Published in D. Catalano and R. De Prisco, Eds., *Security and Cryptography for Networks (SCN 2018)*, vol. 11035 of *Lecture Notes in Computer Science*, pp. 527–542, Springer, 2018.

Function-Revealing Encryption
Definitions and Constructions

Marc Joye[1] and Alain Passelègue[2]

[1] NXP Semiconductors (USA)
[2] UCLA (USA)

Abstract. Multi-input functional encryption is a paradigm that allows an authorized user to compute a certain function—and nothing more—over multiple plaintexts given only their encryption. The particular case of two-input functional encryption has very exciting applications, including comparing the relative order of two plaintexts from their encrypted form (order-revealing encryption).
While being extensively studied, multi-input functional encryption is not ready for a practical deployment, mainly for two reasons. First, known constructions rely on heavy cryptographic tools such as multilinear maps. Second, their security is still very uncertain, as revealed by recent devastating attacks.
In this work, we investigate a simpler approach towards obtaining practical schemes for functions of particular interest. We introduce the notion of function-revealing encryption, a generalization of order-revealing encryption to any multi-input function as well as a relaxation of multi-input functional encryption. We then propose a simple construction of order-revealing encryption based on function-revealing encryption for simple functions, namely orthogonality testing and intersection cardinality. Our main result is an efficient order-revealing encryption scheme with limited leakage based on the standard DLin assumption.

Keywords: Order-revealing encryption; property-preserving encryption; multi-input functional encryption; function-revealing encryption.

1 Introduction

The growing reliance on numerous cloud-based services for storing and processing sensitive data demonstrated limitations of traditional encryption techniques. Specifically, traditional encryption is an all-or-nothing notion: informally, an unauthorized user (i.e., who has not access to the decryption key) should not learn any information whatsoever about a plaintext given its encryption. But in many use cases, there is often a need to get a much more fine-grained control of the decryption policy.

(Multi-Input) Functional Encryption. The paradigm of *functional encryption* [9,30] is an extension of traditional encryption that enables an authorized user to compute a certain function of the plaintext. Each decryption key sk_f

corresponds to a specific function f. Informally, this private key sk_f, given the encryption of a plaintext x, allows her holder to learn $f(x)$, and nothing more. An important subclass of functional encryption is *predicate encryption* [10,24]. A plaintext x is viewed as pair $(I, \dot{x})$ where I is some attribute (associated to the message) and $\dot{x}$ is the message itself; functionality f is then defined as

$$f(I, \dot{x}) = \begin{cases} \dot{x} & \text{if } P(I) = 1, \text{ and} \\ \perp & \text{otherwise} \end{cases}$$

for a given predicate P.

The function can be defined over *multiple* plaintexts given their corresponding ciphertexts. This gives rise to multi-input functional encryption introduced in [19,8]. Of particular interest is the case of two-input functional encryption. Suppose that given two encrypted plaintexts, a cloud-based service wishes to compute their respective ordering. For a *public* comparison function, such a functionality is offered by *order-revealing encryption* (ORE) [6,8]. We note that order-revealing encryption necessarily requires *secret-key* encryption as otherwise a binary search from the encryption of chosen plaintexts would yield bit-by-bit the decryption of a given target ciphertext using the ORE comparison procedure. ORE can thus be seen as a secret-key two-input functional encryption for (public) comparison. It is a very useful primitive as it allows one to answer queries over encrypted data, including range queries, sorting queries, searching queries, and more [1,5].

From OPE to ORE. Order-revealing encryption evolved from *order-preserving encryption* (OPE) [5,6], an encryption primitive that preserves the relative ordering of the plaintexts. Clearly, an OPE scheme cannot achieve the standard security notion of *indistinguishability under chosen-plaintext attacks* (IND-CPA). The best we can hope from an OPE scheme is that the encryption of a sequence of plaintexts reveals nothing beyond their relative ordering, the resulting security notion is termed IND-OCPA. Unfortunately, Boldyreva *et al.* showed in [5] that it is impossible to efficiently meet this natural security notion of IND-OCPA, even when the size of the ciphertext space is exponentially larger than that of the message space.

The situation for ORE schemes is different. In [8], Boneh *et al.* present an ORE scheme actually meeting the analogue of IND-OCPA security. But their construction is mostly of existential nature and as such should be considered as a possibility result. The candidate ORE scheme presented in [8] is hardly implementable since it relies on heavy cryptographic tools, namely $(\ell/2 + 1)$-way multilinear maps for comparing ℓ-bit values. Furthermore, and maybe more importantly, the underlying security assumption is questionable owing to the recent attacks mounted against multilinear maps [15,16].

ORE in Practice. A practical construction for order-revealing encryption is proposed in [14]. It merely requires a pseudorandom function F with output space $\{0, 1, 2\}$. The encryption under secret key K of an ℓ-bit plaintext $x = m_1 m_2 \cdots m_\ell$

with $m_i \in \{0,1\}$, $\mathsf{ct} = (c_1, c_2, \ldots, c_\ell)$, is obtained iteratively as

$$c_i = \left[F\big(K, (i, m_1 m_2 \cdots m_{i-1} \| 0^{\ell-i})\big) + m_i\right] \bmod 3\,, \quad \text{for } 1 \leq i \leq \ell\ .$$

The comparison of two ciphertexts $\mathsf{ct} = (c_1, c_2, \ldots, c_\ell)$ and $\mathsf{ct}' = (c'_1, c'_2, \ldots, c'_\ell)$, corresponding to plaintexts x and x', is conducted by finding the first index i, $1 \leq i \leq \ell$, such that $c'_i \neq c_i$. Then,

$$\begin{cases} x < x' & \text{if there exists such an index } i \text{ and if } c'_i \equiv c_i + 1 \pmod 3 \\ x \geq x' & \text{otherwise} \end{cases}\ .$$

While this construction is very efficient, it has the drawback of leaking an important amount of information, as one obtains immediately, given two ciphertexts, the size of the largest common prefix of the two corresponding plaintexts. In particular, this provides an upper bound on the distance separating the two plaintexts.

Our Contributions. In this work, we investigate a new approach towards building efficient secret-key multi-input functional encryption. We propose the notion of function-revealing encryption, which can be viewed both as a generalization of the notion of property-preserving encryption [13,29] and as a specialization of the notion of multi-input functional encryption. Basically, a function-revealing encryption scheme is a secret-key encryption scheme associated to a k-ary function f. The encryption algorithm takes as input a secret key, a message, and some index $i \in [k]$ and outputs a ciphertext. Moreover, there exists a public procedure such that, given k ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$, each corresponding to an encryption of a message x_i at index i, for $i \in [k]$, one can compute $f(x_1, \ldots, x_k)$. In particular, considering the comparison function defined as:

$$f_<\colon (x, y) \mapsto \begin{cases} 1 & \text{if } x < y \\ 0 & \text{otherwise} \end{cases}\ ,$$

our notion matches precisely the notion of order-revealing encryption.

We note that our general framework slightly generalizes the definition of order-revealing encryption, since the original definition is "symmetric" and ours is "asymmetric" (in the sense that our definition only allows to compare a ciphertext with index 1 with a ciphertext with index 2). This is without loss of generality since a symmetric scheme results immediately from an asymmetric scheme.

We consider two (indistinguishability-based and simulation-based) security notions that take into account a possible leakage. The leakage comprises at least the information resulting from the evaluation function, which is unavoidable. However, contrary to a perfect solution that would only permit this unavoidable leakage (as the one offered in [8]), we allow for additional leakage, provided it is very limited. Doing so, we are able to devise constructions that can be used in practical applications.

We then focus on the particular case of 2-ary functions (so the index is 1 or 2) and specifically on building efficient order-revealing encryption. Our main

construction is an efficient order-revealing encryption scheme with limited leakage, under standard assumptions.

Our Techniques. We first show that one can build an order-revealing encryption scheme given only a function-revealing encryption scheme for the function computing the cardinality of the intersection of two sets $f_{\#}\colon (\mathcal{S}, \mathcal{T}) \mapsto \#(\mathcal{S} \cap \mathcal{T})$. This result follows from a fairly simple technique to compare two bitstrings. Consider two bitstrings of same length $x = x_1 \,\|\, \ldots \,\|\, x_n$ and $y = y_1 \,\|\, \ldots \,\|\, y_n$, then we have $x < y$ if and only if there exists $i \in [n]$ such that $x_j = y_j$ for every $j < i$ and $x_i = 0$ and $y_i = 1$. Thus, we have $x < y$ if and only if there exists a prefix $z \,\|\, 0$ of x with $z \in \{0,1\}^*$ such that $z \,\|\, 1$ is a prefix of y, and one can then compare x and y by checking if the sets $\{z \,\|\, 1 \mid z \,\|\, 0 \text{ is a prefix of } x\}$ and $\{z \,\|\, 1 \mid z \,\|\, 1 \text{ is a prefix of } y\}$ are disjoint.

The next step is then to construct a function-revealing encryption scheme for intersection cardinality. We show that one can build such a scheme with only limited leakage based on the existence of function-revealing encryption for the function checking the orthogonality of two vectors $f_{\perp}\colon (\vec{a}, \vec{b}) \mapsto \langle \vec{a}, \vec{b} \rangle = 0$ (this function outputs the value of the predicate $\langle \vec{a}, \vec{b} \rangle = 0$). This transformation relies on the following technique to compute the cardinality of the intersection. Consider two sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$. A simple way to compute $\#(A \cap B)$ is to evaluate the polynomial $\prod_{i=1}^{n}(X - a_i)$ whose roots are the elements of A on every b_j for $j \in [m]$, and to return the number of times this evaluates to 0. This can be done by computing inner products, since $\prod_{i=1}^{n}(X - a_i)$ is a degree n polynomial that can be written as $\sum_{i=0}^{n} \alpha_i X^i$ and thus, we have

$$\prod_{i=1}^{n}(b - a_i) = \langle (\alpha_0, \ldots, \alpha_n), (1, b, b^2, \ldots, b^n) \rangle \,,$$

so checking if this evaluates to 0 corresponds precisely to checking if the above vectors are orthogonal.

Finally, we show that one can build a function-revealing encryption scheme for $f_{\perp}$ under the standard DLin assumption. In particular, we show that any fully-secure predicate encryption scheme for a class of predicate $\mathcal{P} = \{P_a\colon b \mapsto P(a,b)\}$ can be turned into a function-revealing encryption scheme for the function P.

Yet, there is a small catch in our transform from orthogonality to intersection cardinality. Indeed, the above function-revealing encryption scheme for $f_{\#}$ not only reveals the cardinality of the intersection, but also the elements of B that are in A, as each element b_i of B is encrypted separately (by encrypting the corresponding vector $(1, b_i, b_i^2, \ldots, b_i^n)$). In particular, even if b_i is hidden, intersecting A with B and A' with B might also reveal some information about the intersection of A and A' (for instance, if $b_i \in A$ and $b_i \in A'$, then we also learn that $A \cap A' \neq \varnothing$ which should not have been revealed). Thus, our construction reveals a bit more than what we would like ideally. We briefly discuss how one can reduce this leakage by reducing the efficiency of our construction (though the only way to obtain ideal leakage with our technique is by having exponential-size ciphertexts). Also, we note that our leakage *is* ideal if there

is only one set A that is encrypted at index 1, whatever the number of sets $B, C, \ldots$ encrypted at index 2 (or more generally for bounded ciphertexts at index 1 and unbounded ciphertexts at index 2). Finally, since our transformation from intersection cardinality to relative order is generic, any improvement on the security of the underlying scheme for intersection cardinality (both in terms of efficiency and of leakage) would immediately results in an improved construction of order-revealing encryption.

Of independent interest, we also provide a very simple order-revealing encryption scheme achieving the best possible security for short messages, assuming only the existence of one-way functions.

Related Works. Order-revealing encryption has been an subject to several improvements in the last years. In a recent work by Lewi and Wu [27], the authors proposed a similar construction for short messages and extended it to larger domains by adding leakage and using random oracles. Intuitively, for plaintexts of length $n \cdot k$, they encrypt small blocks of k bits with the perfect scheme (whose complexity is exponential in k), and then compose with the scheme from [14] for each of the n blocks. Thus, the ciphertexts reveal the position of the first differing blocks of k bits.

Another recent work by Durak, DuBuisson, and Cash [18] analyzed what is revealed by perfect order-revealing encryption. They show that the ideal functionality already reveals important information for certain applications of ORE (e.g., when plaintexts come from particular distributions). This work emphasizes that an important leakage could be devastating, so reducing the leakage as much as possible (while preserving good efficiency due to the practical importance of ORE) is of prime interest. Our work proposes a first step towards obtaining smaller leakage (in particular achieving ideal leakage in restricted cases). A recent work by Cash, Liu, O'Neill, and Zhang [12] also makes a step in this direction. In this work, the authors construct an order-revealing encryption scheme with limited leakage under SXDH. Their construction is more efficient than ours (basing our construction on current state-of-the-art fully-secure IPE [25]) but their leakage is slightly worse than ours, our construction beneficing from its asymmetry. They obtain a construction, based on pairings, that only leaks the equality pattern of the most significant differing bit (that is, for any 3 plaintexts m_0, m_1, m_2, whether the most significant differing bit of m_0 and m_1 is the same as the one of m_0 and m_2), while the construction from [14] reveals the position of the most significant differing bit. Despite the similarity of our results, both constructions are significantly different in terms of techniques, as [12] is based on the work by Chenette *et al.* [14] while our work opens a new path. In particular, any improvement of our building blocks (e.g., more efficient fully-secure IPE or construction for cardinality of intersection with smaller leakage, or for disjointness, as we explain in Remark 8) would immediately benefit to our ORE scheme.

Concerning multi-input functional encryption, a recent work by Brakerski, Komargodski, and Segev [11], improved in [26], propose a more general approach that allows going from single-input functional encryption to t-input functional encryption in the private key-setting, as long as t is constant (or poly-logarithmic

assuming quasi-polynomial security). In particular, this allows one to obtain function-revealing encryption scheme for functions with t-arity from LWE [20] (or from low-complexity PRG and public-key encryption [21]) for a bounded number T of ciphertexts for one index and unbounded ciphertexts for the others (where the size of the ciphertexts grow with T and the depth of the circuit computing the function). A similar result for the case of 2-arity functions can also be obtained directly from the reusable garbled circuits construction from [20]. The general case with unbounded ciphertexts at both indexes remains out of reach since it requires unbounded-collusion functional encryption, which is not known from standard assumptions (and implies $i\mathcal{O}$ [4,26] up to subexponential security).

Finally, our notion of function-revealing encryption has also been defined in a recent work (as "revealing encryption") by Haagh, Ji, Li, Orlandi, and Song [22]. In this paper, the authors also propose a function-revealing scheme for the comparison of two vectors ($x_i < y_i$ or $x_i \geq y_i$ for all i or $\vec{x}$ and $\vec{y}$ are incomparable). Their construction is obtained by extending the order-revealing construction from [14] and thus implies as well an important leakage.

2 Definitions

2.1 Function-Revealing Encryption

We introduce the paradigm of function-revealing encryption (FRE), as a generalization of property-preserving encryption defined by Pandey and Rouselakis [29] as well as a weakening of the general notion of multi-input functional encryption [8,19]. Our notion assumes the private-key setting [31] and corresponds to *dedicated* multi-input functional encryption schemes where the evaluation of the function is public (i.e., no functional secret key is involved).

Definition 1 (Function-Revealing Encryption). *A* function-revealing encryption scheme *for a k-ary function f consists of a tuple of algorithms $\mathcal{FRE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}_f)$, defined below.*

- $\mathsf{Setup}(1^\kappa)$ *is a probabilistic algorithm that takes as input the security parameter 1^κ and outputs a secret key* sk *(and public parameters* pp*—including the message space $\mathcal{M}$).*
- $\mathsf{Enc}(i, \mathsf{sk}, x)$ *takes as input an index $i \in [k]$, a key* sk*, and a message $x \in \mathcal{M}$. It outputs a ciphertext* ct*.*
  Index i indicates that the output ciphertext ct *constitutes the i-th input to function f.*
- $\mathsf{Eval}_f(\mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ *takes as input k ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ and outputs a value y in the range of f.*

For correctness, it is required that for all $\mathsf{sk} \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$ *and all $(x_1, \ldots, x_k) \in \mathcal{M}^k$:*

$$\mathsf{Eval}_f(\mathsf{ct}_1, \ldots, \mathsf{ct}_k) = f(x_1, \ldots, x_k) \quad \textit{where } \mathsf{ct}_i = \mathsf{Enc}(i, \mathsf{sk}, x_i) \ .$$

Remark 2. 1. Definition 1 is "asymmetric" in the sense that a given ciphertext is bound to a specific input position in the Eval_f procedure. We could define a "symmetric" version of function-revealing encryption where the encryption algorithm Enc no longer takes in an index $i \in [k]$ so that a ciphertext can be used in any input position for the Eval_f procedure. We do not study this symmetric version further since, as stated in Lemma 5, it is implied by the asymmetric version.

2. We choose not to include a decryption algorithm in our definition, since this omission is without loss of generality. Indeed, if necessary, one could just augment the encryption of a message x with an encryption of x with a CPA-secure symmetric encryption scheme under a specific secret-key. Via CPA-security, this additional information does not compromise the security of the construction.

3. As specified in the introduction, we focus on the three following functions:
   - $f_{\perp}\colon (\vec{a}, \vec{b}) \mapsto \begin{cases} 1 & \text{if } \langle \vec{a}, \vec{b} \rangle = 0 \\ 0 & \text{otherwise} \end{cases}$;
   - $f_{\#}\colon (\mathcal{S}, \mathcal{T}) \mapsto \#(\mathcal{S} \cap \mathcal{T})$;
   - $f_{<}\colon (x, y) \mapsto \begin{cases} 1 & \text{if } x < y \\ 0 & \text{otherwise} \end{cases}$.

2.2 Two Security Flavors

We examine two different security notions and explore the relations between them. The first notion is defined as an indistinguishability-based security game, while the second (and stronger) one as a simulation-based security game. These are generalizations of classical notions considered in the case of property-preserving encryption, e.g. in [3,13,17,29].

The two notions are defined relatively to a leakage function $\mathcal{L}$. As a FRE scheme for a function f has to reveal, via the Eval_f procedure, at least the values of the function f according to any tuple of k messages $x_1, \ldots, x_k$ such that x_i is encrypted for index $i \in [k]$, $\mathcal{L}$ will contain at least this information. This leakage is written $\mathcal{L}_f$ and is defined below.

Definition 3 (Leakage of a Function). *The* leakage $\mathcal{L}_f$ of a k-ary function f *with respect to k vectors $\vec{x}_1, \ldots, \vec{x}_k$ of $q_1, \ldots, q_k$ messages respectively —one vector of messages per position in the input of the function, so $\vec{x}_i = (x_{i,1}, \ldots, x_{i,q_i})$— is defined as:*

$$\mathcal{L}_f(\vec{x}_1, \ldots, \vec{x}_k) = (f(x_{1,i_1}, \ldots, x_{k,i_k}))_{i_1 \in [q_1], \ldots, i_k \in [q_k]} \ .$$

$\mathcal{L}$-Indistinguishability Security. A FRE scheme $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}_f)$ for a k-ary function f is $\mathcal{L}$-indistinguishability secure if, for any two sequences of plaintexts with the same leakage, the corresponding sequences of ciphertexts are computationally indistinguishable. Security is defined by a variant of the standard semantic security game and is depicted in Figure 1.

$\underline{\textbf{proc Initialize}}$
$b \leftarrow \{0,1\}$
$\mathsf{sk} \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$
For $i \in [k]$:
  $\vec{\ell}_i^{(0)}, \vec{\ell}_i^{(1)} \leftarrow ()$

$\underline{\textbf{proc Finalize}(b')}$
Return $b' = b$

$\underline{\textbf{proc LoR}(i, x^{(0)}, x^{(1)})}$
$\vec{\ell}_i^{(0)} \leftarrow \vec{\ell}_i^{(0)}.\mathsf{append}(x^{(0)})$
$\vec{\ell}_i^{(1)} \leftarrow \vec{\ell}_i^{(1)}.\mathsf{append}(x^{(1)})$
If $\mathcal{L}(\vec{\ell}_1^{(0)}, \ldots, \vec{\ell}_k^{(0)}) \neq \mathcal{L}(\vec{\ell}_1^{(1)}, \ldots, \vec{\ell}_k^{(1)})$:
  Return $\perp$
Else:
  $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(i, \mathsf{sk}, x^{(b)})$
  Return ct

Fig. 1. Game defining the $\mathcal{L}$-indistinguishability security of a FRE scheme.

Specifically, the adversary has black-box access to a left-or-right encryption oracle **LoR**. This oracle can be adaptively queried with an index i and a pair of messages $(x^{(0)}, x^{(1)})$ to get $\mathsf{Enc}(i, \mathsf{sk}, x^{(b)})$ with b being a fixed bit and sk being a secret key, initialized by the **Initialize** procedure. At the end, the adversary outputs a bit b' and wins if $b = b'$; namely, $\textbf{Finalize}(b') = 1$. In order to prevent trivial attacks (i.e., attacks resulting from the leakage function), the adversary is restricted as follows. If $((x_{i,1}^{(0)}, x_{i,1}^{(1)}), \ldots, (x_{i,q_i}^{(0)}, x_{i,q_i}^{(1)}))$ denotes the sequence of q_i queries made with index i to the **LoR** oracle then, letting $\vec{x}_i^{(t)} = (x_{i,1}^{(t)}, \ldots, x_{i,q_i}^{(t)})$ for $t \in \{0,1\}$, the sequence of queries made by the adversary has to satisfy:

$$\mathcal{L}\big(\vec{x}_1^{(0)}, \ldots, \vec{x}_k^{(0)}\big) = \mathcal{L}\big(\vec{x}_1^{(1)}, \ldots, \vec{x}_k^{(1)}\big) \ .$$

$\mathcal{L}$-Simulation Security. A FRE scheme $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}_f)$ for a k-ary function f is $\mathcal{L}$-simulation secure if, for any efficient adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_q)$ which is given black-box access to encryption oracle Enc that it queries q times, there exists an efficient stateful simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_q)$ such that the outputs of the two distributions $\mathsf{Real}_{\mathcal{A}}^{\mathcal{FRE}}(\kappa)$ and $\mathsf{Sim}_{\mathcal{A},\mathcal{S},\mathcal{L}}^{\mathcal{FRE}}(\kappa)$, described in Figure 2, are computationally indistinguishable.

$\underline{\textbf{proc } \mathsf{Real}_{\mathcal{A}}^{\mathcal{FRE}}(\kappa)}$
$\mathsf{sk} \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$
$\mathsf{st}_{\mathcal{A}} \leftarrow \mathcal{A}_0(1^\kappa)$
For $i \in [k]$:
  $\vec{\mathsf{ct}}_i \leftarrow ()$
For $C \in [q]$:
  $((i, x), \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_C(\mathsf{st}_{\mathcal{A}}, (\vec{\mathsf{ct}}_1, \ldots, \vec{\mathsf{ct}}_k))$
  $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(i, \mathsf{sk}, x)$
  $\vec{\mathsf{ct}}_i \leftarrow \vec{\mathsf{ct}}_i.\mathsf{append}(\mathsf{ct})$
Return $(\vec{\mathsf{ct}}_1, \ldots, \vec{\mathsf{ct}}_k)$

$\underline{\textbf{proc } \mathsf{Sim}_{\mathcal{A},\mathcal{S},\mathcal{L}}^{\mathcal{FRE}}(\kappa)}$
$\mathsf{st}_{\mathcal{S}} \leftarrow \mathcal{S}_0(1^\kappa)$
$\mathsf{st}_{\mathcal{A}} \leftarrow \mathcal{A}_0(1^\kappa)$
For $i \in [k]$:
  $\vec{\mathsf{ct}}_i \leftarrow ()$; $\vec{x}_i \leftarrow ()$
For $C \in [q]$:
  $((i, x), \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_C(\mathsf{st}_{\mathcal{A}}, (\vec{\mathsf{ct}}_1, \ldots, \vec{\mathsf{ct}}_k))$
  $\vec{x}_i \leftarrow \vec{x}_i.\mathsf{append}(x)$
  $(\mathsf{ct}, \mathsf{st}_{\mathcal{S}}) \xleftarrow{\$} \mathcal{S}_C(\mathsf{st}_{\mathcal{S}}, \mathcal{L}(\vec{x}_1, \ldots, \vec{x}_k))$
  $\vec{\mathsf{ct}}_i \leftarrow \vec{\mathsf{ct}}_i.\mathsf{append}(\mathsf{ct})$
Return $(\vec{\mathsf{ct}}_1, \ldots, \vec{\mathsf{ct}}_k)$

Fig. 2. Game defining the $\mathcal{L}$-simulation security of a FRE scheme.

2.3 Relations Between These Security Notions

As one could expect, simulation security implies indistinguishability security, as stated in the following lemma. Moreover, as already mentioned in Remark 2, for both security notions, the existence of a secure "asymmetric" FRE implies the existence of secure "symmetric" FRE, as stated in Lemma 5. Proofs are detailed in the full version [23].

Lemma 4. *Assuming $\mathcal{FRE}$ is an $\mathcal{L}$-simulation secure function-revealing encryption scheme, then $\mathcal{FRE}$ is an $\mathcal{L}$-indistinguishability secure function-revealing encryption scheme.*

Lemma 5. *Assuming there exists an $\mathcal{L}$-indistinguishability (resp. $\mathcal{L}$-simulation) secure asymmetric function-revealing encryption scheme for a function f, there exists a $\mathsf{sym}_{\mathcal{L}}$-indistinguishability (resp. $\mathsf{sym}_{\mathcal{L}}$-simulation) secure symmetric function-revealing encryption scheme for the function f, with $\mathsf{sym}_{\mathcal{L}}(\vec{x}) = \mathcal{L}(\vec{x}, \ldots, \vec{x})$.*

3 Order-Revealing Encryption with Simulation-Security for Polynomial-Size Message Space

Before starting to build our main construction, which is an efficient function-revealing encryption scheme for the function $f_<$ (i.e., order-revealing encryption scheme) with limited leakage, we would like to start with a simple remark. While it seems extremely hard to obtain an $\mathcal{L}_{f_<}$-indistinguishability secure order-revealing encryption scheme from standard assumptions, there is actually a very simple construction that even achieves simulation-based security assuming only one-way functions. However, this construction is only efficient for polynomial-size message space. To improve efficiency, our construction can be instantiated using a pseudorandom permutation, such as AES. This leads to a very efficient construction for small message spaces (e.g., 10-bit integers).

Let $\{0, \ldots, N-1\}$ denote the message space, and let $F\colon \{0,1\}^\kappa \times \mathcal{D} \to \mathcal{R}$ be a pseudorandom function such that its domain $\mathcal{D}$ contains $\{0, \ldots, N-1\} \times \{0, \ldots, 2(N-1)\}$.

Construction 1 *We define $\mathcal{FRE}_< = (\mathsf{Setup}_<, \mathsf{Enc}_<, \mathsf{Eval}_{f_<})$ as follows:*

- $\mathsf{Setup}_<(1^\kappa)$ *picks* $K \xleftarrow{\$} \{0,1\}^\kappa$ *at random and returns it as the secret key* sk*;*
- $\mathsf{Enc}_<(i, \mathsf{sk}, x)$ *with* $x \in \{0, \ldots, N-1\}$ *is defined as:*

$$\mathsf{Enc}_<(i, K, x) = \begin{cases} \mathsf{shuffle}(F_K(x, x+1), \ldots, F_K(x, x+N-1)) & \text{if } i = 1 \\ \mathsf{shuffle}(F_K(0, x), \ldots, F_K(N-1, x)) & \text{if } i = 2 \end{cases};$$

  [Here $\mathsf{shuffle}$ *is a randomized algorithm that returns a random shuffling of its inputs.]*
- $\mathsf{Eval}_{f_<}(\mathsf{ct}_1, \mathsf{ct}_2)$ *checks whether there is a common value in* ct_1 *and* ct_2*. If so, it outputs* 1 *("<"); if not, it outputs* 0 *("≥").*

Correctness. It is clear that if there is no common value, the output of the evaluation algorithm, "$\geq$", is correct. However, it might happen that there is a common value due to a collision. Hence, to ensure that this does not happen, we might want F_K to be injective (e.g., using a pseudorandom permutation instead of a pseudorandom function, e.g. AES), but one could simply make the range $\mathcal{R}$ big enough so that the probability of a collision is negligible.

Construction 1 being deterministic, it reveals if two ciphertexts encrypted with the same index corresponds to the same plaintext. This is the only extra information, beyond the relative order, that is leaked. However, this extra-information is always leaked in the "symmetric" case, as one can always check, given two ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2$ corresponding to plaintexts x_1, x_2, whether $x_1 \geq x_2$ and $x_2 \geq x_1$. Thus, if $x_1 = x_2$, the equality is revealed. For this reason, we claim that Construction 1 achieves ideal security, and we define its leakage $\mathcal{L}_{<,=}$ as:

$$\mathcal{L}_{<,=}(\vec{x}_1, \vec{x}_2) = (\mathcal{L}_{f_<}(\vec{x}_1, \vec{x}_2), \mathcal{L}_{=}(\vec{x}_1, \vec{x}_2)) \,,$$

with $\mathcal{L}_{=}(\vec{x}_1, \vec{x}_2) = (\mathbb{1}_{=}(x_{b,i_b}, x_{b,j_b}))_{i_b,j_b \in [|\vec{x}_b|], b \in \{1,2\}}$ where $\mathbb{1}_{=}(a,b)$ returns 1 if and only if $a = b$. Precisely, $\mathcal{L}_{f_<}(\vec{x}_1, \vec{x}_2)$ reveals exactly the relative order of messages encrypted with index 1 relatively to messages encrypted with index 2, while $\mathcal{L}_{=}(\vec{x}_1, \vec{x}_2)$ reveals exactly the pairs of equal messages encrypted with the same index.

Theorem 6. *Assuming one-way functions exist, there exists an $\mathcal{L}_{<,=}$-simulation secure function-revealing encryption scheme for the function $f_<$, for polynomial-size message spaces.*

The proof of the above theorem is detailed in the full version [23].

4 Order-Revealing Encryption with Limited Leakage

We now describe how to build an order-revealing encryption scheme (a.k.a. function-revealing encryption scheme for $f_<$) from any function-revealing encryption scheme for $f_\#$. As a preliminary, we explain how one can compare two integers by simply checking the disjointness of two sets.

4.1 From Bitstrings to Sets

We define functions Σ^0 and Σ^1, taking as input an n-bit string x and returning a set of prefixes, as follows:

$$\Sigma^b \colon x \in \{0,1\}^n \longmapsto \Sigma^b(x) = \left\{ x_{n-1} \,\|\, \ldots \,\|\, x_{i+1} \,\|\, 1 \mid x_i = b \right\}_{0 \leq i \leq n-1} \,, \qquad (1)$$

for $b \in \{0,1\}$. That is, $\Sigma^1(x)$ returns the set of every prefix of x that ends with a 1, and $\Sigma^0(x)$ returns the set of every $z \,\|\, 1$ such that $z \,\|\, 0$ is a prefix of x. It is easily seen that $\#\Sigma^1(x) = \mathsf{hw}(x)$ and that $\#\Sigma^0(x) = \mathsf{hw}(\bar{x})$. In particular,

we have $\Sigma^0(1^n) = \Sigma^1(0^n) = \varnothing$ and thus $\#\Sigma^0(1^n) = \#\Sigma^1(0^n) = 0$. It is also immediate that $\#\Sigma^1(x \,\|\, \bar{x}) = \#\Sigma^0(x \,\|\, \bar{x}) = n$, for every $x \in \{0,1\}^n$.

Functions Σ^0 and Σ^1 are useful as they allow computing the relative order of two integers [28]. More precisely, we have:

Lemma 7. *Let x, y be two integers such that $0 \leq x, y < 2^n$ and viewed as n-bit strings. Then*

$$x < y \iff \#\big(\Sigma^0(x) \cap \Sigma^1(y)\big) = 1 \quad \text{and} \quad x \geq y \iff \#\big(\Sigma^0(x) \cap \Sigma^1(y)\big) = 0 \ .$$

Please refer to the full version [23] for the proof.

Remark 8. Since the cardinality of the intersection is always 0 or 1, one could also base our order-revealing encryption scheme on any function-revealing scheme for the disjointness (i.e., that only reveals if two sets intersect or not).

4.2 A Generic Transform from $\mathcal{FRE}_{\#}$ to $\mathcal{FRE}_{<}$

Our transform simply relies on the above technique. Let $\mathcal{FRE}_{\#} = (\mathsf{Setup}_{\#}, \mathsf{Enc}_{\#}, \mathsf{Eval}_{f_{\#}})$ be a function-revealing encryption scheme for the function $f_{\#}$. For simplicity, instead of directly encrypting the sets $\Sigma^0(x)$ or $\Sigma^1(x)$, one encrypts the sets $\Sigma^0(x \,\|\, \bar{x})$ or $\Sigma^1(x \,\|\, \bar{x})$, which are both of size n if x is an n-bit integer. This allows us to assume that the sets encrypted by $\mathcal{FRE}_{\#}$ all have the same size. It is very easy to see that Lemma 7 still holds even if we replace $\Sigma^0(x)$ and $\Sigma^1(y)$ by $\Sigma^0(x \,\|\, \bar{x})$ and $\Sigma^1(y \,\|\, \bar{y})$ respectively.

Construction 2 *We build a function-revealing encryption scheme $\mathcal{FRE}_{<} = (\mathsf{Setup}_{<}, \mathsf{Enc}_{<}, \mathsf{Eval}_{f_{<}})$ for the function $f_{<}$ as follows:*

- $\mathsf{Setup}_{<}$ *takes as input the security parameter κ and outputs* $\mathsf{Setup}_{\#}(1^\kappa) = \mathsf{sk}$*;*
- $\mathsf{Enc}_{<}$ *takes as input an index $i \in \{1,2\}$, a secret key* sk*, and a message x and outputs:*

$$\mathsf{Enc}_{<}(i, \mathsf{sk}, x) = \begin{cases} \mathsf{Enc}_{\#}(1, \mathsf{sk}, \Sigma^0(x \,\|\, \bar{x})) & \text{if } i = 1 \\ \mathsf{Enc}_{\#}(2, \mathsf{sk}, \Sigma^1(x \,\|\, \bar{x})) & \text{if } i = 2 \end{cases} ;$$

- $\mathsf{Eval}_{f_{<}}$ *takes as input a pair of ciphertexts* $(\mathsf{ct}_1, \mathsf{ct}_2)$ *encrypted with index* 1 *and* 2 *respectively, and returns* $\mathsf{Eval}_{f_{\#}}(\mathsf{ct}_1, \mathsf{ct}_2)$*.*

Correctness. The correctness easily follows from the correctness of $\mathcal{FRE}_{\#}$ and from Lemma 7.

Security. Security immediately follows from the security of $\mathcal{FRE}_{\#}$ and the leakage is simply the leakage associated of $\mathcal{FRE}_{\#}$ applied to the encrypted sets, which are either $\Sigma^0(x \,\|\, \bar{x})$ or $\Sigma^1(x \,\|\, \bar{x})$.

Let $\mathcal{L}$ denote a leakage such that $\mathcal{FRE}_{\#}$ is $\mathcal{L}$-indistinguishability secure. Then, we define the leakage of Construction 2 as:

$$\mathscr{L}_{\mathcal{L}}(\vec{x}_1, \vec{x}_2) = \mathcal{L}(\Sigma^0(\vec{x}_1), \Sigma^1(\vec{x}_2)) \,,$$

where $\vec{x}_i = (x_{i,1}, \ldots, x_{i,q_i})$ is the sequence of integers encrypted with index i, for $i \in \{1,2\}$, and where $\Sigma^0(\vec{x}_1) = (\Sigma^0(x_{1,1} \,\|\, \bar{x}_{1,1}), \ldots, \Sigma^0(x_{1,q_1} \,\|\, \bar{x}_{1,q_1}))$, and $\Sigma^1(\vec{x}_2) = (\Sigma^1(x_{2,1} \,\|\, \bar{x}_{2,1}), \ldots, \Sigma^1(x_{2,q_2} \,\|\, \bar{x}_{2,q_2}))$.

Theorem 9. *Assuming there exists an $\mathcal{L}$-indistinguishability secure function-revealing encryption scheme for the function $f_\#$, then there exists an $\mathscr{L}_{\mathcal{L}}$-indistinguishability secure 2-input functional encryption scheme for the function $f_<$.*

Please refer to the full version [23] for the proof.

Remark 10. One could also prove in a very similar manner that the obtained construction is simulation-secure assuming the underlying scheme $\mathcal{FRE}_\#$ is simulation-secure.

4.3 Computing Cardinality of Intersection with Limited Leakage

We now describe the second step in building our efficient order-revealing encryption scheme with limited leakage, which is to build a function-revealing encryption scheme for computing the cardinality of intersection. Specifically, the messages are sets of fixed size n and the function f we target is the function $f_\#\colon (\mathcal{S}_1, \mathcal{S}_2) \mapsto \#(\mathcal{S}_1 \cap \mathcal{S}_2)$. Our construction relies on the existence of a function-revealing encryption scheme for $f_\perp$.

In order to ease the reading, we assume that every set in the message space has a fixed size n. One could circumvent this condition as long as the maximal size of a set is known and fixed in advance, but this is not useful for our purpose.

We compute the cardinality of the intersection of two sets as follows: given two sets of integers $\mathcal{A} = \{a_1, \ldots, a_n\}$ and $\mathcal{B} = \{b_1, \ldots, b_n\}$, one can compute the polynomial $P_\mathcal{A}(X) = \prod_{i=1}^n (X - a_i)$ such that $b \in \mathcal{A} \Leftrightarrow P_\mathcal{A}(b) = 0$. The problem is that this technique does not hide anything about elements in $\mathcal{A}$ and $\mathcal{B}$. To address this issue, one simply notices that, given $P_\mathcal{A}(X) = \sum_{i=0}^n \alpha_i \cdot X^i$, testing $P_\mathcal{A}(b) = 0$ simply consists in checking if $\langle \vec{\alpha}, \vec{\beta} \rangle = 0$, with $\vec{\alpha} = (\alpha_0, \ldots, \alpha_n)$ and $\vec{\beta} = (1, b, b^2, \ldots, b^n)$. Therefore, this can be tested privately using a function-revealing encryption for orthogonality testing.

We denote by $\mathsf{coef}(\mathcal{S})$ the vector $(\alpha_0, \ldots, \alpha_n)$ such that $\prod_{s \in \mathcal{S}}(X - s) = \sum_{i=0}^n \alpha_i \cdot X^i$ and by $\mathsf{exp}(s)$ the vector $(1, s, s^2, \ldots, s^n)$. It is straightforward that, for n being polynomial, computations of $\mathsf{coef}(\mathcal{S})$ and $\mathsf{exp}(s)$ are polynomial-time. Let $\mathcal{FRE}_\perp = (\mathsf{Setup}_\perp, \mathsf{Enc}_\perp, \mathsf{Eval}_{f_\perp})$ be a function-revealing encryption scheme for orthogonality testing.

Construction 3 *We build a function-revealing encryption scheme $\mathcal{FRE}_\# = (\mathsf{Setup}_\#, \mathsf{Enc}_\#, \mathsf{Eval}_{f_\#})$ for the function $f_\#$ as follows:*

- $\mathsf{Setup}_\#$ *takes as input the security parameter κ and outputs $\mathsf{Setup}_\perp(1^\kappa) = \mathsf{sk}$;*
- $\mathsf{Enc}_\#$ *takes as input an index $i \in \{1,2\}$, a secret key sk, and a set $\mathcal{S} = \{s_1, \ldots, s_n\}$ and outputs:*

$$\mathsf{Enc}_\#(i, \mathsf{sk}, \mathcal{S}) = \begin{cases} \mathsf{Enc}_\perp(1, \mathsf{sk}, \mathsf{coef}(\mathcal{S})) & \text{if } i = 1\,; \\ \mathsf{shuffle}(\mathsf{Enc}_\perp(2, \mathsf{sk}, \mathsf{exp}(s_1)), \ldots, \\ \qquad\qquad \mathsf{Enc}_\perp(2, \mathsf{sk}, \mathsf{exp}(s_n))) & \text{if } i = 2\,. \end{cases}$$

– $\mathsf{Eval}_{f_\#}$ *takes as input a pair of ciphertexts* $(\mathsf{ct}_1, \mathsf{ct}_2)$ *encrypted with index* 1 *and* 2 *respectively and with* $\mathsf{ct}_2 = (\mathsf{ct}_{2,1}, \ldots, \mathsf{ct}_{2,n})$*, computes* $y_i = \mathsf{Eval}_{f_\perp}(\mathsf{ct}_1, \mathsf{ct}_{2,i})$ *for* $i = 1, \ldots, n$ *and outputs* $\sum_{i=1}^{n} y_i$*.*

Correctness. Correctness follows immediately from the correctness of $\mathcal{FRE}_\perp$.

Security. To compute the size of the intersection of a set $\mathcal{S}$ encrypted with index 1 with a set $\mathcal{T}$ encrypted with index 2, one checks, for every element $t \in \mathcal{T}$, if $t \in \mathcal{S}$. Therefore, while it clearly allows to compute the size of the intersection, this also leaks more information. Indeed, consider two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ encrypted with index 1 and another set $\mathcal{T}$ encrypted with index 2. Then, for every $t \in \mathcal{T}$, one can check if $t \in \mathcal{S}_1$ and if $t \in \mathcal{S}_2$. Hence, not only the cardinality $\mathcal{T} \cap \mathcal{S}_1$ and $\mathcal{T} \cap \mathcal{S}_2$ is revealed, but also the one of $\mathcal{T} \cap \mathcal{S}_1 \cap \mathcal{S}_2$. More generally, if k sets $\mathcal{S}_1, \ldots, \mathcal{S}_k$ are encrypted with index 1 and a set $\mathcal{T}$ is encrypted with index 2, their encryptions reveal the size of the intersection of $\mathcal{T}$ with any intersection of 1 to k different sets from $\{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$.

We prove that this is exactly the information that is leaked by our construction and define the leakage of our construction, denoted $\mathcal{L}_{\#^*}$, as follows. For two sequences of sets $\vec{\mathcal{S}} = (\mathcal{S}_1, \ldots, \mathcal{S}_{q_1})$ and $\vec{\mathcal{T}} = (\mathcal{T}_1, \ldots, \mathcal{T}_{q_2})$ encrypted respectively with index 1 and 2, we define:

$$\mathcal{L}_{\#^*}(\vec{\mathcal{S}}, \vec{\mathcal{T}}) = (\#(\mathcal{I} \cap \mathcal{T}_i))_{\mathcal{I} \in \vec{\mathcal{S}}^\cap, i \in [q_2]}\,, \tag{2}$$

where $\vec{\mathcal{S}}^\cap = \{\mathcal{S}_{i_1} \cap \cdots \cap \mathcal{S}_{i_j} \mid j \in [q_1], i_j \in [q_1]\}$, so $\vec{\mathcal{S}}^\cap$ contains every intersection of 1 to q_1 different sets encrypted at index 1. In particular, every set $\mathcal{S}_i$ is in $\vec{\mathcal{S}}^\cap$.

Theorem 11. *Assuming there exists an* $\mathcal{L}_\perp$*-indistinguishability secure function-revealing encryption scheme for orthogonality testing, then there exists an* $\mathcal{L}_{\#^*}$*-indistinguishability secure function-revealing encryption scheme for cardinality of intersection.*

The proof of the above theorem is detailed in the full version [23]. Note that, even if $\mathcal{L}_{\#^*}$ is formally an exponential-size vector, checking whether a query made by an adversary is valid or not remains polynomial.

4.4 Orthogonality Testing and Relation with Predicate Encryption

We finally describe how we obtain a function-revealing encryption for orthogonality testing, namely for the function

$$f_\perp\colon\ (\vec{a}, \vec{b}) \in \mathbb{Z}_p^n \mapsto \begin{cases} 1 & \text{if } \langle \vec{a}, \vec{b} \rangle = 0 \\ 0 & \text{otherwise} \end{cases}.$$

This is the last step in building our efficient order-revealing encryption scheme with limited leakage and from standard assumptions.

The existence of such a scheme is actually implied by the existence of a fully-secure secret-key inner-product encryption scheme, which in particular

exists under the DLin assumption [7]; e.g., [25]. More generally, we describe a transformation from any fully-secure secret-key predicate encryption for a class of predicate $\mathcal{F}_f = \{f_a \colon b \in \mathcal{M} \mapsto f(a,b) \in \{0,1\} \mid a \in \mathcal{M}\}$ to a function-revealing encryption scheme for the function f. A very similar result was already proposed in the case of property-preserving encryption in [2,13]. For completeness, definitions of the DLin assumption and of fully-secure secret-key predicate encryption and inner-product encryption are recalled in the full version [23]. In particular, note that by fully-secure, we mean predicate-hiding and attribute-hiding.

Theorem 12. *Let $f \colon \mathcal{M} \times \mathcal{M} \to \{0,1\}$ be any function. Assuming there exists a fully-secure secret-key predicate encryption scheme for the class of predicates $\mathcal{F}_f = \{f_a \colon b \in \mathcal{M} \mapsto f(a,b) \in \{0,1\} \mid a \in \mathcal{M}\}$, then there exists an $\mathcal{L}_f$-indistinguishability secure function-revealing encryption scheme for the function f.*

Please refer to the full version [23] for the proof.

5 Putting Everything Together

We conclude by assembling all our results and obtain an order-revealing encryption scheme with limited leakage assuming the standard DLin assumption. We denote by $\mathcal{L}_\perp$ the (ideal) leakage of the function $f_\perp$ (so $\mathcal{L}_\perp = \mathcal{L}_{f_\perp}$ in the sense of Definition 3).

Corollary 13. *Assuming* DLin*, there exists an $\mathcal{L}_\perp$-indistinguishability secure function-revealing encryption scheme for orthogonality testing.*

Corollary 14. *Assuming* DLin*, there exists an $\mathcal{L}_{\#^*}$-indistinguishability secure function-revealing encryption scheme for the function $f_\#$.*

Corollary 15. *Assuming* DLin*, there exists a $\mathscr{L}_{\mathcal{L}_{\#^*}}$-indistinguishability secure function-revealing encryption scheme for the function $f_<$ (a.k.a. order-revealing encryption scheme).*

A more detailed explanation of our leakage as well as a detailed comparison with the main concurrent work by Cash *et al.* [12] are provided in the full version [23].

5.1 Applications

To conclude this paper, we propose two applications of our constructions. In particular, these applications do not suffer much from our additional leakage.

Membership Testing on a Database and Searchable Encryption. Our notion of function-revealing encryption for the function $f_\#$ naturally yields a solution to test whether some private data is already in a database stored by a given server. Indeed, one could split the database into distinct sets $\mathcal{S}_1, \ldots, \mathcal{S}_q$ of

fixed size n and storing encryptions $\mathsf{Enc}_{\#}(1, \mathsf{sk}, \mathsf{coef}(\mathcal{S}_i))$ for $i \in [q]$. Then, one can simply send to the server $\mathsf{Enc}_{\#}(2, \mathsf{sk}, \mathsf{exp}(a))$ so it can learn whether a is already in the database. One could also use this method with a plaintext x being a tag used to ask the server to return every encrypted data with the same tag.

Similarly, one could associate a vector $\vec{x}$ to a data and perform searchable encryption using our function-revealing encryption scheme for orthogonality (whose leakage is ideal). Doing so, one could query all the data whose tag $\vec{x}$ is orthogonal to some vector $\vec{y}$.

Range Queries. Our notion of function-revealing encryption for the function $f_<$ allows one to perform efficient range queries on a database. One could indeed store encryptions $\mathsf{Enc}_<(1, \mathsf{sk}, x)$ on the server, and makes queries of the form $\mathsf{Enc}_<(2, \mathsf{sk}, a), \mathsf{Enc}_<(2, \mathsf{sk}, b)$ to get encrypted data $x \in [a; b)$. In particular, as our notion is "asymmetric", the server learns only a few extra information, while classical order-revealing encryption let the server knows the complete order of the elements. Due to the form of our leakage, the leaked informationis ideal if only one such query is made by the user. Moreover, as explained above, the asymmetry of our construction benefits to this application. In particular, a fully-secure secret-key IPE scheme with constant-size tokens or ciphertexts would imply a very efficient solution for range queries.

Acknowledgments

The second author was supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

References

1. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order preserving encryption for numeric data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press, Paris, France (Jun 13–18, 2004)
2. Agrawal, S., Agrawal, S., Badrinarayanan, S., Kumarasubramanian, A., Prabhakaran, M., Sahai, A.: Functional encryption and property preserving encryption: New definitions and positive results. Cryptology ePrint Archive, Report 2013/744 (2013), http://eprint.iacr.org/2013/744
3. Agrawal, S., Agrawal, S., Badrinarayanan, S., Kumarasubramanian, A., Prabhakaran, M., Sahai, A.: On the practical security of inner product functional encryption. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 777–798. Springer, Heidelberg (Mar / Apr 2015)

4. Bitansky, N., Nishimaki, R., Passelègue, A., Wichs, D.: From cryptomania to obfustopia through secret-key functional encryption. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 391–418. Springer, Heidelberg (Oct / Nov 2016)
5. Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-preserving symmetric encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 224–241. Springer, Heidelberg (Apr 2009)
6. Boldyreva, A., Chenette, N., O'Neill, A.: Order-preserving encryption revisited: Improved security analysis and alternative solutions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 578–595. Springer, Heidelberg (Aug 2011)
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (Aug 2004)
8. 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.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 563–594. Springer, Heidelberg (Apr 2015)
9. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (Mar 2011)
10. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (Feb 2007)
11. Brakerski, Z., Komargodski, I., Segev, G.: Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 852–880. Springer, Heidelberg (May 2016)
12. Cash, D., Liu, F.H., O'Neill, A., Zhang, C.: Reducing the leakage in practical order-revealing encryption. Cryptology ePrint Archive, Report 2016/661 (2016), http://eprint.iacr.org/2016/661
13. Chatterjee, S., Das, M.P.L.: Property preserving symmetric encryption revisited. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 658–682. Springer, Heidelberg (Nov / Dec 2015)
14. Chenette, N., Lewi, K., Weis, S.A., Wu, D.J.: Practical order-revealing encryption with limited leakage. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 474–493. Springer, Heidelberg (Mar 2016)
15. Cheon, J.H., Fouque, P.A., Lee, C., Minaud, B., Ryu, H.: Cryptanalysis of the new CLT multilinear map over the integers. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 509–536. Springer, Heidelberg (May 2016)
16. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 3–12. Springer, Heidelberg (Apr 2015)
17. Curtmola, R., Garay, J.A., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. In: Juels, A., Wright, R.N., Vimercati, S. (eds.) ACM CCS 06. pp. 79–88. ACM Press (Oct / Nov 2006)
18. Durak, F.B., DuBuisson, T.M., Cash, D.: What else is revealed by order-revealing encryption? In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 16. pp. 1155–1166. ACM Press (Oct 2016)
19. 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: Nguyen, P.Q., Oswald,

E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 578–602. Springer, Heidelberg (May 2014)
20. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 555–564. ACM Press (Jun 2013)
21. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 162–179. Springer, Heidelberg (Aug 2012)
22. Haagh, H., Ji, Y., Li, C., Orlandi, C., Song, Y.: Revealing encryption for partial ordering. In: O'Neill, M. (ed.) 16th IMA International Conference on Cryptography and Coding. LNCS, vol. 10655, pp. 3–22. Springer, Heidelberg (Dec 2017)
23. Joye, M., Passelègue, A.: Function-revealing encryption. Cryptology ePrint Archive, Report 2016/622 (2016), http://eprint.iacr.org/2016/622
24. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (Apr 2008)
25. Kawai, Y., Takashima, K.: Predicate- and attribute-hiding inner product encryption in a public key setting. In: Cao, Z., Zhang, F. (eds.) PAIRING 2013. LNCS, vol. 8365, pp. 113–130. Springer, Heidelberg (Nov 2014)
26. Komargodski, I., Segev, G.: From minicrypt to obfustopia via private-key functional encryption. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 122–151. Springer, Heidelberg (May 2017)
27. Lewi, K., Wu, D.J.: Order-revealing encryption: New constructions, applications, and lower bounds. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 16. pp. 1167–1178. ACM Press (Oct 2016)
28. Lin, H.Y., Tzeng, W.G.: An efficient solution to the millionaires' problem based on homomorphic encryption. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 05. LNCS, vol. 3531, pp. 456–466. Springer, Heidelberg (Jun 2005)
29. Pandey, O., Rouselakis, Y.: Property preserving symmetric encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 375–391. Springer, Heidelberg (Apr 2012)
30. Sahai, A., Waters, B.R.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (May 2005)
31. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (Mar 2009)