scaler rather than a vector.

Theorem 29. *The presented MCFE scheme in Fig. 4, is an one-IND-secure MCFE scheme under the LWE assumption and in the random-oracle model. More precisely:*

$$
\mathsf{Avd}^{\text{one-IND}} \leq q_{\mathsf{Enc}} \cdot \left(6 \, \mathrm{negl}_1(n_0) + 2\mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0) + 2^{-\kappa} \right) + 2\mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0)
$$

where q_{Enc} is the number of random-oracle queries, and the term $2^{-\kappa}$ is appeared due to the fact that in our Gaussian distribution parameters depend on κ. The term negl_1 comes from the advantage of an adversary in Theorem 28.

Proof. We define a sequence of the games started from G_0, which is the real game when the challenger answers to LR queries through the chosen bit $b = 0$, and ended with G_5, which is the real game corresponding with the bit $b = 1$. Thus,

$$
\mathsf{Adv}^{\text{one-IND}}_{\mathsf{MCFE},\mathcal{A}}(n_0, n) = |\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(n_0, n) - \mathsf{Win}^{\mathsf{G}_5}_{\mathcal{A}}(n_0, n)|.
$$

This sequence of games is shown in Figs. 12 and 13.

Game G_0: is the real game where the challenger answer to $\mathsf{QLeftRight}(x_0, x_1, i, \ell)$ by $\mathsf{Enc}(x_0, i, \ell)$.

Note that hash function is modeled as random oracle RO onto $\mathbb{Z}_q^{n_0+m_0}$.

Game	Description
G_0	$\mathsf{ct}_{i,\ell} = \left\lfloor \mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot x_i^0 \right\rceil$
justification	LWE assumption
G_1	$\begin{aligned}\mathsf{ct}_{i,\ell} &= \left\lfloor \mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil \\ &= \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot \boldsymbol{e}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil\end{aligned}$ $\boxed{\mathcal{H}(\ell) = \begin{pmatrix} \boldsymbol{a}_\ell \\ \mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell \end{pmatrix}}$
justification	$\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell$ absorbed by the rounding, requires $q = q_0 B n_0{}^{\omega(1)}$, where $\lvert \boldsymbol{t}_i \cdot \boldsymbol{e}_\ell \rvert \leq B$
G_2	$\mathsf{ct}_{i,\ell} = \left\lfloor \boxed{(\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell} + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil$ $\mathcal{H}(\ell) = \binom{\boldsymbol{a}_\ell}{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell}$
justification	$\boldsymbol{t}_i \cdot \boldsymbol{e}_{\ell_\gamma}$ absorbed by the rounding
$\mathsf{G}_{2.\gamma.1}$	$\mathsf{ct}_{i,\ell} = \begin{cases} \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell > \ell_\gamma \\ \left\lfloor \boxed{\boldsymbol{s}_i \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot (\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell)} + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell = \ell_\gamma \\ \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell < \ell_\gamma \end{cases}$ $\mathcal{H}(\ell) = \binom{\boldsymbol{a}_\ell}{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell}$
justification	LWE assumption
$\mathsf{G}_{2.\gamma.2}$	$\mathsf{ct}_{i,\ell} = \begin{cases} \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell > \ell_\gamma \\ \left\lfloor (\boldsymbol{s}_i \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \left(\boxed{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell + \boldsymbol{u}} \right) + \lfloor \frac{q}{K} \rfloor \boldsymbol{x}_i^0 \right\rceil & \ell = \ell_\gamma \\ \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell < \ell_\gamma \end{cases}$ $\mathcal{H}(\ell) = \begin{cases} \binom{\boldsymbol{a}_\ell}{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell} & \ell \neq \ell_\gamma \\ \binom{\boldsymbol{a}_\ell}{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell + \boxed{\boldsymbol{u}}} & \ell = \ell_\gamma \end{cases}$ where $\boldsymbol{u} = \mathsf{RF}(\ell_\gamma)$
justification	$\boldsymbol{t}_i \cdot \boldsymbol{e}_{\ell_\gamma}$ absorbed by the rounding
$\mathsf{G}_{2.\gamma.3}$	$\mathsf{ct}_{i,\ell} = \begin{cases} \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell > \ell_\gamma \\ \left\lfloor \boxed{(\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot \boldsymbol{u}} + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell = \ell_\gamma \\ \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell < \ell_\gamma \end{cases}$ $\mathcal{H}(\ell) = \begin{cases} \binom{\boldsymbol{a}_\ell}{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell} & \ell \neq \ell_\gamma \\ \binom{\boldsymbol{a}_\ell}{\mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell + \boxed{\boldsymbol{u}}} & \ell = \ell_\gamma \end{cases}$ where $\boldsymbol{u} = \mathsf{RF}(\ell_\gamma)$
justification	Rewriting, same view generated by sampling $\boldsymbol{s}_i'$ instead of $\boldsymbol{s}_i$

Fig. 12: Overview of the games our MCFE scheme based on LWE.
Here $\lfloor . \rceil$ stands for $\lfloor . \rceil_{q_0}$ and $(\boldsymbol{a}_\ell, \mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell) \in \mathbb{Z}_q^{n_0} \times \mathbb{Z}_q$ are LWE samples.

Game	Description
$\mathsf{G}_{2.\gamma.4}$	$\mathsf{ct}_{i,\ell} = \begin{cases} \left\lfloor \boxed{\boldsymbol{s}'_i} \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell > \ell_\gamma \\ \left\lfloor \boxed{\boldsymbol{s}'_i} \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot \boldsymbol{u} + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell = \ell_\gamma \\ \left\lfloor \boxed{\boldsymbol{s}'_i} \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell < \ell_\gamma \end{cases}$ $\mathcal{H}(\ell) = \begin{cases} \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell} & \ell \neq \ell_\gamma \\ \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell+\boxed{\boldsymbol{u}}} & \ell = \ell_\gamma \end{cases}$ where $\boldsymbol{u} = \mathsf{RF}(\ell_\gamma)$
justification	statistical argument, requires $\sigma \geq 10nV$ and $m = \Omega(\log(q))$
$\mathsf{G}_{2.\gamma.5}$	$\mathsf{ct}_{i,\ell} = \begin{cases} \left\lfloor \boldsymbol{s}'_i \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell > \ell_\gamma \\ \left\lfloor \boldsymbol{s}'_i \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot \boldsymbol{u} + \lfloor \frac{q}{K} \rfloor \cdot \boxed{\boldsymbol{x}_i^1} \right\rceil & \ell = \ell_\gamma \\ \left\lfloor \boldsymbol{s}'_i \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell < \ell_\gamma \end{cases}$ $\mathcal{H}(\ell) = \begin{cases} \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell} & \ell \neq \ell_\gamma \\ \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell+\boxed{\boldsymbol{u}}} & \ell = \ell_\gamma \end{cases}$ where $\boldsymbol{u} = \mathsf{RF}(\ell_\gamma)$
justification	Same steps backwards
$\mathsf{G}_{2.\gamma.6}$	$\mathsf{ct}_{i,\ell} = \begin{cases} \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^0 \right\rceil & \ell > \ell_\gamma \\ \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell = \ell_\gamma \\ \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil & \ell < \ell_\gamma \end{cases}$ $\mathcal{H}(\ell) = \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell}$
justification	the next game is $\mathsf{G}_{2.\gamma+1.1}$ and $\mathsf{G}_{2.q_{\mathsf{Enc}}.5} = \mathsf{G}_3$
G_3	$\mathsf{ct}_{i,\ell} = \left\lfloor (\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S}) \cdot \boldsymbol{a}_\ell + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil$ $\mathcal{H}(\ell) = \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell}$
justification	$\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell$ absorbed by the rounding
G_4	$\mathsf{ct}_{i,\ell} = \left\lfloor \left(\boldsymbol{s}_i + \boldsymbol{t}_i \cdot \mathbf{S} \cdot \boldsymbol{a}_\ell + \boxed{\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell} + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right) \right\rceil = \left\lfloor \mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil$ $\mathcal{H}(\ell) = \binom{\boldsymbol{a}_\ell}{\mathbf{S}\cdot\boldsymbol{a}_\ell+\boldsymbol{e}_\ell}$
justification	LWE assumption
G_5	$ct_{i,\ell} = \left\lfloor \mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i^1 \right\rceil$

Fig. 13: Overview of the games our MCFE scheme based on LWE.
Here $\lfloor . \rceil$ stands for $\lfloor . \rceil_{q_0}$ and $(\boldsymbol{a}_\ell, \mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell) \in \mathbb{Z}_q^{n_0} \times \mathbb{Z}_q$ are LWE samples.

Game G_1: is similar to the game G_0, except that, each new RO-query is answered by a fresh sample of $\mathsf{LWE}_{q,\alpha}$. Thus:

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(n_0,n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_0}(n_0,n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0).$$

We note that since the LWE assumption is already involved with polynomially many samples, the upper-bound does not depend to the number of queries. For the indistinguishability of G_0 and G_1, we consider an extension of LWE problem which is as hard as the original definition. This extension considers samples with the same given coefficients but different secrets which would let to have a matrix as the secret.

Game G_2: is similar to the game G_1, except that, the value $\boldsymbol{t}_i\boldsymbol{e}_\ell$ is absorbed in the rounding. If $q \geq q_0 B n_0^{\omega(1)}$ where $|\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell| \leq B$ with overwhelming probability, then games G_1 and G_2 are indistinguishable. The proof of indistinguishability is similar to the proof of Theorem 28. Giving that:

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(n_0,n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(n_0,n)| \leq q_{\mathsf{Enc}} \cdot \mathrm{negl}_1(n_0),$$

where negl_1 is the probability of distinguishing $\boldsymbol{t}_i \cdot e_\ell$ from $\mathbf{0}$ after applying the rounding map $\lfloor \cdot \rceil_{q_0}$. This change would let us remove the value $\boldsymbol{t}_i \cdot \ddots_\ell$ from all the encryption-queries such that this change is indistinguishable for the adversary.

Game G_3: is similar to the game G_2, except that, the encryption queries $\mathsf{QLeftRight}(x_0, x_1, i, \ell)$ are answered by $\mathsf{Enc}(x_1, i, \ell)$. In Theorem 31, we show that these two games are indistinguishable by a hybrid argument on RO-queries. And:

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(n_0,n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(n_0,n)| \leq 2q_{\mathsf{Enc}} \cdot (2\,\mathrm{negl}_1(n_0) + \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0)) + q_{\mathsf{Enc}} \cdot 2^{-\kappa}.$$

Intuitively, by the current change in the RO-queries, we show that one can simultaneously change the distribution of the master secret key $\boldsymbol{t}_i$ as $\boldsymbol{t}_i + \mu(\boldsymbol{x}_1 - \boldsymbol{x}_0)$ for some $\mu \in \mathbb{Z}$ such that this change is indistinguishable for the adversary. Then, we show that if we change the vector $\mathcal{H}(\ell_q)$ associated with the challenge to a random vector $\boldsymbol{u}$, the multiplication of the new master key and $\boldsymbol{u}$ can statistically hide the message in the challenge.

Games $\mathsf{G}_4, \mathsf{G}_5$: Now from here we come back in reverse, in a similar way from G_2 to the game G_0. The last game G_5 is similar to the real game with $b = 1$.

□

Lemma 30 (Transition from G_0 to G_1). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_1}(n_0,n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_0}(n_0,n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0).$$

Proof. At first we note that clearly LWE problem with samples $(\boldsymbol{a}_\ell, \boldsymbol{a}_\ell \cdot \boldsymbol{s}_i + e_{i,\ell})$ for $i = 1, \ldots, m_0$ (i.e., when each equation crosses through m_0 different secrets $s_1, \ldots, s_{m_0}$) is as hard as the original LWE problem. Now, for samples $(\boldsymbol{a}_\ell, b_{i,\ell})$ from its challenger, the adversary $\mathcal{B}$ sends $(\boldsymbol{a}_\ell, \boldsymbol{b}'_\ell)$ to $\mathcal{A}$ where i-th entry of $\boldsymbol{b}'_\ell$ equals $b_{i,\ell}$. If $b_{i,\ell}$ is $\boldsymbol{a}_\ell \cdot \boldsymbol{s}_i + e_{i,\ell}$, then $\boldsymbol{b}'_\ell = \mathbf{S} \cdot \boldsymbol{a}_\ell + \boldsymbol{e}_\ell$ where the i-th row of $\mathbf{S}$ is as $\boldsymbol{s}_i$, which in this case it simulate the game G_1. If each $b_{i,\ell}$ is chosen uniformly, then $\boldsymbol{b}'_\ell$ is uniform, which simulate the game G_0. □

Lemma 31 (Transition from G_2 to G_3). *two mentioned games G_2 and G_3 in Theorem 23 are indistinguishable. More precisely,*

$$|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_3}(n_0,n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_2}(n_0,n)| \leq 2q_{\mathsf{Enc}} \cdot (2\,\mathrm{negl}_1(n_0) + \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0)) + q_{\mathsf{Enc}} \cdot 2^{-\kappa}.$$

Proof. For each $\gamma = 1, \ldots, q_{\mathsf{Enc}}$, six games $\mathsf{G}_{2.\gamma.1}, \ldots, \mathsf{G}_{2.\gamma.6}$ are defined such that $\mathsf{G}_2 \cong \mathsf{G}_{2.1.1}$, and for any γ, $\mathsf{G}_{2.\gamma.1} \cong \mathsf{G}_{2.\gamma.2} \cong \mathsf{G}_{2.\gamma.3} \cong \mathsf{G}_{2.\gamma.4} \cong \mathsf{G}_{2.\gamma.5} \cong \mathsf{G}_{2.\gamma.6}$ and $\mathsf{G}_{2.\gamma.6} \cong \mathsf{G}_{2.\gamma+1.1}$ where $\mathsf{G}_{2.q_{\mathsf{Enc}}.6} \cong \mathsf{G}_3$.

Game $\mathsf{G}_{2.\gamma.1}$: is similar to its previous game, except that, for the label ℓ_γ, the term $\boldsymbol{t}_i \cdot \boldsymbol{e}_{\ell_\gamma}$ is added to the ciphertext. Again the proof of the indistinguishability is similar to the proof of Theorem 28. Thus,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.1}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\overline{\mathsf{G}}_{2.\gamma.1}}(n_0, n)| \leq \mathrm{negl}_1(n_0)$$

where $\overline{\mathsf{G}}_{2.\gamma.1}$ is the game before $\mathsf{G}_{2.\gamma.1}$ (which might be G_2 or $\mathsf{G}_{2.\gamma-1,6}$). The intuition for this change is to add a random value beside the message which can statistically hide the message in the challenge.

Game $\mathsf{G}_{2.\gamma.2}$: is similar to the previous game, except that, RO-query for label ℓ_γ is replaced by $\mathbf{S} \cdot \boldsymbol{a}_{\ell_\gamma} + \boldsymbol{e}_{\ell_\gamma} + \boldsymbol{u}_\ell$. Similar to the transition from game G_0 to G_1, one should consider LWE problem with samples $(\boldsymbol{a}_{\ell_\gamma}, \boldsymbol{a}_{\ell_\gamma} \cdot \boldsymbol{s}'_i + e_{i,\ell_\gamma})$ for $i = 1, \ldots, m_0$. Lemma Theorem 32 formally proves the computational indistinguishability $\mathsf{G}_{2.\gamma.2}$ from its previous game i.e.,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.2}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.1}}(n_0, n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0)$$

Intuitively, this change will let us to remove $\boldsymbol{t}_i \cdot \mathbf{S}$ from the ciphertexts by moving $\boldsymbol{t}_i \cdot \mathbf{S}$ beside $\boldsymbol{s}_i$ where $\boldsymbol{s}_i$ is uniform and can hide $\boldsymbol{t}_i \cdot \mathbf{S}$.

Game $\mathsf{G}_{2.\gamma.3}$: is similar to the previous game, except that, for the label ℓ_γ, the term $\boldsymbol{t}_i \cdot \boldsymbol{e}_{\ell_\gamma}$ is removed from the ciphertext. Again the proof of the indistinguishability is similar to the proof of Theorem 28. Thus,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.3}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.2}}(n_0, n)| \leq \mathrm{negl}_1(n_0)$$

Game $\mathsf{G}_{2.\gamma.4}$: is similar to the game $\mathsf{G}_{2.\gamma.3}$, except that, in the master secret key generation (and thus in all ciphertexts), $\boldsymbol{s}_i$ is computed as $\boldsymbol{s}'_i - \boldsymbol{t}_i \cdot \mathbf{S}$ where we sampled a fresh random $\boldsymbol{s}'_i$. Clearly, this two games are identical, since $\boldsymbol{s}_i$ is uniformly random. I.e.,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.3}}(n_0, n)| = 0$$

Game $\mathsf{G}_{2.\gamma.5}$: is similar to the previous game, except that, the query $\mathsf{QLeftRight}(x_0, x_1, i, \ell_\gamma)$, associated with label ℓ_γ and corresponding to γth RO-query, is answered by $\mathsf{Enc}(x_i^1, \ell_\gamma)$. In Theorem 33 we show:

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.5}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n)| \leq 2^{-\kappa}$$

□

Lemma 32 (Transition from $\mathsf{G}_{2.\gamma.1}$ to $\mathsf{G}_{2.\gamma.2}$). *If the LWE assumption holds, then two games $\mathsf{G}_{2.\gamma.1}$ and $\mathsf{G}_{2.\gamma.2}$ are indistinguishable and:*

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.2}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.1}}(n_0, n)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(n_0)$$

Proof. The adversary $\mathcal{B}$ receives the samples $(\boldsymbol{a}_{\ell_\gamma}, b_{i,\ell_\gamma})$ from its LWE-challenger. It sends the vector $(\boldsymbol{a}_{\ell_\gamma}, b_{\ell_\gamma})$ to the adversary $\mathcal{A}$. If $b_{i,\ell_\gamma} = \boldsymbol{a}_{\ell_\gamma} \cdot \boldsymbol{s}_i + e_{i,\ell_\gamma}$, then it simulates the game $\mathsf{G}_{2.\gamma.1}$ and if b_{i,ℓ_γ} is uniform, it simulate the game $\mathsf{G}_{2.\gamma.2}$ (since $\boldsymbol{u}_{\ell_\gamma}$ is indistinguishable from $\mathbf{S} \cdot \boldsymbol{a}_{\ell_\gamma} + \boldsymbol{b}_{\ell_\gamma} + \boldsymbol{u}_{\ell_\gamma}$ for any uniform vector $\boldsymbol{u}_{\ell_\gamma}$). □

Note that if γth RO-query is not used by any encryption query, then the games $\mathsf{G}_{2.\gamma.4}$ and $\mathsf{G}_{2.\gamma.5}$ are identical. But for the case that it is used by an encryption query, we claim they are indistinguishable. This step is similar to the security proof technique of single-input FE scheme based of LWE in [ALS16][15]. The formal proof is as follows:

[15] Note that in MCFE, we are in the symmetric key setting and the security game is involved with many ciphertexts queries

Lemma 33 (Transition from $\mathsf{G}_{2.\gamma.3}$ to $\mathsf{G}_{2.\gamma.4}$). *If $\Omega(\log q + 4n \log P) \leq m_0$ and $nP^2 < q$ then:*

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.5}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n)| \leq 2^{-\kappa}$$

Proof. Here at first we prove that the selective[16] versions of these two games are indistinguishable and then by a technique similar to the complexity leveraging we lift the security to their adaptive versions. Let $\mathsf{G}^*_{2.\gamma.4}$ and $\mathsf{G}^*_{2.\gamma.5}$ be the selective versions of $\mathsf{G}_{2.\gamma.4}$ and $\mathsf{G}_{2.\gamma.5}$, respectively.

We show that

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.5}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.4}}(n_0, n)| \leq 2^{-\kappa}$$

We define a new game $\mathsf{G}^b_{2.\gamma}$, depending on a random bit $b \xleftarrow{R} \{0,1\}$ such that when $b = 0$ it is the same as $\mathsf{G}^*_{2.\gamma.4}$ and when $b = 1$ is the same as $\mathsf{G}^*_{2.\gamma.5}$. Thus, in the game $\mathsf{G}^b_{2.\gamma}$, we have $\mathsf{QLeftRight}(x_0, x_1, i, \ell_\gamma) = \boldsymbol{s}'_i \cdot \boldsymbol{a}_\ell + \boldsymbol{t}_i \cdot \boldsymbol{u} + \left[\frac{q}{K}\right] \boldsymbol{x}^\gamma_{ib} = \mathsf{ct}_{i\gamma}$ (note that $\boldsymbol{u} = \mathsf{RF}(\ell_\gamma)$ and x^γ_{ib} is the i-th entry of the message $\boldsymbol{x}^\gamma_b$ associated with the challenge ℓ_γ) and all other queries are answered similar to the game $\mathsf{G}^*_{2.\gamma.3}$. We claim that $\mathsf{ct}_{i\gamma}$ for $i = 1, \ldots, n$, statistically hides $b \in \{0,1\}$. To prove this, we try to show that conditioned on all the leaked information, $X \cdot \mathbf{T}^\gamma_b$ can statistically hide bit b where X is an invertible matrix module q and independent of bit b and $\mathbf{T}^\gamma_b$ is as follows:

$$\mathbf{T}^\gamma_b = \begin{pmatrix} \boldsymbol{t}^\gamma_{1b} \\ \vdots \\ \boldsymbol{t}^\gamma_{nb} \end{pmatrix} \quad , \quad \boldsymbol{t}^\gamma_{ib} = \boldsymbol{t}_i \cdot \boldsymbol{u} + \left[\frac{q}{K}\right] \boldsymbol{x}^\gamma_{ib} \tag{2}$$

This can complete the proof. Let $\boldsymbol{x}^\gamma_\beta = (x^\gamma_{1,\beta}, \ldots, x^\gamma_{n,\beta}),\ \beta \in \{0,1\}$ are the challenges associated with label ℓ_γ and $\boldsymbol{x}^\gamma = \frac{1}{g}(\boldsymbol{x}^\gamma_1 - \boldsymbol{x}^\gamma_0)$ where $g = \gcd(x^\gamma_{1,1} - x^\gamma_{1,0}, \ldots, x^\gamma_{n,1} - x^\gamma_{n,0})$. Without loss of generality, we assume the l first entries of $\boldsymbol{x}^\gamma$ are zero, and all remaining entries are non-zero. The matrix X is considered as $X = \begin{bmatrix} X_{top} \\ X_{bot} \end{bmatrix}$ where X_{top} and X_{bot} are as follows:

$$X_{top} = \left(\begin{array}{c|cccccc} I_l & & & & & & \\ \hline & -x^\gamma_{l+2} & x^\gamma_{l+1} & & & & \\ & & -x^\gamma_{l+3} & x^\gamma_{l+2} & & & \\ & & & \ddots & \ddots & & \\ & & & & & x^\gamma_n & x^\gamma_{n-1} \end{array}\right), \qquad X_{bot} = (\boldsymbol{x}^\gamma)^T$$

For this matrix, $\det(XX^T) = (\prod_{i=l+1}^{n-1} (x^\gamma_i)^2) \cdot ||\boldsymbol{x}^\gamma||^4$. Each $(x^\gamma_i)^2$ is small and non-zero (meaning that for all i, $\gcd((x^\gamma_i)^2, q) = 1$). Thus, the term $(\prod_{i=l+1}^{n-1} (x^\gamma_i)^2)$ is non-zero modulo q. On the other hand, $\gcd(||\boldsymbol{x}^\gamma||, q) = 1$ due to the fact that $nP^2 < q$. Putting together, $\det(X)^2 \neq 0 \mod q$ which means X is invertible over $\mathbb{Z}_q$. Coming back to the main goal, we show that $X \cdot \mathbf{T}^\gamma_b \mod q$ hides the bit b. In fact, what we would show is that $X_{top} \cdot \mathbf{T}^\gamma_b \mod q$ is completely independent of b and $X_{bot} \cdot \mathbf{T}^\gamma_b \mod q$ is close to uniform and therefore statistically hides b.

– Step 1: $X_{top} \cdot \mathbf{T}^\gamma_b \mod q$ is completely independent of b:
This is satisfied due to the fact that $X_{top} \cdot (\boldsymbol{x}^\gamma_0 - \boldsymbol{x}^\gamma_1) = 0$ over q. One can check this relation through the construction of matrix X_{top}.

– Step 2: $X_{bot} \cdot \mathbf{T}^\gamma_b \mod q$ is close to uniform:
For this, we show that the residual distribution of following vector, conditioned on all the leaked information, has high minimum entropy.

$$X_{bot} \cdot \mathbf{T} = X_{bot} \cdot \begin{pmatrix} \boldsymbol{t}_1 \\ \vdots \\ \boldsymbol{t}_n \end{pmatrix}$$

[16] In fact, we can prove that their variants for *selective per label* are indistinguishable.

Then using (a variant of) the leftover hash lemma with randomness $X_{bot} \cdot \mathbf{T}$ and seed $\boldsymbol{u}$, we will then conclude that conditioned on all the leaked information, the pair $(\boldsymbol{u}, X_{bot} \cdot \mathbf{T} \cdot \boldsymbol{u})$ is close to uniform and hence it statistically hides bit b in Eq. (2).

Theorem 35 confirms that $X_{bot} \cdot \mathbf{T}$ has the min-entropy $m_0 \log(4/3)$. Thus, thanks to the leftover hash lemma, having the min-entropy conditioned on $I = (X_{top}, X_{top}\mathbf{T})$, the pair $(\boldsymbol{u}, \mathbf{X}_{bot}\mathbf{T} \cdot \boldsymbol{u})$ is within statistical distance $\frac{1}{2}\sqrt{2^{-H_\infty(X_{bot}\cdot\mathbf{T}|I)} \cdot 2^{\log q}}$. Which is less than $2^{-\kappa}$, when $-H_\infty(X_{bot} \cdot \mathbf{T}|I) + \log q \leq -2\kappa$. Resulting in the condition $\log(3/4) \cdot (\log q + 2\kappa) \leq m_0$.

Now by applying a complexity leveraging technique, we have,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.3}}(n_0, n) = P^{2n} \cdot \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.3}}(n_0, n), \quad \text{and} \quad \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n) = P^{2n} \cdot \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.4}}(n_0, n)$$

Thus,

$$\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.3}}(n_0, n) = P^{2n} \cdot (\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.4}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.3}}(n_0, n))$$

Meaning that if $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.4}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}^*_{2.\gamma.3}}(n_0, n)| \leq 2^{-\kappa} \cdot P^{-2n}$ then, $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.3}}(n_0, n)| \leq 2^{-\kappa}$. Clearly, if in the last part of th proof one sets $-H_\infty(X_{bot} \cdot \mathbf{T}|I) + \log q \leq -2(\kappa + 2n \log P)$ or equivalently $\log(3/4) \cdot (\log q + 2\kappa + 4n \log P) \leq m_0$, then $|\mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.4}}(n_0, n) - \mathsf{Win}_{\mathcal{A}}^{\mathsf{G}_{2.\gamma.3}}(n_0, n)| \leq 2^{-\kappa}$ and consequently, the indistinguishability of the adaptive variants would be concluded. □

Lemma 34. *In Theorem 33, conditioned on all the leaked information, the min-entropy of* $X_{bot} \cdot \mathbf{T}$ *is* $\geq m_0 \log(4/3)$.

Proof. Here we describe what are the leaked information about $\mathbf{T}$ in the adversary's view.

1. all the ciphertexts for $\ell \neq \ell_\gamma$: we note that these ciphertexts don't contain any information about $\mathbf{T}$.

2. secret key queries: it is essentially $\Sigma_i y_i \cdot \mathbf{Z}_i$. And conditioning on this information is the same as conditioning on $X_{top} \cdot \mathbf{T}$, since y can be written as a linear combination of rows of X_{top}.
3. corruption queries for slot i: it leaks the key $\mathbf{Z}_i$.

We first consider the distribution of $X_{bot} \cdot \mathbf{T}$ conditioned on $(\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{T})$. Note that in $\mathbf{X}_{top}\mathbf{T}$ and $\mathbf{X}_{bot}\mathbf{T}$ matrices $\mathbf{X}_{top}$ and $\mathbf{X}_{bot}$ act in parallel on the columns of $\mathbf{T}$. We can hence restrict ourselves to the distribution of $\mathbf{X}_{bot}\mathbf{T}_i$ conditioned $(\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{T}_i)$, where $\mathbf{T}_i$ stands for the ith column of $\mathbf{T}$. Fix $\mathbf{T}_i^*$ arbitrary. The distribution of $\mathbf{T}_i$ given $(\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{T}_i)$ is $\mathbf{T}_i^\star + D_{\Lambda,\sigma,-\mathbf{T}_i^\star}$, with $\Lambda = \{\boldsymbol{y} \in \mathbb{Z}^n \ : \ \mathbf{X}_{top}\boldsymbol{y} = \mathbf{0}\}$. By construction of $\mathbf{X}$, we have that $\Lambda = \mathbb{Z}\boldsymbol{x}^\gamma$. As a result, the conditional distribution of $\mathbf{X}_{bot}\mathbf{T}_i$ is $\langle \boldsymbol{x}^\gamma, \mathbf{T}_i^\star \rangle + D_{\|x^\gamma\|^2.\mathbb{Z}, \|x^\gamma\|\sigma, \langle x^\gamma, -\mathbf{T}_i^\star \rangle}$.

Since $\sigma \geq 10 \cdot nP^2 \geq 10 \cdot ||\boldsymbol{x}^\gamma||^2$, we can apply Theorem 35. Thus, after conditioning with respect to $(\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{T})$, each column of $\mathbf{X}_{bot}\mathbf{T}$ has min-entropy $\geq \log(4/3)$. Due to the fact that columns are independent, we have that,

$$H_\infty(\mathbf{X}_{bot}\mathbf{T}|\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{T}) \geq m_0 \log(4/3).$$

□

Lemma 35. *[ALS16,PR06] Let $\Lambda = k\mathbb{Z}$ be a 1-dimensional lattice. For any $\sigma \geq 10 \cdot k$, $b \in \Lambda$ and $c \in \mathbb{R}$, we have that $\mathcal{D}_{\Lambda,\sigma,c}(b) \leq 3/4$. In particular, we have $H_\infty(\mathcal{D}_{\Lambda,\sigma,c}) \geq 0.4$, where $H_\infty(\cdot)$ refers to the mini-entropy.*

C.3 Parameter Setting.

Correctness of scheme. We remind that for the correctness, we had the conditions $q_0 > K(nV+1)$.
Reduction from LWE to our construction. The indistinguishability between games G_1 and G_2 needs $q \geq q_0 {n_0}^{\omega(1)} B$, where $|\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell| \leq B$ with overwhelming probability. To present a precise bound, we find the value B. By Markov's inequality, for a Gaussian variable with mean 0,

$$\Pr[|X| \leq a] \leq 1 - 2\exp(-\frac{1}{2}\sigma^2\lambda^2 - \lambda a) \text{ for any } \lambda$$

Thus, for $a = \sigma$ and $\lambda = 1$,

$$\Pr[|X| \leq \sigma] \leq 1 - 2\exp(-\sigma)$$

Meaning that if $\sigma = \Theta({n_0}^\epsilon), \sigma' = \Theta({n_0}^\epsilon),\ \epsilon > 0$, where σ and $\sigma' = q \cdot \alpha'$ are respectively standard deviations for variables $\boldsymbol{t}_i$ and $\boldsymbol{e}_\ell$, then,

$$\Pr[|\boldsymbol{t}_i| \leq \sigma] \leq 1 - \mathrm{negl}(n_0), \quad \Pr[|\boldsymbol{e}_\ell| \leq \sigma'] \leq 1 - \mathrm{negl}'(n_0)$$

So, $|\boldsymbol{t}_i \cdot \boldsymbol{e}_\ell| \leq \sigma \cdot \sigma'$ with overwhelming probability i.e., we can set $B = \sigma \cdot \sigma'$.

The statistical argument from game $\mathsf{G}_{2.\gamma.3}$ to game $\mathsf{G}_{2.\gamma.5}$ needs $\sigma \geq 10.nP^2$. And also $\Omega(\log q) \leq m_0$ and $\Omega(\log q + 4n \log P) \leq m_0$, respectively for the selective security and the adaptive security. One can set $\kappa = \omega(1)$ where $\omega(1)$ comes from $q > q_0 {n_0}^{\omega(1)} B$.
Reduction from lattice problems to LWE. for this reduction we need $q \geq \Omega(\sqrt{n_0}/\alpha')$. Since module q is super-polynomially-big this condition is already satisfied.

D Decentralized Multi-Client Functional Encryption

Now, we introduce the definition of decentralized multi-client functional encryption (DMCFE) [CDG+18a].

Definition 36. (Decentralized Multi-Client Functional Encryption) Let $\mathcal{F} = \{\mathcal{F}_\rho\}_\rho$ be a family (indexed by ρ) of sets $\mathcal{F}_\rho$ of functions $f \colon \mathcal{X}_{\rho,1} \times \cdots \times \mathcal{X}_{\rho,n_\rho} \to \mathcal{Y}_\rho$.Let $\mathsf{Labels} = \{0,1\}^*$ or $\{\perp\}$ be a set of labels. A decentralized multi-client functional encryption scheme (DMCFE) for the function family $\mathcal{F}$ and the label set Labels is a tuple of six algorithms $\mathsf{DMCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDerShare}, \mathsf{KeyDerComb}, \mathsf{Enc}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\kappa, 1^n)$ is defined as for MCFE in Theorem 1.
$\mathsf{KeyGen}(\mathsf{pp})$**:** Takes as input the public parameters pp and outputs n secret keys $\{\mathsf{sk_i}\}_{i\in[n]}$.
$\mathsf{KeyDerShare}(\mathsf{pp}, \mathsf{sk}_i, f)$**:** Takes as input the public parameters pp, a secret key sk_i from position i and a function $f \in \mathcal{F}_\rho$, and outputs a partial functional decryption key $\mathsf{sk}_{i,f}$.
$\mathsf{KeyDerComb}(\mathsf{pp}, \mathsf{sk}_{1,f}, \ldots, \mathsf{sk}_{n,f})$**:** Takes as input the public parameters pp, n partial functional decryption keys $\mathsf{sk}_{1,f}, \ldots, \mathsf{sk}_{n,f}$ and outputs the functional decryption key sk_f.
$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, \ell)$ is defined as for MCFE in Theorem 1.
$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}_{1,\ell}, \ldots, \mathsf{ct}_{n,\ell})$ is defined as for MCFE in Theorem 1.

A scheme DMCFE is correct, if for all $\kappa, n \in \mathbb{N}$, $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa, 1^n)$, $f \in \mathcal{F}_\rho$, $\ell \in \mathsf{Labels}$, $x_i \in \mathcal{X}_{\rho,i}$, when $\{\mathsf{sk}_i\}_{i\in[n]} \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, $\mathsf{sk}_{i,f} \leftarrow \mathsf{KeyDerShare}(\mathsf{sk}_i, f)$ for $i \in [n]$, and $\mathsf{sk}_f \leftarrow \mathsf{KeyDerComb}(\mathsf{pp}, \mathsf{sk}_{1,f}, \ldots, \mathsf{sk}_{n,f})$, we have

$$\Pr\left[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_1, x_1, \ell), \ldots, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_n, x_n, \ell)) = f(x_1, \ldots, x_n)\right] = 1.$$

We consider a similar security definition for the decentralized multi-client scheme. We point out that contrary to [CDG+18a], we do not differentiate encryption keys from secret keys. This is without loss of generality, as corruptions in [CDG+18a] only allow to corrupt both keys at the same time.

Definition 37. (Security of DMCFE) The xx-yy-zz-IND security notion of a DMCFE scheme is similar to the one of an MCFE scheme (Theorem 2), except that there is no master secret key msk and the key derivation oracle is now defined as:

Key derivation oracle $\mathsf{QKeyD}(f, i)$**:** Output $\mathsf{sk}_{i,f} := \mathsf{KeyDerShare}(\mathsf{sk}_i, f)$.

D.1 Decentralized-MCFE with partial functional-keys

In a decentralized version of MCFE no authority is in charge of generating individual secret keys, but rather each client takes care of its secret keys. The compiler by Abdalla et al. [ABKW19,ABG19] transferring MCFE to DMCFE, uses an information-theoretically secure layer on the functional-keys requiring a square-size of the key in the key-generation phase. Meaning that, we can go from MCFE to DMCFE with no need for a new assumption. We refer the readers to their general compiler while the underlying MCFE scheme is realized by our DCR-based MCFE scheme. As their compiler is general we do not discuss it here in detail, but it worth to mention that their scheme has $O(n^2)$ key-size while the ciphertext-size is of the same order of the underlying MCFE. They have proved the security for the case that the adversary is restricted to ask for all the partial functional-keys.

Chotard et al. [CDG+18a] presented a DMCFE scheme with linear size of the key and based on pairing (for their spacial MCFE based on DDH assumption).[17] As it might be clear, in MCFE the only part which relies on the authority is the functional decryption-key generation which is $\mathsf{sk}_y = \sum y_i s_i$. This value can be computed in a decentralized way as follows; One may see sk_y as $\langle \boldsymbol{sy}, \mathbf{1} \rangle$ which can again be computed by a MCFE with message $t_i = s_i y_i$ for the client i and the decryption-key would be like $\sum s'_i$ where s'_i is the secret-key chosen by the client i. A MPC protocol or a trysted setup preparing $\sum s'_i = 0$, would complete the puzzle for the decentralization. The DMCFE [CDG+18a] needs pairing, this requirement comes from the fact that for their underlying MCFE (based on DDH assumption) the inner-product value (here $\sum s_i y_i$) needs to be small which may not be satisfied as s_i is chosen uniformly from $\mathbb{Z}_p$. One can go around this problem by pairing.

Here we discuss how we can extend their construction for some MCFE schemes which do not need discrete-logarithm computation during the decryption and consequently no pairing would be needed in the DMCFE construction. We try to give a general construction in order to use it for both of our MCFE scheme (DCR-based and LWE-based MCFE). The point about this construction is that, one can prove the security against the static corruption and with no restriction on the completeness of partial functional-keys.

In the following theorem *sel*1 stands for the case that in the security game, there is only a single functional-key query which should be asked at the beginning of the game. This is a very weak security requirement, which not only the existing MCFE scheme can fulfill it, but also one that may come up with more efficient constructions satisfying only this weak version of security.

Theorem 38. *The DMCFE scheme in Fig. 14 is sta-yy-IND secure, if* MCFE_1 *is sta-yy-IND secure and* MCFE_2 *is sel1-sta-*pos$^+$*-IND secure. Concretely,*

$$\mathsf{Adv}^{\text{sta-}yy\text{-IND}}_{\mathcal{A},\mathsf{DMCFE}}(\kappa, n) \leq 2\mathsf{Adv}^{\text{sta-}yy\text{-IND}}_{\mathcal{B},\mathsf{MCFE}_1}(\kappa, n) + 2\mathsf{Adv}^{sel\text{1-sta-}yy\text{-IND}}_{\mathcal{B}',\mathsf{MCFE}_2}(\kappa, n).$$

We proceed through a sequence of the games such that the first game is the real game associated with bit $b = 0$ and the last game is the real game associated with bit $b = 1$. The intuition for the proof is as follows: we note that the encryption algorithm is only involved with MCFE_1. Thus, we can use the security of MCFE_1 to change the message $\boldsymbol{x}^0$ to $\boldsymbol{x}^1$, if we can be sure that the only information leaking through the queries $\mathsf{QKeyD}(\boldsymbol{y}, i)$ is $\sum y_i \mathsf{sk}_{1,i}$ (which is the functional-key

[17] Chotard et al. [CDG+18b] also presented a compiler with liner size of the key but based on an additional assumption CDH. Adding a new assumption may not be a good idea when the underling MCFE is based on other assumptions such as DCR or LWE.

$\underline{\mathsf{Setup}(1^\kappa, 1^n)}$: – run $\mathsf{pp}_1 \leftarrow \mathsf{Setup}_1(1^\kappa, 1^n)$ – run $\mathsf{pp}_2 \leftarrow \mathsf{Setup}_2(1^\kappa, 1^n)$ – choose a hash function $H : \mathcal{Y} \longrightarrow \mathsf{Labels}$ Output $\mathsf{pp} = (\mathsf{pp}_1, \mathsf{pp}_2, H)$. $\underline{\mathsf{KeyGen}(\mathsf{pp})}$: – run $\mathsf{sk}_{1,i} \leftarrow \mathsf{KeyGen}_1(\mathsf{pp}_1)$ – interactively produce $\mathsf{sk}_{2,i}$ st. $\sum \mathsf{sk}_{2,i} = 0$ Return $\mathsf{sk} = (\mathsf{sk}_{1,i}, \mathsf{sk}_{2,i})_i$. $\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}, i, \ell)}$: To encrypt a message $\boldsymbol{x}$ – run $\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i, i, \ell)$. Return $\mathsf{ct}_{i,\ell}$	$\underline{\mathsf{KeyDerShare}(\mathsf{pp}, \mathsf{sk}_i, i, \boldsymbol{y})}$: To generate a key for $\boldsymbol{y}$ Return $\mathsf{sk}_{i,y}$ $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, y_i \mathsf{sk}_{1,i}, i, H(\boldsymbol{y}))$ $\underline{\mathsf{KeyDerComb}(\mathsf{pp}, \{\mathsf{sk}_{i,y}\}_i)}$: – run $\mathsf{sk} \leftarrow \mathsf{Dec}_2(\mathbf{1}, \mathbf{0}, \{\mathsf{sk}_{i,y}\}_i, H(\boldsymbol{y}))$. Rutern sk $\underline{\mathsf{Dec}(\boldsymbol{y}, \mathsf{sk}, \{\mathsf{ct}_{i,\ell}\}_i, \ell)}$: –run $C = \mathsf{Dec}_1(\boldsymbol{y}, \mathsf{sk}, \{\mathsf{ct}_{i,\ell}\}_i, \ell)$ Return C

Fig. 14: DMCFE from MCFE

value for MCFE_1). To prove that queries $\mathsf{QKeyD}(\boldsymbol{y}, i)$ leak no information beyond $\sum y_i \mathsf{sk}_{1,i}$, we should be able to replace the keys $\mathsf{sk}_{1,i}$ with some randoms $\mathsf{sk}'_{1,i}$ in $\mathsf{Enc}_2(\mathsf{sk}_{2,i}, y_i \mathsf{sk}_{1,i}, H(\boldsymbol{y}))$ (which is the response to $\mathsf{QKeyD}(\boldsymbol{y}, i)$), as far as $\sum y_i \mathsf{sk}'_{1,i} = \sum y_i \mathsf{sk}_{1,i}$. The formal description of the games are as follows:

Game G_0: is the real game associated with $b = 0$.

Game G_1: is similar to the previous game, except that, each key $\mathsf{sk}_{i,1}$ is replaced with a new key $\mathsf{sk}'_{i,1}$ such that $\sum y_i \mathsf{sk}_{i,1} = \sum y_i \mathsf{sk}'_{i,1}$. All the queries are answered by the real algorithm based on the new keys $\mathsf{sk}'_{i,1}$. The indistinguishability between game G_1 and G_0 reduces to the security of MCFE_2. Intuitively, there are two main possible cases here. One is that $\mathsf{QKeyD}(\boldsymbol{y}, i)$ are asked over all the slots, which would be directly reduced to the pos$^+$-security of MCFE_2 knowing the last honest slot which takes care of the equality $\sum y_i \mathsf{sk}_{i,1} = \sum y_i \mathsf{sk}'_{i,1}$. And the second case is when for some indices i no query $\mathsf{QKeyD}(\boldsymbol{y}, i)$ or $\mathsf{QCorrupt}(i)$ is issued. For this case, we try to fill up the missing slots such that again $\sum y_i \mathsf{sk}_{i,1} = \sum y_i \mathsf{sk}'_{i,1}$ is satisfied and then again rely on the pos$^+$-security of MCFE_2. The formal proof is given in Theorem 39 expressing that:

$$|\mathsf{Win}^{\mathsf{G}_0}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa, n)| \leq \mathsf{Adv}^{\mathsf{MCFE}_2}_{\mathcal{B}}(\kappa, n).$$

Game G_2: is similar to the previous game, except that, the message $\boldsymbol{x}^0$ is replaced $\boldsymbol{x}^1$. Here we rely on the security of MCFE_1. The simulation of $\mathsf{sk}_{i,y}$ is possible, tanks to the previous change removing the $\mathsf{sk}_{1,i}$ from $\mathsf{sk}_{i,y}$. The formal proof for indistiguishability between G_1 and G_2 is given in Theorem 40 showing that

$$|\mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_2}_{\mathcal{A}}(\kappa, n)| \leq \mathsf{Adv}^{\mathsf{MCFE}_1}_{\mathcal{B}}(\kappa.n).$$

Games G_3, G_4: are backward-version of games G_1 and G_0 with similar reasoning, which end up with the real game associated with $b = 1$.

Lemma 39 (Transition from G_0 to G_1). *In Theorem 38 If MCFE_2 is sel1-sta-pos$^+$-IND secure, then two games G_0 and G_1 are computationally distinguishable.*

Proof. Let $\mathcal{B}$ be the attacker to the security of MCFE_2, and $\mathcal{A}$ is the adversary trying to distinguish between G_0 and G_1. $\mathcal{B}$ samples $i^* \leftarrow \{0, 1, \ldots, n\}$ as its guess for the missing index, expecting that $\mathcal{A}$ will never ask queries $\mathsf{QKeyD}(\boldsymbol{y}, i^*)$ for a vector $\boldsymbol{y}$ or corrupt the slot i^*, otherwise $\mathcal{B}$ just aborts. If $i^* = 0$, it means for a vector $\boldsymbol{y}$, the queries QKeyD are issued for all the positions.

– $\mathcal{B}$ runs $\mathsf{sk}_{1,i} \leftarrow \mathsf{KeyGen}_1(\mathsf{pp}_1)$ for $i = 1, \ldots, n$.

Game	Description	Justification
G_0	$\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i^0, i, \ell), \quad \sum \mathsf{sk}_{2,i} = 0$ $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, y_i \mathsf{sk}_{1,i}, H(\boldsymbol{y}))$	real game $b = 0$
G_1	$\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i^0, i, \ell), \quad \sum \mathsf{sk}_{2,i} = 0$ $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, \boxed{y_i \mathsf{sk}'_{1,i}}, H(\boldsymbol{y}))$ *st.* $\sum y_i \mathsf{sk}'_{1,i} = \sum y_i \mathsf{sk}_{1,i}$	security of MCFE_2
G_2	$\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}_1(\mathsf{sk}_{1,i}, \boxed{x_i^1}, i, \ell), \quad \sum \mathsf{sk}_{2,i} = 0$ $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, y'_i \mathsf{sk}'_{1,i}, H(\boldsymbol{y}'))$	security of MCFE_1
G_3	$\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i^1, i, \ell), \quad \sum \mathsf{sk}_{2,i} = 0$ $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, \boxed{y_i \mathsf{sk}_{1,i}}, H(\boldsymbol{y}))$ *st.* $\sum y_i \mathsf{sk}'_{1,i} = \sum y_i \mathsf{sk}_{1,i}$	security of MCFE_2
G_4	$\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i^1, i, \ell), \quad \sum \mathsf{sk}_{2,i} = 0$ $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, y_i \mathsf{sk}_{1,i}, H(\boldsymbol{y}))$	real game $b = 1$

Fig. 15: Overview of games for Theorem 38

- for each $i \in \mathcal{CS}$, the adversary $\mathcal{B}$, sends a corruption query to its challenger and sends back the response $\mathsf{sk}_{2,i}$ along side $\mathsf{sk}_{1,i}$.
- when $\mathcal{B}$ receives challenges $\mathsf{QLeftRight}(i, x_i^0, x_i^1, \ell)$ from $\mathcal{A}$, it responds with $\mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i^0, i, \ell)$.
- when $\mathcal{B}$ receives $\mathsf{QKeyD}(\boldsymbol{y}, i)$ from $\mathcal{A}$:
  - if $i^* = 0$, the adversary $\mathcal{B}$ sends query $\mathsf{QLeftRight}(i, y_i \mathsf{sk}_{1,i}, y_i \mathsf{sk}_{1,i}, \boldsymbol{y})$ to its challenger and sends back the result to $\mathcal{A}$.
  - if $i^* \neq 0$, for any $j \in \mathcal{HS}$ it samples $\mathsf{sk}'_{1,j}$ and sets $m^0_{j,y} = y_j \mathsf{sk}_{1,j}$, $m^1_{j,y} = y_j \mathsf{sk}'_{1,j}$. For any $j \in \mathcal{CS}$ it sets $m^0_{j,y} = m^1_{j,y} = y_j \mathsf{sk}_{1,j}$

  Then, it sends $\mathsf{QLeftRight}(i, m^0_{i,y}, m^1_{i,y}, y)$ to its challenger and receives $\mathsf{sk}_{i,y} = \mathsf{Enc}_2(\mathsf{sk}_{2,i}, m^b_{i,y}, i, y)$ and sends back $\mathsf{sk}_{i,y}$ to $\mathcal{A}$.
- when $\mathcal{A}$ outputs a bit b', at first $\mathcal{B}$ checks if its guess for i^* is correct where otherwise it aborts. It sets $m^0_{i^*,y}$ and $m^1_{i^*,y}$ and also $m^0_{j,y}, m^1_{j,y}$ for other missing slots j, such that $\sum_{j \in [n]-i^*} m^0_{j,y} + m^0_{i^*,y} = \sum_{j \in [n]-i^*} m^1_{j,y} + m^1_{i^*,y}$ and sends $\mathsf{QLeftRight}(i, m^0_{j,y}, m^1_{j,y}, \boldsymbol{y})$ to its challenger for any missing slot j. Finlay, it outputs bit b'.

□

Lemma 40 (Transition from G_1 to G_2). *If MCFE_1 is sta-yy-IND secure, then two games G_1 and G_2 are indistinguishable. Concretely,*

$$|\mathsf{Win}^{\mathsf{G}_1}_{\mathcal{A}}(\kappa, n) - \mathsf{Win}^{\mathsf{G}_2}_{\mathcal{A}}(\kappa, n)| \leq \mathsf{Adv}^{\mathsf{MCFE}_1}_{\mathcal{B}}(\kappa, n).$$

Let $\mathcal{B}$ be the adversary attacking to the security of MCFE_1, and $\mathcal{A}$ the attacker trying to distinguish between two games G_1 and G_2. The adversary $\mathcal{B}$ simulate the game for the adversary $\mathcal{A}$ as follows:

- *the adversary $\mathcal{B}$ runs $\mathsf{pp}_2 \leftarrow \mathsf{Setup}_2(1^\kappa, 1^n)$ and chooses $\mathsf{sk}_{2,i}$ such that $\sum \mathsf{sk}_{2,i} = 0$. It also chooses random values $\mathsf{sk}'_{1,i}$.*
- *for each $i \in \mathcal{CS}$, the adversary $\mathcal{B}$ sends a corruption query to its challenger and sends back the response $\mathsf{sk}_{1,i}$ along side $\mathsf{sk}_{2,i}$.*
- *when $\mathcal{B}$ receives the challenges $\mathsf{QLeftRight}(i, x_i^0, x_i^1, \ell)$ from $\mathcal{A}$, sends it directly to its challenger and receives $\mathsf{Enc}_1(\mathsf{sk}_{1,i}, x_i^b, i, \ell)$ which would be returned to $\mathcal{A}$.*
- *when $\mathcal{B}$ receives queries $\mathsf{QKeyD}(\boldsymbol{y}, i)$ from $\mathcal{A}$, it simply runs $\mathsf{Enc}_2(\mathsf{sk}_2, i.y_i \mathsf{sk}_{1,i}, H(\boldsymbol{y}))$ and turn back the result to $\mathcal{A}$.*
- *it outputs bit b' given by the adversary $\mathcal{A}$.*

We remark that with a similar trick of guessing a missing slot i^* and answering to the corruption and QKeyD queries on-the-fly but in a consistent way, one can extend the security proof of the DMCFE compiler presented in [ABG19] against partial functional-keys and adaptive corruptions.

E Details of Implementation

E.1 DDH Implementation

Choice of the group. We chose to use an elliptic curve with a prime order that is 256 bits long, already predefined in the openssl library. Hence, we used brainpoolP256t1, however, the design of the implementation allows us to switch easily to another curve by changing the public parameters generation.
Decryption. The decryption is the most constraining part of the implementation because it needs to compute a discrete logarithm. Here, we solve this problem by sequentially testing all numbers. The decryption is thus efficient enough since our output is small, but if the output grows bigger, the decryption time becomes hard to manage. It is possible to trade-off memory for space, using a baby-step giant-step algorithm to compute the discrete logarithm.

E.2 DCR Implementation

Choice of parameters. As this implementation is a proof of concept, we decided to use very conservative parameters, to show that the scheme can run with very large parameters. We advice anyone who wants to use this work for an application to chose more carefully the parameters that fits their requirements for security and efficiency. We used the OpenSSL library for big numbers for all the elements in the scheme, as well as their RSA key generation in order to generate the public parameters, and chose a 4096 bits number N. The discrete Gaussian was also overshot, and was required to be at least as large as N^6. We also required the output of the hash function to be at least 256 bits larger than the modulo, in order to be very close to the uniform distribution (statistical distance less than 2^{-256}).
Discrete Gaussian sampling. One of the main challenges in this implementation was to sample a very large Gaussian distribution over the integers. We used the sampler described in [MW17] for large standard deviations. To keep the implementation simple and readable, we decided to only use integers for computations, so once again, with further work, this stage can be optimized, for both more precise sampling (we overshoot the target a lot) and also faster sampling. We also took a very small $\varepsilon = 2^{-256}$ when taking a bound on the smoothing parameter of $\mathbb{Z}$. This shouldn't be required by most application, so there is room for improvement there also. Our base sampler has standard deviation 64, is implemented using CDT, and has tails cut above 1023 and under -1023. At each step, z_i is s_i divided by 16, which again, leaves space for improvement if taking a more precise bound on the smoothing parameter.
Observations and possible optimizations. The main bottleneck in this implementation is the size of the secret keys. The secret keys are very large, thus requiring a lot of space, and slowing down key generation as well as encryption and decryption. Indeed, the secret keys are used as exponents during encryption and decryption, and they cannot be reduced modulo the order of the group since it has to remain a secret. This implies that those operation get slower and slower as the size of the secret keys grow. A first step to improve the implementation is to get a closer look at the concrete requirements for security of the scheme, and sample a Gaussian distribution with a standard deviation that matches more closely the requirements.

E.3 LWE Implementation

Number representations and modulo. In order to have more efficient code, we chose to work with 128 bits numbers, so we had to choose a modulo that is smaller than 2^{128}. To get the

most efficient possible modulo operation, we picked the twelfth Mersenne prime $q = 2^{127} - 1$. The ciphertexts were stored on 32 bits integers, ensuring big enough rounding for security, but large enough for correctness.

Choice of parameters. As discussed previously, we decided to encrypt vectors of size 100, and have a message space of $K = 1048576$. The dimensions for the keys are 500 for each of the $\mathbf{s}$ and $\mathbf{t}$ parts, and the standard deviation chosen is 1000. Note that this is not the standard deviation for the error for the LWE assumption, the standard deviation for the LWE assumption does not appear in the scheme since we are instead using rounding. We don't give a precise security estimate, since the proof allows for a trade-off in the computational security against statistical security, meaning that the rounding errors can be smaller if we decide to take a smaller standard deviation for the LWE problem, leading to less statistical loss during the proof, but also relying on an easier LWE instance. For a given application, it is possible to optimize for the security requirements, depending on the number of samples given to the adversary, the time it has got to execute its attack and other considerations. We chose those parameters for a security of over a hundred bits for reasonable applications.

F Extensions on MCFE

F.1 Extensions of our DCR-based MCFE

Extension to vectors per slots. In the previous section we presented an MCFE scheme for the case that each client has a single input. This helped to simplify the proof. But our construction can easily be extended to vectors associated with clients (slots). In Fig. 16, we can consider $|y_{i,j}|,\ |x_{i,j}| \leq \sqrt{\frac{N}{2nm}}$ where $y_{i,j},\ x_{i,j}$ are respectively the jth component of ith slot (client) of the key and message vectors. Here $(1+N)^{\boldsymbol{x}_i} \cdot \mathcal{H}^{\boldsymbol{s}_i}(\ell)$ is the column vector $((1+N)^{x_{i,1}} \cdot \mathcal{H}^{s_{i,1}}(\ell), \ldots, (1+N)^{x_{i,m}} \cdot \mathcal{H}^{s_{i,m}}(\ell))$.

$\underline{\mathsf{Setup}(1^\kappa, n, m)}$:
– run $\mathsf{SP}(\kappa)$ to get (p, q)
– compute $N = pq$.
– Let $\mathcal{H} : \mathsf{Labels} \to \mathbb{Z}^*_{N^2}$ be a full-domain hash function.
– set $X < \sqrt{\frac{N}{2nm}}$
Return $\mathsf{pp} = (N, \mathcal{H}, X)$

$\underline{\mathsf{KeyGen}(\mathsf{pp})}$:
– sample $\boldsymbol{s} \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$
– where $\boldsymbol{s} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n),\ \boldsymbol{s}_i \in \mathbb{Z}^m$.
– set $\sigma > \sqrt{\kappa} \cdot N^{5/2}$ for the selective security and $\sigma > \sqrt{\kappa + 2nm\log(2X)} \cdot N^{5/2}$ for the adaptive security.
Return $\mathsf{msk} = \boldsymbol{s}$ and $\mathsf{sk}_i = \boldsymbol{s}_i$.

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i, i, \ell)}$:
To encrypt a message $\boldsymbol{x}_i \in \mathbb{Z}^m$ where with $|x_{ij}| \leq X < \sqrt{\frac{N}{2nm}}$;
– compute $\mathsf{ct}_{i,\ell} = (1+N)^{\boldsymbol{x}_i}.\mathcal{H}(\ell)^{\boldsymbol{s}_i} \mod N^2$.
Return $ct_{i,\ell}$

$\underline{\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \boldsymbol{y})}$:
To generate a key for $\boldsymbol{y} \in \mathbb{Z}^{m\times n}$ where $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$, $\boldsymbol{y}_i \in \mathbb{Z}^m$ with $|y_{ij}| \leq Y$:
– compute $\mathsf{sk}_y = \Sigma_i \boldsymbol{s}_i^T \cdot \boldsymbol{y}_i$
Return sk_y

$\underline{\mathsf{Dec}(\mathsf{pp}, \boldsymbol{y}, \mathsf{sk}, \{\mathsf{ct}_{i,\ell}\}_{i\in[n]}, \ell)}$:
compute $C = \prod_i (\mathsf{ct}_{i,\ell}^T)^{\boldsymbol{y}_i}.\mathcal{H}(\ell)^{\mathsf{sk}}$
Return $\frac{C-1 \mod N^2}{N}$

Fig. 16: DCR-based MCFE (vectors per slots)

Security extension (from one to pos$^+$). Abdalla et al. [ACF+18] gave a general conversion from one-time MIFE to many-challenges MIFE. More precisely, they showed that by having

a one-time MIFE and putting a single-FE layer on it, we can get MIFE secure against many-ciphertexts challenges. Chotard et al. [CDG+18b] used the same idea proving that if each client use another layer of single-input FE over the output of MFCE scheme, then the security can be extended from one-ciphertext per label to many-ciphertexts per label (pos$^+$). The point is that the outer layer and the inner one should be compatible. It means that the ciphertext produced by the inner layer should belong to the message space of the outer layer and the secret key produced by the inner layer should belong to the (functional) key space of the outer layer. As it is also pointed out in [CDG+18b] because of the restriction on the compatibility, the suggested conversion is not general. In Fig. 17 we have directly instantiated the mentioned conversion by a single-input FE compatible with our MCFE construction. The security proof is eliminated due to the similarity to [ACF+18] and [CDG+18b]. Note that this technique works when $m > 2$. Thus, we have considered the inputs of the clients as vectors. In this instantiation as the outer layer, we are using the single-input FE scheme based on DCR assumption such that the message space is an encoding of $\mathbb{Z}$ (as $(1+N)^{\boldsymbol{x}_i} \cdot \mathcal{H}^{s_i}(\ell)$, see Fig. 17).

$\underline{\mathsf{Setup}(1^\kappa, n, m)}$:

– run $\mathsf{SP}(\kappa)$ to get (p, q)

– compute $N = pq$.

– sample $g' \xleftarrow{R} \mathbb{Z}^*_{N^2}$

– compute $g = g'^{2N} \mod N^2$.

– Let $\mathcal{H} : \mathsf{Labels} \to \mathbb{Z}^*_{N^2}$ be a full-domain hash function.

– set $X, Y < \sqrt{\frac{N}{2nm}}$

Return $\mathsf{pp} = (N, g, \mathcal{H}, X)$

$\underline{\mathsf{KeyGen}(\mathsf{pp})}$

– sample $\hat{\boldsymbol{s}} \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times n}, \sigma}$ where $\hat{\boldsymbol{s}}_i \in \mathbb{Z}^m$

– sample $\boldsymbol{s} \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times n}, \sigma^*}$ where $\boldsymbol{s}_i \in \mathbb{Z}^m$

– set $\sigma > \sqrt{\kappa} \cdot N^{5/2}$

– set $\sigma^* = \sigma$ for the selective security and $\sigma^* > \sqrt{\kappa + 2nm\log(2X)} \cdot N^{5/2}$ for the adaptive security.

Return $\mathsf{msk} = (\boldsymbol{s}, \hat{\boldsymbol{s}})$ and $\mathsf{sk}_i = (\boldsymbol{s}_i, \hat{\boldsymbol{s}}_i)$.

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i, i, l)}$:

To encrypt a message $\boldsymbol{x}_i \in \mathbb{Z}^m$ with $|x_{ij}| \leq X <$;

– compute $h_i = g^{\hat{\boldsymbol{s}}_i} \mod N^2$.

– sample $r_i \xleftarrow{R} \{0, \ldots, [\frac{N}{4}]\}$.

– set $\mathsf{ct}_{i,\ell} = \begin{cases} g^{r_i} \bmod N^2, \\ (1+N)^{\boldsymbol{x}_i}.\mathcal{H}(\ell)^{\boldsymbol{s}_i} \cdot h_i^{r_i} \bmod N^2 \end{cases}$

Return $\mathsf{ct}_{i,\ell}$

$\underline{\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \boldsymbol{y})}$:

For $\boldsymbol{y} \in \mathbb{Z}^{m\times n}$ where $\boldsymbol{y}_i \in \mathbb{Z}^m$ with $|y_{ij}| \leq Y$:

– compute $\mathsf{sk}_y = \Sigma_i \boldsymbol{s}_i^T \cdot \boldsymbol{y}_i$ and $\mathsf{sk}_{y_i} = \langle \boldsymbol{y}_i, \hat{\boldsymbol{s}}_i \rangle$

Return $(\mathsf{sk}_y, \{\mathsf{sk}_{y_i}\}_{i\in[n]})$

$\underline{\mathsf{Dec}(\mathsf{pp}, \boldsymbol{y}, \mathsf{sk}, \{\mathsf{ct}_{i,\ell}\}_{i\in[n]}, \ell)}$:

– parse $\boldsymbol{y}$ as $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$, sk as $(\mathsf{sk}_y, \{\mathsf{sk}_{y_i}\}_{i\in[n]})$ and $\mathsf{ct}_{i,\ell}$ as $(\mathsf{ct}^0_{i,\ell}, \mathsf{ct}^1_{i,\ell})$

– compute $C = \prod_{i=1}^{n} ((\mathsf{ct}^1_{i,\ell})^T)^{\boldsymbol{y}_i} \cdot (\mathsf{ct}^0_{i,\ell})^{-\mathsf{sk}_{y_i}} \cdot \mathcal{H}(\ell)^{-\mathsf{sk}}$

Return $\frac{C - 1 \bmod N^2}{N}$

Fig. 17: DCR-based MCFE (pos$^+$ secure)

Security extension (from pos$^+$ to any). Another limitation of the security definition (Theorem 2) is the reliance on the assumption that when the adversary makes a LRencryption query it has to complete the ciphertext. In this section for the simplicity, when there is no yy = one or yy = pos$^+$ restriction in the security definition we call it any-security. Abdalla et al. [ABKW19] and Chotard et al. [CDG+18b] have separately presented two compilers converting pos$^+$-secure-MCFE to any-secure-MCFE, the former has a square-size of the ciphertext[18]. While the latter is using pairing making it possible to achieve linear-size of the ciphertext and is based on Q-fold DBDH assumption. Note that both schemes are relying on the random oracle assumption which is what our MCFE construction is already involved with.

[18] Note that though [ABKW19] is presenting the compiler for DMCFE, it is easy to specific it for MCFE. Simply by unifying the algorithms KeyDerShare and KeyDerComb and replacing it with KeyDer algorithm.

Extension to DMCFE. We note that for the adaptive case during the complexity leveraging phase, for the second layer of MCFE (to compute $\sum s_i y_i$), one needs to guess just a scaler $\sum s_i y_i$. This comes from the fact that in this layer there is only one secret-key query (corresponding with vector $\mathbf{1}$). Thus, $\sigma' > \sqrt{\kappa + \log(1+N'^2)} \cdot N'^{5/2}$. For the correctness of the second layer, we need $N' > \sum s_i y_i$, this means $N' > Y \cdot \sum s_i$. Based on the Markov's inequality $\Pr[|s_i| \leq \sigma] \geq 1 - \mathsf{negl}(\kappa)$ if $\sigma = \Theta(\kappa^\epsilon)$ and $\epsilon > 0$. Then, $N' > Y \cdot n \cdot \sigma^2$ and since $Y < \sqrt{N/2L}$, one can set $N' > \sqrt{NL} \cdot \sigma^2$.

F.2 Extensions of our LWE-based MCFE

Extension to vectors per slot. For the sake of simplicity we proved the security when each client has a single scalar as it input. The construction can be simply extended to vectors-per-slot by considering $K = mnPV$ i.e, one should replace n with mn in the parameters setting.

Security extension (from one to pos$^+$). One can use a single-input FE and an MCFE both based on LWE assumption in the compiler of [CDG+18b] to get pos$^+$ security. The construction is depicted in Fig. 18.

$\underline{\mathsf{Setup}(1^\kappa, m, n.n_0, {n_0}')}$:

– set integers ${m_0}, {m_0}', q', q_0 \geq 2, q > q_0$, $K = mnPV, K' = mq_0V$ and $\alpha, \alpha' \in (0,1)$.

– let $\mathcal{H} : \mathsf{Labels} \to \mathbb{Z}_q^{n_0+m_0}$ be a full-domain hash function

Return $\mathsf{pp} = ({m_0}, {m_0}', q, q', \alpha, \alpha', P, V)$

$\underline{\mathsf{KeyGen}(\mathsf{pp})}$:

– sample $\mathbf{Z}_i = (s_i, t_i) \xleftarrow{R} \mathbb{Z}_q^{m \times n_0} \times \mathcal{D}_{\mathbb{Z}^{m \times m_0}, \alpha q}$

– sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{{m_0}' \times {n_0}'}$, and $\mathbf{Z}'_i \xleftarrow{R} \tau'$ where τ' is a special distribution over $\mathbb{Z}^{m \times {m_0}'}$.

Return $\mathsf{msk} = (\mathbf{Z}_i, \mathbf{Z}'_i)_{i \in [n]}$ and $\mathsf{sk}_i = (\mathbf{Z}'_i, \mathbf{Z}_i)$.

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \boldsymbol{x}_i, i, \ell)}$:

To encrypt $\boldsymbol{x}_i \in \{0, \ldots, P-1\}^m$:

– set $\overline{X}_i = \mathbf{Z}_i \cdot \mathcal{H}(\ell) + \lfloor \frac{q}{K} \rfloor \cdot \boldsymbol{x}_i \mod q$ s.t. $\overline{X}_i \in \{0, \ldots, q\}$

– set $X_i = \left\lfloor \overline{X}_i \right\rceil_{q_0} \in \{0, \ldots, q_0\}^m$

– sample $\boldsymbol{s}'_i \xleftarrow{R} \mathbb{Z}_{q'}^{{n_0}'}, \boldsymbol{e}'_i \xleftarrow{R} \mathcal{D}_{\mathbb{Z}^{{m_0}'}, \alpha' q'}$

– set $\mathsf{ct}^0_{i,\ell} = \mathbf{A} \cdot \boldsymbol{s}'_i + \boldsymbol{e}'_i \mod q'$

– compute $\mathsf{ct}^1_{i,\ell} = \lfloor \frac{q'}{K'} \rfloor X_i + \mathbf{Z}'_i \mathbf{A} \boldsymbol{s}'_i \mod q'$

Return $\mathsf{ct}_{i,\ell} = (\mathsf{ct}^0_{i,\ell}, \mathsf{ct}^1_{i,\ell})$

$\underline{\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \boldsymbol{y})}$:

For the vector $\boldsymbol{y}$ with $\boldsymbol{y} \in \{0, \ldots, V\}^m$:

– compute $\mathsf{sk}_y = \sum_{i \in [n]} y_i^T \cdot \mathbf{Z}_i$

– compute $\mathsf{sk}_{y_i} = y_i^T \mathbf{Z}'_i$.

Return $\mathsf{sk} = (\mathsf{sk}_y, \{\mathsf{sk}_{y_i}\}_{i \in [n]})$

$\underline{\mathsf{Dec}(\mathsf{pp}, \boldsymbol{y}, \mathsf{sk}, \{\mathsf{ct}_{i,\ell}\}_{i \in [n]}, \ell)}$:

– compute

– set $\mu_i = \boldsymbol{y}_i^T \mathsf{ct}^1_{i,\ell} - \langle \mathsf{sk}_{\boldsymbol{y}_i}, \mathsf{ct}^0_{i,\ell} \rangle \mod q'$

– find $\mu_i \in \{-K'+1, \ldots, K'-1\}$ that minimize $|\lfloor q'/K' \rfloor \cdot \mu_i - \mu'_i|$

$\mu' = \sum_{i \in [n]} \mu_i - \left\lfloor \langle \mathsf{sk}_y, \mathcal{H}(\ell) \rangle \right\rceil_{q_0} \mod q_0$

Return $\mu \in \{-K+1, \ldots, K-1\}$ that minimizes $|\frac{q_0}{q} \lfloor \frac{q}{K} \rfloor \cdot \mu - \mu'|$.

Fig. 18: MCFE based on LWE (pos$^+$ secure)