Efficient Function-Hiding Functional Encryption: From Inner-Products to Orthogonality

Manuel Barbosa[1], Dario Catalano[2], Azam Soleimanian[3]*, and Bogdan Warinschi[4]

[1] INESC TEC and FCUP, Portugal
mbb@fc.up.pt
[2] Università di Catania, Italy.
catalano@dmi.unict.it
[3] Kharazmi University of Tehran, Iran and École Normale Supérieure, Paris, France
std_a.soleimani@khu.ac.ir
[4] University of Bristol, United Kingdom.
csxbw@bristol.ac.uk

Abstract. We construct functional encryption (FE) schemes for the orthogonality (OFE) relation where each ciphertext encrypts some vector **x** and each decryption key, associated to some vector **y**, allows to determine if **x** is orthogonal to **y** or not. Motivated by compelling applications, we aim at schemes which are function hidding, i.e. **y** is not leaked.

Our main contribution are two such schemes, both rooted in existing constructions of FE for inner products (IPFE), i.e., where decryption keys reveal the inner product of **x** and **y**. The first construction builds upon the very efficient IPFE by Kim et al. (SCN 2018) but just like the original scheme its security holds in the generic group model (GGM). The second scheme builds on recent developments in the construction of efficient IPFE schemes in the standard model and extends the work of Wee (TCC 2017) in leveraging these results for the construction of FE for Boolean functions. Conceptually, both our constructions can be seen as further evidence that shutting down leakage from inner product values to only a single bit for the orthogonality relation can be done with little overhead, not only in the GGM, but also in the standard model.

We discuss potential applications of our constructions to secure databases and provide efficiency benchmarks. Our implementation shows that the first scheme is extremely fast and ready to be deployed in practical applications.

1 Introduction

Consider the following scenario inspired from the literature on privacy preserving cryptographic role-based access control. The file storage of an organization is structured following a role-based access control, where users have associated one or more roles and each file can be accessed by users with a certain role (or

*Work done while visiting student at the Università di Catania, Italy.

combination of roles). Storage of the files is outsourced to a cloud which needs to serve files to users that request them. In particular, the cloud needs to determine, for each request, if it complies with the access control structure. In this scenario it is important to empower the cloud to perform such checks but, crucially, the cloud should not have information regarding the roles that can access each file. Indeed, access privileges may indicate which files are critical and may be linked, semantically, with the content of the files (e.g. revealing which patient files can be accessed by psychiatrists is clearly undesirable).

A similar scenario arises in the context of outsourcing file storage in a way that enables keyword search. A solution is to reveal to the cloud, for each file deterministic encryptions of the keywords which occur in that file. Even if the actual kewords are hidden, this solution reveals co-occurrence information, i.e. which files share keywords and how many keywords are shared. In turn this may reveal sensitive information about the semantics of the encrypted keywords.

The two scenarios are conceptually quite close and, unsurprisingly, share a similar solution. The information associated to a file f can be encoded as a binary vector $\boldsymbol{r}_f$ which encodes the subset of roles that can access a file. Similarly, to each user u one can then associate a binary vector $\boldsymbol{r}_u$, which encodes the roles associated to that user. User u has access to file f if $\langle \boldsymbol{r}_u, \boldsymbol{r}_f \rangle \neq 0$.[5] The challenge is to encode $\boldsymbol{r}_u$ and $\boldsymbol{r}_f$ in a way that prevents unnecessary leaks. In particular, given encodings of $\boldsymbol{r}_{f_1}$ and $\boldsymbol{r}_{f_2}$ the precise relation between the vectors (i.e. their dot-product) should not be revealed. More interestingly, while the cloud should learn that $\langle \boldsymbol{r}_u, \boldsymbol{r}_f \rangle \neq 0$ it should not learn the precise value of $\langle \boldsymbol{r}_u, \boldsymbol{r}_f \rangle$: this reveals the number of roles associated to a user that allow accessing that file.

Technically, the above functionality can be achieved using functional encryption for the orthogonality relation (OFE). Here, each ciphertext encrypts a vector $\mathbf{x}$ in $\mathbb{Z}_q^n$. Each secret key $\mathsf{sk}_{\mathbf{y}}$ is also associated with a vector $\mathbf{y}$ in $\mathbb{Z}_q^n$ defines a function $f_{\mathbf{y}}(\mathbf{x})$ that returns 1 iff $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, and returns 0 otherwise. We write $\mathbf{x} \perp \mathbf{y}$ for the orthogonality predicate between vectors $\mathbf{x}$ and $\mathbf{y}$.[6]

Despite the close relation between orthogonality and inner products, OFE is a different primitive from Functional Encryption for Inner-Products(IPFE): in the latter schemes a decryption key permits recovering the value of the inner-product $\langle \mathbf{x}, \mathbf{y} \rangle$. Intuitively, IPFE schemes should be easier to construct than OFE schemes, since they leak much more information about the encrypted data. A cursory look at the state of the art shows that this the case. For IPFE schemes, recent works [2,3,4,5,6,1] propose surprisingly efficient constructions of IPFE schemes with strong security guarantees and comparatively simple security proofs. More recent extensions to these constructions also covered the function-hiding case where decryption keys do not reveal information about the function to which they are associated. The most efficient scheme to date offering this level of security is

[5] Here $\langle \cdot, \cdot \rangle$ denotes the inner-product.

[6] In other works this type of OFE has been referred to predicate-hiding attribute-hiding predicate-only predicate encryption, but we prefer the view that we are dealing with a particular case of functional encryption rather than a particular case of attribute-based or predicate encryption.

based on a modular construction proposed by Lin [15] that converts two instances of a non-function hiding IPFE into a function hiding IPFE in an elegant way.

In contrast, most of the existing OFE schemes are instantiated in (three-factors) composite-order bilinear groups [10,12] or dual paring vector space on prime-order bilinear groups [17,18]. All of these schemes share an uncomfortably high level of conceptual complexity which explains perhaps the slow progress in this area. Indeed, till the recent work by Wee [22] there had been little progress on the design of (non function-hiding) OFE schemes. Wee shows that it is possible to port the rationale underlying the family of constructions of IPFE initiated by Abdalla et al. [1] to build more efficient OFE schemes from standard assumptions and using simpler proof techniques. The main result of this line of work is a family of simple OFE schemes in prime-order bilinear groups under the matrix-DDH (MDDH) assumption, using an insightful randomization technique to reduce inner-product leakage (in the exponents) to the orthogonality leakage allowed by OFE.

In this paper we extend this line of works, by considering the following two main questions in the context of OFE schemes that are function hiding:

Question 1 Can the relation between OFE and IPFE hinted at by Wee's construction be generalized to obtain black-box constructions of OFE from IPFE simply by "shutting down" the excessive leakage?

Question 2 Can one combine the new techniques by Lin [15] and Wee [22] in the construction of OFE, giving rise to new families of schemes and proof techniques?

1.1 Our Contributions

Simple Constructions: Good and Bad. We start by looking at the relation between OFE and IPFE and give a negative result that excludes a simplistic approach to constructing a function hiding OFE from any IPFE. Specifically, we look at black-box constructions that deterministically encodes the key $\mathbf{y}$ for the orthogonality relation as a set of keys $\{\mathbf{y}_1, \mathbf{y}_1, \ldots, \mathbf{y}_k\}$ for the inner product computation. We show that, even starting from a secure IPFE that also guarantees function hiding (FH-IPFE, for short), it is impossible to construct in this way a function hiding OFE even if security should only hold for a single ciphertext. We then extend the results to the case where the transformation is randomized, but multiple challenge queries are allowed. We stress that other black-box transformations, e.g. some which combine multiple instances of an IPFE scheme, are not ruled out by these results.

Next, we show that this negative result is tight: we provide a construction of an OFE from a FH-IPFE via a randomized transformation which is secure but only for the single-challenge case. While not all-encompassing, these negative results suggest ways around them. On the positive side, we first show how to overcome this negative result when working in the generic group model and slightly deviating from the simplistic black-box construction above. We give a highly efficient secure OFE in the generic group model (that also achieves function hiding) via a simple modification of the FH-IPFE scheme put forth by Kim et al. [14]. After these warm up results we move on to construct a fully secure OFE from standard assumptions. Our solution builds on results by Wee [22] and Lin [15] and extends them to the setting of function hiding OFE. We start by briefly discussing these two results separately.

Recent Developments in IPFE. In [22] Wee shows how a family of (public-key) IPFE schemes can be constructed from the MDDH assumption. The schemes are inspired by recent results in constructing IPFE in which the inner-product result is recovered in the exponent. Wee's crucial observation is that it is possible to use randomization to preserve the orthogonality relation in the decrypted result, while ensuring that no additional leakage exists under the DDH assumption. The resulting schemes are elegant and have a relatively simple security proof when compared to constructions relying on alternative techniques such as composite-order bilinear groups and dual pairing vector spaces over prime-order bilinear groups. The caveat is that these schemes are semi-adaptive secure (selective after seeing the master public key)

Lin [15] gave a generic construction of (secret-key) FH-IPFE from (public-key) IPFE schemes with a particular structure (similar in spirit to those explored by Wee). The construction (roughly) uses two instances of the same scheme on top of each other (the encryption algorithm of one scheme is used to protect keys and the other scheme is used to encrypt messages) and then takes advantage of the algebraic structure of such schemes to ensure the correctness of the construction via a combination of key extraction and decryption. Again, the security proof is simple and elegant. [7]

Main construction. We show how to combine the two techniques by Wee and Lin to give a modular construction of a new family of function-hiding OFE via the following partial results, which add up to our main technical contribution.

First, we extend Lin's generic construction from the IPFE to the OFE setting, showing that the construction also works if one starts from two instances of a OFE scheme to obtain a (weakly) function hiding OFE. We also observe that this transformation has a downside: if starting from a semi-adaptively secure OFE, one obtains a weakly secure OFE, where the adversary must be restricted to selectively commit to both keys and indices. Interestingly, our transform differs from Lin's original one in two main points. First, it does not induce additional levels of multi-linearity. Starting from two OFE in the bilinear group setting, the transformation produces a (weak) function hiding OFE that also relies on pairings. This is in sharp contrast with the basic IPFE setting [15] and similar to the multi-input IPFE setting [3]. Second, to guarantee correctness, the two underlying OFE need to be instantiated with different, but matching parameters.

[7] A (small) caveat of Lin's transform is that it only achieves weak function hiding. This is a relaxation of the FH notion that imposes some additional constraints on the key derivation queries that the adversary is allowed to ask. This restriction is not too severe as generic (yet efficient) transforms to fully fledged (strong) function hiding are known [16].

Scheme	Ours (GGM)	Ours (SM)	[20]	[13]
security	full	full*	selective	full
group order	prime	prime	composite	prime
assumption	GGM	MDDH, DDH	C3DH, DLIN	DLIN
key size	n	$6n+6$	$4n+4$	$6n$
ciphertext size	n	$6n+6$	$4n+4$	$6n$
key extraction	n	$12n+9$	$32n+4$	$6n$
encryption	n	$12n+9$	$24n+16$	$6n$
decryption	n	$6n+6$	$4n+4$	$6n$

Table 1. Comparison of our generic group model (GGM) and standard model (SM) constructions with prior constructions. Full security refers to unrestricted indistinguishability-based function-hiding. For the case of our standard-model scheme, we signal with * the (controlled) impact of complexity leveraging in our proof of security. Selective security refers to the setting where the attacker commits to the challenge message ahead of time. Sizes are given in terms of group element counts and the costs of key generation, extraction and encryption are expressed in group operation counts. For our standard model scheme we take $k = 2$.

.

Thus, to concretely instantiate our transform we modify Wee's OFE construction in two ways: i. we make it compatible with our extension of Lin's construction and ii. we use complexity leveraging to get adaptive (rather than semi-adaptive) security. As a result we get a new family of (function hiding) OFE schemes based on the MDDH assumption with a simple and modular proof of security and whose practical efficiency compares favorable with existing solutions (see table 1 for comparisons with previous work). We remark that our usage of complexity leveraging does not degrade security too much (at least when restricting, as we do in our applications, to small norm vectors). To see why this is the case let us describe our techniques a bit more in detail.

Just Enough Complexity Leveraging In general, any selective (or semi-adaptive) secure scheme can be turned into an adaptively secure one by essentially guessing the challenges in advance. Complexity leveraging typically induces an exponential factor (in the length of the challenge) loss in the quality of the reduction, often resulting in meaningless security guarantee for practical parameters. At the same time if one applies complexity leveraging to small size challenges, the security loss might become tolerable, thus making the technique relevant also from a practical perspective. A naive application of complexity leveraging to the scheme resulting from our transformation would lead to an unacceptably high security loss. Indeed, as we are dealing with a symmetric and function-hiding scheme, the reduction would need to guess in advance all the challenge messages and secret key queries that the adversary is allowed to ask. Even when restricting to small norm vectors this results in a huge exponential loss that destroys security completely. Our key observation is to "anticipate" complexity leveraging to a stage where it can be made much less harmful. Concretely, we apply the complexity leveraging step to the basic (semi-adaptive secure) OFE scheme. This scheme is secure in the public-key setting and therefore only one challenge query needs to be guessed by the reduction.[8] Moreover, we show that the next steps in our construction (namely our Lin-style function-hiding transform) easily extends to the adaptive setting without introducing exponential losses. Hence the final loss essentially matches the possibilities for a single message vector, which is tolerable for small norm vectors.

Applications and Implementation As a final result we put forward applications of OFE in the area of access-control and conjunctive keyword search. We focus on applications where our usage of complexity leveraging step does not reduce security too much, which is the case for both applications because they depend only on the ability to compute the subset relation. Indeed, when encoding the subset relation over n keywords/roles we can show that our loss in reduction tightness is only 2^{2n} and is independent of the size of the finite-field in which the orthogonality relation is computed.

We implement both our scheme in the generic group model and our main constructions and give benchmarking results for subset keyword search. The generic group model construction is very fast and it can be used in practical applications: all operations are in the range of 100 milliseconds for vectors of size 256. Operations in our standard model construction are roughly 6 times slower.

Organization After we establish notation and introduce preliminary definitions in Section 2 we present our generic group model construction is presented in Section 4 and our standard model construction in Section 5. Finally in Sections 6 and 7, respectively, we present our experiment results and discuss applications of our schemes.

2 Preliminaries

We write $y \leftarrow x$ for assigning a value to variable x and $x \twoheadleftarrow X$ when sampling x from the set X uniformly at random. For an integer n, we let $[n]$ denote the set $\{1, \ldots, n\}$. If $\mathcal{A}$ is a probabilistic algorithm, we also write $y \twoheadleftarrow \mathcal{A}(x_1, \ldots, x_n)$ for the action of running $\mathcal{A}$ on inputs $x_1, \ldots, x_n$ with random coins chosen uniformly at random, and assigning the result to y. We use ppt for probabilistic polynomial-time. All algorithms are ppt unless stated otherwise.

We use lowercase bold font for vectors $\boldsymbol{x}$ and uppercase bold font for matrices $\boldsymbol{A}$. $|\boldsymbol{x}|$ denotes vector length and $\boldsymbol{x} \parallel \boldsymbol{y}$ is used for vector concatenation. We use $\langle \mathsf{\mathbf{x}}, \mathsf{\mathbf{y}} \rangle$ to denote the inner-product of two vectors. We write $\mathsf{\mathbf{x}} \perp \mathsf{\mathbf{y}}$ for orthogonality of two vectors, which takes the value 1 if $\langle \mathsf{\mathbf{x}}, \mathsf{\mathbf{y}} \rangle = 0$, and 0 otherwise.

Throughout we let $\mathcal{PG} = (e, G_1, G_2, G_T, q, g_1, g_2)$ denote a pairing group, where G_1, G_2, G_T are cyclic groups of prime order q, g_1 and g_2 are generators of G_1 and G_2 respectively, and $e : G_1 \times G_2 \to G_T$ is an admissible bilinear map. For $a \in \mathbb{Z}_q$ and $i = \{1, 2, T\}$ we write $[a]_i$ for encoding a using the group operation $[a]_i = g_i^a$ and extend this notation naturally for the component-wise

[8] Recall that in the public key setting, adaptive single message indistinguishability implies adaptive many message indistinguishability via a standard hybrid argument

encoding of vectors and matrices. We will assume that the following computational assumption holds in both G_1 and G_2.[9]

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

Definition 2 ($\mathcal{D}_{l,k}$-Matrix Diffie-Hellman Assumption [11]). Let $\mathcal{D}_{l,k}$ be a matrix distribution. We say that the $\mathcal{D}_{l,k}$-Matrix Diffie-Hellman Assumption ($\mathcal{D}_{l,k}$-MDDH) holds in G if, for all ppt adversaries D, this definition of advantage is small

$$\mathsf{Adv}_{\mathcal{D}_{l,k},G}(D) := \Pr[D(\mathcal{G}, [A], [As]) = 1] - \Pr[D(\mathcal{G}, [A], [c]) = 1] \,.$$

The probability space is that induced by the following sampling operations $A \leftarrow\leftarrow \mathcal{D}_{l,k}$, $s \twoheadleftarrow \mathbb{Z}_q^k$, and $c \twoheadleftarrow \mathbb{Z}_q^l$ and the coin tosses of adversary D.

In this paper we consider the case $l = k + 1$ referred as $\mathcal{D}_k$-MDDH assumption. Note that, to simplify notation, we omit the security parameter in the previous assumption and throughout the paper. Asymptotic definitions of security can be recovered by considering a family of bilinear groups indexed by the security parameter.

Functional Encryption We briefly overview relevant concepts from the area of functional encryption, following the formalization introduced by Boneh, Sahai, Waters [9] and O'Neill [19]. We start with the syntax of this primitive.

Syntax A functional encryption scheme $\mathcal{FE}$ for a family of functions $F_{\mathsf{y}} X \rightarrow \Sigma$, for $\mathsf{y} \in Y$, is a tuple $\mathcal{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of ppt algorithms, where:

- $\mathsf{Setup}(\,)$ is the setup algorithm, which outputs a master public key mpk and a master secret key msk.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{y})$, is the key extraction algorithm, which on input a master secret key msk and key $\mathsf{y} \in Y$ outputs a secret key $\mathsf{sk_y}$ associated with F_{y}.
- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{x})$ is the encryption algorithm, which on input a public key mpk and a message $\mathsf{msk} \in X$ outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk_y})$ is the deterministic decryption algorithm, which on input a master publik key mpk, a ciphertext ct and a secret key $\mathsf{sk_y}$ outputs $z \in \Sigma$ or an abort symbol $\perp$.

We note that when $\Sigma = \{0, 1\}$ the syntax considered above matches predicate-only encryption schemes [12].

Correctness. A scheme $\mathcal{FE}$ as above is correct if, for all $(\mathsf{mpk}, \mathsf{msk})$ in the range of $\mathsf{Setup}(\,)$, all $\mathsf{x}, \mathsf{y} \in X$, all $\mathsf{sk_y}$ in the range of $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{y})$ and all ct in the range of $\mathsf{Enc}(\mathsf{mpk}, \mathsf{x})$, we have that $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk_y}) = F(\mathsf{x}, \mathsf{y})$.

[9] This implies that our scheme requires an asymmetric Type-III pairing group.

```
Experiment $\mathsf{IND}^{b}_{\mathcal{FE},\mathcal{A}}(\,)$:
$(\mathsf{mpk}, \mathsf{msk}) \twoheadleftarrow \mathsf{Setup}(\,)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{LoR_y}(\cdot,\cdot),\mathsf{LoR_x}(\cdot,\cdot)}(\alpha)$
Output $b'$
```

Fig. 1. Game defining indistiguishability-based security of a functional encryption scheme. An admissible adversary will ensure that $F_{\mathsf{y}_0^j}(\mathsf{x}_0^i) = F_{\mathsf{y}_1^j}(\mathsf{x}_1^i)$ for all i queries to $\mathsf{LoR_x}$ and all j queries to $\mathsf{LoR_y}$. Furthermore, we also impose that the attacker never queries the all-zeroes to either the key extraction or the encryption oracle.

Indistinguishability-based security Consider the experiment defined in Figure 1, parametrised by a functional encryption scheme $\mathcal{FE}$, an attacker $\mathcal{A}$ and a secret bit b. The $\mathsf{LoR_x}$ oracle receives two messages $(\mathsf{x}_0, \mathsf{x}_1)$ and returns a fresh encryption of x_b and the $\mathsf{LoR_y}$ oracle receives two keys $(\mathsf{y}_0, \mathsf{y}_1)$ and returns a secret key sk_b corresponding to a fresh extraction of y_b.

Several variants of IND-based security can be defined based on this experiment:

- Public-key security: the input to the attacker is $\alpha = \mathsf{mpk}$. In the secret key setting, we have $\alpha = \epsilon$. We use SK to refer to the latter weaker setting.
- Semi-adaptive security: the attacker places all calls to $\mathsf{LoR_x}$ before calling $\mathsf{LoR_y}$. We use SAD to refer to the weaker setting where this restriction is enforced.
- Non function-hiding (standard) security: the attacker is restricted to making $\mathsf{y}_0 = \mathsf{y}_1$ in all calls to $\mathsf{LoR_y}$. We use FH to denote the stronger setting where this restriction is not enforced.
- Weak function-hiding: the attacker is restricted by the stronger requirement $F_{\mathsf{y}_0^j}(\mathsf{x}_0^i) = F_{\mathsf{y}_1^j}(\mathsf{x}_0^i) = F_{\mathsf{y}_1^j}(\mathsf{x}_1^i)$ for all i queries to $\mathsf{LoR_x}$ and all j queries to $\mathsf{LoR_y}$. We use wFH to distinguish this case from the full function-hiding case.
- Single-message security: the attacker places only one call to $\mathsf{LoR_x}$. We will use one to indicate when we are in the weaker setting where this restriction is enforced.

For all such variants, the advantage of an an attacker $\mathcal{A}$ against $\mathcal{FE}$ is defined by the following difference of conditional probabilities, where xx will specify the security variant according to the above conventions.

$$\mathbf{Adv}^{\mathsf{xx\text{-}IND}}_{\mathcal{FE},\mathcal{A}}(\,) = \left|\Pr[\mathsf{IND}^{1}_{\mathcal{FE},\mathcal{A}}(\,) \Rightarrow 1] - \Pr[\mathsf{IND}^{0}_{\mathcal{FE},\mathcal{A}}(\,) \Rightarrow 1]\right| .$$

Discussion As examples of the use of our notation for security definitions, the strongest notion of security is function hiding public-key FE, denoted $\mathsf{FH\text{-}IND}$, which is actually impossible to achieve; the weakest notion is single-message, semi-adaptive single-key security in the secret-key setting, denoted $\mathsf{one\text{-}SAD\text{-}SK\text{-}IND}$. Note that in the public key setting the single-message and multi-message are equivalent via a standard hybrid argument (for all variants of security) whereas in the symmetric key setting this is not the case since the attacker cannot obtain arbitrary encryptions of chosen messages. Note also that, as mentioned above,

```
Experiment $\mathsf{Real}_{\mathcal{FE},\mathcal{A}}(\,)$:
    $(\mathsf{mpk}, \mathsf{msk}) \twoheadleftarrow \mathsf{Setup}(\,)$
    $b \twoheadleftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot),\mathsf{Enc}(\mathsf{mpk},\cdot)}(\alpha)$
    Output $b$
```

```
Experiment $\mathsf{Ideal}_{\mathcal{FE},\mathcal{A},\mathcal{S}}(\,)$:
    $(\mathsf{mpk}, \mathsf{msk}) \twoheadleftarrow \mathsf{Setup}(\,)$
    $b \twoheadleftarrow \mathcal{A}^{\mathcal{S}(\mathsf{key},\Phi(\cdot)),\mathcal{S}(\mathsf{msg},\Phi(\cdot))}(\alpha)$
    Output $b$
```

Fig. 2. Games defining simulation-based security of an FE scheme. On the i-th (resp. j-th) Enc query (resp. KeyGen query) the (stateful) simulator $\mathcal{S}$ receives as side information leakage Φ: a matrix of values such that $\Phi[i, j] = F_{\mathsf{y}_j}(\mathsf{x}_i)$, for all (i, j) combinations of all key extraction and encryption queries placed by $\mathcal{A}$ (including the current one). Furthermore, we also impose that the attacker never queries the all zeroes vector to either the key extraction or the encryption oracle.

function-hiding functional encryption cannot be satisfied in the public-key setting: once an adversary is provided with a secret key sk_y for some y and public encryption key mpk, it can learn $F_\mathsf{y}(\mathsf{x})$ for arbitrary x. Finally, note that in the secret-key setting, semi-adaptive security is the same as selective security, where the adversary needs to commit to the LoR_x queries without any side information about the global parameters. A further weakening of this notion is fully selective security, where all queries are provided upfront and the adversary gets a set of challenge ciphertexts and keys in batch to conduct its attack.

Simulation-based security Consider the experiments defined in Figure 2, which are parametrised by functional encryption scheme $\mathcal{FE}$, adversary $\mathcal{A}$ and simulator $\mathcal{S}$.
As before, the following variants of simulation-based security can be defined based on this experiment:

- Public-key security: the attacker is parametrised with $\alpha = \mathsf{mpk}$. In the secret key setting (SK), we have $\alpha = \epsilon$.
- Semi-adaptive security (SAD): the attacker places all calls to Enc before calling KeyGen.
- Non function-hiding (standard) security: leakage Φ is extended to also provide the inputs to the KeyGen oracle (i.e., the keys are explicitly given to the simulator). Again we use FH to denote the stronger function-hiding setting.
- Single-message security (one): the attacker places only one call to LoR_x.

For all such variants, the advantage of an an attacker $\mathcal{A}$ against $\mathcal{FE}$ is defined by the following difference of probabilities, where xx will specify the security variant according to the above conventions.

$$\mathbf{Adv}^{\mathsf{xx\text{-}SIM}}_{\mathcal{FE},\mathcal{A}}(\,) = \left|\Pr[\mathsf{Real}_{\mathcal{FE},\mathcal{A}}(\,) \Rightarrow 1] - \Pr[\mathsf{Ideal}_{\mathcal{FE},\mathcal{A},\mathcal{S}}(\,) \Rightarrow 1]\right| \,.$$

For the same set of adversarial restrictions, simulation-based security implies indistinguishability-based security. To see this, observe that any IND attacker $\mathcal{A}$ can be used to construct a SIM attacker $\mathcal{B}$ as follows. $\mathcal{B}$ initially chooses a bit b uniformly at random and converts the left-right calls placed by $\mathcal{A}$ into encryption and key extractions calls x_b (resp. y_b) that depend on b. By giving the oracle answers back to $\mathcal{A}$, our SIM adversary ensures that, when running in the real world, it perfectly simulates the environment in the IND experiment for $\mathcal{A}$. The output of $\mathcal{A}$, which $\mathcal{B}$ uses as its own will therefore be correlated with b in a visible way if $\mathcal{A}$ is a successful IND attacker. Consider now the ideal world and any simulator Sim. It is easy to see that, given the restrictions on the left-or-right calls placed by $\mathcal{A}$, the input to the simulator will be information-theoretically independent of b, which means that the output of $\mathcal{A}$ will also be independent of b. The bias in the real-world output would therefore give $\mathcal{B}$ a visible advantage in breaking SIM security. In other words, the existence of an IND attacker with large advantage contradicts the existence of a successful simulator.

3 IPFE vs OFE

Perhaps the first question elicited by the close relationship between IPFE and OFE is whether generic transformations of one scheme into the other one are possible. We briefly explore a couple of simple transformations where one attempts to construct an OFE from an IPFE by somehow encoding an OFE key $\mathbf{y}$ as a vector of keys $\mathbf{y}_i$ for the underlying IPFE. We provide negative results which show that no deterministic transformation (even one which depends on a secret key) cannot yield a function-hidding OFE, independent of the security level offered by the starting IPFE.

These negative results heavily rely on the determinism of the transformation and suggest that one way around them would be to consider randomized transformations. Indeed, for warm-up we present a simple OFE scheme constructed, generically, from an IPFE scheme: the OFE key for some vector $\mathbf{y}$ is simply the IPFE key for $r \cdot \mathbf{y}$ for some randomly selected scalar r: decryption of a ciphertext which encrypts $\mathbf{x}$ is either 0 when $\mathbf{x} \perp \mathbf{y}$ or uniformly random otherwise. Clearly, as soon as the adversary has more than one ciphertext, which each encrypts messages known to the adversary, then can recover information about r and $\mathbf{y}$. In effect, we can only prove that the scheme is $\mathsf{one\text{-}SAD\text{-}FH\text{-}IND}$-secure.

For space reasons we describe the negative results and the construction in the full version of this paper. Nonetheless, even the cursory discussion above indicates that one needs additional randomization also in the ciphertexts. The scheme which we present next implements this intuition.

4 A construction in the generic group model

In this section we describe a simple construction which satisfies simulation-based security in the generic group model (GGM). Our starting point is recent work by Kim et al. [14] who propose a FH-IPFE scheme that is simulation-based secure in the GGM. The construction follows the pattern of recent schemes where the inner-product is recovered by solving a discrete logarithm problem over a small domain by exhaustive search. Here we show that, by a simple adaptation where we omit one group element in both keys and ciphertexts (which are the values used to compute the basis for the discrete logarithm problem) we obtain a fully secure OFE. Indeed, the information leaked by the scheme of Kim et al. is accessible to the GGM attacker only via a zero-testing oracle which becomes useless if the basis for the discrete logarithm problem is hidden.

Our construction works as follows:

- $\mathsf{Setup}(1^\lambda, n)$: On input the security parameter λ, the setup algorithm samples an asymmetric bilinear group (G_1, G_2, G_T, q, e) and chooses generators $g_1 \in G_1$ and $g_2 \in G_2$. Then, it samples an invertible square matrix $\boldsymbol{B} \in \mathbb{Z}_q^{n \times n}$ uniformly at random and sets $\boldsymbol{B}^\star = \mathsf{det}(\boldsymbol{B}) \cdot (\boldsymbol{B}^{-1})^\top$. The algorithm outputs the public parameters $\mathsf{pp} = (G_1, G_2, G_T, q, e, n)$ and the master secret key $\mathsf{msk} = (\mathsf{pp}, g_1, g_2, \boldsymbol{B}, \boldsymbol{B}^\star)$.
- $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{y})$: On input the master secret key msk and a vector $\boldsymbol{y} \in \mathbb{Z}_q^n$, the key generation algorithm chooses an element $\alpha \in \mathbb{Z}_q$ uniformly at random and outputs $\mathsf{sk}_{\boldsymbol{y}} = [\alpha \cdot \boldsymbol{y}^\top \cdot \boldsymbol{B}]_1$, i.e., a vector of encodings in G_1.
- $\mathsf{Enc}(\mathsf{msk}, \boldsymbol{x})$: On input the master secret key msk and a vector $\boldsymbol{x} \in \mathbb{Z}_q^n$, the encryption algorithm chooses an element $\beta \in \mathbb{Z}_q$ uniformly at random and outputs $\mathsf{ct} = [\beta \cdot \boldsymbol{x}^\top \cdot \boldsymbol{B}^\star]_2$, i.e., a vector of encodings in G_2.
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$: On input the public parameters pp, a secret key sk and a ciphertext ct, the algorithm computes $\prod_{i=1}^{n} e(\mathsf{sk}[i], \mathsf{ct}[i])$ and returns $\top$ if the result is equal to $\mathbf{1}_{G_T}$ and $\perp$ otherwise.

Correctness of the scheme follows from the fact that the output value computed by decryption encodes $[\alpha\beta \cdot \mathbf{x}^\top \cdot \mathbf{B} \cdot {\mathbf{B}^\star}^\top \cdot \mathbf{y}]_t$, which therefore includes $\langle \mathbf{x}, \mathbf{y} \rangle$ as a multiplicative factor. The following theorem establishes the security of the scheme.

Theorem 1. The above OFE scheme is simulation-based secure OFE in the GGM.

Sketch. The proof is an adaptation of the original argument in [14]. Specifically, we describe a simulator that, not only answers key extraction and encryption queries in a way which is identical to what happens in the real world, it also simulates the operation of the generic bilinear group operations in a way which is indistinguishable from what the attacker sees in the real world. Due to the operation of the generic group model, all queries that the adversary makes can be perfectly simulated by returning fresh random labels for all group elements resulting from key extraction, encryption, and bilinear group operations bar zero testing. Simulating zero-test queries in the source groups is natural: the simulator answers zero if and only if the queried label corresponds to a formal polynomial that is identically zero; all non-zero answers can be justified by the Schwartz-Zippel lemma. The more intricate part of the simulation lies in zero-test queries for the target group, where one must take into account that formal polynomials that are not identically zero in the simulator's view, correspond to cancellations in the real world. Here we show that the simulator can identify honest evaluations of inner products between orthogonal vectors (these cases can be detected because orthogonality is revealed in the leakage provided to the simulator) and correctly answer zero to linear combinations of such cases.

```
$\mathsf{Setup}(\,)$:
$\boldsymbol{A} \twoheadleftarrow \mathbb{Z}_q^{k+1 \times k}$
For $i \in [n]$:
  $\boldsymbol{W}_i \twoheadleftarrow \mathbb{Z}_q^{k+1 \times k+1}$
$\mathsf{msk} \leftarrow (\boldsymbol{A}, \{\boldsymbol{W}_i\}_{i=1}^n)$
$\mathsf{mpk} \leftarrow ([\boldsymbol{A}^\top]_1, \{[\boldsymbol{A}^\top \boldsymbol{W}_i]_1\}_{i=1}^n)$
Return $(\mathsf{msk}, \mathsf{mpk})$
```

```
$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x})$:
$\boldsymbol{s} \twoheadleftarrow \mathbb{Z}_q^{k}$
$\boldsymbol{U} \twoheadleftarrow \mathbb{Z}_q^{k+1 \times k+1}$
$\boldsymbol{M}_0 \leftarrow \boldsymbol{s}^\top \boldsymbol{A}^\top$
$\mathsf{ct} \leftarrow [\boldsymbol{M}_0 || \{\boldsymbol{M}_0(\mathbf{x}_i \boldsymbol{U} + \boldsymbol{W}_i)\}_{i=1}^n]_1$
Return $\mathsf{ct}$
```

```
$\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$:
$\boldsymbol{r} \twoheadleftarrow \mathbb{Z}_q^{k+1}$
$\mathsf{sk} \leftarrow [-\sum_{i=0}^n \mathbf{y}_i \boldsymbol{W}_i \boldsymbol{r} || \{\mathbf{y}_i \boldsymbol{r}\}_{i=1}^n]_2$
Return $\mathsf{sk}$
```

```
$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$:
Return $\langle \mathsf{ct}, \mathsf{sk} \rangle = \mathbf{1}$
```

Fig. 3. First variant of Wee's scheme. Decryption is presented using inner-product notation, denoting in compact form the pointwise pairing of ciphertext and key components (each comprising $(n+1)(k+1)$ group elements), followed by a product to obtain a single group element.

We adapt the argument in [14] to show that all other cases can be answered as non-zero. The details are deferred to the full version of this paper.

□

5 A construction in the standard model

In this section we show a construction of a function hiding OFE that is provably secure in the standard model. Our construction is developed in several steps.

Intuitively, our goal is to adapt a technique originally developed by Lin [15] in the context of functional encryption for inner products to the case of OFE. Recall that Lin's technique allows to combine two instances of a functional encryption scheme for inner products to obtain a (secret key) functional encryption scheme for inner products that also provides function hiding guarantees.

Aiming at the simplest possible solution, the natural approach would be to try to combine Lin's technique with the clever OFE recently proposed by Wee in [22] . Interestingly, adapting Lin's transform to the orthogonality setting is not at all immediate. Indeed, to guarantee correctness, the two instances of the OFE need to be instantiated with different, but matching, parameters. This is in sharp contrast with the basic IPFE setting where the transformation is less demanding on the underlying encryption schemes. In particular, we need to develop two novel variants of the basic Wee's scheme, both of which we discuss next.

5.1 First Scheme

The first scheme closely follows the blueprint of Wee's original scheme. The difference is that matrices U and W_i are uniformly chosen in $\mathbb{Z}_q^{k+1 \times k+1}$, rather than in $\mathbb{Z}_q^{k+1 \times k}$ as in Wee's scheme. This is shown in Figure 3. Correctness follows from the fact that the result of decryption includes $\langle \mathbf{x}, \mathbf{y} \rangle$ as a multiplicative factor

```
$\mathsf{Setup}()$:
$\boldsymbol{A} \twoheadleftarrow \mathbb{Z}_q^{k+1\times k}$
For $i \in [n]$:
  $\boldsymbol{W}_i \twoheadleftarrow \mathbb{Z}_q^{k+1\times k+1}$
$\mathsf{msk} \leftarrow (\boldsymbol{A}, \{\boldsymbol{W}_i\}_{i=1}^n)$
$\mathsf{mpk} \leftarrow ([\boldsymbol{A}^\top]_2, \{[\boldsymbol{A}^\top \boldsymbol{W}_i]_2\}_{i=1}^n)$
Return $(\mathsf{msk}, \mathsf{mpk})$
```

```
$\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{X})$:
$\boldsymbol{s} \twoheadleftarrow \mathbb{Z}_q^k$
$\boldsymbol{U} \twoheadleftarrow \mathbb{Z}_q^{k+1}$
$\boldsymbol{M}_0 \leftarrow \boldsymbol{s}^\top \boldsymbol{A}^\top$
$\mathsf{ct} \leftarrow [\boldsymbol{M}_0 || \{\boldsymbol{M}_0(\boldsymbol{U}\boldsymbol{X}_i + \boldsymbol{W}_i)\}_{i=1}^n]_2$
Return $\mathsf{ct}$
```

```
$\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$:
$r \twoheadleftarrow \mathbb{Z}_q$
$\mathsf{sk} \leftarrow [-\sum_{i=0}^n r\boldsymbol{Y}_i^\top \boldsymbol{W}_i^\top || \{r\boldsymbol{Y}_i^\top\}_{i=1}^n]_1$
Return $\mathsf{sk}$
```

```
$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$:
Return $\langle \mathsf{ct}, \mathsf{sk} \rangle = \mathbf{1}$
```

Fig. 4. Second variant of Wee's scheme. Decryption notation is as in Figure 3.

in the exponent. Indeed, decryption computes in the exponents:

$$\sum_{i=1}^n \mathbf{y}_i \boldsymbol{M}_0(\mathbf{x}_i \boldsymbol{U} + \boldsymbol{W}_i)\boldsymbol{r} - \boldsymbol{M}_0 \sum_{i=0}^n \mathbf{y}_i \boldsymbol{r}^\top \boldsymbol{W}_i^\top = \boldsymbol{M}_0 \boldsymbol{U} \boldsymbol{r} \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}_q \,.$$

The following theorem establishes security and follows an argument similar to Wee's construction [22]. A sketch of the proof is given in the full version of this paper.

Theorem 2. If MDDH and DDH assumptions hold respectively in G_1 and G_2 then the modified scheme of Wee in figure 3 is one-SAD-SIM secure.

5.2 Second Scheme

The second construction modifies Wee's scheme in the sense that it allows to compute $\sum_i \boldsymbol{X}_i \boldsymbol{Y}_i$ for $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ and $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n)$ where for all $i \in [n]$, $\boldsymbol{X}_i \in \mathbb{Z}_q^{1\times k+1}$ and $\boldsymbol{Y}_i \in \mathbb{Z}_q^{k+1}$. Intuitively, this corresponds precisely to the computation carried out in the exponents by the decryption algorithm of the first variant of Wee's scheme we presented above. The scheme can be found in Figure 3.

Correctness can be verified by rewriting the decryption operation as

$$\sum_{i=1}^n (r\boldsymbol{M}_0(\boldsymbol{U}\boldsymbol{X}_i + \boldsymbol{W}_i)\boldsymbol{Y}_i) - \boldsymbol{M}_0 \sum_{i=0}^n r\boldsymbol{W}_i \boldsymbol{Y}_i = r\boldsymbol{M}_0 \boldsymbol{U} \sum_{i=1}^n \boldsymbol{X}_i \boldsymbol{Y}_i$$

Again, the following theorem shows that these modifications do not affect security. The proof is similar to the scheme of Wee [22] and is given in the full version of this paper.

Theorem 3. If DDH and MDDH assumptions hold respectively in G_1 and G_2, then the modification of Wee's scheme in figure 4 is one-SAD-SIM secure.

As a simple corollary of theorems 2 and 3 we have the following

Corollary 1. The two modifications of Wee's scheme are (many) SAD-IND secure.

5.3 Weak Function-Hiding Functional Encryption for Orthogonality

Now, we can give the details of our new Lin-like transform for orthogonality. For simplicity, we present our results in the fully selective setting, but the proof easily generalises to the fully adaptive setting if the underlying constructions are themselves fully adaptive. Moreover, for clarity of exposition, we present the transform in an abstract, generic way. In particular we first establish a set of conditions (see Definition 3 below) for which the transformation works and then show that our two schemes from sections 5.1 and 5.2 trivially satisfy these conditions. We stress that the transformation produces a scheme that is weakly function hiding. Still, this is enough for us as we can move to a full-fledged FH solution using the efficient Lin-Vaikuntanathan [16] compiler.[10]

Definition 3. Let $\mathit{\Gamma} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a Functional Encryption scheme for orthogonality (OFE), we say that $\mathit{\Gamma}$ is $[\cdot]_{\alpha\beta}$-OFE, for $\alpha, \beta \in \{1,2\}$ if the following properties are satisfied.

1. There are ppt algorithms RowKey and RowEnc such that,

$$\mathsf{Enc}(\mathsf{mpk}, \cdot) = [\mathsf{RowEnc}(\mathsf{msk}, \cdot)]_\alpha \text{ and } \mathsf{KeyGen}(\mathsf{msk}, \cdot) = [\mathsf{RowKey}(\mathsf{msk}, \cdot)]_\beta$$

   for all $(\mathsf{mpk}, \mathsf{msk})$ in the support of $\mathsf{Setup}(\,)$.
2. There are efficiently computable functions F_e and F_k such that

$$\mathsf{Enc}(\mathsf{mpk}, \cdot) = F_e(\mathsf{mpk}, [\cdot]_\alpha) \text{ and } \mathsf{KeyGen}(\mathsf{msk}, \cdot) = F_k(\mathsf{msk}, [\cdot]_\beta)\,.$$

3. For both schemes, and for all ciphertexts in the support of $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x})$ and keys in the support of $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$, there exists some scalar δ that is a function of the randomness used in algorithms Enc and KeyGen, such that decryption returns $[\langle \mathbf{x}, \mathbf{y} \rangle]_\top^\delta$ computed as

$$\mathsf{Dec}(\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}), \mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})) = [\langle \mathsf{RowEnc}(\mathsf{mpk}, \mathbf{x}), \mathsf{RowKey}(\mathsf{msk}, \mathbf{y}) \rangle]_\top\,.$$

It is easy to see that our first and second modification of Wee's scheme are respectively $[\cdot]_{12}$-OFE and $[\cdot]_{21}$-OFE schemes. We now show that, if $\mathit{\Gamma}_1$ and $\mathit{\Gamma}_2$ are two $[\cdot]_{12}$ and $[\cdot]_{21}$ OFE schemes, respectively, then the generic OFE construction in Figure 5 is a secret-key (weakly) function hiding OFE. Correctness of the construction follows from the following derivation:

$$\mathit{\Gamma}_2.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = [\langle \mathsf{sk}_2, \mathsf{ct}_1 \rangle]_T^{\delta_2} = [\langle \mathbf{x}, \mathbf{y} \rangle]_T^{\delta_1 \delta_2}$$

Theorem 4. If $\mathit{\Gamma}_1$ and $\mathit{\Gamma}_2$ are $\mathsf{SAD\text{-}IND}$ secure OFE schemes then our scheme is selectively secure OFE with (weak) function hiding.

[10] The compiler has been proposed in the IPFE setting, but trivially extends to the OFE setting.

$\underline{\mathsf{Setup}(\,)}$:	$\underline{\mathsf{Enc}(\mathsf{msk}, \mathbf{x})}$:
$(\mathsf{msk}_1, \mathsf{mpk}_1) \twoheadleftarrow \mathit{\Gamma}_1.\mathsf{Setup}(n)$ $(\mathsf{msk}_2, \mathsf{mpk}_2) \twoheadleftarrow \mathit{\Gamma}_2.\mathsf{Setup}(n+1)$ Return $(\mathsf{msk}_1, \mathsf{msk}_2, \mathsf{mpk}_2)$	$(\mathsf{msk}_1, \mathsf{msk}_2) \leftarrow \mathsf{msk}$ $\mathsf{ct}_1 \twoheadleftarrow \mathit{\Gamma}_1.\mathsf{RowEnc}(\mathsf{msk}_1, \mathbf{x})$ $\mathsf{ct} \twoheadleftarrow \mathit{\Gamma}_2.\mathsf{KeyGen}(\mathsf{msk}_2, \mathsf{ct}_1)$ Return ct
$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})}$: $(\mathsf{msk}_1, \mathsf{msk}_2) \leftarrow \mathsf{msk}$ $\mathsf{sk}_1 \twoheadleftarrow \mathit{\Gamma}_1.\mathsf{RowKey}(\mathsf{msk}_1, \mathbf{y})$ $\mathsf{sk} \twoheadleftarrow \mathit{\Gamma}_2.\mathsf{Enc}(\mathsf{mpk}_2, \mathsf{sk}_1)$ Return sk	$\underline{\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})}$: Return $\mathit{\Gamma}_2.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

Fig. 5. Lin-like transform for orthogonality. We slightly abuse notation by using $\mathit{\Gamma}_1.\mathsf{Setup}(n)$ and $\mathit{\Gamma}_2.\mathsf{Setup}(n+1)$ to denote the size of message and key vectors supported by each scheme when constructing a function-hiding OFE for vectors of size n.

Proof. The proof follows from a sequence of games, where Game_0 is the real game in the definition of indistinguishability-based security, when $b = 0$, Game_1 is the same game when $b = 1$, and Game_h is a hybrid game that proceeds as Game_0, except that Enc is run on inputs $\mathbf{x}_1^j$. Thus, for the security proof it is enough to prove that Game_h is computationally indistinguishable from both Game_0 and Game_1.

Indistinguishability of Game_0 and Game_h: Let $\mathcal{A}_{0-h}$ be any adversary that is able to distinguish between these two games. We construct $\mathcal{B}$ that breaks the SAD-IND-security of $\mathit{\Gamma}_1$. $\mathcal{B}$ runs $\mathcal{A}_{0-h}$, interpolating between the two games while interacting with the experiment SAD-IND, as follows.

$\mathcal{B}$ gets the the public key mpk_1 of scheme $\mathit{\Gamma}_1$ and challenges $(\mathbf{x}_0^j, \mathbf{x}_1^j)$ and $(\mathbf{y}_0^i, \mathbf{y}_1^i)$ from $\mathcal{A}_{0-h}$. Then $\mathcal{B}$ runs $\mathit{\Gamma}_2.\mathsf{Setup}$ itself to get a pair $(\mathsf{msk}_2, \mathsf{mpk}_2)$, calls the external LoR_x oracle to get encryptions under $\mathit{\Gamma}_1$ of all the challenges $[\mathsf{ct}_1^j]_1 = \mathit{\Gamma}_1.\mathsf{Enc}(\mathsf{mpk}_1, \mathbf{x}_b^j)$, and computes $\mathsf{ct}^j = \mathit{\Gamma}_2.\mathsf{KeyGen}(\mathsf{msk}_2, \mathsf{ct}_1^j) = F_k(\mathsf{msk}_2, [\mathsf{ct}_1^j]_1)$, where F_k comes from definition 3. Then, $\mathcal{B}$ sends queries $\mathbf{y}_0^i$ to key extraction in the external game, receives secret keys $[\mathsf{sk}_1^i]_2 = \mathit{\Gamma}_1.\mathsf{KeyGen}(\mathsf{msk}_1, \mathbf{y}_0^i)$ and computes

$$\mathsf{sk}^i = \mathit{\Gamma}_2.\mathsf{Enc}(\mathsf{mpk}_2, \mathsf{sk}^i) = F_e(\mathsf{mpk}_2, [\mathsf{sk}_1^i]_2)$$

It provides all ciphertexts and keys to the attacker, waits for the adversary's choice, and uses this as it's own output. It is easy to see that any change in the behaviour of $\mathcal{A}_{0-h}$ between the two games is immediately translated into a distinguishing advantage against $\mathit{\Gamma}_1$. This is because all queries placed by $\mathcal{B}$ are admissible: $\mathcal{B}$ must satisfy restriction $\mathbf{x}_0^j \perp \mathbf{y}_0^i = \mathbf{x}_1^j \perp \mathbf{y}_0^i$ on all queries and this is guaranteed because $\mathcal{A}_{0-h}$ has output challenges that satisfy $\mathbf{x}_0^j \perp \mathbf{y}_0^i = \mathbf{x}_1^j \perp \mathbf{y}_0^i = \mathbf{x}_1^j \perp \mathbf{y}_1^i$.

Indistinguishability of Game_h and Game_1: Let $\mathcal{A}_{h-1}$ be any adversary that is able to distinguish between these two games. We construct $\mathcal{B}$ that breaks the

SAD-IND-security of $\mathit{\Gamma}_2$. $\mathcal{B}$ runs $\mathcal{A}_{h-1}$, interpolating betweeen the two games while interacting with the experiment SAD-IND, as follows.

$\mathcal{B}$ gets the the public key mpk_2 of scheme $\mathit{\Gamma}_2$ and challenges $(\mathbf{x}_0^j, \mathbf{x}_1^j)$ and $(\mathbf{y}_0^i, \mathbf{y}_1^i)$ from $\mathcal{A}_{h-1}$. Then $\mathcal{B}$ runs $\mathit{\Gamma}_1.\mathsf{Setup}$ itself to get a pair $(\mathsf{msk}_1, \mathsf{mpk}_1)$, computes

$$\mathsf{sk}_{1,c}^i = \mathit{\Gamma}_1.\mathsf{RowKey}(\mathsf{msk}_1, \mathbf{y}_c^i) \text{ for all } i \text{ and } c \in \{0,1\}\,,$$

and calls LoR_x in the external game on $(\mathsf{sk}_{1,0}^i, \mathsf{sk}_{1,1}^i)$ to get $\mathsf{sk}^i = \mathit{\Gamma}_2.\mathsf{Enc}(\mathsf{mpk}_2, \mathsf{sk}_{1,b}^i)$. $\mathcal{B}$ then computes $\mathsf{ct}_1^j = \mathit{\Gamma}_1.\mathsf{RowEnc}(\mathsf{mpk}_1, \mathbf{x}_1^j)$, calls key extraction in the external game to obtain $\mathsf{ct}^j = \mathit{\Gamma}_2.\mathsf{KeyGen}(\mathsf{msk}_2, \mathsf{ct}_1^j)$. Finally, $\mathcal{B}$ provides all ciphertexts and keys to the attacker, waits for the adversary's choice, and uses this as it's own output.

It is easy to see that any change in the behaviour of $\mathcal{A}_{h-1}$ between the two games is immediately translated into a distinguishing advantage against $\mathit{\Gamma}_2$. This is because all queries placed by $\mathcal{B}$ are admissible, which we now justify. $\mathcal{B}$ must satisfy restriction $\mathsf{ct}_1^j \perp \mathsf{sk}_{1,0}^i = \mathsf{ct}_1^j \perp \mathsf{sk}_{1,1}^i$ on all queries. Note that $[\langle \mathsf{ct}_1^j, \mathsf{sk}_{1,b}^i \rangle]_T = [\langle \mathbf{x}_1^j, \mathbf{y}_b^i \rangle]_T^{\delta_1}$, so restriction $\mathsf{ct}_1^j \perp \mathsf{sk}_{1,0}^i = \mathsf{ct}_1^j \perp \mathsf{sk}_{1,1}^i$ is equivalent to $\mathbf{x}_1^j \perp \mathbf{y}_0^i = \mathbf{x}_1^j \perp \mathbf{y}_1^i$. Furthermore, $\mathcal{A}_{h-1}$ outputs challenges that satisfy $\mathbf{x}_0^j \perp \mathbf{y}_0^i = \mathbf{x}_1^j \perp \mathbf{y}_0^i = \mathbf{x}_1^j \perp \mathbf{y}_1^i$. Thus, all queries placed by $\mathcal{B}$ are admissible. □

5.4 Achieving adaptive security

An obvious way to make the scheme given in section 5.3 adaptive secure, would be to employ complexity leveraging.

However, a naive application of complexity leveraging to the scheme from section 5.3 would result in a security loss 2^τ where $\tau = q_e|\mathbf{x}| + q_s|\mathbf{y}|$, (here q_e and q_s are, respectively, the maximum number of encryption queries and secret key queries allowed). This is because the scheme is selective both with respect to challenge messages and with respect to challenge keys. Furthermore, since it lives in the symmetric setting we need to guess all the challenges in advance. Notice that, while in our setting both $|\mathbf{x}|$ and $|\mathbf{y}|$ might be small, this is not necessarily the case for τ.

We overcome this by "anticipating" the complexity leveraging step to the basic schemes. Recall that the construction from section 5.3 builds upon two schemes $\mathit{\Gamma}_1$ and $\mathit{\Gamma}_2$ that are in the public key setting. These latter schemes, in turn, are assumed to guarantee $\mathsf{SAD\text{-}IND}$ security, which means they also guarantee $\mathsf{one\text{-}SAD\text{-}IND}$ security.

Our key observation is to apply complexity leveraging to these basic $\mathsf{one\text{-}SAD\text{-}IND}$ secure building blocks. This means that assuming that $\mathbf{x}$ (resp. $\mathbf{y}$) is sufficiently small, complexity leveraging induces only a polynomial $2^{2|\mathbf{x}|}$ (resp. $2^{2|\mathbf{y}|}$) loss, as one single challenge query has to be guessed. Next, we build our way towards a fully fledged (adaptively secure) construction via the following two observations. First, in the public key setting, $\mathsf{one\text{-}IND}$ implies (many) IND via a standard hybrid argument that only induces a polynomial loss in the security reduction. Second, Theorem 4 trivially extends to the adaptive setting without introducing additional losses.

All these observations combined mean that the resulting scheme achieves adaptive security with only a $\max(2^{2|\mathsf{x}|}, 2^{2|\mathsf{y}|})$ security loss with respect to the selective secure solution we started from. In what follows we prove this formally. We start with the following theorem (its proof appears in the full version of this paper).

Theorem 5. Let n, be a integer bound on the max size of admissible messages. If $\varGamma$ is a ϵ one-SAD-IND-secure functional encryption for orthogonality (where ϵ denotes the advantage of adversary attacking the security of the scheme), then $\varGamma$ is also $2^{2n}\epsilon$ one-IND-secure.

Claim. If $\varGamma$ is a ϵ'-one-IND-secure functional encryption for orthogonality, then it is also $(q+1)\epsilon'$-IND-secure (where q is the number of ciphertext challenges).

The proof is a straightforward hybrid argument.

Claim. If $\varGamma_1$ and $\varGamma_2$ are respectively ϵ_1 and ϵ_2-IND-secure functional encryption schemes for orthogonality, then the construction from section 5.3 is $(\epsilon_1+\epsilon_2)$-IND-secure.

The proof is the same as that given in section 5.3 and is, therefore, omitted. Putting together all the claims we have the following result.

Corollary 2. If $\varGamma_1$ and $\varGamma_2$ are ϵ-one-SAD-IND-secure OFE, then our proposed construction is $2^{2n}((q_\mathsf{x}+1)+(q_\mathsf{y}+1))\epsilon$-IND-secure FH-OFE scheme (where n is the length of the messages and q_x and q_y are respectively the number of ciphertext and secret key challenges).

Thus, the total factor of security that we will lose is $2^{2n}((q_\mathsf{x}+1)+(q_\mathsf{y}+1))$.

6 Experimental evaluation

We have implemented our new OFE schemes in C++ starting from Shoup's Number Theory Library[11] (NTL) on top of the GNU Multiprecision Library[12] (GMP), and in integration with and the SCIPR Lab's library for Finite Fields and Elliptic Curves[13] (libff). We used NTL to deal with matrix and vector operations carried out in the exponents, and libff as a provider for the pairing group. Conversions between the NTL representations and the libff representations make the implementation sub-optimal in terms of performance in key generation and encryption. No such conversions are needed for decryption. We used the pairing group over a curve known as BN128 from libff, aka BN254,[14] which is deployed

[11] https://www.shoup.net/ntl/

[12] https://gmplib.org/

[13] https://github.com/scipr-lab/libff

[14] https://github.com/zcash/zcash/issues/2502

	GGM			SM		
N	Extract	Encrypt	Decrypt	Extract	Encrypt	Decrypt
16	6	2	10	36	15	60
32	12	4	19	71	28	116
64	22	9	37	139	60	231
128	46	20	73	270	112	463
256	100	44	155	558	229	968

	GGM		SM	
N	Keys	Cph	Keys	Cph
16	0,99	0,50	6,34	3,18
32	1,99	1,00	12,30	6,16
64	3,98	1,99	24,23	12,14
128	7,95	3,98	48,09	24,09
256	15,91	7,97	95,81	48,00

Table 2. Benchmarking results for our generic-group-model construction (GGM) and our standard-model construction (SM). On the left-hand side, timing values are given in milliseconds. On the righ-hand side, key and ciphertext lengths are given in kilobytes. Each row corresponds to an increasing vector size N. Although similar in terms of group operations, the execution times and sizes for keys and ciphertexts differ due to the different sizes of representations of G_1 and G_2 elements in an asymmetric pairing.

for example in ZCash but gradually being abandoned due to the fact that it offers less than 128 bits of security.[15] All our implementations are single-threaded, and could be further optimized via parallelization. For all of these reasons, we present this implementation as a proof of concept, aiming to give an approximate idea of the performance one might get if deploying such schemes. The implementation is available upon request.

Our benchmarking results were collected in a standard MacBook Pro machine with a 2.9 GHz Intel Core i5 and 16 GB or RAM. For every chosen set of parameters, we repeated the experiment 10 times, and took the median of the timings. In all cases we observed a coefficient of variation below 10%. Table 2 provides execution times and key/ciphertext lengths for growing sizes of key/message vectors. For our standard model construction, note that we are actually using double-sized vectors, in order to guarantee full security according to the discussion in Section 5. We observe the linear growth in both execution times and key/ciphertext length, which is to be expected, and highlight the fact that the overhead of going for a standard-model security guarantee is roughly 6-fold. The most insteresting conclusion we can draw, although not surprising due to the close match between our GGM scheme and that proposed in [14], is that our implementation is roughly twice as fast for the same security level (112-bits) than the results reported for the original inner-product encryption scheme. This shows that we bridged the gap between the two primitives with essentially no efficiency loss (this is explained by the fact that we deal with a generic attacker).

7 Applications of Function-Hiding OFE

Our function-hiding OFE constructions can be applied in all the scenarios where secret-key functional encryption for hyperplane-membership [12,8] and hidden-vector encryption [10] are used. These include outsourcing of computations of CNF/DNF Boolean formulas, outsourcing subset relations and range queries on encrypted data. In particular, in the latter example no information is leaked

[15] https://twitter.com/pbarreto/status/779852921135476738

about encrypted data and the query, besides the value of the predicate itself. Indeed, since our constructions are function-hiding, they also imply property-revealing encryption schemes [7] for such predicates. To see this, consider the construction of a property-revealing encryption scheme where an encryption of message $\boldsymbol{x}$ consists of both an encryption and a key token for $\boldsymbol{x}$ under our function-hiding OFE. Then, the orthogonality relation can be publicly computed over all pairs of encrypted messages as in the property revealing setting. In fact, this construction gives rise to a single-key two-input functional encryption scheme, which in turn implies a property-revealing encryption scheme [14].

Furthermore, both our GGM construction and our standard model construction are the most efficient to date under comparable assumptions. However, our standard model construction comes with a message space constraint due to the application of a complexity leveraging argument that we use to achieve full adaptivity.

We therefore focus our attention on applications of function-hiding OFE where this constraint is not a limitation. Our goal is to emphasize that the optimized complexity leveraging argument that we give in Section 5 is crucial to validate our standard model construction for applications where adaptive security is a requirement.

We recall that all our schemes can securely operate over message sizes of roughly $|\mathcal{M}| = q^n$, where q is the cardinality of the cyclic groups over which the schemes are implemented and n is the vector length. However, our standard model scheme from Section 5 achieves only selective security for both keys and messages. A naive complexity leveraging argument to obtain adaptive security would therefore lead to a security loss in the range of $|\mathcal{M}|^{k+1}$, where k is an upper bound on the number of key extraction queries that the scheme should tolerate. However, in Section 5 we have shown how to obtain adaptive security with only $|\mathcal{M}|$ loss. This motivates our analysis of applications of function-hiding OFE where only a small fraction of the full message space $|\mathcal{M}| \approx 2^n \ll q^n$ is used. We stress that no such restrictions apply to our GGM construction, which therefore can be used to replace with better performance all applications of OFE proposed in the literature.

Privacy-preserving subset relation Let us consider a universe $\mathcal{U}$ of n elements $u_1, \ldots, u_n$ and the following two representations of sets $A, B \subseteq \mathcal{U}$ in this universe as vectors $\boldsymbol{x}$, $\boldsymbol{y}$ of length $n+1$ such that

$$\mathsf{mRep}(A) := \begin{cases} \boldsymbol{x}_i = 1 & \text{if } u_i \in A, 1 \leq i \leq n \\ \boldsymbol{x}_i = 0 & \text{if } u_i \notin A, 1 \leq i \leq n \\ \boldsymbol{x}_{n+1} = -1 & \end{cases}$$

$$\mathsf{kRep}(B) := \begin{cases} \boldsymbol{y}_i = 1 & \text{if } u_i \in B, 1 \leq i \leq n \\ \boldsymbol{y}_i = 0 & \text{if } u_i \notin B, 1 \leq i \leq n \\ \boldsymbol{y}_{n+1} = |B| & \end{cases}$$

Clearly, $\langle \mathsf{mRep}(A), \mathsf{kRep}(B) \rangle = 0$ if and only if $B \subseteq A$. Furthermore, the power set $\mathcal{P}(\mathcal{U})$ has size 2^n and both of these representations give injective mappings from $\mathcal{P}(\mathcal{U})$ to $\mathbb{F}_q^{n+1}$. This means that, by using these encodings to compute the subset relation over $\mathcal{P}(\mathcal{U})$, we are in effect operating over a message space of size 2^n.

The computation of the subset relation over a universe of small size can therefore be securely outsourced to an untrusted server with full adaptivity (i.e., new messages can be encrypted interleaved with query evaluations) with the guarantee that the orthogonality predicate over all message/key pairs is leaked to the untrusted server. Furthermore, no information is leaked to an external observer or a snapshot adversary that just observes encrypted messages at rest.

One direct application of this primitive is to allow topological sorting over encrypted data, as any partial order can be computed by using the subset relation. Another application of the subset relation is conjunction keyword search: fix a dictionary of keywords of size n and for each document in a database, encrypt the set of keywords that match that document; then the subset relation can be used to identify all the documents that match all the keywords in the set associated with an extracted key. This subsumes the simplest form of single-key symmetric searchable encryption and reduces leakage for conjunctive queries by hiding the size of the matched subset. However, the security loss of our scheme requires impractically small dictionaries. Next however, we consider two other applications of the subset relation where this is not the case.

Range Queries A standard method to encode range queries of the sort $a < x < b$ is to partition the range of values that x can take into n disjoint intervals of equal size $0 < i_1 < i_2 < \ldots$, and then encode x as the singleton $\{i_k\}$ such that $i_{k-1} \leq x < i_k$. Let I_x be the representation of a value x. Then, the check $i_a \leq x < i_b$ can be computed as $I_x \subseteq \{i_a, \ldots, i_b\}$. This also applies to cases where x is represented in generalized form as belonging to a range of more than one intervals. Our standard-model function-hiding OFE therefore permits dealing with range queries whenever the granularity of the used intervals is acceptable for reasonably small n. In particular, for x coming from a small domain, the same technique can be used to implement the comparison operator and therefore implies a standard order revealing encryption scheme. For implications and optimized variants of these techniques we refer the interested reader to, e.g., [21].

Access Control It is well known that access-control and, more generally, dataflow control restrictions can be represented as partial orders, and therefore implemented using a set representation and the subset relation. Then, the enforcement of an access control mechanism can be outsourced to an untrusted remote server, while keeping the details of the security lattice secret. For example, consider a database of encrypted resources stored in the remote server, each along with an encryption of the point in the access-control lattice that defines the minimal set of permissions A required to access it. Then, by providing the server with a decryption key for an OFE that encodes the set of permissions assigned to a user B, the server can decide whether the operation is allowed by computing $A \subseteq B$. Any security lattice with n nodes is isomorphic to a partially ordered subset of the power set $\mathcal{P}([n])$, and can be therefore outsourced with our standard model scheme if n is reasonably small.

Acknowledgements. This work was supported in part by Royal Society grant for international collaboration and by the European Union Horizon 2020 Research and Innovation Programme under grant agreement 780108 (FENTEC). The first author is financed by Project NanoSTIMA (NORTE-01-0145-FEDER-000016) through the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement and the ERDF.

References

1. Abdalla, M., Bourse, F., Caro, A.D., Pointcheval, D.: Simple functional encryption schemes for inner products. Public-Key Cryptography - PKC 2015. Proceedings, pp. 733–751 (2015)
2. Abdalla, M., Bourse, F., Caro, A.D., Pointcheval, D.: Better security for functional encryption for inner product evaluations. IACR Cryptology ePrint Archive 2016, 11 (2016)
3. Abdalla, M., Catalano, D., Fiore, D., Gay, R., Ursu, B.: Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. Advances in Cryptology - CRYPTO 2018. Proceedings, Part I, pp. 597–627 (2018)
4. Abdalla, M., Gay, R., Raykova, M., Wee, H.: Multi-input inner-product functional encryption from pairings. Advances in Cryptology - EUROCRYPT 2017. Proceedings, Part I, pp. 601–626 (2017)
5. Agrawal, S., Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption: New perspectives and lower bounds. Advances in Cryptology - CRYPTO 2013. Proceedings, Part II, pp. 500–518 (2013)
6. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. Advances in Cryptology - CRYPTO 2016. Proceedings, Part III, pp. 333–362 (2016)
7. Boneh, D., Lewi, K., Raykova, M., Sahai, A., Zhandry, M., Zimmerman, J.: Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. E. Oswald, M. Fischlin (eds.) Advances in Cryptology - EUROCRYPT 2015. Proceedings, Part II, Lecture Notes in Computer Science, vol. 9057, pp. 563–594. Springer (2015). DOI 10.1007/978-3-662-46803-6_19. URL https://doi.org/10.1007/978-3-662-46803-6_19
8. Boneh, D., Raghunathan, A., Segev, G.: Function-private subspace-membership encryption and its applications. K. Sako, P. Sarkar (eds.) Advances in Cryptology - ASIACRYPT 2013. Proceedings, Part I, Lecture Notes in Computer Science, vol. 8269, pp. 255–275. Springer (2013). DOI 10.1007/978-3-642-42033-7_14. URL https://doi.org/10.1007/978-3-642-42033-7_14
9. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. Theory of Cryptography. Proceedings, pp. 253–273 (2011)
10. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. Theory of Cryptography. Proceedings, pp. 535–554 (2007)
11. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. J. Cryptology 30(1), 242–288 (2017)
12. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. J. Cryptology 26(2), 191–224 (2013)
13. Kawai, Y., Takashima, K.: Predicate- and attribute-hiding inner product encryption in a public key setting. Pairing-Based Cryptography - Pairing 2013, Revised Selected Papers, pp. 113–130 (2013)

14. Kim, S., Lewi, K., Mandal, A., Montgomery, H.W., Roy, A., Wu, D.J.: Function-hiding inner product encryption is practical. IACR Cryptology ePrint Archive 2016, 440 (2016)
15. Lin, H.: Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 prgs. Advances in Cryptology - CRYPTO 2017. Proceedings, Part I, pp. 599–629 (2017)
16. Lin, H., Vaikuntanathan, V.: Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016. Proceedings, pp. 11–20 (2016)
17. Okamoto, T., Takashima, K.: Adaptively attribute-hiding (hierarchical) inner product encryption. Advances in Cryptology - EUROCRYPT 2012. Proceedings, pp. 591–608 (2012)
18. Okamoto, T., Takashima, K.: Efficient (hierarchical) inner-product encryption tightly reduced from the decisional linear assumption. IEICE Transactions 96-A(1), 42–52 (2013)
19. O'Neill, A.: Definitional issues in functional encryption. IACR Cryptology ePrint Archive 2010, 556 (2010)
20. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. Theory of Cryptography - TCC 2009. Proceedings, pp. 457–473 (2009)
21. Shi, E., Bethencourt, J., Chan, T.H.H., Song, D., Perrig, A.: Multi-dimensional range query over encrypted data. Proceedings of the 2007 IEEE Symposium on Security and Privacy, SP '07, pp. 350–364. IEEE Computer Society, Washington, DC, USA (2007). DOI 10.1109/SP.2007.29. URL https://doi.org/10.1109/SP.2007.29
22. Wee, H.: Attribute-hiding predicate encryption in bilinear groups, revisited. Theory of Cryptography - TCC 2017. Proceedings, Part I, pp. 206–233 (2017)