Multi-Input Functional Encryption: Efficient Applications From Symmetric Primitives (extended version)

Alexandros Bakas and Antonis Michalas

Tampere University of Technology,
Tampere, Finland
{alexandros.bakas, antonios.michalas}@tuni.fi

Abstract. Functional Encryption (FE) allows users who hold a specific secret key (known as the functional key) to learn a specific function of encrypted data whilst learning nothing about the content of the underlying data. Considering this functionality and the fact that the field of FE is still in its infancy, we sought a route to apply this potent tool to design efficient applications. To this end, we first built a symmetric FE scheme for the ℓ_1 norm of a vector space, which allows us to compute the sum of the components of an encrypted vector. Then, we utilized our construction, to design an Order-Revealing Encryption (ORE) scheme and a privately encrypted database. While there is room for improvement in our schemes, this work is among the first attempts that seek to utilize FE for the solution of practical problems that can have a tangible effect on people's daily lives.

Keywords: Differential Privacy · Functional Encryption · Order Revealing Encryption

1 Introduction

Functional Encryption (FE) is an emerging cryptographic technique that allows selective computations over encrypted data. FE schemes provide a key generation algorithm that outputs decryption keys with remarkable capabilities. More precisely, each decryption key sk_f is associated with a function f. In contrast to traditional cryptographic techniques, using sk_f on a ciphertext $\mathsf{Enc}(x)$ does *not* recover x but a function $f(x)$ – thus keeping the actual value x private. While the first constructions of FE allowed the computation of a function over a *single* ciphertext, more recent works [24] introduced the more general notion of multi-input FE (MIFE). In a MIFE scheme, given ciphertexts $\mathsf{Enc}(x_1), \ldots, \mathsf{Enc}(x_n)$, a user can use sk_f to recover $f(x_1, \ldots, x_n)$. The function f can allow only highly processed forms of data to learned by the functional key holder. Unfortunately, while MIFE seems to be a perfect fit for many real-life applications – especially cloud-based ones where multiple users store large volumes of data in remote and

possibly corrupted entities [33,34] – most of the works in the field revolve around constructing *generic* schemes that do not support specific functions. Hence, while the concept of FE has the potential to unleash new, creative, useful and emerging applications, it still holds a promise that remains *largely untapped* from a practical perspective. Having identified the importance of FE and believing that it is a family of modern encryption schemes that can push us into an uncharted technological terrain, we try to make a first attempt to smooth out the identified asymmetries between theory and practice. To do so, we first design a MIFE scheme for the ℓ_1 norm of a vector space based on [1]. Then, using our MIFE scheme we attempt a first approach in embedding FE into the problems of Order-Revealing-Encryption (ORE) [11] and Differentially Private Databases [32].

An ORE scheme simply requires that there exists a publicly computable function that compares two ciphertexts. ORE is an important cryptographic primitive in the study of symmetric searchable encryption because it allows for efficient range queries, sorting, and threshold filtering on encrypted data. An encryption scheme is called order-revealing if the comparison of two ciphertexts leaks *only* the ordering of the corresponding plaintexts and nothing else. Approaching the problem of ORE using MIFE is *not* a novel idea. In [11], authors proposed a MIFE-based ORE construction. However, their scheme relies on multi-linear maps [12] and thus, is impractical for real-life applications. While subsequent designs in the filed of ORE [28] are significantly more efficient, they do not rely on functional encryption. With this in mind, we sought to design the first order-revealing encryption scheme based solely on MIFE from symmetric primitives. Continuing our research into proving that FE can be a fundamental tool for a slew of applications, we turned our attention to the important problem of differential privacy [19]. While cryptography ensures the confidentiality of the data stored in a database, it does not ensure the privacy of the users. This is one of the big hurdles that security researchers as well as big industrial players have wrestled with for decades [5,26]. While there are numerous works that deal with either the problem of private databases [21], or that of encrypted databases [20], there are only a handful [3] that focus on combining differential privacy with cryptography to construct privately encrypted databases. This fact motivated us to attempt a first approach towards a multi-input functionally encrypted private database. In particular, we design an application in which a data owner (curator) encrypts her data locally using MIFE, before outsourcing them to the cloud. Then, we allow a *possibly malicious* analyst to query the database for various functions of her choice (i.e. sum of the entries, averages, etc.). To make our construction private, we use the well-studied *Laplace Mechanism* [22].

1.1 Contribution

The contribution of this paper is twofold: *(1)* First, we design a MIFE scheme in the symmetric key setting for the ℓ_1 norm of a vector, based on the single-client MIFE for inner products presented in [1]. Then, we show how our scheme can be transformed from the single-client to the multi-client setting. This transformation requires the users to perform a Multi-Party Computation (MPC). More

precisely, each user generates their own symmetric keys independently and then they collaborate to calculate a functional decryption key sk_f that is derived from a combination of all the generated symmetric keys. This result is quite remarkable since users generate their private keys locally and independently. As a result, their symmetric keys are never exposed to unauthorized parties, and thus no private information about the content of the underlying ciphertexts is revealed. At the same time, sufficient information to generate the functional decryption key is provided. *(2)* Our second contribution derives from the identified need to create a dialogue between the theoretical concept of FE and real life applications. As a result, we tried to provide a pathway towards new prospects that show the direct and realistic applicability of this promising encryption technique when applied to concrete obstacles. To this end, we showed how our MIFE scheme can be used to provide a solution to two disjointed but important problems. More precisely, we show how our scheme can be used to provide realistic solutions to the problems of Order-Revealing-Encryption [11] and Differentially Private Databases [3].

1. First, we design a MIFE scheme in the symmetric key setting for the ℓ_1 norm of a vector, based on the single-client MIFE for inner products presented in [1]. Then, we show how our scheme can be transformed from the single-client to the multi-client setting. This transformation requires the users to perform a Multi-Party Computation (MPC). More precisely, each user generates their own symmetric keys independently and then they collaborate to calculate a functional decryption key sk_f that is derived from a combination of all the generated symmetric keys. This result is quite remarkable since users generate their private keys locally and independently. As a result, their symmetric keys are never exposed to unauthorized parties, and thus no private information about the content of the underlying ciphertexts is revealed. At the same time, sufficient information to generate the functional decryption key is provided.
2. Our second contribution derives from the identified need to create a dialogue between the theoretical concept of FE and real life applications. As a result, we tried to provide a pathway towards new prospects that show the direct and realistic applicability of this promising encryption technique when applied to concrete obstacles. To this end, we showed how our MIFE scheme can be used to provide a solution to two disjointed but important problems. More precisely, we show how our scheme can be used to provide realistic solutions to the problems of Order-Revealing-Encryption [11], an encryption technique closely related to the problem of searching on ecnrypted data [6,8,18] and Differentially Private Databases [3].

Order Revealing Encryption: ORE [11] is an encryption technique that allows direct comparison of two ciphertexts, yielding the ordering of the corresponding plaintexts. This is done by a using a comparison functions CMP. This is done with the help of a comparison function CMP defined as follows:

$$\text{CMP}(x_i, x_j) = \begin{cases} 1 & \text{if } x_i > x_j \\ 0 & \text{if } x_i = x_j \\ -1 & \text{if } x_i < x_j \end{cases}$$

In our case, the CMP function takes as input two ciphertexts encrypted under our MIFE scheme and, with the help of a secret functional key sk_f, outputs the result $r \in \{-1, 0, 1\}$ depending on the ordering of the corresponding plaintexts. Our first ORE scheme, MORE, makes direct use of our MIFE scheme for the ℓ_1 norm resulting to a very efficient and straight forward solution. However, MORE is susceptible to the inference attacks presented in [32]. To this end, we also present TWOMORE – a two-layered encryption ordering revealing scheme that is based on MORE and is susceptible to the attacks presented in [32]. More precisely, we prove that TWOMORE satisfies the *best-possible* security for an ORE scheme as defined in [9] which states that the ciphertexts do not leak any information beyond the ordering of the plaintexts.

Privately Encrypted Databases: We design a Privately Encrypted Database similar to the one presented in [3]. The main difference between the two works, is that in [3], authors use as a basis Structured Encryption [15], which can be seen as a generalization of Symmetric Searchable Encryption [37] while we solely base our construction on our MIFE scheme for the ℓ_1 norm. Moreover, we choose to use the Laplace Mechanism [22] as a privacy mechanism. The reason for choosing the Laplace Mechanism is that it is defined using the ℓ_1 norm of a database (i.e. the sum of all entries) and thus, along with our MIFE scheme, they compliment each other very well. Finally, we prove the security of our construction assuming the existence of a *statistical* adversary.

1.2 Organization

In Section 2, we present well-established works in the area of Functional Encryption and Multi-Input Functional Encryption. In Section 3 we provide the basic notation and definitions that are used throughout the paper. In Section 4, we present the first result of this paper, a multi-client MIFE scheme for the ℓ_1 norm in the symmetric key setting. Then, we show how our MIFE scheme ban be leveraged to design an order revealing encryption scheme (Section 5) and a privately encrypted database (Section 6) In Section 7 we discuss future directions for the applications presented in this paper and finally, Section 8 concludes the paper.

2 Related Work

While numerous studies with general definitions and generic constructions of FE have been proposed [35,27,38,25,14] there is a clear lack of work proposing FE schemes supporting specific functions – a necessary step that would allow

FE to transcend its limitations and provide the foundations for reaching its full potential. To the best of our knowledge, the only works that have shown how to efficiently run specific functions on ciphertexts is [1,2] which calculates inner-product and [36] which successfully executes computations with quadratic polynomials. While [1] and [14] are symmetric FE schemes (i.e. efficient), their actual application in real-life scenarios can be considered as limited since both are limited to supporting the single-client model. Our work is heavily influenced by the symmetric key MIFE scheme for inner products presented in [1] where authors designed a scheme that can be regarded as the FE equivalent of the one-time-pad. More precisely, using [1] as a basis, we constructed a symmetric key MIFE scheme for the ℓ_1 norm of an arbitrary vector space. Most importantly, we show that our construction can also support the multi-client model while preserving exactly the same security properties as the MIFE for inner-product in [1]. This is a significant result as it proves that functional encryption can be efficiently applied to solve more complex problems.

3 Preliminaries

Notation If $\mathcal{Y}$ is a set, we use $y \xleftarrow{\$} \mathcal{Y}$ if y is chosen uniformly at random from $\mathcal{Y}$. The cardinality of a set $\mathcal{Y}$ is denoted by $|\mathcal{Y}|$. If $\mathcal{X}$ and $\mathcal{Y}$ are two sets, we denote by $[\mathcal{X}, \mathcal{Y}]$ all the functions from $\mathcal{X}$ to $\mathcal{Y}$ and by $\overline{[\mathcal{X}, \mathcal{Y}]}$ all the injective functions from $\mathcal{X}$ to $\mathcal{Y}$. For a positive integer m, $[m]$ denotes the set $\{1, \ldots, m\}$. If $m \in \mathbb{Z}$, we denote by $m[i]$ the digit in the i-th position of m where $m[0]$ is the rightmost digit. The number of digits of m in base n is $\lfloor log_n m \rfloor + 1$. Vectors are denoted in bold as $\mathbf{x} = [x_1, \ldots, x_n]$. A probabilistic polynomial time (PPT) adversary $\mathcal{ADV}$ is a randomized algorithm for which there exists a polynomial $p(z)$ such that for all input z, the running time of $\mathcal{ADV}(z)$ is bounded by $p(|z|)$. A database is denoted by DB, an encrypted database by EDB and a private encrypted database by PEDB. An invertible pseudorandom function (IPRF) [10] is defined as follows:

Definition 1 (Invertible Pseudorandom Function (IPRF)). *An IRPF with key-space $\mathcal{K}$, domain of definition $\mathcal{X}$ and range $\mathcal{Y}$ consists of two functions $G : (\mathcal{K} \times \mathcal{X}) \to \mathcal{Y}$ and $G^{-1} : (\mathcal{K} \times \mathcal{Y}) \to \mathcal{X} \cup \{\perp\}$. Moreover, let $\mathsf{G.Gen}(1^\lambda)$ be an algorithm that given the security parameter λ, outputs $k \in \mathcal{K}$. The functions G and G^{-1} satisfy the following properties:*

1. $G^{-1}(k, G(k, x)) = x, \forall x \in \mathcal{X}$.
2. $G^{-1}(k, y) = \perp$ *if y is not an image of G.*
3. *G and G^{-1} can be efficiently computed by deterministic polynomial algorithms.*
4. $G(k, \cdot) \in \overline{[\mathcal{X}, \mathcal{Y}]}, G^{-1}(k, \cdot) \in \overline{[\mathcal{Y}, \mathcal{X}]}$

The function $G : (\mathcal{K} \times \mathcal{X}) \to \mathcal{Y}$ is an IPRF if $\forall$ PPT adversary $\mathcal{A}$:

$$|Pr[k \leftarrow \mathsf{G.Gen}(1^\lambda) : \mathcal{A}^{G(k,\cdot), G^{-1}(k,\cdot)}(1^\lambda) = 1] - Pr[k' \xleftarrow{\$} \overline{[\mathcal{X}, \mathcal{Y}]} : \mathcal{A}^{R(\cdot), R^{-1}(\cdot)}(1^\lambda) = 1| = negl(\lambda) \quad (1)$$

Definition 2 (ℓ_1 norm). *The ℓ_1 norm of $\mathbb{Z}^n$ is a function $\|\cdot\|_1$ defined by:*

$$f(x) = \|\mathbf{x}\|_1 = \sum_{i=1}^{i=n} x_i = x_1 + \ldots + x_n, \text{ for } \mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{Z}^n$$

3.1 Multi-Input Functional Encryption

Definition 3 (Multi-Input Functional Encryption in the Symmetric Key Setting). *Let $\mathcal{F} = \{f_1, \ldots, f_n\}$ be a family of n-ary functions where each f_i is defined as follows: $f_i : \mathbb{Z}^n \to \mathbb{Z}$. A multi-input functional encryption scheme for $\mathcal{F}$ consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda)$: *Takes as input a security parameter λ and outputs a secret key $\mathbf{K} = [\mathsf{k}_1, \ldots, \mathsf{k}_n] \in \mathbb{Z}^n$.*
- $\mathsf{Enc}(\mathsf{K}, i, x_i)$: *Takes as input $\mathbf{K}$, an index $i \in [n]$ and a message $x_i \in \mathbb{Z}$ and outputs a ciphertext ct_i.*
- $\mathsf{KeyGen}(\mathbf{K}, f)$: *Takes as input $\mathbf{K}$ and a description of a function f_i and outputs a functional decryption key sk_{f_i}.*
- $\mathsf{Dec}(\mathsf{sk}_{f_i}, ct_1, \ldots, ct_n)$: *Takes as input a decryption key sk_{f_i} for a function f_i and n ciphertexts and outputs a value $y \in \mathbb{Z}$.*

For the needs of our work, we will draw on the one-adaptive (one-AD) and one-selective (one-SEL) security definitions from [1] that were first formalized in [4]. Informally, in the one-AD-IND security game, the adversary $\mathcal{ADV}$ receives the encryption key of the MIFE scheme and then adaptively queries the corresponding oracle for functional decryption keys of her choice. Furthermore, $\mathcal{ADV}$ outputs two messages x_0 and x_1 to the encryption oracle, who flips a random coin and outputs an encryption of $x_\beta, \beta \in \{0, 1\}$. If the functional keys are associated with functions that do not distinguish between the messages ($i.e. f(x_0) = f(x_1)$) then $\mathcal{ADV}$ should *not* be able to distinguish between the encryptions of x_0 and x_1. In the case of the one-SEL-IND security, the game is identical to the one-AD-IND case, with the only difference being that $\mathcal{ADV}$ needs to decide on the x_0 and x_1 messages *before* seeing the encryption key. The *"one"* in both security games determines that the encryption oracle can only be queried once for each slot i (i.e. the adversary is not allowed to issue multiple queries to the encryption oracle for the same x_i).

Definition 4 (one-AD-IND-secure MIFE).

For every MIFE scheme for $\mathcal{F}$, every PPT adversary $\mathcal{ADV}$, every security parameter $\lambda \in \mathbb{N}$ we define the following experiment for $\beta \in \{0, 1\}$:

Adaptive Security

one-AD-IND$_\beta^{MIFE}(1^\lambda, \mathcal{ADV})$:
$\mathbf{K} \leftarrow \mathsf{Setup}(1^\lambda)$
$\alpha \leftarrow \mathcal{ADV}^{\mathsf{KeyGen}(\mathbf{K}), \mathsf{Enc}(\cdot,\cdot,\cdot)}$
Output α

Where $\mathsf{Enc}(\cdot,\cdot,\cdot)$ *is an oracle that on input* (i, x_i^0, x_i^1)*, flips a random coin* β *and outputs* $\mathsf{Enc}(\mathsf{K}, i, x_i^\beta), \beta \in \{0,1\}$*. Moreover,* $\mathcal{ADV}$ *is restricted to only make queries to the* KeyGen *oracle satisfying* $f(x_1^0, \ldots, x_n^0) = f(x_1^1, \ldots, x_n^1)$*. A MIFE scheme is said to be one-AD-IND secure if for all PPT adversaries* $\mathcal{ADV}$*, their advantage is negligible in* λ *where the advantage is defined as:*

$$Adv^{one-AD-IND}(\lambda, \mathcal{ADV}) = |Pr[one\text{-}AD\text{-}IND_0^{MIFE}(1^\lambda, \mathcal{ADV}) = 1] - Pr[one\text{-}AD\text{-}IND_1^{MIFE}(1^\lambda, \mathcal{ADV}) = 1]|$$

Definition 5 (one-SEL-IND-secure MIFE).

For every MIFE scheme for $\mathcal{F}$*, every PPT adversary* $\mathcal{ADV}$*, and every security parameter* $\lambda \in \mathbb{N}$ *we define the following experiment for* $\beta \in \{0,1\}$*:*

Selective Security

$one\text{-}SEL\text{-}IND_\beta^{MIFE}(1^\lambda, \mathcal{ADV})$:
$\{x_i^b\}_{i\in[n], b\in\{0,1\}} \leftarrow \mathcal{ADV}(1^\lambda, f_i)$
$\mathbf{K} \leftarrow \mathsf{Setup}(1^\lambda)$
$ct_i = \mathsf{Enc}(\mathbf{K}, x_i^\beta)$
$\alpha \leftarrow \mathcal{ADV}^{\mathsf{KeyGen}(\mathbf{K})}(\{ct_i\})$
Output α

$\mathcal{ADV}$ *is restricted to only make queries to the* KeyGen *oracle satisfying* $f(x_1^0, \ldots, x_n^0) = f(x_1^1, \ldots, x_n^1)$*. A MIFE scheme is said to be one-SEL-IND secure if for all PPT adversaries* $\mathcal{ADV}$*, their advantage is negligible in* λ *where the advantage is defined as:*

$$Adv^{one-SEL-IND}(\lambda, \mathcal{ADV}) = |Pr[one\text{-}SEL\text{-}IND_0^{MIFE}(1^\lambda, \mathcal{ADV}) = 1] - Pr[one\text{-}SEL\text{-}IND_1^{MIFE}(1^\lambda, \mathcal{ADV}) = 1]|$$

4 Multi-Input Functional Encryption for the ℓ_1 Norm

In this section, we present the first result and an important contribution of our work. In particular, in the first part of this section, we show how the one-AD-IND-secure MIFE scheme for inner-products from [1], can be transformed to a one-AD-IND-secure MIFE scheme for the ℓ_1 norm (MIFE_{ℓ_1}), while preserving exactly the same security properties. Then, we show how we can transform our construction from the single-client model to the multi-client one. For purposes of completeness, we briefly recall the one-AD-IND-secure MIFE scheme for inner-products in Figure 1. The security of both MIFE schemes (inner products and ℓ_1 norm), is derived from the fact that they behave as the functional encryption equivalent of the one-time-pad. Note that, just like in the case of the one-time-pad, to achieve perfect secrecy, we require that $|\mathsf{k_i}| \geq |x_i|$, where k_i is the encryption key and x_i, the message to be encrypted.

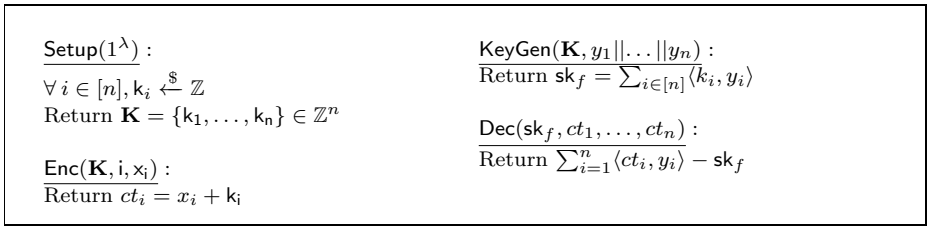

Fig. 1: one-AD-IND-secure MIFE for inner products.

In the previous scheme, by fixing $\mathbf{y}$ to be $\mathbf{y} = [1, \ldots, 1]$, we compute $\langle \mathbf{x}, 1 \rangle = \|\mathbf{x}\|_1$ for $\mathbf{x} \in \mathbb{Z}^n$. By doing so, we manage to transform the original inner products MIFE to a new construct that successfully computes the ℓ_1 norm. Our construction is illustrated in Figure 2. Our construction, illlustrated in Figure 2 is a special case of the scheme presented in [1] and hence, it is straight forward that the security proofs of the two schemes will be similar. By fixing y to be $y = [1, \ldots, 1]$, we compute $\langle x, 1 \rangle = \|x\|_1$ for $x \in \mathbb{Z}^n$ and thus, transform the MIFE for inner products to MIFE for the ℓ_1 norm. Our construction is illustrated in Figure 2. Since our construction is a special case of the scheme in Figure 1, it is straight forward that the security proof of our scheme will be very similar to the one presented in [1].

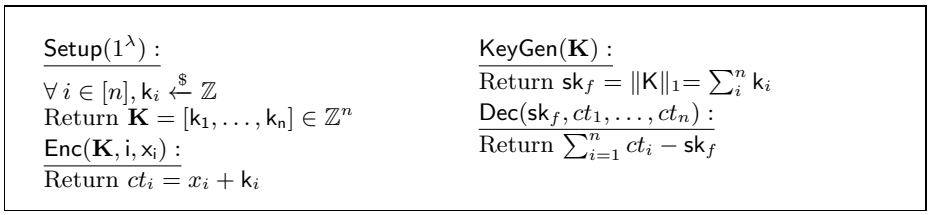

Fig. 2: one-AD-IND-secure MIFE for the ℓ_1 norm (MIFE_{ℓ_1}).

Theorem 1. *The MIFE scheme for the ℓ_1 norm (described in Figure 2) is one-AD-IND-secure. That is, for all PPT adversaries $\mathcal{ADV}$:*

$$Adv_{\mathcal{ADV}}^{one-AD-IND}(\lambda) = 0$$

Proof. The proof consists of two parts. First we construct a selective distinguisher $\mathcal{B}$ whose advantage for the one-SEL-IND experiment is an upper bound for the advantage of any adaptive distinguisher $\mathcal{ADV}$. Then, using the fact that the MIFE for the ℓ_1 norm behaves like the one-time-pad, we prove that the advantage of $\mathcal{B}$ is zero.

For the first part of the proof we will use a complexity argument. In particular, let $\mathcal{B}$ be an adversary that guesses the challenge $\{x_i^b\}$ and then simulates

the one-AD-IND experiment of $\mathcal{ADV}$. If $\mathcal{B}$ successfully guesses $\mathcal{ADV}$'s challenge then she can simulate $\mathcal{ADV}$'s view. Otherwise it outputs $\perp$. Hence, $\mathcal{ADV}$'s advantage maximizes when $\mathcal{B}$ guesses correctly the challenge. If the input space is $\mathcal{X}$, then $\mathcal{B}$ can guess successfully with probability exactly $|\mathcal{X}|^{-1}$. Hence:

$$Adv_{\mathcal{ADV}}^{one-AD-IND} \leq |X|^{-1} Adv_{\mathcal{B}}^{one-SEL-IND}$$

From the above, it can be seen that if the input space $\mathcal{X}$ is very large, the advantage of $\mathcal{ADV}$ tends to zero independently of the value of $Adv_{\mathcal{B}}^{one-SEL-IND}$ $\big(i.e.\ |\mathcal{X}| \to \infty \Rightarrow Adv_{\mathcal{ADV}}^{one-AD-IND} \to 0\big)$. We will still show that no matter the cardinality of $\mathcal{X}$, $Adv_{\mathcal{ADV}}^{one-AD-IND} = 0$. To do so, we first need to prove that the advantage of the selective adversary is zero, or $Adv_{\mathcal{B}}^{one-SEL-IND} = 0$. This will directly imply that $Adv_{\mathcal{ADV}}^{one-AD-IND} = 0$, since we already know that $Adv_{\mathcal{ADV}}^{one-AD-IND} \leq Adv_{\mathcal{B}}^{one-SEL-IND}$. In Figure 3 we present a hybrid game that is identical to the one-SEL-IND security game. This is derived from the fact that if $u \xleftarrow{\$} \mathbb{Z}$, then $\{u_i\}$ and $\{u_i - x_i^{\beta}\}$ are identical distributions. It is easy to see that the only information leaking about β, is $\|\mathbf{r} - \mathbf{x}^{\beta}\|$, which is independent of β according to the definition of the security game and the restrictions of the adversary.

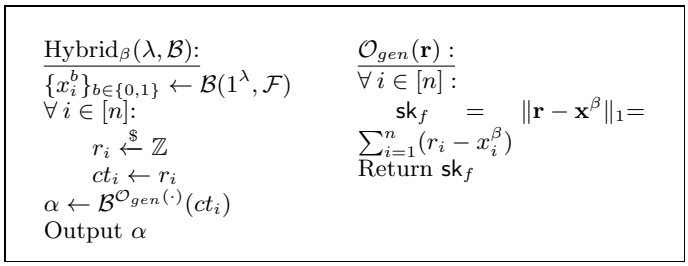

Fig. 3: Hybrid games for the proof of Theorem 1

While we showed that a MIFE scheme for inner products can be transformed to a MIFE scheme for the ℓ_1 norm, our construction is still inadequate for an e-voting protocol. This is due to the the fact that our construction only supports the single-client model. In Section 4.1, we show how our single client MIFE can be extended to support the multi-client model.

4.1 From Single-Client to Multi-Client MIFE

We are now ready to describe how we can transform our single-user MIFE_{ℓ_1} to the multi-user MIFE for the ℓ_1 norm (MUMIFE_{ℓ_1}). The idea is the following: Each user generates a symmetric key $\mathsf{k}_i \in \mathbb{Z}$ which uses it to encrypt a plaintext x_i as $ct_i = \mathsf{k}_i + x_i$. All the generated symmetric keys, form a vector

$\mathbf{K} = [\mathsf{k_1}, \ldots, \mathsf{k_n}] \in \mathbb{Z}^n$, where n is the number of users. The functional decryption key sk_f is then $\|\mathbf{K}\|_1$ and decryption works as follows:

$$\sum_{i=1}^{n} ct_i - \mathsf{sk}_f = \sum_{1}^{n}(\mathsf{k}_i + x_i) - \sum_{1}^{n} \mathsf{k}_i = \sum_{1}^{n} x_i = \|\mathbf{x}\|_1$$

A third party decryptor who would get access to sk_f should only learn $\|\mathbf{x}\|_1$ and *not* each individual x_i. In addition to that, the users should never reveal their symmetric keys. To achieve this, we assume the existence of a trusted authority that will allow users to perform an MPC in order to jointly compute a masked version of sk_f without revealing each distinct k_i. Our construction is presented in Figure 4.

In our system model, we assume the existence of a Trusted Execution Environment (TEE) that will allow the TA to run in a trusted state. Hence, the existence of a trusted entity relies on more realistic assumptions.

Trusted Execution Environments: A TEE is a secure, integrity-protected environment, with processing, memory and storage capabilities, isolated from an untrusted, Rich Execution Environment that comprises the OS and installed applications. While there are several different TEEs in our work we rely on the use of Intel SGX whose main functionalities are *(1)* Isolation, *(2)* Sealing and *(3)* Attestation. Due to space constraints, we ommit their formal description (more details can be found in [17]).

Trusted Authority (TA) TA is running in an enclave and is responsible for generating and distributing a unique random number s_i to each user u_i. The users will use the received random values to mask their symmetric keys. By doing so, and considering the fact that TA is running in an enclave and thus it is trusted, they will be able to jointly compute a masked version of the functional decryption key sk_f which will be used by the evaluator to calculate sk_f .

Evaluator (EV) EV is an untrusted entity responsible for collecting users' ciphertexts $\{ct_1, \ldots, ct_n\}$, generating the functional decryption key sk_f based on the masked value that will receive from users and finally, calculate $f(x_1, \ldots, x_n)$ without learning any information about each individual x_i.

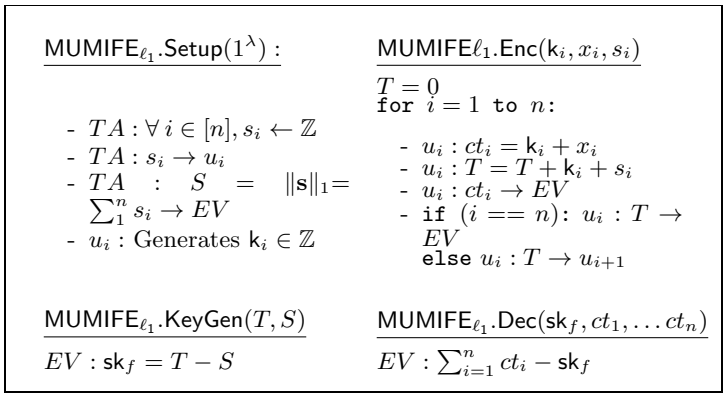

Fig. 4: Multi-Input MIFE for the ℓ_1 norm (MUMIFE$_{\ell_1}$)

Correctness: The correctness of the MUMIFE$_{\ell_1}$ scheme presented in Figure 4 follows directly since:

$$\sum_{i=1}^{n} ct_i - \mathsf{sk}_f = \sum_{i=1}^{n} ct_i - T + S =$$
$$\sum_{i=1}^{n} \mathsf{k}_i + \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} (\mathsf{k}_i + s_i) + \sum_{i=1}^{n} s_i = \|\mathbf{x}\|_1$$

Theorem 2. *The Multi-User Multi-Input Functional Encryption scheme for the ℓ_1 norm (described in Figure 4) is one-AD-IND-secure. That is, for all PPT adversaries $\mathcal{ADV}$:*

$$Adv_{\mathcal{ADV}}^{one-AD-IND}(\lambda) = 0$$

Proof (Proof Sketch.). The proof is omitted since it is a direct result from Theorem 1. This can be seen by the fact that the Encryption and KeyGen oracles are identical to the ones described in Figure 3. The only difference is that in the case of MUMIFE$_{\ell_1}$, the Setup algorithm is executed by multiple users instead of one, since each user generates a distinct symmetric key. Without loss of generality, we can assume that this is exactly the same procedure since in the case of MIFE$_{\ell_1}$, one user samples n random numbers from $\mathbb{Z}$ resulting to a vector $\mathbf{K} = [\mathsf{k}_1, \ldots, \mathsf{k}_n]$, and in case of MUMIFE$_{\ell_1}$, n users sample one random number from $\mathbb{Z}$ each, resulting to a vector $\mathbf{K'} = [\mathsf{k}'_1, \ldots, \mathsf{k}'_n]$. However, the distributions $\{\mathsf{k}_i\}$ and $\{\mathsf{k}'_i\}$ are identical and thus we conclude that we can use exactly the same Hybrid game as the one in Figure 3.

5 Order Revealing Encryption

We now show how our MIFE$_{\ell_1}$ scheme can be utilized to perform Order Revealing Encryption (ORE). Informally, ORE is an encryption technique that allows

direct comparison of two ciphertexts yielding the order of the plaintexts. The notion was first introduce in [11] where a construction based on multi-linear maps was presented. However fascinating, such an approach is impractical for realistic scenarios. An improved ORE scheme was presented in [28] where each query requires time linear to the total number of ciphertexts. Our MORE scheme is formally presented in Figure 5.

In this section, we present MORE – a Multi-Input Functional Order Revealing Encryption scheme that takes as input two encrypted numbers and returns their ordering. MORE is solely based on the MIFE_{ℓ_1} presented in Section 4.

Definition 6 (MORE). *MORE is a two party scheme between a curator (data owner) and a CSP that consists of three polynomial-time protocols that work as follows:*

- $EDB \leftarrow \mathsf{Setup}(1^\lambda, DB)$*: The Data Owner (or Curator) gives as input the security parameter* λ *and a sequence of plaintexts. The CSP receives a sequence of ciphertexts.*
- $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{k}_i, \mathsf{k}_j)$*: The curator gives as input two encryption keys* k_i *and* k_j *and receives* sk_f*.*
- $(r \in \{-1, 0, 1\}) \leftarrow \mathsf{Query}(ct_i, ct_j, \mathsf{sk}_f)$*: The curator gives as input two ciphertexts and a functional key* sk_f*. The server performs the comparison of the ciphertexts and returns 1 if* $c_i > c_j$*, -1 if* $c_i < c_j$ *and 0 if* $c_i = c_j$*.*

Our MORE scheme is formally presented in Figure 5.

$\underline{\mathsf{MORE.Setup}(1^\lambda, DB):}$
Curator:
$\forall\, x_i \in DB$ Run:
- $\mathsf{MIFE}_{\ell_1}.\mathsf{Setup}(1^\lambda)$
- $\mathsf{MIFE}_{\ell_1}.\mathsf{Enc}(\mathbf{K}, i, x_i)$

$\underline{\mathsf{MORE.KeyGen}(ct_i, ct_j)}$
Curator:
- Run $\mathsf{MIFE.KeyGen}(ct_i, ct_j)$
- Return $\mathsf{sk}_f = \mathsf{k}_j - \mathsf{k}_i$

$\underline{\mathsf{MORE.Query}(\mathsf{sk}_f, ct_i, ct_j)}$
Curator:
- Send $(\mathsf{sk}_f, ct_i, ct_j)$ to the CSP.

CSP:
- Run $\mathsf{MIFE.Dec}(\mathsf{sk_f}, \mathsf{ct_i}, \mathsf{ct_j})$
  Return $r \in \{-1, 0, 1\}$

Fig. 5: Order Revealing Encryption based on MIFE_{ℓ_1}

Correctness: The correctness MORE scheme follows since:

$$\begin{aligned} ct_i - ct_j + \mathsf{sk}_f &= (x_i + \mathsf{k}_i) - (x_j + \mathsf{k}_j) + \mathsf{sk}_f \\ &= x_i + \cancel{\mathsf{k}_i} - x_j - \cancel{\mathsf{k}_j} + \cancel{\mathsf{k}_j} - \cancel{\mathsf{k}_i} \\ &= x_i - x_j \end{aligned} \tag{2}$$

Theorem 3. *The MORE scheme described in Figure 5 is one-AD-IND-secure.*

Proof is omitted as it is a direct result of Theorem 1.

While our construction may be simplistic, it is susceptible to the inference attacks presented by Naveed et al. in [32]. These attacks rely heavily on the fact that Order Preserving encrypted ciphertexts enable equality comparisons by design. In the next subsection, we present an extension of the MORE scheme that is secure against these attacks.

5.1 Two-Layered Encryption

We start by describing the core security properties that our enhanced scheme has to fulfil. First we give the definition of *best-possible* security for an ORE scheme [9].

Definition 7 (Best-Possible Security). *An ORE scheme satisfies the notion of best-possible security with respect to a leakage function $\mathcal{L}_{CMP}$, if $\mathcal{L}_{CMP}$ only reveals the ordering of the plaintexts and nothing else. In particular:*

$$\mathcal{L}_{CMP}(ct_1, \ldots, ct_n) = \{i, j, \mathrm{CMP}(x_i, x_j)\} \tag{3}$$

Where $\mathrm{CMP}(x_i, x_j)$ *is a comparison function defined as:*

$$\mathrm{CMP}(x_i, x_j) = \begin{cases} 1 & \text{if } x_i > x_j \\ 0 & \text{if } x_i = x_j \\ -1 & \text{if } x_i < x_j \end{cases}$$

Definition 8 (TWOMORE with Leakage).

Let TWOMORE $= (\mathsf{IPRFSetup}, \mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Query})$ *be an ORE scheme, and let $\mathcal{A}_q$ be an adversary for some $q = poly(n)$. Moreover, let $\mathcal{S}$ be a simulator and $\mathcal{L}(\cdot)$ a leakage function. We define the following experiments:*

Real Experiment

$\mathsf{K}_G \leftarrow \mathrm{TWOMORE}.\mathsf{IPRFSetup}(1^{\check{}})$
$(i, j) \leftarrow \mathcal{A}_1(1^\lambda)$
$r \leftarrow \mathrm{TWOMORE}(\mathsf{sk}_f, ct_i^{(2)}, ct_j^2)$
Output a bit b indicating whether this is the real or the ideal experiment

Ideal Experiment

$st_{\mathcal{S}} \leftarrow \mathcal{S}(1^\lambda)$
$(i, j) \leftarrow \mathbb{A}_1(1^\lambda)$
$r \leftarrow \mathrm{TWOMORE}(st_{\mathcal{S}}, \mathcal{L}_{CMP}(ct_1^{(2)}, \ldots, ct_N^{(2)}))$
Output a bit b indicating whether this is the real or the ideal experiment

We say that the ORE scheme is secure with repsect to the best-possible leakage function $\mathcal{L}_{CMP}$ if and only if no PPT adversary $\mathcal{ADV}$ can distinguish between the real and ideal experiments.

Definition 9 (TWOMORE). *TWOMORE is a two party protocol between a curator and a CSP that consists of four polynomial-time protocols that work as follows:*

- $\mathsf{K}_G \leftarrow \mathsf{IPRFSetup}(1^\lambda)$*: The Curator gives as input a security parameter* λ *and receives a key* K_G *for an IPRF* G*.*
- $EDB \leftarrow \mathsf{Setup}(1^\lambda, DB)$*: The curator gives as input the security parameter* λ *and a sequence of plaintexts. The CSP receives a sequence of ciphertexts.*
- $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{k}_i, \mathsf{k}_j)$*: The curator gives as input two encryption keys* k_i *and* k_j *and receives a functional key* sk_f*.*
- $(r \in \mathbb{Z}) \leftarrow \mathsf{Query}(ct_i, ct_j, \mathsf{sk}_f)$*: The curator gives as input two ciphertexts and a functional key* sk_f*. The server performs the comparison of the ciphertexts and returns* $r = c_j - c_i$*.*

A TWOMORE scheme is used as follows. First the curator generates a key K_G for the IPRF G. Then she runs the $\mathsf{MIFE}_{\ell_1}.\mathsf{Enc}$ for all $x_i \in DB$ to get the first layer ciphertexts $ct_i^{(1)}$. After she receives the corresponding ciphertexts, she applies G on each $ct_i^{(1)}$ to get the second level ciphertexts $ct_i^{(2)}$. Finally, she sends the encrypted database EDB to the CSP. To hide the exact difference of the two compared numbers, the curator can simply send a functional key sk_f that is off by a certain value. Then, after she receives r from the CSP, she can add that value to r to find the correct result. A detailed construction is illustrated in Figure 6.

Correctness: The correctness of the MORE scheme follows from the fact that:

$$G^{-1}\left(\mathsf{K}_G, G\left(\mathsf{K}_G, ct_i^{(1)}\right), G\left(\mathsf{K}_G, ct_j^{(1)}\right)\right) = \left(ct_i^{(1)}, ct_j^{(2)}\right)$$

And then directly from the correctness of the MORE construction.

5.2 Security Analysis

We will now prove that our TWOMORE satisfies the *best-possible* security property. Like in similar works [28,16,11], we assume that the CSP is honest but curious. In particular, we will show that we can build a simulator $\mathcal{S}$ which given as input only the leakage function $\mathcal{L}_{CMP}$, it can simulate the real functionalities of TWOMORE.

Theorem 4. *TWOMORE is secure with the best-possible leakage function* $\mathcal{L}_{CMP}$ *assuming that* G *is a secure IPRF - modelled as a truly random function.*

We start by describing the simulator $\mathcal{S}$. Since $\mathcal{S}$ has access to $\mathcal{L}_{CMP}$, it knows the ordering of the plaintexts. To reply consistently to $\mathcal{A}$'s queries, $\mathcal{S}$ maintains a dictionary Dict, where it stores the functional keys sk_f for each $\{i, j\}$ pair that $\mathcal{A}$ queries, along with the $\{i, j\}$ pair.

To prove Theorem 4 we make use of a Hybrid Argument. **Hybrid 0:** This is the real experiment.

Let $G : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a secure IPRF.

$\underline{\mathsf{TWOMORE.IPRFSetup}(1^\lambda)}$

- $\mathsf{K}_G \leftarrow \{0,1\}^\lambda$

$\underline{\mathsf{TWOMORE.Setup}(1^\lambda, DB) :}$
Curator:
$\forall\, x_i \in DB$ Run:

- $\mathbf{K} \leftarrow \mathsf{MIFE}_{\ell_1}.\mathsf{Setup}(1^\lambda)$
- $ct_i^{(1)} \leftarrow \mathsf{MIFE}_{\ell_1}.\mathsf{Enc}(\mathbf{K}, i, x_i)$
- $ct_i^{(2)} \leftarrow G(\mathsf{K}_G, ct_i^{(1)})$

$\underline{\mathsf{MORE.KeyGen}(ct_i, ct_j)}$
Curator:

- Run $\mathsf{MIFE.KeyGen}(\mathbf{K}, \{i, j\})$
- Return $\mathsf{sk}_f = \mathsf{k}_j - \mathsf{k}_i$

$\underline{\mathsf{MORE.Query}(\mathsf{sk}_f, ct_i^{(2)}, ct_j^{(2)})}$
Curator:

- Send $(\mathsf{sk}_f, ct_i^{(2)}, ct_j^{(2)})$ to the CSP.

CSP:

- Run
  $\mathsf{MIFE.Dec}\left(\mathsf{sk}_f, G^{-1}\left(\mathsf{K}_G, G\left(\mathsf{K}_G, ct_i^{(1)}\right), G\left(\mathsf{K}_G, ct_j^{(1)}\right)\right)\right)$
- Return $r \in \{-1, 0, 1\}$

Fig. 6: Two-Layered Order Revealing Encryption based on MIFE_{ℓ_1}

Hybrid 1: Like Hybrid 0 but instead of an IPRF, $\mathcal{S}$ uses a truly random function. The indistinguishability of Hybrids 0 and 1 follows directly from the security of the IPRF.
Hybrid 2: This is the ideal experiment.

Lemma 1. *Hybrid 2 is indistinguishable from Hybrid 1*

Proof. On each query that includes $\left(\mathsf{sk}_f, ct_i^{(2)}, ct_j^{(2)}\right)$, $\mathcal{S}$ stores the functional key along with the indices of the ciphertexts in Dict. This does not affect $\mathcal{A}$'s point of view. As a next step, $\mathcal{S}$ invokes $\mathcal{L}_{CMP}(i,j)$ and sends back to $\mathcal{A}$ a response $r \in \{-1, 0, 1\}$.

6 Functionally Encrypted Private Databases

In this section, we extend the MIFE-ORE construction presented in Section 5 to also ensure the privacy of the users instead of only protecting the confidentiality of their data. In particular, we assume that a curator encrypts her data locally before outsourcing them to the cloud, using our MIFE shcme for the ℓ_1 norm. Then, we allow an analyst to perform different kinds of queries (i.e. ORE queries, sums, averages) on the curator's encrypted data. To achieve our goal we describe a new primitive called *Multi-Input Functionally Encrypted Private Databases* (MIFE-PDB).

6.1 Background

We proceed by providing the main definitions of ϵ-differential privacy (ϵ-DP) and the main properties of the Laplace mechanism.

Definition 10 (ℓ_1-distance). *The ℓ_1-distance between two databases DB and DB' is given by:*

$$|||DB - DB'|||_1 \tag{4}$$

The ℓ_1-distance counts the number of entries on which the two databases differ.

Definition 11. *Two databases DB and DB' are neighbouring if:*

$$|||DB - DB'|||_1 \leq 1 \tag{5}$$

Definition 12 (ϵ-DP). *A privacy mechanism $\mathcal{M} : \mathbb{N}^{\mathcal{D}} \rightarrow Im(\mathcal{M})$ is ϵ-DP if $\forall\, \mathcal{S} \subset Im(\mathcal{M})$ and $\forall$ neighboring databases $DB, DB' \in \mathbb{N}^{|\mathcal{D}|}$:*

$$Pr[\mathcal{M}(DB) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(DB) \in \mathcal{S}]$$

Definition 13 (ℓ_1 sensitivity). *The ℓ_1 sensitivity of a query $q : \mathbb{N}^{|\mathcal{D}|} \rightarrow \mathbb{R}$ is:*

$$\Delta q = \max_{d(DB,DB') \leq 1} \|q(DB) - q(DB')\|_1$$

Definition 14 (Laplace distribution). *The Laplace distribution centered at 0 and with scale parameter b is given by:*

$$Lap(z) = \frac{1}{2b} e^{-\frac{|z|}{b}}$$

where the mean is 0 and the variance is $2b^2$.

We are now ready to proceed with the definition of the Laplace Mechanism [22].

Definition 15 (Laplace Mechanism). *Given a query* $q : \mathbb{N}^{|\mathcal{D}|} \to \mathbb{R}$*, the Laplace Mechanism is:*

$$M_L(DB, q, \epsilon) = q(DB) + Y_i,$$

where $Y_i \sim Lap(b)$

A proof showing that the Laplace Mechanism is ϵ-differentially private can be found in [22].

6.2 A Multi-Input Functionally Encrypted Database

We now proceed with the definition and construction of our Multi-Input Functionally Encrypted Database (MIFE-PDB).

Definition 16 (MIFE-PDB). *A MIFE-PDB scheme consists of four polynomial-time protocols* $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Query}$ *and* PQuery *that work as follows:*

- $PEDB \leftarrow \mathsf{Setup}(1^\lambda, DB)$*: The curator gives as input the security parameter* λ *and a sequence of plaintexts. The CSP receives a sequence of ciphertexts.*
- $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathbf{K}, f)$*: The curator gives as input two encryption keys* k_i *and* k_j *and receives a functional key* sk_f.
- $r \leftarrow \mathsf{Query}(\mathsf{sk}_f)$*: The curator a functional key* sk_f *for a function of her choice. The server replies with a result* r.
- $r \leftarrow \mathsf{PQuery}(\mathsf{sk_f})$*: This is a three party protocol between the Analyst, the Curator and the CSP. The Analyst requests a functional key* sk_f *for a function* f *of her choice. The Curator generates the key and sends it back to the Analyst. The Analyst then performs a private query to the CSP and receives back a result* r.

The MIFE-PDB scheme works as follows: The $\mathsf{Setup}, \mathsf{KeyGen}$ and Query protocols, have the same flow as in the MIFE-ORE construction described in Section 5. For the analyst to perform a private query, she first contacts the curator providing her with a description of the function f for which she wants a functional key sk_f. The curator generates the corresponding samples $\gamma \xleftarrow{\$} Lap(\frac{\Delta q}{\epsilon})$ and adds it to the key. Finally, the curator sends back $\mathsf{sk}'_f = \mathsf{sk}_f + \gamma$. The analyst can now contact the CSP to perform her private query. The MIFE-PDB scheme is presented in Figure 7.

$\underline{\mathsf{MIFE-PDB.Setup}(1^\lambda, DB):}$
Curator:
$\forall\, x_i \in DB$ Run:

- $\mathsf{MIFE}_{\ell_1}.\mathsf{Setup}(1^\lambda)$
- $\mathsf{MIFE}_{\ell_1}.\mathsf{Enc}(\mathbf{K}, i, x_i)$

$\underline{\mathsf{MIFE-PDB.KeyGen}(f)}$
Curator:

- Run $\mathsf{MIFE.KeyGen}(f)$
- Return sk_f

$\underline{\mathsf{MIFE-PDB.Query}(\mathsf{sk}_f, f)}$
Curator:

- Send (sk_f, f) to the CSP.

CSP:

- Run $\mathsf{MIFE.Dec}(\mathsf{sk_f}, \mathsf{PEDB})$
  Return a result r

$\underline{\mathsf{MIFE-PDB.PQuery}(\mathsf{sk}_f)}$
Analyst:

- Request a functional key for a function f from the curator.

Curator:

- Run $\mathsf{sk}_f \leftarrow \mathsf{MIFE-PDB.KeyGen}(f)$
- Sample noise $\gamma \xleftarrow{\$} Lap(\frac{\Delta q}{\epsilon})$
- Send $\mathsf{sk}'_f = \mathsf{sk}_f + \gamma$ back to the analyst.

Analyst:

- Send a query to the the CSP.

CSP:

- Run $\mathsf{MIFE.Dec}(\mathsf{sk}_f, PEDB)$
- Return a result r to the analyst

Fig. 7: Multi-Input Functionally Encrypted Private Database

Threat Model: For this application of our MIFE scheme, we need to revisit our threat model as we do not longer face an adversary that aims at breaking the cryptography. More specifically, we prove that our construction is differentially private by assuming the existence of a statistical adversary that corrupts the analyst. The statistical adversary can see the responses to the private queries but does not have access to the encrypted database nor to the queries of the curator.

Theorem 5. *Let $PDB, PDB' \in \mathbb{N}^{|\mathcal{D}|}$ be arbitrary neighbouring databases, let $q : \mathbb{N}^{|\mathcal{D}|} \to \mathbb{R}$ be an arbitrary query and let $r, r' \in \mathbb{R}$. Moreover, let M_L be the Laplace Mechanism. Then, the* PQuery *protocol in MIFE-PDB is ϵ-differentially private as per Definition 12.*

Proof. Our goal is to prove that issuing a private query q to PDB reveals no more information than what is allowed by the privacy factor ϵ. In our construction, we have that a query to the database returns a result r'. In other words, $q(PDB) = r'$. However, the query contains the secret functional key $\mathsf{sk}'_f = \mathsf{sk}_f + \gamma$, where $\gamma \leftarrow Lap(\frac{\Delta q}{\epsilon})$. Hence when the Laplace Mechanism is applied on the query we get

$$\begin{aligned} M_L(q, PDB, \epsilon) \ &= r = r' + \gamma \Rightarrow \\ \gamma \ &= r - r' \Rightarrow \\ \gamma &= r - q(PDB). \end{aligned}$$

However, since γ is arbitrarily chosen from the Laplace distribution ($\gamma \xleftarrow{\$} Lap(\frac{\Delta q}{\epsilon})$), then, without loss of generality, we can replace γ with $Lap(\frac{\Delta q}{\epsilon})$. Hence, we get:

$$\begin{aligned}
\frac{Pr[M_L(PDB, q, \epsilon) = r]}{Pr[M_L(PDB', q, \epsilon) = r]} &= \frac{Pr[Lap(\frac{\Delta q}{\epsilon}) = r - q(PDB)]}{Pr[Lap(\frac{\Delta q}{\epsilon}) = r - q(EDB')]} \\
&= \frac{\frac{\epsilon}{2\Delta q} exp(-\frac{|r-q(PDB)|}{\Delta q}\epsilon)}{\frac{\epsilon}{2\Delta q} exp(-\frac{|r-q(PDB')|}{\Delta q}\epsilon)} \\
&= exp\left(-\frac{\epsilon|r - q(PDB')| - |r - q(PDB)|)}{\Delta q}\right) \\
&= exp\left(\frac{\epsilon|q(PDB') - q(PDB)|}{\Delta q}\right) \leq e^{\epsilon}
\end{aligned}$$

7 Future Directions

While this work is amongst the first that utilize FE to address real-life problems, we acknowledge that it faces certain limitations. However, it is our firm belief that our schemes can serve as the basis for more advanced applications.

Range Queries Our TWOMORE scheme is a good candidate for a Range Queries scheme. In such a scheme, a user is enabled to issue encrypted search queries of the form "*Find all values in the range* $[a, b]$". Achieving such a scheme from an ORE scheme is not trivial since the contents of the query must also be protected like in the case of SSE schemes. Moreover, we plan to further enhance our construction with the important properties of forward and backward privacy [13] to ensure that updates on the database leak as little information as possible.

Functionally Encrypted Dynamic Private Databases Our main goal for this application is to make our database dynamic so that it would support cloud storage designs like the ones presented in [30,31,7,29]. A good first step towards such a database is to try modifying the differential privacy definitions accordingly. In particular, we would need to prove that our construction satisfies the following definition of differential privacy under continual observations [23]:

Definition 17. *Let M be a privacy mechanism, let $S \subset Im(M^{\lambda})$ and let* DB_0 *be the starting database. M is said to be differentially private over continual observations is for all neighboring sequences of curator operations $\sigma = (\sigma_1, \ldots, \sigma_{\lambda})$ and $\sigma' = (\sigma'_1, \ldots, \sigma'_{\lambda})$:*

$$\begin{aligned}
&Pr\left[(M(DB_1), \ldots, M(DB_{\lambda})) \in S\right] \leq \\
&e^{\epsilon} Pr\left[(M(DB'_1), \ldots, M(DB'_{\lambda})) \in S\right]
\end{aligned}$$

Where DB_i results from applying the update operation σ_i to DB_{i-1} and DB'_i from applying σ'_i to DB'_{i-1}.

With this in mind, we hope that the scheme presented in this work can be regarded as a first step towards the first Dynamic Functionally Encrypted Private Database.

8 Conclusion

The future will inevitably bring to the fore the need to exploit the power and applicability of FE. In addition to that, we strongly believe that cloud-based services will rely less on traditional decryption of information, and more on computations over encrypted data. We hope that this work will kick-start a period of greater research in the area of privacy-preserving computations in untrusted clouds and will allow us to see things concealed beneath the surface of the theoretical concepts of FE.

References

1. Abdalla, M., Catalano, D., Fiore, D., Gay, R., Ursu, B.: Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 597–627. Springer International Publishing, Cham (2018)
2. Abdalla, M., Gay, R., Raykova, M., Wee, H.: Multi-input inner-product functional encryption from pairings. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 601–626. Springer (2017)
3. Agarwal, A., Herlihy, M., Kamara, S., Moataz, T.: Encrypted databases for differential privacy. Proceedings on Privacy Enhancing Technologies **2019**(3), 170–190 (2019)
4. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Annual Cryptology Conference. pp. 657–677. Springer (2015)
5. Apple: Differential Privay. Tech. rep., https://www.apple.com/lae/privacy/docs/Differential_Privacy_Overview.pdf
6. Bakas, A., Michalas, A.: Power range: Forward private multi-client symmetric searchable encryption with range queries support. In: The 25th IEEE International Conference on Communications (ISCC'20) (2020)
7. Bakas, A., Michalas, A.: Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and sgx. In: International Conference on Security and Privacy in Communication Systems. pp. 472–486. Springer (2019)
8. Bakas, A., Michalas, A.: Multi-client symmetric searchable encryption with forward privacy. Cryptology ePrint Archive, Report 2019/813 (2019), https://eprint.iacr.org/2019/813
9. Boldyreva, A., Chenette, N., Lee, Y., O'neill, A.: Order-preserving symmetric encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 224–241. Springer (2009)
10. Boneh, D., Kim, S., Wu, D.J.: Constrained keys for invertible pseudorandom functions. In: Theory of Cryptography Conference. pp. 237–263. Springer (2017)

11. Boneh, D., Lewi, K., Raykova, M., Sahai, A., Zhandry, M., Zimmerman, J.: Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015. pp. 563–594. Springer (2015)
12. Boneh, D., Silverberg, A.: Applications of multilinear forms to cryptography. Contemporary Mathematics **324**(1), 71–90 (2003)
13. Bost, R., Minaud, B., Ohrimenko, O.: Forward and backward private searchable encryption from constrained cryptographic primitives. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 1465–1482. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3133980, https://doi.org/10.1145/3133956.3133980
14. Brakerski, Z., Komargodski, I., Segev, G.: Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. Journal of Cryptology **31**(2), 434–520 (2018)
15. Chase, M., Kamara, S.: Structured encryption and controlled disclosure. In: International conference on the theory and application of cryptology and information security. pp. 577–594. Springer (2010)
16. Chenette, N., Lewi, K., Weis, S.A., Wu, D.J.: Practical order-revealing encryption with limited leakage. In: International conference on fast software encryption. pp. 474–493. Springer (2016)
17. Costan, V., Devadas, S.: Intel sgx explained. IACR Cryptology ePrint Archive **2016**(086), 1–118 (2016)
18. Dang, H.V., Ullah, A., Bakas, A., Michalas, A.: Attribute-based symmetric searchable encryption. Cryptology ePrint Archive, Report 2020/999 (2020), https://eprint.iacr.org/2020/999
19. Desfontaines, D., Pejó, B.: Sok: Differential privacies. Proceedings on Privacy Enhancing Technologies **2020**(2), 288–313 (2020)
20. Dowsley, R., Michalas, A., Nagel, M., Paladi, N.: A survey on design and implementation of protected searchable data in the cloud. Comput. Sci. Rev. **26**(C), 17–30 (Nov 2017). https://doi.org/10.1016/j.cosrev.2017.08.001, https://doi.org/10.1016/j.cosrev.2017.08.001
21. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. pp. 1–19. Springer (2008)
22. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference. pp. 265–284. Springer (2006)
23. Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N.: Differential privacy under continual observation. In: Proceedings of the forty-second ACM symposium on Theory of computing. pp. 715–724 (2010)
24. Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 578–602. Springer (2014)
25. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to run turing machines on encrypted data. In: Annual Cryptology Conference. pp. 536–553. Springer (2013)
26. Google: Enabling developers and organizations to use differential privacy. Tech. rep. (09 2019), https://developers.googleblog.com/2019/09/enabling-developers-and-organizations.html

27. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Annual Cryptology Conference. pp. 162–179. Springer (2012)
28. Lewi, K., Wu, D.J.: Order-revealing encryption: New constructions, applications, and lower bounds. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1167–1178. CCS '16, ACM, New York, NY, USA (2016)
29. Michalas, A.: The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. pp. 146–155 (2019)
30. Michalas, A., Bakas, A., Dang, H.V., Zalitko, A.: Access control in searchable encryption with the use of attribute-based encryption and sgx. In: Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop. pp. 183–183 (2019)
31. Michalas, A., Bakas, A., Dang, H.V., Zaltiko, A.: Microscope: enabling access control in searchable encryption with the use of attribute-based encryption and sgx. In: Nordic Conference on Secure IT Systems. pp. 254–270. Springer (2019)
32. Naveed, M., Kamara, S., Wright, C.V.: Inference attacks on property-preserving encrypted databases. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 644–655 (2015)
33. Paladi, N., Gehrmann, C., Michalas, A.: Providing user security guarantees in public infrastructure clouds. IEEE Transactions on Cloud Computing **5**(3), 405–419 (July 2017). https://doi.org/10.1109/TCC.2016.2525991
34. Paladi, N., Michalas, A., Gehrmann, C.: Domain based storage protection with secure access control for the cloud. In: Proceedings of the 2014 International Workshop on Security in Cloud Computing. ASIACCS '14, ACM, New York, NY, USA (2014)
35. Sahai, A., Seyalioglu, H.: Worry-free encryption: functional encryption with public keys. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 463–472 (2010)
36. Sans, E.D., Gay, R., Pointcheval, D.: Reading in the dark: Classifying encrypted digits with functional encryption. IACR Cryptology ePrint Archive **2018**, 206 (2018)
37. Song, D., Wagner, D., Perrig, A.: Practical techniques for searching on encrypted data. In: IEEE Symposium on Research in Security and Privacy. pp. 44–55
38. Waters, B.: A punctured programming approach to adaptively secure functional encryption. In: Annual Cryptology Conference. pp. 678–697. Springer (2015)