

Revealing Encryption for Partial Ordering

Helene Haagh¹, Yue Ji², Chenxing Li², Claudio Orlandi¹, and Yifan Song²

¹ Aarhus University

² IIIS, Tsinghua University*

Abstract. We generalize the cryptographic notion of *Order Revealing Encryption (ORE)* to arbitrary functions and we present a construction that allows to determine the (partial) ordering of two vectors i.e., given $E(\mathbf{x})$ and $E(\mathbf{y})$ it is possible to learn whether $\mathbf{x} = \mathbf{y}$, $\mathbf{x} > \mathbf{y}$, $\mathbf{x} < \mathbf{y}$ or whether \mathbf{x} and \mathbf{y} are incomparable. This is the first non-trivial example of a *Revealing Encryption (RE)* scheme with output larger than one bit, and which does not rely on cryptographic obfuscation or multilinear maps.

Keywords: secret-key cryptography, order-revealing encryption, revealing encryption

1 Introduction

Computing on encrypted data is a promising approach to privacy preserving cloud computing. Using techniques such as (fully) homomorphic encryption [RAD78, Gen09], a client can upload sensitive data on a partially untrusted cloud which can perform computation on the data without learning anything about the data, *including* the result of the computation. However in many applications it is desirable for the server to learn the result of the computation, so that the server can make decisions based on this result without further interaction with the client. Imagine as an example a server running an encrypted spam filter: using homomorphic encryption the server can, given an encrypted message, determine whether the message is spam or not but, since the server does not learn this bit, the server is unable to place the encrypted message in the user’s spam folder.

Revealing Encryption. To solve the above class of problems a different kind of cryptographic primitive is needed, which we refer to as *revealing encryption* or RE. Intuitively, an RE scheme is an encryption scheme that allows to compute (selected) functions of the plaintexts by having access to the encrypted data only. In other words, given a target function f we want to construct an encryption scheme E and a public function F such that if $X_1 = E(K, x_1)$ and $X_2 = E(K, x_2)$ (for a random key K) then we have that $F(X_1, X_2) = f(x_1, x_2)$.

Order Preserving Encryption. The first attempt towards building RE was taken by Agrawal et al. [AKSX04] when they introduced *order preserving encryption (OPE)*, which using our language can be phrased as the very special case of RE where both f and F are numeric comparison. The “preserving” part of OPE is both a strength and a weakness: since $f = F$ it is very easy to use OPE in practical applications (a client outsourcing an encrypted database using OPE does not even need to inform the server that the database is encrypted, as the database can compare encrypted data in the exact same way as it would compare plaintext data). Unfortunately preserving numeric ordering implies that OPE cannot achieve strong security guarantees, as shown by [BCLO09, BCO11]. To overcome this limitation order revealing encryption (ORE) was introduced by Boneh et al. [BLR+15]. The main conceptual contributions of this paper is to generalize the notion of ORE to arbitrary functions (the formal definition of RE is given in Section 3).

* Work done while visiting Aarhus University.

While the first (fully-secure) ORE schemes could only be instantiated using extremely heavy cryptographic tools (see below) and were therefore completely impractical, Chenette et al. [CLWW16] proposed a very elegant and simple construction of ORE which is extremely efficient in practice (at the price of leaking slightly more information than in the ideal case).

Obfuscation & Co. On the other end of the scale, it is trivial to construct secure RE for any function using *ideal circuit obfuscation*. In a nutshell, one can let F be an obfuscated circuit that takes as input two ciphertexts X_1, X_2 , contains a (hardwired) secret key K , and outputs

$$F(X_1, X_2) = f(D(K, X_1), D(K, X_2))$$

i.e., the obfuscated program simply outputs the output of f evaluated on the result of the decryption of its inputs.

Unfortunately general purpose *ideal obfuscation* or even *virtual black-box obfuscation* does not exist [BGI⁺01]. While a weaker notion of obfuscation (called *indistinguishability obfuscation*), might be plausibly instantiated under cryptographic assumptions (as shown by the fascinating research direction started by Garg et al. [GGH⁺13]), it seems unlikely that this will turn into a practical solution in the foreseeable future. Note that using obfuscation it is possible to instantiate *multi-input functional encryption (MIFE)* [GGG⁺14, BLR⁺15, BKS16]: using MIFE, one can implement RE in a similar way as we sketched above, where the obfuscated program is replaced by a MIFE secret key sk_f for the function f .

Note that despite the fact that MIFE implies RE, RE does not imply MIFE³. It is therefore plausible that RE can be instantiated *more efficiently* and *under weaker assumptions* than MIFE, and our results show that this is indeed the case.

Our Contributions. Given the state of affairs, it is natural to ask:

For which functions can we construct practically efficient revealing encryption (RE) schemes?

In this paper we begin answering the question by showing a construction of revealing encryption for partial order of vectors. This is a naturally interesting function motivated by concrete applications such as *privacy-preserving skyline queries* [BKS01, PTFS03] (or enhancing privacy in any other algorithm based on the partial order relation). In particular, given a dataset of d -dimensional vectors, the goal of a skyline query is to determine the set of dominating vectors. As a classic example, in a skyline query a client (e.g., a department committee) wants to evaluate a number of different offers (e.g., job candidates for a faculty position) based on a set of incomparable parameters (e.g., teaching experience, research output, funding, etc.). In this case the department committee is interested in evaluating all candidates for which there does not exist another candidate who is better qualified *under all parameters*, which is exactly the output of a skyline query. Using RE it is possible to compute the output of a skyline query by performing the partial-comparison directly in the encrypted domain i.e., without having to first decrypt the vectors. We note that there are plenty of protocols and algorithms in the computer science literature which use the partial ordering relation (lattice-based access control, timestamps based on vector clocks, topological ordering, etc.), and we therefore believe that the notion of RE for partial order is as natural as the case of RE for the total order relation.

³ A MIFE scheme must not reveal any information (e.g., satisfy IND-CPA security) until a secret key for a function f is released, while in an RE scheme anyone can compute the authorized function on the encrypted data.

Note that the notion of revealing encryption has also been independently introduced by Joye and Passelgue [JP16]. (Interestingly, we came to the notion of RE by generalizing ORE while they reached the same notion by simplifying 2-input FE). In their work they present RE constructions for functions different than those considered in this paper, including: comparison (ORE), orthogonality testing and cardinality of intersection.

Technical Overview. The starting point of our solution is the recent ORE scheme of Chenette et al. [CLWW16]. In this scheme, a value $x \in \{0, 1\}^n$ is encrypted using n evaluations of a pseudorandom function (PRF) F_K for key K , i.e., for each index $i = 1, \dots, n$ the encryption algorithm outputs a value

$$c_i = F_K(i, \text{prefix}(x, i - 1)) + x_i$$

where $\text{prefix}(x, i)$ is the function that outputs the i most significant bits of x , x_i is the i th most significant bit, and where $+$ is integer addition.

Now, take two values x and y and let i^* be the largest index such that

$$\text{prefix}(x, i^* - 1) = \text{prefix}(y, i^* - 1)$$

i.e., i^* is the smallest index such that $x_{i^*} \neq y_{i^*}$. Then the first $i^* - 1$ ciphertexts will be identical for both x, y (since the PRF is evaluated on exactly the same value, and the added bit is the same), while the i^* -th ciphertext will be “in the right order” (since the PRF is evaluated on exactly the same value but in only one of the two cases 1 will be added) and therefore one can compare x and y by finding the first ciphertext component in which the encryptions differ and perform a simple numerical comparison of this value. For security, note that the bottom $n - i^* - 1$ ciphertexts will be independently random since the PRF is evaluated on different values. Therefore, the scheme reveals the order as well as the first position in which the value differs. A very recent work shows that it is possible to limit this leakage [CLOZ16], but unfortunately their construction requires heavy public key operations (we believe that similar techniques could be applied to our scheme as well).

In a nutshell, we generalize the construction presented by Chenette et al. [CLWW16] in the following way: consider for simplicity the 2-dimensional case $\mathbf{x} = (x_1, x_2)$. Then for each pair of indices i, j we compute

$$c_{i,j} = F_K(i, j, \text{prefix}(x_1, i - 1), \text{prefix}(x_2, j - 1)) + \alpha_{i,j}(x_1, x_2)$$

where $\alpha_{i,j}$ is a carefully chosen function that allows to perform the comparison between two vectors in such a way that no information is leaked when the vectors are incomparable. The main challenge in coming up with the right function α , is that we are trying to encode a non-binary output (i.e., $\mathbf{x} = \mathbf{y}$, $\mathbf{x} > \mathbf{y}$, $\mathbf{x} < \mathbf{y}$, or incomparable) into a binary relation (i.e., the numerical comparison between the scalars $\alpha(\mathbf{x})$ and $\alpha(\mathbf{y})$). Details of the constructions are given in Section 4 and in Section 5 we give a performance analysis of our scheme.

Revealing Encryption Beyond Partial Ordering. We think that discovering which functions admit revealing encryption schemes is an exciting and important future research direction. In Appendix C, we discuss simple (unconditionally secure) examples of revealing encryptions for *absolute distance* and for *hamming distance* (which unfortunately is only secure for a limited number of queries).

Other related work. During recent years, OPE and ORE have been active research areas: Bun and Zhandry [BZ16] have studied the connection between ORE and differentially private learning [DMNS06, KLN+11]. Concurrent with this work, Lewi and Wu [LW16] presented a new and

efficient ORE construction based on the work of Chenette et al. [CLWW16]. This construction splits the message in blocks (i.e. a sequence of bits) and the scheme leaks the position of the first block in which the messages differ. Roche et al. [RACY16] proposed a new primitive called *partial order preserving encoding*, which achieves ideal OPE security (IND-OCPA [BCLO09]) while providing fast insertion and search in an encrypted database. Furthermore, *interactive OPE* [PLZ13, KS14, Ker15] was introduced to achieve stronger security guarantees (like ideal security) for OPE schemes. In these schemes, ciphertexts are mutable, meaning that whenever a new value is encrypted the existing ciphertexts can be updated.

During the last couple of decades there has been a long line of work concerning encryption schemes, where either the ciphertexts preserve some information about the underlying messages or it is possible to perform a public test that reveals some information about the encrypted data: *searchable encryption* [SWP00, GSW04, BBO07, BHJP14] allows users to outsource their data in a private manner, while maintaining the possibility to do efficient search over it. Variants of searchable encryption are *public-key encryption with keyword search* [BCOP03, CGKO06], *secure indexes* [Goh03], and *(privacy-preserving) attribute-based searchable encryption* [WLLX13, KHY13, ZXA14, CD15b]. Other related encryption schemes are *prefix preserving encryption* [XFAM02, XY12] and *format preserving encryption* [BRRS09, WRB15], which are concerned with preserving some specific information about the encrypted data. *Property preserving symmetric encryption* [PR12, CD15a] is a generalization of OPE that enables users to learn the properties of a massive data set.

The applications of RE schemes is closely related to the applications of encryption schemes, like *attribute-based encryption* [GPSW06, GVW13], *functional encryption* [BSW11], *predicate encryption* [KSW08], *(anonymous) identity-based encryption* [Sha84, KSW08], and *access control encryption* [DHO16]. All these encryption schemes deal with payload privacy, user privacy, computation on outsourced encrypted data, fine-grained access control on data, etc.

Finally, in Appendix D, we review the (in)security of some existing systems which offer alternative solutions to *privacy-preserving skyline queries*.

2 Preliminaries

For $n, n_1, n_2 \in \mathbb{N}$, let $[n_1 : n_2]$ be the set $\{n_1, n_1 + 1, \dots, n_2 - 1, n_2\}$ and $[n]$ be the set $[1 : n]$. For $x \in \mathbb{Z}$, let $|x|$ denote the absolute value of x . Let $x \leftarrow_{\S} S$ denote that x is sampled uniform random from the set S .

Definition 1 (Pseudorandom Function). *We say that $F : \{0, 1\}^{\kappa} \times \{0, 1\}^* \rightarrow \{0, 1\}^{\kappa}$ is a pseudorandom function (PRF) if for all PPT adversaries \mathcal{A}*

$$\text{adv}_{\mathcal{A}} = 2 \cdot |\Pr[\mathcal{A}^{\mathcal{O}_b(\cdot)}(1^{\kappa}) = b] - 1/2| < \text{negl}(\kappa)$$

with \mathcal{O}_0 a uniform random function and $\mathcal{O}_1 = F_K$ for some key $K \in \{0, 1\}^{\kappa}$.

We interpret $x \in \{0, 1\}^n$ both as a string of bits i.e. $x = (x_1, \dots, x_n)$ and as an integer $x = \sum_{i=0}^{n-1} 2^i x_{n-i}$ i.e., x_1 is the most significant bit of x . Given such an x and an index $i \in [n]$ it is convenient to define the function $\text{prefix} : \{0, 1\}^n \times [0 : n] \rightarrow \{0, 1\}^n \times [0 : n]$

$$\text{prefix}(x, i) := (x_1, \dots, x_i, 0^{n-i}, i)$$

so that $\text{prefix}(x, 0) = (0^n, 0)$, $\text{prefix}(x, 1) = (x_1, 0^{n-1}, 1)$ and so on. Note that prefix has the useful property that for all $x \in \{0, 1\}^n$ $\text{prefix}(x, i) \neq \text{prefix}(x, j)$ if $i \neq j$. Given a d -dimensional vector $\mathbf{m} = (x_1, \dots, x_d) \in (\{0, 1\}^n)^d$ we define prefix to output the vector⁴

$$\text{prefix}(\mathbf{m}, (i_1, \dots, i_d)) := (\text{prefix}(x_1, i_1), \dots, \text{prefix}(x_d, i_d)).$$

Given two strings $x, y \in \{0, 1\}^n$ we define $\text{pos}(x, y)$ to return the largest i such that $\text{prefix}(x, i - 1) = \text{prefix}(y, i - 1)$ or equivalently the smallest i such that $x_i \neq y_i$. If $x = y$, then we define $\text{pos}(x, y)$ to output $n + 1$. Given two d -dimensional vectors $\mathbf{m}^{(1)} = (x_1^{(1)}, \dots, x_d^{(1)})$, $\mathbf{m}^{(2)} = (x_1^{(2)}, \dots, x_d^{(2)})$ we define pos to output the vector

$$\text{pos}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) := (\text{pos}(x_1^{(1)}, x_1^{(2)}), \dots, \text{pos}(x_d^{(1)}, x_d^{(2)})).$$

3 Revealing Encryption

In this section we formally define Revealing Encryption (RE).

Authorized Function. Let \mathcal{M} be the input space and \mathcal{I} the output space, then a RE scheme is parametrized by ℓ -ary authorized function

$$f : \mathcal{M}^\ell \rightarrow \mathcal{I}.$$

Revealing Encryption. Given an authorized function f , a RE scheme for f is a triple of algorithms $\Pi_f = (\text{Setup}, \text{Enc}, \text{Eval})$ defined as follows:

Setup: On input the security parameter κ , the randomized algorithm Setup outputs a secret key sk and the public parameters pp .

Encryption: On input a message $m \in \mathcal{M}$ and a secret key sk , the randomized algorithm Enc outputs a ciphertext c .

Eval: On input ℓ ciphertexts $\{c_i = \text{Enc}(sk, m_i)\}_{i \in [\ell]}$ and the public parameters pp , the Eval algorithm outputs $f(m_1, \dots, m_\ell) \in \mathcal{I}$.

Remark 1. Note that here and in the rest of the paper we do not mention the decryption algorithm, since any RE can be enhanced to allow for decryption by appending an IND-CPA secure encryption to the RE ciphertext.

Definition 2 (Correctness). Let f be an authorized function and κ be the security parameter. Let $\Pi_f = (\text{Setup}, \text{Enc}, \text{Eval})$ be a RE scheme for f . We say that Π_f is correct if for all messages $\{m_i\}_{i \in [\ell]} \in \mathcal{M}^\ell$ the following probability

$$\Pr [\text{Eval}(pp, \{\text{Enc}(sk, m_i)\}_{i \in [\ell]}) \neq f(\{m_i\}_{i \in [\ell]})]$$

is negligible in κ , where $(sk, pp) \leftarrow \text{Setup}(1^\kappa)$ and the probabilities are taken over the random coins of all algorithms.

⁴ To ease the notation we use bold subscripts to indicate the entries in the vector.

Leakage Function. Following the work of Chenette et al. [CLWW16], our definition also allows for a leakage function $\mathcal{L} : \mathcal{M}^* \rightarrow \{0, 1\}^*$ that exactly characterizes the information leaked by our constructions. In the best case $\mathcal{L}(\{m_i\}_{i \in [q]})$ outputs $f(\{m_j\}_{j \in S})$ for every subset $S \subset [q]$ of size ℓ , and in this case we talk about *optimal leakage*. Note that the work of Chenette et al. leaks extra information as well (the first digit at which two integers x, y are different) and our main construction inherits this leakage.

Definition 3 (Security, [CLWW16]). Let κ be the security parameter, let $q \in \mathbb{N}$, and let f be an authorized function. Let $\Pi_f = (\text{Setup}, \text{Enc}, \text{Eval})$ be a RE scheme for f . Consider the following experiments, where $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_q)$ is an adversary, $\mathcal{S} = (\mathcal{S}_0, \dots, \mathcal{S}_q)$ is a simulator, and $\mathcal{L}(\cdot)$ is a leakage function.

Security Experiments	
$\text{REAL}_{\mathcal{A}}^{\Pi_f}(\kappa)$: 1. $(sk, pp) \leftarrow \text{Setup}(1^\kappa)$; 2. $(m_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(1^\kappa, pp)$; 3. $c_1 \leftarrow \text{Enc}(sk, m_1)$; 4. for $2 \leq i \leq q$; a. $(m_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, c_1, \dots, c_{i-1})$; b. $c_i \leftarrow \text{Enc}(sk, m_i)$; 5. output (c_1, \dots, c_q) and $st_{\mathcal{A}}$;	$\text{IDEAL}_{\mathcal{A}, \mathcal{S}, \mathcal{L}}^{\Pi_f}(\kappa)$: 1. $(st_{\mathcal{S}}, pp) \leftarrow \mathcal{S}_0(1^\kappa)$; 2. $(m_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(1^\kappa, pp)$; 3. $(c_1, st_{\mathcal{S}}) \leftarrow \mathcal{S}_1(st_{\mathcal{S}}, \mathcal{L}(m_1))$; 4. for $2 \leq i \leq q$; a. $(m_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, c_1, \dots, c_{i-1})$; b. $(c_i, st_{\mathcal{S}}) \leftarrow \mathcal{S}_i(st_{\mathcal{S}}, \mathcal{L}(m_1, \dots, m_i))$; 5. output (c_1, \dots, c_q) and $st_{\mathcal{A}}$;

We say that Π_f is a q -secure RE scheme wrt $\mathcal{L}(\cdot)$ if for all adversaries \mathcal{A} that makes no more than q queries, there exists a simulator \mathcal{S} such that the output distributions of the two experiments are computationally indistinguishable

$$\text{REAL}_{\mathcal{A}}^{\Pi_f}(\kappa) \sim^c \text{IDEAL}_{\mathcal{A}, \mathcal{S}, \mathcal{L}}^{\Pi_f}(\kappa)$$

We say a scheme is simply secure if it is q -secure for every $q = \text{poly}(\kappa)$.

Definition 3 captures the requirement that given an a priori bounded number of ciphertexts, the adversary should not be able to learn more than the allowed leakage. The security experiments formalize this requirement by creating the challenge ciphertexts either as real encryptions of the adversarial chosen plaintexts or simulated based on the allowed leakage of the adversarial chosen plaintexts. Note that the output of the experiment contains an arbitrary output from the adversary (i.e., $st_{\mathcal{A}}$), which is a very conservative way of allowing the adversary to output any information that might be useful to distinguish between the ideal experiment and the real experiment.

4 Partial Order Revealing Encryption (PORE)

In this section, we present a construction of revealing encryption for partial ordering of vectors. For the sake of presentation, we will start by showing our construction in the 2-dimensional case (which already requires a significant amount of notation and indices). In Appendix B, we generalize to the multidimensional case. Let $\mathcal{M} = \{0, 1\}^n \times \{0, 1\}^n$ be the message domain, and the authorized function for a 2-dimensional PORE is

$$f : \mathcal{M} \times \mathcal{M} \rightarrow \{(0, 0), (0, 1), (1, 0), (1, 1)\}$$

For $\mathbf{m}^{(1)} = (x^{(1)}, y^{(1)}) \in \mathcal{M}$ and $\mathbf{m}^{(2)} = (x^{(2)}, y^{(2)}) \in \mathcal{M}$ we define a function that determines the order

$$\text{ord}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) := \begin{cases} 1 & \text{if } x^{(1)} \leq x^{(2)} \wedge y^{(1)} \leq y^{(2)} \\ 0 & \text{otherwise} \end{cases}$$

Then we can define the authorized function as

$$f(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) := (\text{ord}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}), \text{ord}(\mathbf{m}^{(2)}, \mathbf{m}^{(1)}))$$

which means that

$$f(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) := \begin{cases} (1, 1) & \text{if } \mathbf{m}^{(1)} = \mathbf{m}^{(2)} \\ (1, 0) & \text{if } \mathbf{m}^{(1)} < \mathbf{m}^{(2)} \\ (0, 1) & \text{if } \mathbf{m}^{(1)} > \mathbf{m}^{(2)} \\ (0, 0) & \text{if they are incomparable} \end{cases}$$

We will prove the security of our scheme with respect to the following leakage function (with f as defined above and pos as defined in Section 2):

$$\mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(q)}) = \left\{ f(\mathbf{m}^{(i)}, \mathbf{m}^{(j)}), \text{pos}(\mathbf{m}^{(i)}, \mathbf{m}^{(j)}) \mid i, j \in [q] \right\}$$

i.e. the construction reveals the order as well as the first position in which each coordinate differ.

Given a pseudorandom function $F : \{0, 1\}^\kappa \times \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$, we define the following four functions:

$$\begin{aligned} F_{1,K}, F_{2,K} &: \mathcal{M} \times [n+1]^2 \rightarrow \{0, 1, 2\} \\ F_{3,K}, F_{4,K} &: \{0, 1\}^n \times [n] \rightarrow \{0, 1\} \end{aligned}$$

where given a plaintext $\mathbf{m} = (x, y) \in \mathcal{M}$ and two indices $i, j \in [n+1]$ we define

$$\begin{aligned} F_{1,K}(\mathbf{m}, (i, j)) &= F_K(1, \text{prefix}(x, i-1), \text{prefix}(y, j-1)) \pmod{3} \\ F_{2,K}(\mathbf{m}, (i, j)) &= F_K(2, \text{prefix}(x, i-1), \text{prefix}(y, j-1)) \pmod{3} \\ F_{3,K}(x, i) &= F_K(3, \text{prefix}(x, i-1)) \pmod{2} \\ F_{4,K}(y, j) &= F_K(4, \text{prefix}(y, j-1)) \pmod{2} \end{aligned}$$

On a high level, the construction works as follows: given a point $\mathbf{m} = (x, y)$ and a key K for the pseudorandom function F , then the ciphertext consist of a two-dimensional matrix $\mathbf{cm} \in \{0, 1, 2\}^{(n+1) \times (n+1)}$ and two bit strings $bx, by \in \{0, 1\}^n$ (i.e. one bit string for each entry in the point). These bit strings are constructed to fulfil the property: given encryptions of two points, we can for each entry (x and y) determine at which position they differ, but without revealing the order. These positions can then be used to look up an entry in the matrix \mathbf{cm} , which will determine the partial order of the two points.

Construction 1 Fix a security parameter $\kappa \in \mathbb{N}$. We define a PORE scheme for two dimensions $\Pi_{\text{PORE}} = (\text{Setup}, \text{Enc}, \text{Eval})$ as follows

Setup: On input $\kappa \in \mathbb{N}$, sample and output a key $K \leftarrow_{\mathfrak{s}} \{0, 1\}^\kappa$.

Encryption: Given a point $\mathbf{m} = (x, y) \in \mathcal{M}$ and a secret key K compute for all $i, j \in [n + 1]$

$$cm_{i,j} = F_{2,K}(\mathbf{m}, (i, j)) + \alpha(\mathbf{m}, (i, j)) \pmod 3$$

where α is a function defined as follows

$$\alpha(\mathbf{m}, (i, j)) := \begin{cases} 0 & \text{if } (x_i, y_j) = (0, 0) \\ 1 & \text{if } (x_i, y_j) = (1, 1) \\ x_i & \text{if } i \leq n, j = n + 1 \\ y_j & \text{if } i = n + 1, j \leq n \\ 0 & \text{if } i = n + 1, j = n + 1 \\ z_{i,j} & \text{otherwise} \end{cases}$$

where $z_{i,j} = F_{1,K}(\mathbf{m}, (i, j))$. Next, we compute for all $i, j \in [n]$

$$\begin{aligned} bx_i &= F_{3,K}(x, i) + x_i \pmod 2 \\ by_j &= F_{4,K}(y, j) + y_j \pmod 2 \end{aligned}$$

Then, output the ciphertext $C = (\mathbf{cm}, bx, by)$, where

$$\begin{aligned} \mathbf{cm} &:= (cm_{i,j})_{i,j \in [n+1]} \in \{0, 1, 2\}^{(n+1) \times (n+1)} \\ bx &:= (bx_1, \dots, bx_n) \in \{0, 1\}^n \\ by &:= (by_1, \dots, by_n) \in \{0, 1\}^n \end{aligned}$$

Evaluation: On input two ciphertexts

$$\begin{aligned} C^{(1)} &= (\mathbf{cm}^{(1)}, bx^{(1)}, by^{(1)}) = \text{Enc}(K, \mathbf{m}^{(1)}) \\ C^{(2)} &= (\mathbf{cm}^{(2)}, bx^{(2)}, by^{(2)}) = \text{Enc}(K, \mathbf{m}^{(2)}) \end{aligned}$$

Compute $\text{pos}(bx^{(1)}, bx^{(2)}) = \ell_x$ and $\text{pos}(by^{(1)}, by^{(2)}) = \ell_y$. If $\ell_x = n + 1$ and $\ell_y = n + 1$, the algorithm outputs $(1, 1)$ (since $\mathbf{m}^{(1)} = \mathbf{m}^{(2)}$). Otherwise, lookup the two entries $cm_{\ell_x, \ell_y}^{(1)}$ and $cm_{\ell_x, \ell_y}^{(2)}$ and compute

$$t = cm_{\ell_x, \ell_y}^{(1)} - cm_{\ell_x, \ell_y}^{(2)} \pmod 3$$

Next, the algorithm branches on the value of t :

- If $t = -1$, output $(1, 0)$ (since $\mathbf{m}^{(1)} < \mathbf{m}^{(2)}$);
- If $t = 1$, output $(0, 1)$ (since $\mathbf{m}^{(1)} > \mathbf{m}^{(2)}$);
- Otherwise output $(0, 0)$, since the two points are *incomparable*.

Correctness. Let $\mathbf{m}^{(1)} = (x^{(1)}, y^{(1)})$ and $\mathbf{m}^{(2)} = (x^{(2)}, y^{(2)})$ be two plaintexts such that

$$\text{pos}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) = (\ell_x, \ell_y)$$

(i.e. $x^{(1)}$ and $x^{(2)}$ differ at position ℓ_x and $y^{(1)}$ and $y^{(2)}$ differ at position ℓ_y). We consider the encryptions of these messages

$$C^{(1)} = (\mathbf{cm}^{(1)}, bx^{(1)}, by^{(1)}) = \text{Enc}(K, \mathbf{m}^{(1)})$$

$$C^{(2)} = (\mathbf{cm}^{(2)}, bx^{(2)}, by^{(2)}) = \text{Enc}(K, \mathbf{m}^{(2)})$$

We first argue that $bx_i^{(1)} = bx_i^{(2)}$ for $i < \ell_x$. This is easy to see:

$$\begin{aligned} bx_i^{(1)} &= F_{3,K}(x^{(1)}, i) + x_i^{(1)} \pmod{2} \\ &= F_K(3, \text{prefix}(x^{(1)}, i-1)) + x_i^{(1)} \pmod{2} \\ &= F_K(3, \text{prefix}(x^{(2)}, i-1)) + x_i^{(2)} \pmod{2} \\ &= bx_i^{(2)} \end{aligned}$$

Since by definition of ℓ_x we know that $\forall i < \ell_x$, $\text{prefix}(x^{(1)}, i-1) = \text{prefix}(x^{(2)}, i-1)$ and $x_i^{(1)} = x_i^{(2)}$. The same can be argued about the y part. We then argue that if $\ell_x < n+1$, then there $\exists i < n+1$ such that $bx_i^{(1)} \neq bx_i^{(2)}$. This is easy to see since by definition of ℓ_x the output of prefix is the same but $x_{\ell_x}^{(1)} \neq x_{\ell_x}^{(2)}$.

So, we turn our attention to the comparison between $cm_{\ell_x, \ell_y}^{(1)}$ and $cm_{\ell_x, \ell_y}^{(2)}$ by computing

$$t = cm_{\ell_x, \ell_y}^{(1)} - cm_{\ell_x, \ell_y}^{(2)} \pmod{3}$$

Note that by definition of ℓ_x, ℓ_y , the output of prefix is the same for both ciphertexts and therefore the output of $F_{2,K}$ is the same so we can rewrite this as

$$t = \alpha(\mathbf{m}^{(1)}, (\ell_x, \ell_y)) - \alpha(\mathbf{m}^{(2)}, (\ell_x, \ell_y)) \pmod{3}$$

We now have the following cases:

1. $\ell_x < n+1 \wedge \ell_y < n+1$: In this case we know that $x_{\ell_x}^{(1)} \neq x_{\ell_x}^{(2)} \wedge y_{\ell_y}^{(1)} \neq y_{\ell_y}^{(2)}$, which means that we are either in the case (comparable)

$$\begin{aligned} (x_{\ell_x}^{(1)}, y_{\ell_y}^{(1)}) &= (0, 0), & (x_{\ell_x}^{(2)}, y_{\ell_y}^{(2)}) &= (1, 1) & \text{or} \\ (x_{\ell_x}^{(1)}, y_{\ell_y}^{(1)}) &= (1, 1), & (x_{\ell_x}^{(2)}, y_{\ell_y}^{(2)}) &= (0, 0) \end{aligned}$$

or (incomparable)

$$\begin{aligned} (x_{\ell_x}^{(1)}, y_{\ell_y}^{(1)}) &= (0, 1), & (x_{\ell_x}^{(2)}, y_{\ell_y}^{(2)}) &= (1, 0) & \text{or} \\ (x_{\ell_x}^{(1)}, y_{\ell_y}^{(1)}) &= (1, 0), & (x_{\ell_x}^{(2)}, y_{\ell_y}^{(2)}) &= (0, 1) \end{aligned}$$

In the comparable case we have that one of the α is 1 and the other is 0. When we are in the case $\mathbf{m}^{(1)} < \mathbf{m}^{(2)}$ (i.e. the first of the two comparable cases), then we have that $\alpha(\mathbf{m}^{(1)}, (\ell_x, \ell_y)) = 0$ and $\alpha(\mathbf{m}^{(2)}, (\ell_x, \ell_y)) = 1$, thus $t = -1$ and the evaluation algorithm will output $(1, 0)$ meaning that $\mathbf{m}^{(1)} < \mathbf{m}^{(2)}$. Similar, we argue that the evaluation algorithm correctly outputs $(0, 1)$ (since $t = 1$) when $\mathbf{m}^{(1)} > \mathbf{m}^{(2)}$.

In the incomparable case we have that $t = 0$ since the value $z_{i,j}$ is the same in both cases (since as argued before prefix 's output is the same and so is $F_{1,K}$'s output).

2. $\ell_x = n+1 \wedge \ell_y < n+1$: following a similar reasoning in this case $x_{\ell_x}^{(1)} = x_{\ell_x}^{(2)} \wedge y_{\ell_y}^{(1)} \neq y_{\ell_y}^{(2)}$ therefore $t = y_{\ell_y}^{(1)} - y_{\ell_y}^{(2)} = -1$ when $\mathbf{m}^{(1)} < \mathbf{m}^{(2)}$, and $t = 1$ when $\mathbf{m}^{(1)} > \mathbf{m}^{(2)}$.

3. $\ell_x < n + 1 \wedge \ell_y = n + 1$: following a similar reasoning in this case $x_{\ell_x}^{(1)} \neq x_{\ell_x}^{(2)} \wedge y_{\ell_y}^{(1)} = y_{\ell_y}^{(2)}$ therefore $t = x_{\ell_y}^{(1)} - x_{\ell_y}^{(2)} = -1$ when $\mathbf{m}^{(1)} < \mathbf{m}^{(2)}$, and $t = 1$ when $\mathbf{m}^{(1)} > \mathbf{m}^{(2)}$.
4. $\ell_x = n + 1 \wedge \ell_y = n + 1$: by the definition of the `pos` function this means that $\mathbf{m}^{(1)} = \mathbf{m}^{(2)}$, and by definition the evaluation algorithm will output $(1, 1)$ meaning that the two messages are equal.

4.1 Security

To prove the security of the construction, we present a simulator that constructs the ciphertexts based on information provided by the leakage function and the previous generated ciphertext.

On a high level the simulator works as follows: each component of the ciphertext $C^{(i)}$ is generated using the leakage $\mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(i)})$. For each $j < i$, the leakage reveals at which position ℓ_x (resp. ℓ_y) the messages $\mathbf{m}^{(i)}$ and $\mathbf{m}^{(j)}$ differ for coordinate x (resp. y) and the partial order of the messages. Given this information, we can construct the entries $cm_{\ell_x, \ell_y}^{(i)}$, $bx_{\ell_x}^{(i)}$ and $by_{\ell_y}^{(i)}$ in $\mathbf{cm}^{(i)}$, $bx^{(i)}$ and $by^{(i)}$ such that they reveal the order. Next, for each entry $cm_{s,t}^{(i)}$, $bx_s^{(i)}$ and $by_t^{(i)}$ with $s < \ell_x$ and $t < \ell_y$, the content of the entry will be equivalent to (some of) the previous generated ciphertexts (which are contained in the state of the simulator), since the encryption algorithm uses a pseudorandom function. Finally, for each entry $cm_{s,t}^{(i)}$, $bx_s^{(i)}$ and $by_t^{(i)}$ with $s > \ell_x$ and $t > \ell_y$, the content of the entry will be generated uniformly at random, since the corresponding input to the pseudorandom function has never been used before.

Simulator. Denote the adversarial chosen message as $\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(q)}$, where $\mathbf{m}^{(i)} = (x^{(i)}, y^{(i)}) \in \mathcal{M}$. Initially, simulator \mathcal{S}_0 is empty and \mathcal{S}_1 sets $C^{(1)} = (\mathbf{cm}^{(1)}, bx^{(1)}, by^{(1)})$, where $\mathbf{cm}^{(1)}$, $bx^{(1)}$, $by^{(1)}$ are all drawn uniformly at random. Furthermore, it sets the state $\text{st}_{\mathcal{S}} = (C^{(1)})$. Next, define the simulator \mathcal{S}_i (for $2 \leq i \leq q$) as in Figure 1.

Theorem 1. *The RE scheme Π_{PORE} from Construction 1 is secure with leakage function \mathcal{L} .*

The complete proof of Theorem 1 is presented in Appendix A.

5 Efficiency of PORE

In this section we analyze the efficiency of our PORE construction.

5.1 Theoretical Efficiency

Let κ be the security parameter, d the number of dimensions and n the bit length of each entry. Then we can compute the storage and computational complexity of our scheme.

Storage Complexity. The bit length of a ciphertext in our PORE scheme is exactly:

$$1.6(n + 1)^d + nd = O(n^d)$$

Computational Overhead. Performing an encryption requires

$$2(n + 1)^d + nd = O(n^d)$$

calls to a PRF (with unbounded domain). Note that running the evaluation algorithm requires no invocation of the PRF (only d binary searches into vectors of n bits each and a single addition modulo 3).

$$(C^{(i)}, \text{st}_S) \leftarrow \mathcal{S}_i(\text{st}_S, \mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(i)}))$$

For all $j < i$ we have that $\text{pos}(\mathbf{m}^{(i)}, \mathbf{m}^{(j)}) = (\ell_x^{(j)}, \ell_y^{(j)})$ is provided by $\mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(i)})$.

For each $s, t \in [n+1]$ define entry $cm_{s,t}^{(i)} \in \mathbf{cm}^{(i)}$ as:

1. If $\exists j < i$ such that $\ell_x^{(j)} > s$ and $\ell_y^{(j)} > t$, then set $cm_{s,t}^{(i)} := cm_{s,t}^{(j)}$.
2. Else if $\exists j < i$ such that $\ell_x^{(j)} = s$ and $\ell_y^{(j)} = t$, then
 - if $\mathbf{m}^{(i)} > \mathbf{m}^{(j)}$, set $cm_{s,t}^{(i)} := cm_{s,t}^{(j)} + 1 \pmod 3$;
 - if $\mathbf{m}^{(i)} < \mathbf{m}^{(j)}$, set $cm_{s,t}^{(i)} := cm_{s,t}^{(j)} - 1 \pmod 3$;
 - if they are incomparable, set $cm_{s,t}^{(i)} := cm_{s,t}^{(j)}$.
3. Else set $cm_{s,t}^{(i)} \leftarrow_{\S} \{0, 1, 2\}$.

For each $s \in [n]$ define entry $bx_s^{(i)} \in bx^{(i)}$ as:

4. If $\exists j < i$ such that $\ell_x^{(j)} > s$, then set $bx_s^{(i)} := bx_s^{(j)}$.
5. Else if $\exists j < i$ such that $\ell_x^{(j)} = s$, then set $bx_s^{(i)} := bx_s^{(j)} + 1 \pmod 2$.
6. Else set $bx_s^{(i)} \leftarrow_{\S} \{0, 1\}$.

For each $t \in [n]$ define entry $by_t^{(i)} \in by^{(i)}$ as:

7. If $\exists j < i$ such that $\ell_y^{(j)} > t$, then set $by_t^{(i)} := by_t^{(j)}$.
8. Else if $\exists j < i$ such that $\ell_y^{(j)} = t$, then set $by_t^{(i)} := by_t^{(j)} + 1 \pmod 2$.
9. Else set $by_t^{(i)} \leftarrow_{\S} \{0, 1\}$.

Output $C^{(i)} = (\mathbf{cm}^{(i)}, bx^{(i)}, by^{(i)})$ and $\text{st}_S = (C^{(1)}, \dots, C^{(i)})$.

Fig. 1. Simulator \mathcal{S}_i (for $2 \leq i \leq q$) for 2-dimensional PORE.

5.2 Implementation Choices

In this section we describe the result of our experimental validation of the efficiency of our PORE scheme.

Plaintext Space. We have implemented our scheme for a range of parameters d and n . We report here the results for all combinations (d, n) with $d \in \{2, \dots, 8\}$ and $n = 2^i$ for $i \in \{1, \dots, 13\}$ s.t. the ciphertext size is less than 20MB.

PRF Choice. We implement the PRF $F : \{0, 1\}^\kappa \times \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ using AES-CBC mode, with key size $\kappa = 128$ bits. This is a particularly convenient choice thanks to the AES native instruction in modern CPUs.

Note that in the theoretical analysis we stated that the complexity of the encryption is $O(n^d)$ when measured as the number of calls to a PRF with unbounded domain. However in practice, when instantiating F with AES in CBC mode the running time (in terms of number of calls to AES) grows linearly with the number of blocks needed for the plaintext, namely $\lceil dn/128 \rceil$. Therefore, a naïve implementation would be significantly slower than promised. We notice, however, that thanks

$n \backslash d$	2	3	4	5
2	2.0 (± 0.42) μs	4.0 (± 0.61) μs	18.2 (± 4.56) μs	45.1 (± 7.53) μs
4	7.0 (± 0.76) μs	23.9 (± 1.98) μs	100.2 (± 4.81) μs	411.4 (± 36.60) μs
8	16.2 (± 0.98) μs	107.5 (± 4.31) μs	749.3 (± 95.20) μs	5.6 (± 0.60) ms
16	49.2 (± 1.81) μs	622.3 (± 63.24) μs	7.6 (± 1.12) ms	110.6 (± 6.49) ms
32	154.8 (± 5.05) μs	3.5 (± 0.37) ms	93.0 (± 6.40) ms	3.2 (± 0.01) s
64	546.8 (± 47.95) μs	21.9 (± 2.21) ms	1.4 (± 0.01) s	
128	1.8 (± 0.22) ms	162.5 (± 8.32) ms		
256	6.5 (± 0.83) ms	1.3 (± 0.02) s		
512	21.8 (± 2.53) ms			
1024	83.3 (± 5.95) ms			
2048	326.5 (± 7.58) ms			
4096	1.3 (± 0.02) s			
8192	5.3 (± 0.03) s			

$n \backslash d$	6	7	8
2	124.1 (± 7.18) μs	342.8 (± 25.00) μs	744.3 (± 21.90) μs
4	1.6 (± 0.22) ms	7.4 (± 1.03) ms	33.7 (± 4.03) ms
8	39.3 (± 0.59) ms	358.0 (± 12.67) ms	
16	1.9 (± 0.01) s		

Table 1. Encryption time and standard deviation

to the special structure of the inputs of our PRF it is possible to get rid of this extra factor. In particular, we note that in our matrix of ciphertexts we evaluate the PRF on inputs of the form

$$F_K(\text{prefix}(x_1, i_1), \dots, \text{prefix}(x_d, i_d))$$

where each value $\text{prefix}(x_k, i_k)$ is given as input to n different PRFs. Therefore we modify the way we evaluate the PRF by first precomputing

$$u_{k,i} = F_K^k(\text{prefix}(x_k, i)) \quad \forall k \in [d], i \in [n]$$

and then implement

$$F_K(\text{prefix}(x_1, i_1), \dots, \text{prefix}(x_d, i_d)) = F_K^0(u_{1,i_1} \oplus \dots \oplus u_{d,i_d})$$

so that the inputs to F_K^0 is of fixed length 128. Therefore (even adding the $O(n^2d)$ extra AES invocations on “long” n -bit values used to precompute the u ’s), the total number of calls to AES and hence the running time is $O(n^d)$ as initially promised.

Note, the XOR operation over d strings takes $O(d)$ time. However, the points which are in the same position in the first k dimensions shares the value $u_{1,i_1} \oplus \dots \oplus u_{k,i_k}$. By making these values reusable, we can reduce the amortized complexity to $\sum_{i=1}^d \frac{1}{n^{i-1}} = O(1)$.

5.3 Experimental Setup

The reported encryption timings (Table 1) are the average taken over 100 executions of the encryption algorithm. For the evaluation timings (Table 2), we randomly pick 500 pairs from the 100 ciphertexts and take the average of the 500 executions of the evaluation algorithm. To measure

$n \backslash d$	2	3	4	5
2	0.27 (± 0.02)	0.56 (± 0.05)	0.59 (± 0.05)	0.62 (± 0.06)
4	0.54 (± 0.05)	0.57 (± 0.05)	0.61 (± 0.05)	0.54 (± 0.06)
8	0.54 (± 0.05)	0.58 (± 0.06)	0.43 (± 0.05)	0.37 (± 0.05)
16	0.55 (± 0.05)	0.42 (± 0.05)	0.35 (± 0.04)	0.91 (± 0.57)
32	0.43 (± 0.04)	0.32 (± 0.02)	0.30 (± 0.22)	0.35 (± 0.28)
64	0.42 (± 0.05)	0.56 (± 0.51)	0.95 (± 0.79)	
128	0.37 (± 0.04)	0.71 (± 0.62)		
256	0.30 (± 0.04)	0.80 (± 0.73)		
512	0.39 (± 0.31)			
1024	0.40 (± 0.35)			
2048	0.52 (± 0.60)			
4096	0.50 (± 0.44)			
8192	0.24 (± 0.02)			

$n \backslash d$	6	7	8	
2	0.61 (± 0.06)	0.64 (± 0.73)	0.48 (± 0.06)	
4	0.49 (± 0.07)	0.41 (± 0.06)	3.78 (± 3.24)	
8	0.91 (± 0.53)	1.40 (± 0.76)		
16	1.27 (± 0.78)			

Table 2. Evaluation time and standard deviation (μs)

the size of the ciphertexts (Table 3), we keep track of the size of the required space each time the encryption algorithm applies the memory.

Hardware. The experiments were executed on a machine with the following characteristics:

- OS: Linux TitanX1 3.19.0-15-generic #15-Ubuntu SMP
- CPU: Intel(R) Xeon(R) CPU E5-2675 v3 1.80GHz
- Memory: 128GB
- GCC: gcc version 4.9.2 (Ubuntu 4.9.2-10ubuntu13) (Compile option -O2)

5.4 Results

Encryption Complexity. Table 1 shows how long it takes to encrypt a single plaintext for different values of d and n . As expected, we observe that the encryption time grows as the dimension d and bit lengths n increases.

Evaluation Complexity. Note that the theoretical complexity of the evaluation algorithm is $O(d)$. However, the actual running time of the evaluation algorithm from Table 2 indicates that the algorithm is so fast that for most choices of parameters it is hard to appreciate the theoretical complexity.

When the combined size of all 100 ciphertext from the experiments does not exceed 6MB (i.e. each ciphertext does not exceed 60kB), then all ciphertexts fits inside the L2 cache of the CPU. By observing the variation of the evaluation timings in Table 2 and the ciphertext size in Table 3, we can conclude that there is a tendency that when the ciphertexts fits inside the L2 cache, then the variation stays below $0.07 \mu s$ (this observation is indicated in the tables by splitting the columns in two).

$\begin{array}{c} d \\ \backslash \\ n \end{array}$	2	3	4	5	6	7	8
2	32 B	84 B	232 B	668 B	1.9 kB	5.7 kB	17.1 kB
4	48 B	212 B	1016 B	4.9 kB	24.4 kB	122.1 kB	610.4 kB
8	80 B	660 B	5.7 kB	51.3 kB	461.3 kB	4.1 MB	
16	144 B	2.3 kB	38.4 kB	652.5 kB	10.8 MB		
32	536 B	17.0 kB	561.5 kB	18.1 MB			
64	1.5 kB	99.0 kB	6.3 MB				
128	5.1 kB	650.1 kB					
256	18.1 kB	4.5 MB					
512	68.3 kB						
1024	264.5 kB						
2048	1.0 MB						
4096	4.0 MB						
8192	16.1 MB						

Table 3. The size of a ciphertext

6 Conclusion

In this work, we introduced a generalization of order-revealing encryption (ORE) called revealing encryption (RE), which is an encryption scheme that allows to compute a (selected) function f of the plaintexts given only the encrypted data. We adopt the simulation-based security notion presented by Chenette et al. [CLWW16], which define security with respect to a leakage function. This enables one to determine the exact information that the ciphertexts leak about the underlying messages (which will always include the function f evaluated on all possible ciphertexts).

Revealing encryption is of special interest in relation to applications like computation or queries on outsourced encrypted data. However, these encryption schemes leak potentially sensitive information about the encrypted data depending on the actual application in which RE is used. This means that before using RE in a concrete application one should make a proper analysis to understand whether the leakage provided is problematic or not. A recent line of work has been concerned with attacking applications built on top of this kind of encryption schemes [NKW15, DDC16, GSB⁺17].

Acknowledgements

This research received funding from the Danish National Research Foundation and The National Science Foundation of China (grant 61361136003) for the Sino-Danish Center for the Theory of Interactive Computation (CTIC), the Danish Independent Research Council under Grant-ID DFF-6108-00169 (FoCC), and the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731583 (SODA).

References

- AKSX04. Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Order-preserving encryption for numeric data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 563–574, 2004.
- BBO07. Mihir Bellare, Alexandra Boldyreva, and Adam O’Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 535–552, 2007.

- BCLO09. Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O’Neill. Order-preserving symmetric encryption. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 224–241, 2009.
- BCO11. Alexandra Boldyreva, Nathan Chenette, and Adam O’Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 578–595, 2011.
- BCOP03. Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. *IACR Cryptology ePrint Archive*, 2003:195, 2003.
- BGI⁺01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001. Proceedings*, pages 1–18, 2001.
- BHJP14. Christoph Bösch, Pieter H. Hartel, Willem Jonker, and Andreas Peter. A survey of provably secure searchable encryption. *ACM Comput. Surv.*, 47(2):18:1–18:51, 2014.
- BKS01. Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The skyline operator. In *Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001, Heidelberg, Germany*, pages 421–430, 2001.
- BKS16. Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 852–880, 2016.
- BKV13. Suvarna Bothe, Panagiotis Karras, and Akrivi Vlachou. eskyline: Processing skyline queries over encrypted data. *Proc. VLDB Endow.*, 6(12):1338–1341, August 2013.
- BLR⁺15. Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 563–594, 2015.
- BRRS09. Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. Format-preserving encryption. In *Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, pages 295–312, 2009.
- BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 253–273, 2011.
- BZ16. Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 176–206, 2016.
- CD15a. Sanjit Chatterjee and M. Prem Laxman Das. Property preserving symmetric encryption revisited. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 658–682, 2015.
- CD15b. Payal Chaudhari and Maniklal Das. Privacy-preserving attribute based searchable encryption. *Cryptology ePrint Archive*, Report 2015/899, 2015. <http://eprint.iacr.org/2015/899>.
- CGKO06. Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 79–88, 2006.
- CLOZ16. David Cash, Feng-Hao Liu, Adam O’Neill, and Cong Zhang. Reducing the leakage in practical order-revealing encryption. *Cryptology ePrint Archive*, Report 2016/661, 2016. <http://eprint.iacr.org/2016/661>.
- CLWW16. Nathan Chenette, Kevin Lewi, Stephen A. Weis, and David J. Wu. Practical order-revealing encryption with limited leakage. In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 474–493, 2016.

- DDC16. F. Betül Durak, Thomas M. DuBuisson, and David Cash. What else is revealed by order-revealing encryption? In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1155–1166, 2016.
- DHO16. Ivan Damgård, Helene Haagh, and Claudio Orlandi. Access control encryption: Enforcing information flow with cryptography. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 547–576, 2016.
- DMNS06. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 265–284, 2006.
- Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.
- GGG⁺14. Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 578–602, 2014.
- GGH⁺13. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49, 2013.
- Goh03. Eu-Jin Goh. Secure indexes. Cryptology ePrint Archive, Report 2003/216, 2003. <http://eprint.iacr.org/2003/216>.
- GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006*, pages 89–98, 2006.
- GSB⁺17. Paul Grubbs, Kevin Sekniqi, Vincent Bindschaedler, Muhammad Naveed, and Thomas Ristenpart. Leakage-abuse attacks against order-revealing encryption. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 655–672, 2017.
- GSW04. Philippe Golle, Jessica Staddon, and Brent R. Waters. Secure conjunctive keyword search over encrypted data. In *Applied Cryptography and Network Security, Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004, Proceedings*, pages 31–45, 2004.
- GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554, 2013.
- JP16. Marc Joye and Alain Passelgue. Function-revealing encryption. Cryptology ePrint Archive, Report 2016/622, 2016. <http://eprint.iacr.org/2016/622>.
- Ker15. Florian Kerschbaum. Frequency-hiding order-preserving encryption. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 656–667, 2015.
- KHY13. Dongyoung Koo, Junbeom Hur, and Hyunsoo Yoon. Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage. *Computers & Electrical Engineering*, 39(1):34–46, 2013.
- KLN⁺11. Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.
- KS14. Florian Kerschbaum and Axel Schröpfer. Optimal average-complexity ideal-security order-preserving encryption. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 275–286, 2014.
- KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, pages 146–162, 2008.
- LLM⁺16. Ximeng Liu, Rongxing Lu, Jianfeng Ma, Le Chen, and Haiyong Bao. Efficient and privacy-preserving skyline computation framework across domains. *Future Generation Comp. Syst.*, 62:161–174, 2016.
- LW16. Kevin Lewi and David J. Wu. Order-revealing encryption: New constructions, applications, and lower bounds. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1167–1178, 2016.

- NKW15. Muhammad Naveed, Seny Kamara, and Charles V. Wright. Inference attacks on property-preserving encrypted databases. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 644–655, 2015.
- PLZ13. Raluca A. Popa, Frank H. Li, and Nikolai Zeldovich. An ideal-security protocol for order-preserving encoding. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 463–477, 2013.
- PR12. Omkant Pandey and Yannis Rouselakis. Property preserving symmetric encryption. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 375–391, 2012.
- PTFS03. Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*, pages 467–478, 2003.
- RACY16. Daniel S. Roche, Daniel Apon, Seung Geol Choi, and Arkady Yerukhimovich. POPE: partial order preserving encoding. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1131–1142, 2016.
- RAD78. Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.
- Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 47–53, 1984.
- SWP00. Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 14-17, 2000*, pages 44–55, 2000.
- WLLX13. Changji Wang, Wentao Li, Yuan Li, and Xi-Lei Xu. A ciphertext-policy attribute-based encryption scheme supporting keyword search function. In *Cyberspace Safety and Security - 5th International Symposium, CSS 2013, Zhangjiajie, China, November 13-15, 2013, Proceedings*, pages 377–386, 2013.
- WRB15. Mor Weiss, Boris Rozenberg, and Muhammad Barham. Practical solutions for format-preserving encryption. *CoRR*, abs/1506.04113, 2015.
- XFAM02. Jun (Jim) Xu, Jinliang Fan, Mostafa H. Ammar, and Sue B. Moon. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *10th IEEE International Conference on Network Protocols (ICNP 2002), 12-15 November 2002, Paris, France, Proceedings*, pages 280–289, 2002.
- XY12. Liangliang Xiao and I-Ling Yen. Security analysis and enhancement for prefix-preserving encryption schemes. *IACR Cryptology ePrint Archive*, 2012:191, 2012.
- ZXA14. Qingji Zheng, Shouhuai Xu, and Giuseppe Ateniese. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, pages 522–530, 2014.

A Proof of Theorem 1

In the proof we replace the pseudorandom function F with a truly random function $g : \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$, and we define the following four functions

$$\begin{aligned} g_1, g_2 &: \mathcal{M} \times [n+1]^2 \rightarrow \{0, 1, 2\} \\ g_3, g_4 &: \{0, 1\}^n \times [n] \rightarrow \{0, 1\} \end{aligned}$$

where given a plaintext $\mathbf{m} = (x, y) \in \mathcal{M}$ and two indices $i, j \in [n+1]$ we define

$$\begin{aligned} g_1(\mathbf{m}, (i, j)) &= g(1, \text{prefix}(x, i-1), \text{prefix}(y, j-1)) \pmod 3 \\ g_2(\mathbf{m}, (i, j)) &= g(2, \text{prefix}(x, i-1), \text{prefix}(y, j-1)) \pmod 3 \\ g_3(x, i) &= g(3, \text{prefix}(x, i-1)) \pmod 2 \\ g_4(y, j) &= g(4, \text{prefix}(y, j-1)) \pmod 2 \end{aligned}$$

These functions fulfil the following property, which follows directly from the definition of the functions g_1, g_2, g_3, g_4 , **prefix** and **pos**.

Lemma 1. *For all points $\mathbf{m}^{(1)} = (x^{(1)}, y^{(1)})$ and $\mathbf{m}^{(2)} = (x^{(2)}, y^{(2)})$ in \mathcal{M} , if $\text{pos}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) = (\ell_x, \ell_y)$, then for all $i \leq \ell_x$ and all $j \leq \ell_y$ it holds that*

$$\begin{aligned} g_1(\mathbf{m}^{(1)}, (i, j)) &= g_1(\mathbf{m}^{(2)}, (i, j)) \\ g_2(\mathbf{m}^{(1)}, (i, j)) &= g_2(\mathbf{m}^{(2)}, (i, j)) \\ g_3(x^{(1)}, i) &= g_3(x^{(2)}, i) \\ g_4(y^{(1)}, j) &= g_4(y^{(2)}, j) \end{aligned}$$

Proof (of Theorem 1). We prove that the simulator defined in Figure 1 generates ciphertexts, which are indistinguishable from the actual ciphertexts. We start by defining a series of hybrid games:

H₀: The real experiment: $\text{REAL}_{\mathcal{A}}^{\text{PORE}}(\kappa)$, where the ciphertexts are generated by the encryption algorithm.

H₁: Same as **H₀**, except we replace the PRF F with a truly random function g (i.e. we replace $F_{1,K}, \dots, F_{4,K}$ with g_1, \dots, g_4).

H₂: The ideal experiment: $\text{IDEAL}_{\mathcal{A}, \mathcal{S}, \mathcal{L}}^{\text{PORE}}(\kappa)$, where the ciphertexts are generated by the simulator \mathcal{S} .

From the definition of pseudorandom function it is given that **H₁** is indistinguishable from **H₀** (the real experiment). Next, we prove by induction that the ciphertexts $(C^{(1)}, \dots, C^{(q)})$ generated by the simulator have the same distribution as the ciphertexts $(\widehat{C}^{(1)}, \dots, \widehat{C}^{(q)})$ generated by **H₁** (i.e. that **H₁** is indistinguishable from **H₂**). From the construction of hybrid **H₁**, we notice that the distribution of $cm_{i,j}$, bx_i and by_i are independent of each other for all $i, j \in [n+1]$. Thus, to prove that the distributions are indistinguishable, we can look at each part separately (i.e. we look at each of the nine cases defined in the simulator, separately).

In the base case of the induction we show that $C^{(1)}$ is indistinguishable from $\widehat{C}^{(1)}$: note that the simulator chooses each part of $C^{(1)}$ uniformly random, while in hybrid **H₁** each part of $\widehat{C}^{(1)}$ is computed using the truly random function g with fresh inputs to each evaluation. Thus, we can

conclude that $C^{(1)}$ is indistinguishable from $\widehat{C}^{(1)}$ (note that we do not need to consider the leakage, since we only have one ciphertext and thus not enough to perform an evaluation).

Assume that $(C^{(1)}, \dots, C^{(i-1)})$ (generated by the simulator) is indistinguishable from $(\widehat{C}^{(1)}, \dots, \widehat{C}^{(i-1)})$ (generated by the H_1) for some $1 < i \leq q$. Then, we prove that $C^{(i)} = (\mathbf{cm}^{(i)}, bx^{(i)}, by^{(i)})$ and $\widehat{C}^{(i)} = (\widehat{\mathbf{cm}}^{(i)}, \widehat{bx}^{(i)}, \widehat{by}^{(i)})$ are indistinguishable distributed. Denote the adversarial chosen message by $\mathbf{m}^{(i)} = (x^{(i)}, y^{(i)})$ for $i = 1, \dots, q$. For all $j < i$, let $(\ell_x^{(j)}, \ell_y^{(j)}) = \text{pos}(\mathbf{m}^{(i)}, \mathbf{m}^{(j)})$ (which is provided by the leakage function).

For each entry $cm_{s,t}^{(i)}$ in $\mathbf{cm}^{(i)}$:

1. If $\exists j < i$ such that $\ell_x^{(j)} > s$ and $\ell_y^{(j)} > t$, then we have that $x_s^{(i)} = x_s^{(j)}$ and $y_t^{(i)} = y_t^{(j)}$. Thus, from the definition of hybrid H_1 and by Lemma 1 we get

$$\begin{aligned} \widehat{cm}_{s,t}^{(i)} &= g_2(\mathbf{m}^{(i)}, (s, t)) + \alpha(\mathbf{m}^{(i)}, (s, t)) \\ &= g_2(\mathbf{m}^{(j)}, (s, t)) + \alpha(\mathbf{m}^{(j)}, (s, t)) \\ &= \widehat{cm}_{s,t}^{(j)} \end{aligned}$$

From the definition of the simulator (in Figure 1) it is given that $cm_{s,t}^{(i)} = cm_{s,t}^{(j)}$, and by assumption we have $C^{(j)} \sim \widehat{C}^{(j)}$, which means that $cm_{s,t}^{(j)} \sim \widehat{cm}_{s,t}^{(j)}$. Thus, we can conclude that $cm_{s,t}^{(i)} = cm_{s,t}^{(j)} \sim \widehat{cm}_{s,t}^{(j)} = \widehat{cm}_{s,t}^{(i)}$.

2. Else if $\exists i < j$ such that $\ell_x^{(j)} = s$ and $\ell_y^{(j)} = t$, then $x_s^{(i)} \neq x_s^{(j)}$, $y_t^{(i)} \neq y_t^{(j)}$ and $g_2(\mathbf{m}^{(i)}, (s, t)) = g_2(\mathbf{m}^{(j)}, (s, t))$. The relation between $\widehat{cm}_{s,t}^{(i)}$ and $\widehat{cm}_{s,t}^{(j)}$ is defined from the relation between $\mathbf{m}^{(i)}$ and $\mathbf{m}^{(j)}$ as follows
 - If $\mathbf{m}^{(i)} > \mathbf{m}^{(j)}$, then $(x_s^{(i)}, y_t^{(i)}) = (0, 0)$ and $(x_s^{(j)}, y_t^{(j)}) = (1, 1)$, which means that the following must hold $\alpha(\mathbf{m}^{(i)}, (s, t)) = 0$ and $\alpha(\mathbf{m}^{(j)}, (s, t)) = 1$. Thus

$$\widehat{cm}_{s,t}^{(i)} = \widehat{cm}_{s,t}^{(j)} + 1 \pmod{3}$$

- If $\mathbf{m}^{(i)} < \mathbf{m}^{(j)}$, then $(x_s^{(i)}, y_t^{(i)}) = (1, 1)$ and $(x_s^{(j)}, y_t^{(j)}) = (0, 0)$, which means that the following must hold $\alpha(\mathbf{m}^{(i)}, (s, t)) = 1$ and $\alpha(\mathbf{m}^{(j)}, (s, t)) = 0$. Thus

$$\widehat{cm}_{s,t}^{(i)} = \widehat{cm}_{s,t}^{(j)} - 1 \pmod{3}$$

- If $\mathbf{m}^{(i)}$ and $\mathbf{m}^{(j)}$ are incomparable, then we have

$$[(x_s^{(i)}, y_t^{(i)}), (x_s^{(j)}, y_t^{(j)})] = [(0, 1), (1, 0)] \text{ or } [(1, 0), (0, 1)].$$

Thus, by Lemma 1 we get that that $\widehat{cm}_{s,t}^{(i)} = \widehat{cm}_{s,t}^{(j)}$, since

$$\alpha(\mathbf{m}^{(i)}, (s, t)) = g_1(\mathbf{m}^{(i)}, (s, t)) = g_1(\mathbf{m}^{(j)}, (s, t)) = \alpha(\mathbf{m}^{(j)}, (s, t))$$

By the definition of the simulator (see Figure 1) and the assumption that $C^{(j)} \sim \widehat{C}^{(j)}$, we can conclude that $cm_{s,t}^{(i)}$ and $\widehat{cm}_{s,t}^{(i)}$ are indistinguishable in all three cases.

3. Else $\forall j < i$, we consider the remaining cases

- If $\ell_x^{(j)} < s$ or $\ell_y^{(j)} < t$, then $\widehat{cm}_{s,t}^{(i)}$ is uniformly random, since at least one of the inputs to g_2 has never been used before.
- If $\ell_x^{(j)} = s$ or $\ell_y^{(j)} = t$, then either $x_s^{(i)} = x_s^{(j)}$, $y_t^{(i)} \neq y_t^{(j)}$ or $x_s^{(i)} \neq x_s^{(j)}$, $y_t^{(i)} = y_t^{(j)}$. Thus, exactly one of $\alpha(\mathbf{m}^{(i)}, (s, t))$ and $\alpha(\mathbf{m}^{(j)}, (s, t))$ is random, and the other one is fixed.⁵

Thus, we can conclude that $\widehat{cm}_{s,t}^{(i)}$ is uniformly random and independent from $\widehat{cm}_{s,t}^{(j)}$. Since the simulator choose $cm_{s,t}^{(i)}$ uniformly random, we can conclude that $\widehat{cm}_{s,t}^{(i)}$ and $cm_{s,t}^{(i)}$ are indistinguishable.

For each entry $bx_s^{(i)}$ in $bx^{(i)}$:

4. If $\exists j < i$ such that $\ell_x^{(j)} > s$, then $x_s^{(i)} = x_s^{(j)}$. Thus, from the definition of hybrid H_1 and by Lemma 1 we get

$$\widehat{bx}_s^{(i)} = g_3(x^{(i)}, s) + x_s^{(i)} = g_3(x^{(j)}, s) + x_s^{(j)} = \widehat{bx}_s^{(j)}$$

Thus, by the definition of the simulator and the assumption that $C^{(j)} \sim \widehat{C}^{(j)}$, we can conclude that $bx_s^{(i)}$ and $\widehat{bx}_s^{(i)}$ are indistinguishable.

5. Else if $\exists j < i$ such that $\ell_x^{(j)} = s$, then $x_s^{(i)} \neq x_s^{(j)}$, and by the definition of hybrid H_1 we have

$$\widehat{bx}_s^{(i)} = g_3(x^{(i)}, s) + x_s^{(i)}, \quad \widehat{bx}_s^{(j)} = g_3(x^{(j)}, s) + x_s^{(j)}$$

Thus, we can conclude that $\widehat{bx}_s^{(i)} \neq \widehat{bx}_s^{(j)}$, which implies that

$$\widehat{bx}_s^{(i)} = \widehat{bx}_s^{(j)} + 1 \pmod{2}$$

By the definition of the simulator and the assumption that $C^{(j)} \sim \widehat{C}^{(j)}$, we can conclude that $bx_s^{(i)}$ and $\widehat{bx}_s^{(i)}$ are indistinguishable.

6. Else $\forall j < i$, we have that $\ell_x^{(j)} < s$. In this case, the input to g_3 has never appeared before, thus $\widehat{bx}_s^{(i)}$ is uniform random. Since the simulator choose $bx_s^{(i)}$ uniformly at random, they are indistinguishable.

For each entry $by_t^{(i)}$ in $by^{(i)}$ the arguments follow closely the arguments for case 4-6. Thus, $C^{(i)}$ and $\widehat{C}^{(i)}$ are indistinguishable, if $(C^{(1)}, \dots, C^{(i-1)})$ and $(\widehat{C}^{(1)}, \dots, \widehat{C}^{(i-1)})$ are indistinguishable distributed. By induction, we can conclude that the simulator generates a distribution, which is indistinguishable from the one generated by H_1 .

B d -dimensional PORE

In this section we will generalize the 2-dimensional construction from Section 4 into d dimensions: the matrix \mathbf{cm} will be extended from 2 dimensions to d dimensions, and instead of two n -bit string,

⁵ This follows directly from the way $\alpha(\mathbf{m}^{(i)}, (s, t))$ and $\alpha(\mathbf{m}^{(j)}, (s, t))$ is chosen in the encryption algorithm, and the fact that the two messages differs in exactly one coordinate (e.g. $(x_s^{(i)}, y_t^{(i)}) = (0, 0)$ and $(x_s^{(j)}, y_t^{(j)}) = (0, 1)$).

we now have d n -bit strings bx_1, \dots, bx_d . Thus, given two ciphertexts $C^{(1)} = (\mathbf{cm}^{(1)}, bx_1^{(1)}, \dots, bx_d^{(1)})$ and $C^{(2)} = (\mathbf{cm}^{(2)}, bx_1^{(2)}, \dots, bx_d^{(2)})$, we can use the bitstrings $bx_k^{(1)}$ and $bx_k^{(2)}$ for $k \in [d]$ to determine the entry in $\mathbf{cm}^{(1)}$ and $\mathbf{cm}^{(2)}$ that reveals the partial order of the vectors.

The message space is defined as $\mathcal{M} = (\{0, 1\}^n)^d$, and the authorized function for a d -dimensional PORE is

$$f : \mathcal{M} \times \mathcal{M} \rightarrow \{(0, 0), (0, 1), (1, 0), (1, 1)\}$$

For $\mathbf{m}^{(1)} = (x_1^{(1)}, \dots, x_d^{(1)}) \in \mathcal{M}$ and $\mathbf{m}^{(2)} = (x_1^{(2)}, \dots, x_d^{(2)}) \in \mathcal{M}$ we define a function that determines the order

$$\text{ord}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) := \begin{cases} 1 & \text{if } x_k^{(1)} \leq x_k^{(2)} \ \forall k \in [d] \\ 0 & \text{otherwise} \end{cases}$$

Then we can define the authorized function similar to the 2-dimensional case:

$$f(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) := (\text{ord}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}), \text{ord}(\mathbf{m}^{(2)}, \mathbf{m}^{(1)}))$$

We will prove the security of our scheme with respect to the following leakage function (similar to the one defined for the 2-dimensional case):

$$\mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(q)}) = \left\{ f(\mathbf{m}^{(i)}, \mathbf{m}^{(j)}), \text{pos}(\mathbf{m}^{(i)}, \mathbf{m}^{(j)}) \mid i, j \in [q] \right\}$$

Given a pseudorandom function $F : \{0, 1\}^\kappa \times \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$, we define the following $d + 2$ functions:

$$\begin{aligned} F_{1,K}, F_{2,K} &: \mathcal{M} \times [n+1]^d \rightarrow \{0, 1, 2\} \\ F_{k+2,K} &: \{0, 1\}^n \times [n] \rightarrow \{0, 1\} \quad \text{for } k \in [d] \end{aligned}$$

where given a plaintext $\mathbf{m} = (x_1, \dots, x_d) \in \mathcal{M}$ and d indices $i_1, \dots, i_d \in [n+1]$ we define

$$\begin{aligned} F_{1,K}(\mathbf{m}, (i_1, \dots, i_d)) &= F_K(1, \text{prefix}(x_1, i_1 - 1), \dots, \text{prefix}(x_d, i_d - 1)) \pmod 3 \\ F_{2,K}(\mathbf{m}, (i_1, \dots, i_d)) &= F_K(2, \text{prefix}(x_1, i_1 - 1), \dots, \text{prefix}(x_d, i_d - 1)) \pmod 3 \end{aligned}$$

and for $k \in [d]$ we define

$$F_{k+2,K}(x_k, i_k) = F_K(k+2, \text{prefix}(x_k, i_k - 1)) \pmod 2$$

Construction 2 Fix a security parameter $\kappa \in \mathbb{N}$. We define a PORE for d -dimensional points $\Pi_{\text{PORE}} = (\text{Setup}, \text{Enc}, \text{Eval})$ as follows

Setup: On input $\kappa \in \mathbb{N}$, sample and output a key $K \leftarrow_{\mathcal{S}} \{0, 1\}^\kappa$.

Encryption: Given a point $\mathbf{m} = (x_1, \dots, x_d) \in \mathcal{M}$ and a secret key K . Compute for all $i_1, \dots, i_d \in [n+1]$:

$$cm_{i_1 \dots i_d} = F_{2,K}(\mathbf{m}, (i_1, \dots, i_d)) + \alpha(\mathbf{m}, (i_1, \dots, i_d)) \pmod 3$$

where α is a function defined as follows

$$\alpha(\mathbf{m}, (i_1, \dots, i_d)) = \begin{cases} 0 & \text{if } x_{k, i_k} = 0 \ \forall k \in S \vee S = \emptyset \\ 1 & \text{if } x_{k, i_k} = 1 \ \forall k \in S \wedge S \neq \emptyset \\ F_{1,K}(\mathbf{m}, (i_1, \dots, i_d)) & \text{otherwise} \end{cases}$$

where $S = \{k \in [d] \mid i_k \leq n\}$,⁶ and $x_{\mathbf{k}, i_k}$ is the i_k -th bit in $x_{\mathbf{k}}$. Next, compute for all $k \in [d]$

$$bx_{\mathbf{k}, i_k} = F_{k+2, K}(x_{\mathbf{k}}, i_k) + x_{\mathbf{k}, i_k} \pmod{2} \quad \forall i_k \in [n]$$

Then output $C = (\mathbf{cm}, bx_1, \dots, bx_d)$, where

$$\begin{aligned} \mathbf{cm} &= \{cm_{i_1 \dots i_d}\}_{i_1, \dots, i_d \in [n+1]} \in (\{0, 1, 2\}^{(n+1)})^d \\ bx_{\mathbf{k}} &= (bx_{\mathbf{k}, 1}, \dots, bx_{\mathbf{k}, n}) \in \{0, 1\}^n \quad \forall k \in [d] \end{aligned}$$

Evaluation: On input two ciphertexts

$$\begin{aligned} C^{(1)} &= (\mathbf{cm}^{(1)}, bx_1^{(1)}, \dots, bx_d^{(1)}) = \text{Enc}(K, \mathbf{m}^{(1)}) \\ C^{(2)} &= (\mathbf{cm}^{(2)}, bx_1^{(2)}, \dots, bx_d^{(2)}) = \text{Enc}(K, \mathbf{m}^{(2)}) \end{aligned}$$

For all $k \in [d]$, compute $\text{pos}(bx_{\mathbf{k}}^{(1)}, bx_{\mathbf{k}}^{(2)}) = \ell_k$. If $\ell_k = n + 1$ for all $k \in [d]$, the algorithm outputs $(1, 1)$ (since $\mathbf{m}^{(1)} = \mathbf{m}^{(2)}$). Otherwise, lookup the entry defined by ℓ_1, \dots, ℓ_d in $\mathbf{cm}^{(1)}$ and $\mathbf{cm}^{(2)}$, and compute

$$t = cm_{\ell_1 \dots \ell_d}^{(1)} - cm_{\ell_1 \dots \ell_d}^{(2)} \pmod{3}$$

Next, the algorithm branches on the value of t :

- If $t = -1$, output $(1, 0)$ (since $\mathbf{m}^{(1)} < \mathbf{m}^{(2)}$);
- If $t = 1$, output $(0, 1)$ (since $\mathbf{m}^{(1)} > \mathbf{m}^{(2)}$);
- Otherwise output $(0, 0)$, since the two points are *incomparable*.

Correctness. Given $\mathbf{m}^{(1)} = (x_1^{(1)}, \dots, x_d^{(1)})$ and $\mathbf{m}^{(2)} = (x_1^{(2)}, \dots, x_d^{(2)})$ such that $\text{pos}(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}) = (\ell_1, \dots, \ell_d)$. Then, by the same arguments as in the 2-dimensional case, we can prove for all $k \in [d]$ that $bx_{\mathbf{k}, i_k}^{(1)} = bx_{\mathbf{k}, i_k}^{(2)}$ for $i_k < \ell_k$, and if $\ell_k < n + 1$ then there exists $i_k < n + 1$ such that $bx_{\mathbf{k}, i_k}^{(1)} \neq bx_{\mathbf{k}, i_k}^{(2)}$. Thus, we can identify the entry (l_1, \dots, l_d) in $\mathbf{cm}^{(1)}$ and $\mathbf{cm}^{(2)}$ that determines the partial order of the vectors. Next, we can do the same case analysis as in the proof for 2 dimensions by a natural extensions to d dimensions.

B.1 Security.

The security proof of the d -dimensional PORE scheme is a direct generalization of the security proof for the 2-dimensional PORE.

Simulator. Denote the adversarial chosen message as $\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(q)}$, where $\mathbf{m}^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)}) \in \mathcal{M}$. Initially, simulator \mathcal{S}_0 is empty, and simulator \mathcal{S}_1 sets $C^{(1)} = (\mathbf{cm}^{(1)}, bx_1^{(1)}, \dots, bx_d^{(1)})$, where $\mathbf{cm}^{(1)}, bx_{\mathbf{k}}^{(1)}$ for $k \in [d]$ are all drawn uniformly at random. Furthermore, it sets $\text{st}_{\mathcal{S}} = (C^{(1)})$. Define the simulator \mathcal{S}_i (for $2 \leq i \leq q$) as in Figure 2.

Theorem 2. *The RE scheme Π_{PORE} from Construction 2 is secure with leakage function \mathcal{L} .*

⁶ Given i_1, \dots, i_d , the set S contains all the k 's such that $i_k \leq n$.

$$(C^{(1)}, \text{st}_S) \leftarrow \mathcal{S}_i(\text{st}_S, \mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(i)}))$$

For all $j < i$ we have that $\text{pos}(\mathbf{m}^{(i)}, \mathbf{m}^{(j)}) = (\ell_1^{(j)}, \dots, \ell_d^{(j)})$ is provided by $\mathcal{L}(\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(i)})$.

For each $i_1, \dots, i_d \in [n+1]$ define the entry $cm_{i_1 \dots i_d}^{(i)} \in \mathbf{cm}^{(i)}$ as:

1. If $\exists j < i$ such that $\ell_k^{(j)} \geq i_k$ for all $k \in [d]$, and $\exists k$ such that $\ell_k^{(j)} > i_k$, then set $cm_{i_1 \dots i_d}^{(i)} := cm_{i_1 \dots i_d}^{(j)}$.
2. Else if $\exists j < i$ such that $(\ell_1^{(j)}, \dots, \ell_d^{(j)}) = (i_1, \dots, i_d)$, then
 - if $\mathbf{m}^{(i)} > \mathbf{m}^{(j)}$, set $cm_{i_1 \dots i_d}^{(i)} := cm_{i_1 \dots i_d}^{(j)} + 1 \pmod 3$
 - if $\mathbf{m}^{(i)} < \mathbf{m}^{(j)}$, set $cm_{i_1 \dots i_d}^{(i)} := cm_{i_1 \dots i_d}^{(j)} - 1 \pmod 3$
 - if they are incomparable, set $cm_{i_1 \dots i_d}^{(i)} := cm_{i_1 \dots i_d}^{(j)}$
3. Else set $cm_{i_1 \dots i_d}^{(i)} \leftarrow_{\S} \{0, 1, 2\}$.

For all $k \in [d]$: for each $i_k \in [n]$ define the entry $bx_{\mathbf{k}, i_k}^{(i)} \in bx_{\mathbf{k}}^{(i)}$, :

4. If $\exists j < i$ such that $\ell_k^{(j)} > i_k$, then set $bx_{\mathbf{k}, i_k}^{(i)} := bx_{\mathbf{k}, i_k}^{(j)}$.
5. Else if $\exists j < i$ such that $\ell_k^{(j)} = i_k$, then set $bx_{\mathbf{k}, i_k}^{(i)} := bx_{\mathbf{k}, i_k}^{(j)} + 1 \pmod 2$.
6. Else set $bx_{\mathbf{k}, i_k}^{(i)} \leftarrow_{\S} \{0, 1\}$.

Output $C^{(i)} = (\mathbf{cm}^{(i)}, bx_{\mathbf{1}}^{(i)}, \dots, bx_{\mathbf{d}}^{(i)})$ and $\text{st}_S = (C^{(1)}, \dots, C^{(i)})$

Fig. 2. Simulator \mathcal{S}_i (for $2 \leq i \leq q$) for the d -dimensional PORE.

Proof. We state a series of hybrid games, which are similar to the 2-dimensional case:

H₀: The real experiment: $\text{REAL}_{\mathcal{A}}^{\text{PORE}}(\kappa)$, where the ciphertexts are generated by the encryption algorithm.

H₁: Same as **H₀**, except we replace the PRF F with a truly random function.

H₂: The ideal experiment: $\text{IDEAL}_{\mathcal{A}, \mathcal{S}, \mathcal{L}}^{\text{PORE}}(\kappa)$, where the ciphertexts are generated by the simulator.

The first step of the proof is to replace the pseudorandom function F with a truly random function. Thus, from the property of the pseudorandom function, we get that hybrid **H₀** (the real experiment) and hybrid **H₁** are indistinguishable. Next, we prove by induction that hybrid **H₁** generates ciphertexts, which are indistinguishable from simulated ciphertexts (hybrid **H₂**). This is proven in the same manner as for the 2-dimensional PORE. Separately, we study each entry in the d -dimensional matrix \mathbf{cm} and each entry in the n -dimensional vectors $bx_{\mathbf{1}}, \dots, bx_{\mathbf{d}}$, and prove that the entry created using the random function is indistinguishable from the simulated version. From the definition of the simulator and hybrid **H₁** we get that each entry is independent from the others. Thus, we can conclude that the construction is secure with leakage function \mathcal{L} .

C Revealing Encryption For Other Functions

In this section we present some ideas for constructing simple revealing encryption schemes for other natural functions.

C.1 Hamming distance

Given a plaintext space $\{0, 1\}^n$, we define a RE scheme for the function $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{Z}_n$

$$f(x, y) = d_H(x, y)$$

where $d_H(x, y) = |\{x_j \neq y_j | j \in \mathbb{Z}_n\}|$ is the Hamming distance between the bit vectors x and y .

Our construction is as follows: the setup algorithm outputs $pp = n$ and $sk = (\pi, r)$, where $\pi : [n] \rightarrow [n]$ is a random permutation and $r \leftarrow_{\$} \{0, 1\}^n$ is a random n -bit string. The encryption algorithm on input $m \in \{0, 1\}^n$ outputs

$$c = \text{Enc}(sk, m) = (m_{\pi(1)}, \dots, m_{\pi(n)}) \oplus r$$

(i.e. we permute the bits of the message m and XOR the result with a random value r). Given two ciphertexts c_1, c_2 the evaluation algorithm outputs

$$\text{Eval}(pp, c_1, c_2) = d_H(c_1, c_2)$$

Note that when computing the Hamming distance between the two ciphertexts, the random value r will cancel out. This leaves the permuted plaintexts, which has the same Hamming distance as the original plaintexts. Thus, the scheme enjoys correctness. Next, the scheme can be proven secure according to the following leakage function for $q \leq 3$

$$\mathcal{L}(m_1, \dots, m_q) = \{d_H(m_i, m_j) | i, j \in [q]\}$$

To prove that the scheme is secure we take a look at the general case for an arbitrary q , and investigate what the ciphertexts leak about the structure and relation between the queried messages m_1, \dots, m_q . For all $s \in \{0, 1\}^q$ and $i \in [n]$, define A_s as follows:

$$i \in A_s \text{ iff } s = (m_{1,i}, \dots, m_{q,i})$$

where $m_{j,i}$ denotes the i th bit of message m_j for $j \in [q]$. Denote the leaked structure by

$$\mathcal{T}(m_1, \dots, m_q) = \{(s, |A_s| + |A_{\bar{s}}|) \mid s_1 = 0\}$$

where \bar{s} is defined such that $\bar{s}_j \neq s_j$ for all $j \in [q]$. Thus, we define a new leakage function

$$\mathcal{L}^*(m_1, \dots, m_q) = \mathcal{L}(m_1, \dots, m_q) \cup \mathcal{T}(m_1, \dots, m_q)$$

The simulator then proceeds by picking random ciphertexts c_1, \dots, c_q under the condition that $\mathcal{T}(m_1, \dots, m_q) = \mathcal{T}(c_1, \dots, c_q)$. Then it can be proven that c_1, \dots, c_q is indistinguishable from real encryptions of messages m_1, \dots, m_q under leakage function \mathcal{L}^* . Finally, we can prove that for $q \leq 3$ the information leaked by \mathcal{L}^* can be computed given the information leaked by \mathcal{L} .

Insecurity when $q > 3$. We will now give a concrete example of why leakage function \mathcal{L} is not enough for $q > 3$. For two different set of queried messages $\{m_1, \dots, m_q\}$ and $\{m'_1, \dots, m'_q\}$ with the same leakage under \mathcal{L} , they can have different structure of \mathcal{T} . For example (for $q = 4$):

$$\begin{array}{ll}
m_1 = 0000 & m'_1 = 0000 \\
m_2 = 0011 & m'_2 = 0011 \\
m_3 = 0101 & m'_3 = 0101 \\
m_4 = 1001 & m'_4 = 0110
\end{array}$$

Here we observe that $d_H(m_i, m_j) = d_H(m'_i, m'_j)$ for all $1 \leq i < j \leq 4$. However, for $s = (0, 0, 0, 0)$ we note that $(s, 0) \in \mathcal{T}(m_1, m_2, m_3, m_4)$, while $(s, 1) \in \mathcal{T}(m'_1, m'_2, m'_3, m'_4)$. Thus, the two sets of queries have different structure, which for $q > 3$ cannot be computed given only the information provided by leakage function \mathcal{L} .

C.2 Difference Revealing Encryption

Modular Difference. Given a plaintext space \mathbb{Z}_n (for any integer n), it is easy to see that *one-time pad* encryption, with key re-use, is a perfectly secure RE scheme for the function $f : \mathbb{Z}_n \times \mathbb{Z}_n \rightarrow \mathbb{Z}_n$

$$f(x, y) = x - y \bmod n$$

In particular, let $k \leftarrow \mathbb{Z}_n$ be a random key and $pp = n$, then given a plaintext $m_i \in \mathbb{Z}_n$

$$c_i = \text{Enc}(k, m_i) = m_i + k \bmod n$$

Given two ciphertexts c_i, c_j it is now possible to compute

$$\text{Eval}(pp, c_i, c_j) = c_i - c_j \bmod n = m_i - m_j = f(m_i, m_j)$$

The scheme can be easily proven secure according to the *optimal* leakage function

$$\mathcal{L}(m_1, \dots, m_q) = \{f(m_i, m_j) \mid i, j \in [q]\}$$

since the simulator only needs to pick a random ciphertext $c_1 \leftarrow_{\S} \mathbb{Z}_n$ to start with, and then compute each following ciphertext c_2, \dots, c_q as

$$c_j = c_1 - f(m_1, m_j)$$

Absolute Difference. More interestingly, the above simple construction can be turned into a revealing encryption for absolute difference between integers of bounded magnitude B i.e., for the function $f(x, y) : [B] \times [B] \rightarrow [0 : B - 1]$ defined as

$$f(x, y) = |x - y|$$

(Note that the challenge here is to construct a scheme where the output of the *Eval* function should be the same no matter what the order of its input is). Our construction is as follows: the setup algorithm outputs a secret key $sk = (s, k)$, where $k \leftarrow_{\S} [2B - 1]$ and $s \leftarrow_{\S} \{-1, +1\}$, and $pp = B$. The encryption algorithm on input a plaintext $m_i \in [B]$ outputs

$$c_i = \text{Enc}(sk, m_i) = s \cdot m_i + k \bmod 2B - 1$$

and given two ciphertexts c_i, c_j the evaluation function outputs

$$\text{Eval}(pp, c_i, c_j) = \min\{|c_i - c_j|, 2B - 1 - |c_i - c_j|\}$$

For correctness, we observe that

$$c_i - c_j \bmod 2B - 1 = s(m_i - m_j) \bmod 2B - 1$$

Given that $m_i, m_j \in [B]$ we have that $-B < s(m_i - m_j) < B$. Thus we can conclude that the evaluation algorithm outputs the absolute difference of the two messages:

$$\text{Eval}(pp, c_i, c_j) = |s(m_i - m_j)| = f(m_i, m_j)$$

Also in this case the scheme can be proven secure according to the *optimal* leakage function

$$\mathcal{L}(m_1, \dots, m_q) = \{f(m_i, m_j) | i, j \in [q]\}$$

using the following simulation strategy: start by picking a random ciphertext $c_1 \leftarrow_{\S} [2B - 1]$, and for any $i \in [q]$ such that $f(m_i, m_1) = 0$, set $c_i = c_1$. Let $k \in [q]$ be the smallest index such that $f(m_k, m_1) \neq 0$, then let $c_k = c_1 + s \cdot f(m_k, m_1)$ for $s \leftarrow_{\S} \{-1, +1\}$. Then for $2 \leq i \leq q$ do the following

1. if $f(m_k, m_i) = |f(m_k, m_1) - f(m_i, m_1)|$, then compute $c_i = c_1 + s \cdot f(m_i, m_1)$,
2. otherwise compute $c_i = c_1 - s \cdot f(m_i, m_1)$.

Note, the reason why we distinguish between these two cases is to determine whether m_i is on the same side (or opposite side) of m_1 compared to m_k . In case 1) m_1 is the maximum or minimum among m_1, m_k, m_i , thus, m_k and m_i are on the same side. In case 2) they are on opposite sides.

D Review of Existing Privacy-Preserving Skyline Queries Systems

In this section we review the security of two existing systems for performing privacy-preserving skyline queries.

D.1 eSkyline

Bothe et al. [BKV13] present a system called eSkyline with the goal of processing skyline queries over encrypted data. They propose a deterministic secret-key encryption scheme to encrypt each data vector. However, the scheme is clearly not IND-CPA secure (as the authors also observe themselves), since a chosen-plaintext attack will allow an adversary to determine the encryption key. Furthermore, an encryption of the zero-vector will always result in the zero-vector. Thus, the encryption scheme reveals too much unwanted information, even to an adversary that only is allowed to observe the encrypted data.

D.2 EPSC

Liu et al. [LLM⁺16] propose a new system called EPSC (efficient and privacy-preserving skyline computation). To implement this system they design a new additive homomorphic public key encryption scheme as follows: let τ, q and η be large primes, and compute $C_0 = \tau^{-1} \bmod q$, $p = C_0 + k_0 \cdot q$ such that p is a prime, and $\Phi = p \cdot \eta$. Let $pk = (\Phi, q)$ be the public key, and $sk = (p, \tau, \eta)$ be the private key. Then they propose to encrypt a message x as follows: choose a random number r (of size significantly smaller than q) and compute $C = \Phi \cdot r + x \bmod q$.

In the paper, the following parameters are suggested: $|q| = 1024$, $|\Phi| = 2048$ and $|r| = 512$. This encryption scheme is unfortunately not secure: given a ciphertext C , we can determine whether C encrypts x' by computing

$$a = (C - x') \cdot (\Phi^{-1} \bmod q) = r + (x - x') \cdot \Phi^{-1} \bmod q.$$

If $x = x'$ then $a = r$, which means that a will be small (i.e. $a \leq 2^{512}$ with probability 1), while in all other cases a will be large (i.e., $a > 2^{512}$ with overwhelming probability). Thus, the system does not satisfy IND-CPA security.