Practical Order-Revealing Encryption with Limited Leakage

Nathan Chenette[1], Kevin Lewi[2], Stephen A. Weis[3], and David J. Wu[2]

[1]Rose-Hulman Institute of Technology
[2]Stanford University
[3]Facebook, Inc.

Abstract

In an order-preserving encryption scheme, the encryption algorithm produces ciphertexts that preserve the order of their plaintexts. Order-preserving encryption schemes have been studied intensely in the last decade, and yet not much is known about the security of these schemes. Very recently, Boneh et al. (Eurocrypt 2015) introduced a generalization of order-preserving encryption, called order-revealing encryption, and presented a construction which achieves this notion with best-possible security. Because their construction relies on multilinear maps, it is too impractical for most applications and therefore remains a theoretical result.

In this work, we build efficiently implementable order-revealing encryption from pseudorandom functions. We present the first efficient order-revealing encryption scheme which achieves a simulation-based security notion with respect to a leakage function that precisely quantifies what is leaked by the scheme. Moreover, we show how composing our construction with existing order-preserving encryption schemes results in order-revealing encryption that is strictly more secure than all preceding order-preserving encryption schemes.

1 Introduction

A symmetric encryption scheme is order-preserving if the ciphertexts preserve the numeric ordering of their underlying plaintexts. The notion of order-preserving encryption (OPE) was introduced by Agrawal et al. [AKSX04] who showed how it could be used to efficiently answer range queries over encrypted data, as well as sorting queries, searching queries, and more. Indeed, existing OPE solutions have been implemented in practice [Sky, PRZB11] for these exact purposes. Since the introduction of OPE, there has been a plethora of work on analyzing the security of various OPE schemes, found both in the cryptography community and the database community. However, it is troubling that in spite of the numerous practical applications of OPE, the security of the best candidate OPE schemes is still not well understood.

Prior work. The first OPE construction by Agrawal et al. [AKSX04] relied on heuristics and lacked a formal security analysis. Subsequently, Boldyreva et al. [BCLO09] gave the first formal security definitions for OPE schemes. Boldyreva et al. introduced two primary notions for security of an OPE scheme. The first notion of security for an OPE scheme is called indistinguishability under an ordered chosen plaintext attack (IND-OCPA). The IND-OCPA definition can be viewed as a generalization of semantic security [GM84], and effectively says that encryptions of a sequence of

messages should reveal nothing about the underlying messages other than their ordering. However, in the same work, Boldyreva et al. showed that *any* order-preserving encryption scheme that is IND-OCPA secure must have a ciphertext space which is exponentially large in the size of the plaintext space.

In light of this lower bound for OPE schemes that satisfy IND-OCPA security, Boldyreva et al. introduced a weaker notion of security (POPF-CCA security) where the encryption function for the OPE scheme is compared to a random order-preserving function—that is, the encryption algorithm for an OPE scheme behaves like a truly random order-preserving function. Under this definition, an OPE scheme inherits the properties of a random order-preserving function.[1] In the same work, Boldyreva et al. gave an explicit construction of an OPE scheme that satisfies POPF-CCA security. However, the POPF-CCA security definition does not precisely specify the information that is leaked by an OPE scheme that achieves this definition. In fact, a scheme that achieves this notion of security does not even satisfy semantic security for a single encryption, and indeed, in subsequent work, Boldyreva et al. [BCO11] showed that ciphertexts in their OPE scheme leak approximately the first half of the bits of the underlying plaintexts. In addition, they introduce several new security definitions in order to better quantify the information leakage of OPE schemes that are POPF-CCA secure.

Recently, Boneh et al. [BLR+15] proposed a generalization of OPE called order-revealing encryption (ORE). In an OPE scheme, the ciphertexts are numeric-valued, and the ordering of the underlying plaintexts is determined by numerically comparing the ciphertexts. In contrast, in an ORE scheme, the ciphertexts are not constrained to any particular form, and instead, there is a publicly computable comparison function which takes two ciphertexts and outputs the numeric ordering of the underlying plaintexts.[2] Although this generalization may at first seem subtle, Boneh et al. constructed an ORE scheme from multilinear maps that achieves the "best-possible" notion of security, which is equivalent to the IND-OCPA security notion for order-preserving encryption.

The main drawback of the Boneh et al. ORE construction is that it relies on complicated tools and strong assumptions on these tools, and as such, is currently impractical to implement.

1.1 Our Contributions

We now summarize the main contributions of this work, which include a new simulation-based security notion for ORE, along with a practical construction of an ORE scheme which achieves this security notion. We also show how our new construction can be used to achieve a strictly stronger notion of security compared to other stateless and efficiently implementable (e.g., constructions that rely on powerful primitives such as multilinear maps and indistinguishability obfuscation) OPE and ORE encryption schemes.

Security model. In our work, we take the general approach of Boneh et al. in constructing an ORE scheme, except we take a more efficient route. Our first contribution is a new security definition for order-revealing encryption schemes that both allows for and explicitly models the leakage in the scheme. Our design goals for introducing this new security model are twofold: first, the security model should enable constructions that are efficiently implementable, and second, it

[1] This definition is inspired by the similar definition for PRF security [GGM84], which compares the output of a keyed function to that of a truly random function.

[2] This application was also observed and independently achieved by Goldwasser et al. [GGG+14] using indistinguishability obfuscation.

should provide a precise quantification of any information leaked by the scheme. The two primary notions of security, IND-OCPA and POPF-CCA, introduced by Boldyreva et al. [BCLO09] each satisfy one of these two properties. In particular, all non-interactive, stateless[3] ORE schemes that achieve IND-OCPA security require strong cryptographic primitives such as multilinear maps or indistinguishability obfuscation [GGG+14, BLR+15], and thus, are not efficiently implementable today. At the other end of the spectrum, it is difficult to precisely quantify the leakage of schemes that satisfy POPF-CCA security. The work by Boldyreva et al. [BCO11] provides some concrete lower and upper bounds for the leakage under the strong assumption that the plaintexts are drawn from a uniform distribution. For more general distributions, the leakage remains unclear.

In our work, we give a simulation-based definition of security for ORE with respect to a leakage function $\mathcal{L}$. In other words, our definition states that whatever an adversary is able to deduce from seeing encryptions of messages $m_1, \ldots, m_n$, it could also deduce given only the leakage $\mathcal{L}(m_1, \ldots, m_n)$. The "best-possible" security for ORE would correspond to the case where the leakage function simply outputs whether $m_i < m_j$ for all pairs of messages m_i, m_j. By allowing for the possibility of additional leakage, it becomes possible to construct practical ORE schemes from standard assumptions. Thus, our constructions provide a concrete trade-off between security and efficiency. Our security definitions are similar to the simulation-based definitions that have been considered previously in the searchable symmetric encryption literature [CM05, CGKO06].

Constructions. In our main construction, we show how to construct an ORE scheme from one-way functions (more precisely, from pseudorandom functions (PRFs) [GGM84]). This particular ORE scheme reveals slightly more information than just the ordering of the underlying messages. Specifically, two ciphertexts encrypting messages m_1 and m_2 also reveal the index of the first bit in m_1 and m_2 that differ. In other words, our ORE scheme leaks some information about the relative distance between the underlying messages.

We give a brief overview of our PRF-based construction. The secret key in our scheme consists of two PRF keys k_1 and k_2. A ciphertext is a tuple of PRF evaluations on the prefixes of the message. More precisely, to encrypt an n-bit message $m = m_1 m_2 \cdots m_n$, the encryption algorithm computes the following PRF outputs for each $i \in [n]$:

$$u_i = F(k_1, m_1 m_2 \cdots m_i) \qquad \text{and} \qquad v_i = F(k_2, m_1 m_2 \cdots m_{i-1}) + m_i \pmod 3.$$

The ciphertext is the concatenation of the PRF outputs: $\mathsf{ct} = (u_1 \| v_1, \ldots, u_n \| v_n)$.

To compare encryptions $\mathsf{ct} = (u_1 \| v_1, \ldots, u_n \| v_n)$ and $\mathsf{ct}' = (u'_1 \| v'_1, \ldots, u'_n \| v'_n)$ of messages m and m', the evaluator first finds the first index i for which $u_i \neq u'_i$. Since u_i and u'_i are PRF evaluations on the length-i prefixes of m and m', the first index i for which $u_i \neq u'_i$ is the first bit of m and m' that differ. After identifying the i^{th} bit that differs, the evaluator uses v_i and v'_i to determine which message has 0 as the i^{th} bit and which message has 1.[4] Conversely, if $u_i = u'_i$ for all i, then $\mathsf{ct}_i = \mathsf{ct}'_i$, and so $m = m'$. Correctness and security of this construction both follow from security of the PRF (Theorems 3.1 and 3.2).

Ciphertexts in our candidate scheme are $O(\lambda n)$ bits, where λ is a security parameter and n is the bit-length of the message. As a point of comparison, ciphertexts in the OPE scheme of

[3] There are "mutable" order-preserving encryption schemes [PLZ13, KS14, Ker15] that do satisfy IND-OCPA, but they require stateful encryption, and oftentimes, an interactive protocol to "update" ciphertexts. We survey some of these constructions in Section 1.2.

[4] Either $v_i + 1 = v'_i \pmod 3$, in which case $m < m'$, or $v_i - 1 = v'_i \pmod 3$, in which case $m > m'$.

Boldyreva et al. [BCLO09] are only $O(n)$-bit. While the ciphertexts in our scheme are longer (by a multiplicative factor λ), we note that the security provided by the Boldyreva et al. scheme does not increase by a noticeable amount if we further increase the size of the ciphertext space.[5]

We also describe a simple extension that allows for shorter ciphertexts in our ORE scheme. In particular, we show that if we view the messages in base-d rather than base-2, we can (after making appropriate changes to the scheme) achieve a $O(\log d)$ reduction in the size of the ciphertexts. However, this results in greater leakage, and so, our extension provides a space/security trade-off that may be appropriate depending on the application.

Comparison with existing schemes. First, we note in Section 2.3 that the security of any OPE scheme can be "augmented" by applying ORE encryption on top of OPE encryption. The resulting scheme is at least as secure as the underlying OPE scheme, and moreover, inherits the security properties of the ORE scheme. Hence, by composing our ORE construction with existing OPE constructions, we obtain ORE schemes that are at least as secure.

While composing an OPE scheme with an ORE scheme yields a scheme that is at least as secure as the underlying OPE scheme, we show that even without this composition, our basic ORE scheme still achieves stronger security guarantees according to the one-wayness metrics introduced by Boldyreva et al. [BCO11] for analyzing the leakage of random order-preserving functions (and by extension, any OPE scheme that is POPF-CCA secure). In our work, we introduce two more-general one-wayness notions and show that under a uniform plaintext distribution,[6] our basic ORE scheme achieves strictly stronger security compared to OPE schemes that are POPF-CCA secure. Specifically, Boldyreva et al. [BCO11] show that a random order-preserving function leaks half of the most-significant bits of the messages with probability close to 1. In contrast, under the same settings, we can show that our basic ORE scheme will not leak *any constant* fraction of the message bits with *overwhelming* probability.

1.2 Related Work

In recent years, there have been numerous works on order-preserving encryption and related notions [AKSX04, BCLO09, BCO11, PR12, PLZ13, TYM14, KS14, Ker15, MCO+15, RACY15]. In this section, we survey some of these works.

Security definitions. Though the POPF-CCA security definition introduced by Boldyreva et al. [BCLO09] is similar in flavor to PRF security, it is not immediately evident what kind of information the output of a random order-preserving function leaks about its input. In a follow-up work [BCO11], Boldyreva et al. introduce several notions (based on definitions of one-wayness [Gol01] for one-way functions) to capture the information leakage in schemes that are POPF-CCA secure. They show that a random order-preserving function leaks at least half of the bits in each message.

Teranishi et al. [TYM14] also introduce a stronger indistinguishability-based notion (stronger than the one-wayness definitions from [BCO11], but weaker than IND-OCPA) for OPE schemes, as well as a construction that achieves these stronger notions. Notably, their definition ensures that under a uniform message distribution, any fraction of the low-order bits of the messages being encrypted are hidden.

[5] Boldyreva et al. [BCO11] remark that for n-bit messages, it suffices for security that ciphertexts have $2n$ bits. They note that further increasing the ciphertext length has little to no effect on the security of the scheme.

[6] This is the only distribution for which we have concrete analysis of the leakage in any POPF-CCA secure scheme.

Modular OPE. Boldyreva et al. also introduced the notion of modular OPE as a possible extension of standard OPE [BCO11]. In modular OPE, a modular shift is applied to each plaintext before applying OPE—so the scheme is not order-preserving, but naturally supports "wrap-around" range queries. Their modular OPE scheme adds an extra layer of security to vanilla OPE, but it is worth noting that leakage of a small amount of information (say, a single plaintext-ciphertext pair) reveals the shift value and nullifies this added security. Subsequently, Mavroforakis et al. [MCO+15] designed several protocols to avoid leaking the shift value while using modular OPE schemes in practice.

Mutable OPE. Popa et al. [PLZ13] introduced a related notion of a mutable order-preserving encoding scheme which can be viewed as a two-party protocol that allows a user to insert and store encrypted values in a database such that the database is able to perform comparisons and range queries on the encrypted values without learning anything more about the values. Their construction is interactive and leverages stateful encryption. By working in this setting, the authors are able to circumvent the Boldyreva et al. [BCLO09] lower bound for order-preserving encryption and show that their scheme is IND-OCPA secure.

In subsequent work, Kerschbaum and Schröpfer [KS14] improved on the communication complexity of the Popa et al. construction at the expense of increasing the amount of client-side state. Specifically, in their construction, the amount of persistent state the client has to maintain increases linearly in the number of elements inserted into the database. More recently, Kerschbaum [Ker15] introduced a new notion of frequency-hiding OPE that introduces additional randomness to hide whether multiple ciphertexts encrypt the same value. Their notions provide a strictly stronger guarantee than IND-OCPA.

Very recently, Roche et al. [RACY15] introduced the notion of partial order-preserving encodings, which optimizes for the setting where there are a huge number of insertion queries but only a moderate number of range queries. Their protocol improves upon the round-complexity for insertions compared to the Popa et al. protocol [PLZ13], and requires the client to maintain less state than the Kerschbaum-Schröpfer construction [KS14]. All of the schemes described here require stateful encryption and employ an interactive encryption procedure.

ORE. Order-revealing encryption schemes, as introduced by Boneh et al. [BLR+15] provide another method of circumventing the Boldyreva et al. lower bound [BCLO09]. In an ORE scheme, the public comparison operation is not required to correspond to numerically comparing the ciphertexts, and in fact, the ciphertexts themselves need not be elements of a numeric, well-ordered set. This type of relaxation was previously considered by Pandey and Rouselakis [PR12] in the context of property-preserving encryption. In a property-preserving encryption scheme, there is a publicly computable function that can be evaluated on ciphertexts to determine the value of some property on the underlying plaintexts. Order-revealing encryption can thus be viewed as a property-preserving encryption scheme for the comparison operation. Pandey and Rouselakis introduce and explore several indistinguishability-based notions of security for property-preserving encryption; however, they do not construct an order-revealing encryption scheme.

To the best of our knowledge, all existing ORE schemes that provide IND-OCPA security either rely on very strong (and currently impractical) cryptographic primitives such as indistinguishability obfuscation [GGG+14] and cryptographic multilinear maps [BLR+15], or only achieve a weaker notion of security [AJ15] when instantiated with simple cryptographic primitives such as public

key cryptography. For the constructions based on indistinguishability obfuscation or multilinear maps [GGG+14, BLR+15], security of the ORE scheme is conditional on the conjectured security of cryptographic multilinear maps [BS03, GGH13a, CLT13, GGH15, CLT15].[7] However, in the last few months, numerous attacks [CHL+15, BWZ14, CGH+15, HJ15, CLR15, MF15, Cor15] on these multilinear maps have emerged, raising some doubts about the security of constructions that leverage them.

To avoid multilinear maps in favor of more well-studied number-theoretic or lattice-based assumptions, one can apply Ananth and Jain's arity-amplification technique [AJ15] to a single-input functional encryption scheme based on public-key encryption [SS10, GVW12]. However, due to limitations of the underlying functional encryption schemes, the resulting ORE scheme only provides "bounded-message" security—that is, security only holds if there is an *a priori* (polynomial) bound on the maximum number of messages that will be encrypted. Moreover, the length of the ciphertexts in this scheme grows *polynomially* in the bound on the number of messages that will be encrypted. These constraints severely limit the practicality of the resulting ORE scheme. To obtain full semantic security, it would be necessary to apply the arity-amplification transformation to a more powerful functional encryption scheme, but to date, the only known candidates of such schemes rely again on indistinguishability obfuscation [GGH+13b] or multilinear maps [GGHZ14].

Recently, Bun and Zhandry [BZ15] investigated the connection between order-revealing encryption and problems in learning theory.

Other schemes. Numerous ad hoc or heuristic order-preserving encryption schemes [BHF09, KAK10, XYH12] have been proposed in the literature, but most lack formal security analysis.

2 Order-Revealing Encryption

In this section, we establish and review some conventions that we use in this work, and also formally define our security notions for our encryption schemes.

Preliminaries. For $n \in \mathbb{N}$, we write $[n]$ to denote the set of integers $\{1, \ldots, n\}$. If $\mathcal{P}(x)$ is a predicate on x, we write $\mathbf{1}(\mathcal{P}(x))$ to denote the indicator function for $\mathcal{P}$: that is, $\mathbf{1}(\mathcal{P}(x)) = 1$ if and only if $\mathcal{P}(x) = 1$, and 0 otherwise. If $x, y \in \{0,1\}^*$ are bit-strings, we write $x \| y$ to denote the concatenation of x and y. For a finite set S, we write $\mathsf{Unif}(S)$ to denote the uniform distribution on S. We say a function $f(\lambda)$ is negligible in a security parameter λ if $f = o(1/\lambda^c)$ for all $c \in \mathbb{N}$. We write $\mathrm{negl}(\lambda)$ to denote a negligible function in λ and $\mathrm{poly}(\lambda)$ to denote a polynomial in λ. We say an event occurs with negligible probability if the probability of the event is $\mathrm{negl}(\lambda)$, and it occurs with overwhelming probability if the complement of the event occurs with negligible probability. Finally, we review the definition of a pseudorandom function (PRF) [GGM84]. Let $\mathsf{Funs}[\mathcal{D}, \mathcal{R}]$ denote the set of all functions from a domain $\mathcal{D}$ to a range $\mathcal{R}$. Unless otherwise noted, we specialize the domain of our PRFs to $\{0,1\}^n$ and the range to $\{0,1\}^\lambda$.

Definition 2.1 (Pseudorandom Function [GGM84])**.** Fix a security parameter λ. A PRF $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^\lambda$ with key space $\mathcal{K}$, domain $\{0,1\}^n$, and range $\{0,1\}^\lambda$ is secure if for all

[7] To date, the only concrete instantiations of indistinguishability obfuscation [GGH+13b, BGK+14, AB15, Zim15] leverage multilinear maps.

efficient adversaries $\mathcal{A}$,

$$\left| \Pr\left[k \xleftarrow{R} \mathcal{K} : \mathcal{A}^{F(k,\cdot)}(1^\lambda) = 1\right] - \Pr\left[f \xleftarrow{R} \mathsf{Funs}[\{0,1\}^n, \{0,1\}^\lambda] : \mathcal{A}^{f(\cdot)}(1^\lambda) = 1\right]\right| = \mathrm{negl}(\lambda).$$

2.1 Order-Revealing Encryption

An order-revealing encryption (ORE) scheme is a tuple of algorithms $\Pi = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ defined over a well-ordered domain $\mathcal{D}$ with the following properties:

- $\mathsf{ORE.Setup}(1^\lambda) \to \mathsf{sk}$. On input a security parameter λ, the setup algorithm $\mathsf{ORE.Setup}$ outputs a secret key sk.
- $\mathsf{ORE.Encrypt}(\mathsf{sk}, m) \to \mathsf{ct}$. On input the secret key sk and a message $m \in \mathcal{D}$, the encrypt algorithm $\mathsf{ORE.Encrypt}$ outputs a ciphertext ct.
- $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2) \to b$. On input two ciphertexts ct_1, ct_2, the compare algorithm $\mathsf{ORE.Compare}$ outputs a bit $b \in \{0,1\}$.

Remark 2.2 (Public Parameters)**.** In general, the setup algorithm of an ORE scheme can also output public parameters pp which are then passed as an additional input to the comparison algorithm, as is done in Boneh et al. [BLR+15]. However, none of our constructions require these public parameters, so we omit them in this work for simplicity.

Remark 2.3 (Support for Decryption)**.** As described, our definition of an order-revealing encryption scheme does not include a "decryption" function. However, this omission is without loss of generality. To decrypt a message, the holder of the secret key can use the secret key to encrypt messages of her choosing, apply the comparison algorithm, and perform binary search to recover the message. An alternative method that avoids the need for binary search is to augment each ORE encryption of a message m with an encryption of m under a CPA-secure symmetric encryption scheme. The secret key of the ORE scheme would also include the key for the symmetric encryption scheme. As long as the underlying encryption scheme is CPA-secure, including this additional ciphertext does not compromise security. For the remainder of this work, we use the schema described above that does not explicitly specify a decryption function.

Correctness. Fix a security parameter λ. An ORE scheme $\Pi = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ over a well-ordered domain $\mathcal{D}$ is correct if for $\mathsf{sk} \leftarrow \mathsf{ORE.Setup}(1^\lambda)$, and all messages $m_1, m_2 \in \mathcal{D}$,

$$\Pr[\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2) = \mathbf{1}(m_1 < m_2)] = 1 - \mathrm{negl}(\lambda),$$

where $\mathsf{ct}_1 \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_1)$ and $\mathsf{ct}_2 \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_2)$, and the probability is taken over the random coins in $\mathsf{ORE.Setup}$ and $\mathsf{ORE.Encrypt}$.

Security. We now give our simulation-based notion of security for an ORE scheme. As described in Section 1.1, our security definition is parameterized by a leakage function $\mathcal{L}$, which exactly specifies what is leaked by an ORE scheme.

Definition 2.4 (Security of ORE with Leakage)**.** Fix a security parameter $\lambda \in \mathbb{N}$. Let $\Pi_{\mathsf{ore}} = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ be an ORE scheme. Let $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_q)$ be an adversary for some $q \in \mathbb{N}$. Let $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_q)$ be a simulator, and let $\mathcal{L}(\cdot)$ be a leakage function. We define the experiments $\mathsf{REAL}^{\mathrm{ORE}}_{\mathcal{A}}(\lambda)$ and $\mathsf{SIM}^{\mathrm{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}}(\lambda)$ as follows:

$\mathsf{REAL}^{\mathrm{ORE}}_{\mathcal{A}}(\lambda)$:

1. $\mathsf{sk} \leftarrow \mathsf{ORE.Setup}(1^\lambda)$
2. $(m_1, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_1(1^\lambda)$
3. $c_1 \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_1)$
4. for $2 \le i \le q$:
   (a) $(m_i, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_i(\mathsf{st}_{\mathcal{A}}, c_1, \ldots, c_{i-1})$
   (b) $c_i \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_i)$
5. output $(c_1, \ldots, c_q)$ and $\mathsf{st}_{\mathcal{A}}$

$\mathsf{SIM}^{\mathrm{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}}(\lambda)$:

1. $\mathsf{st}_{\mathcal{S}} \leftarrow \mathcal{S}_0(1^\lambda)$
2. $(m_1, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_1(1^\lambda)$
3. $(c_1, \mathsf{st}_{\mathcal{S}}) \leftarrow \mathcal{S}_1(\mathsf{st}_{\mathcal{S}}, \mathcal{L}(m_1))$
4. for $2 \le i \le q$:
   (a) $(m_i, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_i(\mathsf{st}_{\mathcal{A}}, c_1, \ldots, c_{i-1})$
   (b) $(c_i, \mathsf{st}_{\mathcal{S}}) \leftarrow \mathcal{S}_i(\mathsf{st}_{\mathcal{S}}, \mathcal{L}(m_1, \ldots, m_i))$
5. output $(c_1, \ldots, c_q)$ and $\mathsf{st}_{\mathcal{A}}$

We say that Π_{ore} is a secure ORE scheme with leakage function $\mathcal{L}(\cdot)$ if for all polynomial-size adversaries $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_q)$ where $q = \mathrm{poly}(\lambda)$, there exists a polynomial-size simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_q)$ such that the outputs of the two distributions $\mathsf{REAL}^{\mathrm{ORE}}_{\mathcal{A}}(\lambda)$ and $\mathsf{SIM}^{\mathrm{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}}(\lambda)$ are computationally indistinguishable.

Remark 2.5 (IND-OCPA Security)**.** We briefly note how the IND-OCPA definition of security is captured by this definition. Let $\mathcal{L}$ be the following leakage function:

$$\mathcal{L}(m_1, \ldots, m_n) = \{\mathbf{1}(m_i < m_j) : 1 \le i < j \le n\}.$$

If an ORE scheme is secure with leakage $\mathcal{L}$, then it is IND-OCPA secure.

2.2 Order-Preserving Encryption (OPE)

An OPE scheme [AKSX04, BCLO09] is a special case of an ORE scheme, where the ciphertext space is required to be a well-ordered range $\mathcal{R}$ and moreover, for two ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2 \in \mathcal{R}$, the comparison algorithm outputs 1 if $\mathsf{ct}_1 < \mathsf{ct}_2$. For simplicity, we can write an OPE scheme as a tuple of algorithms $\Pi = (\mathsf{OPE.Setup}, \mathsf{OPE.Encrypt})$ defined over a well-ordered domain $\mathcal{D}$ and well-ordered range $\mathcal{R}$ with the following properties:

- $\mathsf{ORE.Setup}(1^\lambda) \to \mathsf{sk}$. On input a security parameter λ, the setup algorithm $\mathsf{ORE.Setup}$ outputs a secret key sk.
- $\mathsf{ORE.Encrypt}(\mathsf{sk}, m) \to \mathsf{ct}$. On input the secret key sk and a message $m \in \mathcal{D}$, the encrypt algorithm $\mathsf{OPE.Encrypt}$ outputs a ciphertext $\mathsf{ct} \in \mathcal{R}$.

Correctness. An OPE scheme $\Pi = (\mathsf{OPE.Setup}, \mathsf{OPE.Encrypt})$ over a well-ordered domain $\mathcal{D}$ and well-ordered range $\mathcal{R}$ is correct if $\mathsf{sk} \leftarrow \mathsf{OPE.Setup}(1^\lambda)$, and all messages $m_1, m_2 \in \mathcal{D}$,

$$m_1 < m_2 \quad \iff \quad \mathsf{OPE.Encrypt}(\mathsf{sk}, m_1) < \mathsf{OPE.Encrypt}(\mathsf{sk}, m_2).$$

2.3 Composing OPE with ORE

By composing an ORE scheme with an OPE scheme, we obtain an ORE scheme whose security is at least as strong as the security of the underlying OPE scheme. Let $\Pi_{\mathsf{ope}} = (\mathsf{OPE.Setup}, \mathsf{OPE.Encrypt})$ be an OPE scheme and $\Pi^{\mathsf{in}}_{\mathsf{ore}} = (\mathsf{ORE}^{\mathsf{in}}.\mathsf{Setup}, \mathsf{ORE}^{\mathsf{in}}.\mathsf{Encrypt}, \mathsf{ORE}^{\mathsf{in}}.\mathsf{Compare})$ be an ORE scheme. Consider the following composed construction $\Pi_{\mathsf{ore}} = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ of an ORE scheme with an OPE scheme:

- $\mathsf{ORE.Setup}(1^\lambda)$. The setup algorithm runs $\mathsf{sk}_1 \leftarrow \mathsf{OPE.Setup}(1^\lambda)$ and $\mathsf{sk}_2 \leftarrow \mathsf{ORE}^{\mathsf{in}}.\mathsf{Setup}(1^\lambda)$. The secret key is $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2)$.
- $\mathsf{ORE.Encrypt}(\mathsf{sk}, \mathsf{m})$. The encryption algorithm outputs $\mathsf{ORE}^{\mathsf{in}}.\mathsf{Encrypt}(\mathsf{sk}_2, \mathsf{OPE.Encrypt}(\mathsf{sk}_1, \mathsf{m}))$.
- $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2)$. The compare algorithm outputs $\mathsf{ORE}^{\mathsf{in}}.\mathsf{Compare}(\mathsf{ct}_1, \mathsf{ct}_2)$.

Correctness of Π_{ore} follows immediately from the correctness of $\Pi^{\mathsf{in}}_{\mathsf{ore}}$ and Π_{ope}. Furthermore, we note that under our simulation-based definition of security, the composed scheme Π_{ore} is at least as secure as Π_{ope}. This intuition is formalized in the following remark, whose proof follows immediately by construction.

Remark 2.6 (Security of Composed Scheme)**.** For any leakage function $\mathcal{L}(\cdot)$, if the OPE scheme Π_{ope} is secure with leakage function $\mathcal{L}(\cdot)$, then the ORE scheme Π_{ore} is also secure with leakage function $\mathcal{L}(\cdot)$.

3 Main Construction

In this section, we give a construction of an ORE scheme for the set of n-bit positive integers with the following leakage function:

$$\mathcal{L}_{\mathsf{f}}(m_1, \ldots, m_t) := \{(\mathsf{ind}_{\mathsf{diff}}(m_i, m_j),\ \mathbf{1}(m_i < m_j)) : 1 \le i < j \le t\}, \tag{3.1}$$

where $\mathsf{ind}_{\mathsf{diff}}(x, y)$ gives the index of the first bit where x and y differ. If $x = y$, we set $\mathsf{ind}_{\mathsf{diff}}(x, y) = n + 1$. In other words, for $x \neq y$, if $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$, then $\mathsf{ind}_{\mathsf{diff}}(x, y)$ is the smallest index $\ell \in [n]$ for which $x_\ell \neq y_\ell$.

Construction. Fix a security parameter $\lambda \in \mathbb{N}$. Let $F : \mathcal{K} \times ([n] \times \{0,1\}^n) \to \{0,1\}^\lambda$ be a secure PRF. In the following, we will sometimes view the output of F as the binary representation of a λ-bit integer. Specifically, we will write $F(\cdot, \cdot) \pmod 3$ to denote the value obtained when taking the λ-bit output of F, treating it as an λ-bit integer, and reducing modulo 3. We define our ORE scheme $\Pi_{\mathsf{ore}} = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ as follows:

- $\mathsf{ORE.Setup}(1^\lambda)$. The setup algorithm chooses two uniformly random PRF keys k_1, k_2 for F. The secret key is $\mathsf{sk} = (k_1, k_2)$.
- $\mathsf{ORE.Encrypt}(\mathsf{sk}, m)$. Let $b_1 \cdots b_n$ be the binary representation of m. For each $i \in [n]$, the encryption algorithm computes

$$\begin{aligned} u_i &= F(k_1, (i, b_1 b_2 \cdots b_i \| 0^{n-i})) \\ v_i &= F(k_2, (i-1, b_1 b_2 \cdots b_{i-1} \| 0^{n-i+1})) + b_i \pmod 3, \end{aligned}$$

  and outputs $(u_1 \| v_1, \ldots, u_n \| v_n)$.

- $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2)$. The compare algorithm first parses

$$\mathsf{ct}_1 = (u_1 \| v_1, \ldots, u_n \| v_n)$$
$$\mathsf{ct}_2 = (u'_1 \| v'_1, \ldots, u'_n \| v'_n),$$

  where $u_1, \ldots, u_n, u'_1, \ldots, u'_n \in \{0,1\}^\lambda$ and $v_1, \ldots, v_n, v'_1, \ldots, v'_n \in \mathbb{Z}_3$. Let i be the smallest index where $u_i \neq u'_i$. If no such index exists, output 0. Otherwise, if $v'_i = v_i + 1 \pmod 3$, output 1. Otherwise, output 0.

We now show that the above ORE scheme Π_{ore} is correct and secure against the leakage function $\mathcal{L}_{\mathsf{f}}$ from Equation (3.1).

Theorem 3.1. *The ORE scheme Π_{ore} is correct under the PRF security of F.*

Proof. Fix any two messages $m_1, m_2 \in \{0,1\}^n$. To show correctness, we show that for $\mathsf{sk} \leftarrow \mathsf{ORE.Setup}(1^\lambda)$, and $\mathsf{ct}_i \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_i)$ for $i \in \{1,2\}$, then

$$\Pr[\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2) = \mathbf{1}(m_1 < m_2)] = 1 - \mathrm{negl}(\lambda). \tag{3.2}$$

To show this, we define a sequence of hybrid distributions over the ciphertexts $\{\mathsf{ct}_1, \mathsf{ct}_2\}$:

- **Hybrid** H_0: This is the real distribution. Specifically, $\mathsf{sk} \leftarrow \mathsf{ORE.Setup}(1^\lambda)$, and for $i \in \{1,2\}$, $\mathsf{ct}_i \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_i)$.

- **Hybrid** H_1: During the setup procedure, a uniformly random function $f_1 \xleftarrow{\mathrm{R}} \mathsf{Funs}[([n] \times \{0,1\}^n), \{0,1\}^\lambda]$ is sampled instead of the PRF key k_1. The PRF key k_2 is sampled as usual. When constructing the ciphertexts ct_1 and ct_2 in $\mathsf{ORE.Encrypt}$, the function $f_1(\cdot)$ is used in place of $F(k_1, \cdot)$. Evaluations of $F(k_2, \cdot)$ remain unchanged.

- **Hybrid** H_2: During the setup procedure, two uniformly random functions $f_1, f_2 \xleftarrow{\mathrm{R}} \mathsf{Funs}[([n] \times \{0,1\}^n), \{0,1\}^\lambda]$ are sampled. In the $\mathsf{ORE.Encrypt}$ function, $F(k_1, \cdot)$ is replaced by $f_1(\cdot)$ and $F(k_2, \cdot)$ is replaced by $f_2(\cdot)$.

Hybrids H_0 and H_1 are computationally indistinguishable by PRF security. The same holds for H_1 and H_2. Thus, it suffices to show that Equation (3.2) holds for the ciphertexts constructed as in H_2.

Let $b_1 \cdots b_n$ be the binary representation of m_1 and let $b'_1 \cdots b'_n$ be the binary representation of m_2. Next, write $\mathsf{ct}_1 = (u_1 \| v_1, \ldots, u_n \| v_n)$ and $\mathsf{ct}_2 = (u'_1 \| v'_1, \ldots, u'_n \| v'_n)$. We now show that with overwhelming probability, $m_1 < m_2$ if and only if $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2) = 1$.

- Suppose $m_1 < m_2$. Thus, there exists some index $i \in [n]$ such that $b_i = 0$ and $b'_i = 1$ and for all $j < i$, $b_j = b'_j$. By construction of ct_1 and ct_2 in H_2, for all $j < i$,

$$u_j = f_1(j, b_1 b_2 \cdots b_j \| 0^{n-j}) = f_1(j, b'_1, \ldots, b'_j \| 0^{n-j}) = u'_j.$$

  Next, since f_1 is a truly random function with range $\{0,1\}^\lambda$, and $b_i \neq b'_i$, the values of u_i and u'_i are independent and uniform over $\{0,1\}^\lambda$. Thus, $u_i = u'_i$ with probability $1/2^\lambda$, and so with probability $1 - \mathrm{negl}(\lambda)$, i is the first index in ct_1 and ct_2 where $u_i \neq u'_i$. Finally, since $b_1 \cdots b_{i-1} = b'_1 \cdots b'_{i-1}$, and using the fact that $b_i = 0$ and $b'_i = 1$, we have that $v'_i = v_i + 1 \pmod 3$, and $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2)$ outputs 1, as desired.

- Suppose that $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2) = 1$. This means that there exists an index $i \in [n]$ such that $u_i \neq u'_i$ and for all $j < i$, $u_j = u'_j$. We first argue that for all $j < i$, $b_j = b'_j$. Suppose there exists an index $\ell < i$ where $b_\ell \neq b'_\ell$. Since $\ell < i$, we know that

$$f_1(\ell, b_1 b_2 \cdots b_\ell \| 0^{n-\ell}) = u_\ell = u'_\ell = f_1(\ell, b'_1 \cdots b'_\ell \| 0^{n-\ell}).$$

  However, since f_1 is a truly random function, and $b_\ell \neq b'_\ell$, these two values are equal with probability $1/2^\lambda = \mathrm{negl}(\lambda)$. Using a union bound, we conclude that for all $\ell < i$, $b_j = b'_j$ with overwhelming probability. It suffices to argue that $b_i = 0$ and $b'_i = 1$. Since $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2) = 1$, we have that $v'_i = v_i + 1 \pmod 3$, and since $b_1 \cdots b_{i-1} = b_1 \cdots b'_{i-1}$, this implies that $b'_i = b_i + 1$. The claim follows. □

Next, we state the security theorem for Π_{ore}, but we defer the formal proof to Appendix A.

Theorem 3.2. *The order-revealing encryption scheme Π_{ore} is secure with respect to leakage function $\mathcal{L}_{\mathsf{f}}$ (Definition 2.4) under the PRF security of F.*

3.1 Space Usage and Security Trade-offs

For the security parameter λ, the order-revealing encryption scheme Π_{ore} on n-bit inputs produces encryptions of size $n(\lambda + 2)$. By considering the "d-ary" generalization of Π_{ore}, we obtain an order-revealing encryption scheme which on n-bit inputs produces ciphertexts of size $\lceil n/\log_2(d) \rceil \cdot (\lambda + \lfloor \log_2(d) \rfloor + 1)$. However, the reduction in ciphertext size comes with a loss in security. In particular, this d-ary generalization of Π_{ore} is secure with respect to the following more expressive leakage function:

$$\mathcal{L}^d_{\mathsf{f}}(m_1, \ldots, m_n) := \left\{ \left(\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_i, m_j),\, \mathbf{1}(m_i < m_j) \right) : 1 \leq i < j \leq n \right\}, \tag{3.3}$$

where $\mathsf{ind}^{(d)}_{\mathsf{diff}}(x, y)$ is defined as follows. For two d-ary inputs $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$ where $x \neq y$, we define

$$\mathsf{ind}^{(d)}_{\mathsf{diff}}(x, y) = (k, x_k - y_k),$$

where k is the smallest index where $x_k \neq y_k$. If $x = y$, then $\mathsf{ind}^{(d)}_{\mathsf{diff}}(x, y) = (n + 1, 0)$.

Generalized construction. Fix a security parameter $\lambda \in \mathbb{N}$. Let $F : \mathcal{K} \times ([n] \times \{0, \ldots, d-1\}^n) \to \{0,1\}^\lambda$ be a secure PRF. In the following, we will sometimes view the output of F as the d-ary representation of a λ-bit integer. Specifically, we will write $F(\cdot, \cdot) \pmod{2d-1}$ to denote the value obtained when taking the λ-bit output of F, treating it as a λ-bit integer, and reducing modulo $2d - 1$. We define our ORE scheme $\Pi_{\mathsf{gore}} = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ as follows:

- $\mathsf{ORE.Setup}(1^\lambda)$. The setup algorithm chooses two uniformly random PRF keys k_1, k_2 for F. The secret key is $\mathsf{sk} = (k_1, k_2)$.
- $\mathsf{ORE.Encrypt}(\mathsf{sk}, m)$. Let $b_1 \cdots b_n$ be the d-ary representation of m. For each $i \in [n]$, the encryption algorithm computes

$$\begin{aligned} u_i &= F(k_1, (i, b_1 b_2 \cdots b_i \| 0^{n-i})) \\ v_i &= F(k_2, (i-1, b_1 b_2 \cdots b_{i-1} \| 0^{n-i+1})) + b_i \pmod{2d-1}, \end{aligned}$$

  and outputs $(u_1 \| v_1, \ldots, u_n \| v_n)$.

- $\mathsf{ORE.Compare}(\mathsf{ct}_1, \mathsf{ct}_2)$. The compare algorithm first parses

$$\mathsf{ct}_1 = (u_1 \| v_1, \ldots, u_n \| v_n)$$
$$\mathsf{ct}_2 = (u'_1 \| v'_1, \ldots, u'_n \| v'_n),$$

  where $u_1, \ldots, u_n, u'_1, \ldots, u'_n \in \{0,1\}^\lambda$ and $v_1, \ldots, v_n, v'_1, \ldots, v'_n \in \mathbb{Z}_{2d-1}$. Let i be the smallest index where $u_i \neq u'_i$. If no such index exists, output 0. Otherwise, if there exists some $k \in [d-1]$ such that $v'_i = v_i + k \pmod{2d-1}$, output 1. Otherwise, output 0.

Correctness of the ORE scheme Π_{gore} follows almost identically as the correctness of Π_{ore} by adapting the correctness proof to deal with the d-ary representation of the message as opposed to the binary representation. The proof of security is similar to that for Theorem 3.2, and is deferred to Appendix B.

Theorem 3.3. *The order-revealing encryption scheme Π_{gore} is secure with respect to leakage function $\mathcal{L}^d_{\mathsf{f}}$ under the PRF security of F.*

4 Comparison to Existing OPE Schemes

We now compare the leakage of our order-revealing encryption scheme to that of existing order-preserving encryption schemes by Boldyreva et al. [BCLO09, BCO11]. As explained in Section 2.3, composing any existing OPE scheme with an ORE scheme results in a new ORE scheme which is at least as secure as the underlying OPE scheme.[8] In this section, we show that even *without* the composition, our construction still achieves stronger security according to the metrics proposed by Boldyreva et al.

The security definition achieved by an order-preserving encryption scheme is that the encryption function behaves like a random order-preserving function (ROPF) from the plaintext space to the ciphertext space. While this definition has the same flavor as that for PRFs, the behavior of a truly random function is very different from that of a random order-preserving function. In particular, the output of an order-preserving function is not independent of its input, and thus, reveals some information about the input. It turns out that quantifying the exact information leakage is a non-trivial task in general. However, under certain assumptions (for example, if the messages are drawn from a uniform distribution), it is possible to obtain concrete upper bounds on the information leakage [BCO11]. In particular, Boldyreva et al. propose two security notions, window one-wayness and window distance one-wayness, to analyze the security of an OPE scheme. In our setting, the nature of our security definition allows us to analyze the construction under a more generalized set of definitions compared to [BCO11]. We present these here.

4.1 One-Wayness

One of the most basic requirements of an encryption scheme is that it is one-way. Given a ciphertext, an adversary that does not have the secret key should not be able to recover the underlying message. In the standard definition of one-wayness [Gol01], the adversary is given the encryption of a random

[8] In most cases, the security of the composed scheme is strictly greater than that of the base OPE scheme since our ORE construction provides semantic security for a single ciphertext, whereas existing OPE schemes generally do not.

message, and its goal is to guess the message. This is a very weak notion of security, and even if an encryption is one-way, the adversary might still be able to deduce nontrivial information about the message given only the ciphertext. To address this, Boldyreva et al. [BCLO09] introduce a more general notion of one-wayness where the adversary is allowed to guess a contiguous interval (a window) in the one-wayness challenge. The adversary succeeds if the message is contained within the interval. Moreover, the adversary is given multiple encryptions (of random messages) and succeeds if it outputs an interval that contains at least one of the messages.

The notion of window one-wayness is useful for arguing that an adversary does not learn many of the *most significant* bits of the message, but if all bits of the message are equally sensitive, then this definition is less useful. In our work, we present a more general definition of one-wayness, where instead of outputting an interval, the adversary is allowed to specify a set of guesses. To allow the adversary to specify a super-polynomially-sized set of guesses, we instead require the adversary to submit a circuit C that encodes its set ($C(x) = 1$ if and only if x is in the set). By requiring that the circuit encodes a contiguous interval, we recover the window one-wayness definition by Boldyreva et al. [BCO11]. We now give our generalized definition.

Definition 4.1 (Generalized One-Wayness)**.** Fix a plaintext space $\mathcal{D}$ and let $\Pi = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ be an ORE over $\mathcal{D}$. The (r, z)-*generalized one-wayness advantage* of an adversary $\mathcal{A}$ against Π is given by

$$\mathsf{Adv}^{\mathsf{gow}}_{r,z,\Pi}(\mathcal{A}) \stackrel{\mathsf{def}}{=} \Pr[\mathsf{Expt}^{\mathsf{gow}}_{r,z,\Pi,\mathcal{A}}(1^\lambda) = 1],$$

where the (r, z)-generalized one-wayness experiment $\mathsf{Expt}^{\mathsf{gow}}_{r,z,\Pi,\mathcal{A}}(1^\lambda)$ is defined as follows:

Experiment $\mathsf{Expt}^{\mathsf{gow}}_{r,z,\Pi,\mathcal{A}}(1^\lambda)$**:**

1. $\mathsf{sk} \leftarrow \mathsf{ORE.Setup}(1^\lambda)$
2. sample $m_1, \ldots, m_z$ uniformly from $\mathcal{D}$ without replacement
3. for $i \in [z]$, $\mathsf{ct}_i \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_i)$
4. $C \leftarrow \mathcal{A}(\mathsf{ct}_1, \ldots, \mathsf{ct}_z)$, where $C : \mathcal{D} \to \{0, 1\}$ is a circuit
5. output 1 if $C(m_i) = 1$ for some $i \in [z]$ and $|\{x \in \mathcal{D} : C(x) = 1\}| \leq r$; otherwise, output 0

Remark 4.2 (Comparison with Existing One-Wayness Notions)**.** By restricting the parameters (r, z) and the classes of circuits the adversary is allowed to output, Definition 4.1 captures many existing notions of one-wayness. For example, when $r = z = 1$, we recover the usual notion of one-wayness [Gol01]. When the underlying plaintext space is the ring $\mathbb{Z}_M$ for some integer M and we require that the circuit output by the adversary encodes a contiguous interval of length at most r in $\mathbb{Z}_M$, our definition corresponds to the notion of window one-wayness introduced by Boldyreva et al. [BCO11].

We now state our security theorem, but defer the proof to Appendix C.

Theorem 4.3. *Fix a security parameter λ and a plaintext space $\{0, 1\}^n$ where $n = \omega(\log \lambda)$. Let Π_{ore} be the ORE scheme given at the beginning of Section 3. Then, for any constant $\varepsilon \in (0, 1]$, any $z = \mathrm{poly}(\lambda)$, and all efficient adversaries $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{gow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda) = \mathrm{negl}(\lambda),$$

where $r = 2^{n(1-\varepsilon)}$.

Comparison to existing schemes. When discussing the notion of one-wayness, we will always assume that the message-space is super-polynomial in the security parameter. Otherwise, the trivial adversary that just guesses a random point in the message space will succeed with non-negligible probability.

In [BCO11], Boldyreva et al. give an upper bound on the one-wayness advantage of any (possibly computationally unbounded) adversary $\mathcal{A}$ against a random order-preserving function ROPF. This corresponds to setting $r = 1$ in our definition. They show [BCO11, Theorem 4.1] that for $z = \text{poly}(\lambda)$, $\mathsf{Adv}^{\mathsf{gow}}_{1,z,\mathsf{ROPF},\mathcal{A}} = \text{negl}(\lambda)$. The same statement holds for our ORE construction assuming a computationally bounded adversary: simply instantiate Theorem 4.3 with $\varepsilon = 1$.

In addition to giving an upper bound on an adversary's ability to guess the plaintext from the ciphertext, Boldyreva et al. also give a lower bound on the advantage for the case when r is large. In particular, they exhibit an efficient adversary $\mathcal{A}$ against an ROPF such that $\mathsf{Adv}^{\mathsf{gow}}_{r,z,\mathsf{ROPF},\mathcal{A}}(1^\lambda) = 1 - 2e^{-b^2/2}$ for a constant b when $r = O(\sqrt{2^n})$ and for any z [BCO11, Theorem 4.2].[9] In other words, the authors describe a concrete adversary that is able to break the generalized one-wayness of any ROPF scheme (with probability close to 1) if the adversary is allowed to specify a set with $r = O(\sqrt{2^n})$ elements, even when $z = 1$. An intuitive way to understand this result is that given the output of an ROPF, an adversary can deduce roughly half of the bits of the associated input. In contrast, in our ORE scheme, if the adversary only sees a polynomial number of ciphertexts ($z = \text{poly}(\lambda)$), then invoking Theorem 4.3 with $\varepsilon = 1/2$, we have that for all efficient adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{gow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda) = \text{negl}(\lambda)$ where $r = \sqrt{2^n}$. In fact, as Theorem 4.3 demonstrates, the adversary's advantage remains negligible even if we further increase the size of the sets the adversary is allowed to submit.

Intuitively, our results show that if the adversary only sees a polynomial number of ciphertexts, then it does not learn any constant fraction ε of the bits in the underlying plaintext from each ciphertext. In contrast, with an ROPF, and correspondingly, any OPE scheme that realizes a ROPF, each ciphertext alone leaks *half* of the most-significant bits of the underlying plaintext.

Similarly, while the OPE scheme by Teranishi et al. [TYM14] can be shown to hide any constant fraction of the least significant bits of the plaintext, no such guarantee exists for the other bits of the plaintext. Note though that the security notion proposed in [TYM14] is indistinguishability-based and hence, stronger than the one-wayness security notions. In fact, our basic ORE construction (by itself) does not achieve their indistinguishability-based definition. However, by composing our ORE construction with their OPE construction, we obtain a resulting ORE scheme which is strictly more secure, since it inherits the security properties of the underlying OPE scheme as well as semantic security for a single ciphertext (Section 2.3, Remark 2.6).

4.2 Distance One-Wayness

Boldyreva et al. [BCO11] also introduce an additional metric they use to analyze the security of an OPE scheme called "window distance one-wayness." This metric helps quantify the extent to which an OPE scheme reveals information about the distance between the underlying plaintexts. In a semantically-secure ORE scheme, recall that no distance information between ciphertexts is revealed aside from information that can be directly inferred from the ordering of the ciphertexts.

[9] Strictly speaking, the adversary they describe is for the window one-wayness experiment, but any adversary that succeeds in the window one-wayness experiment also succeeds in the generalized one-wayness experiment (Definition 4.1).

As was the case for one-wayness, we present here a more general notion of window distance one-wayness, and then show that our proposed ORE scheme is robust under this definition.

Definition 4.4 (Generalized Distance One-Wayness)**.** Fix a plaintext space $\mathbb{Z}_M$ and let $\Pi = (\mathsf{ORE.Setup}, \mathsf{ORE.Encrypt}, \mathsf{ORE.Compare})$ be an ORE over $\mathbb{Z}_M$. The (r, z)-*generalized distance one-wayness advantage* of an adversary $\mathcal{A}$ against Π is given by

$$\mathsf{Adv}^{\mathsf{gdow}}_{r,z,\Pi}(\mathcal{A}) \stackrel{\mathsf{def}}{=} \Pr[\mathsf{Expt}^{\mathsf{gdow}}_{r,z,\Pi,\mathcal{A}}(1^\lambda) = 1],$$

where the (r, z)-generalized distance one-wayness experiment $\mathsf{Expt}^{\mathsf{gdow}}_{r,z,\Pi,\mathcal{A}}(1^\lambda)$ is defined as follows:

Experiment $\mathsf{Expt}^{\mathsf{gdow}}_{r,z,\Pi,\mathcal{A}}(1^\lambda)$**:**

1. $\mathsf{sk} \leftarrow \mathsf{ORE.Setup}(1^\lambda)$
2. sample $m_1, \ldots, m_z$ uniformly from $\mathbb{Z}_M$ without replacement
3. for $i \in [z]$, $\mathsf{ct}_i \leftarrow \mathsf{ORE.Encrypt}(\mathsf{sk}, m_i)$
4. $C \leftarrow \mathcal{A}(\mathsf{ct}_1, \ldots, \mathsf{ct}_z)$, where $C : \mathbb{Z}_M \to \{0, 1\}$ is a circuit
5. output 1 if there exists distinct $i, j \in [z]$ such that $C(d_{ij}) = 1$ where $d_{ij} = m_j - m_i \pmod M$ and $|\{x \in \mathbb{Z}_M : C(x) = 1\}| \leq r$; otherwise, output 0

Remark 4.5 (Comparison with Window Distance One-Wayness [BCO11])**.** In Definition 4.4, if we restrict the adversary to outputting circuits C that represent a contiguous interval of length at most r in $\mathbb{Z}_M$, then we recover the notion of windowed distance one-wayness from [BCO11, §3].

We now state our security theorem, but defer the proof to Appendix D.

Theorem 4.6. *Fix a security parameter λ and a plaintext space $\{0, 1\}^n$ (or equivalently, the ring $\mathbb{Z}_{2^n}$) where $n = \omega(\log \lambda)$. Let Π_{ore} be the ORE scheme given at the beginning of Section 3. Then, for any constant $\varepsilon \in (0, 1]$, any $z = \mathrm{poly}(\lambda)$, and all efficient adversaries $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{gdow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda) = \mathrm{negl}(\lambda),$$

where $r = 2^{n(1-\varepsilon)}$.

Comparison to existing schemes. As was done with one-wayness, Boldyreva et al. [BCO11] also give a lower and upper bound for the advantage of an adversary for a random order-preserving function in the distance one-wayness game. Specifically, they show [BCO11, Theorem 4.3] that when $r = 1$ and $z = \mathrm{poly}(\lambda)$, for all (not necessarily efficient) adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{gdow}}_{r,z,\mathsf{ROPF},\mathcal{A}} = \mathrm{negl}(\lambda)$. A matching upper bound holds for the advantage of all computationally bounded adversaries for our ORE scheme (simply instantiate Theorem 4.6 with $\varepsilon = 1$). Intuitively, this statement says that given encryptions of several uniformly sampled ciphertexts, the adversary cannot guess the *exact* distance between any pair of ciphertexts. This does not rule out learning *partial* information about the distances between ciphertexts, and as it turns out, some partial information is in fact leaked.

Similar to the case of one-wayness, when r is large, there exists an efficient adversary against the ROPF that can win the generalized distance one-wayness game with advantage close to 1.

Specifically, when $r = O(\sqrt{2^n})$, and any $z \geq 2$, there exists an efficient adversary $\mathcal{A}$ where $\mathsf{Adv}^{\mathsf{gdow}}_{r,z,\mathsf{ROPF},\mathcal{A}}(1^\lambda) = 1 - c$ where c is a constant (independent of λ) [BCO11, Theorem 4.4]. Note that the results in [BCO11] are with respect to adversaries restricted to submitting circuits that encode a contiguous interval in the ring $\mathbb{Z}_{2^n}$. Conceptually, this means that two ciphertexts encrypted under an OPE scheme that implements an ROPF reveal roughly half the bits of the distance between the underlying plaintext values.

In contrast, instantiating Theorem 4.6 with $\varepsilon = 1/2$, we have that for $r = \sqrt{2^n}$, and for all efficient adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{gdow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}$ is negligible. In fact, Theorem 4.6 states that given polynomially many ciphertexts, no efficient adversary is able to even learn a *constant* fraction of the bits in the distance between any pair of encrypted messages. Thus, in this setting, our ORE scheme provides a provably stronger security guarantee.

In the same work [BCO11], Boldyreva et al. also introduced the notion of a modular OPE (MOPE) scheme to obtain stronger security guarantees over a standard OPE scheme. At a high level, in an MOPE scheme, a secret and fixed modular offset is added to the plaintext value before encryption with an OPE scheme. On the one hand, an MOPE scheme substantially improves the one-wayness security of the underlying encryption scheme. Specifically, Boldyreva et al. show that the (r, w) window one-wayness advantage (Remark 4.2) of any adversary is optimal (no better than the trivial adversary that outputs a random window of size r). However, this security guarantee is unstable in the sense that a small piece of information (such as a single plaintext-ciphertext pair) reveals the secret offset, reverting the one-wayness security to that of the underlying OPE scheme. Moreover, information about the secret offset can be inferred from range query distribution in a naïve practical implementation [BCO11].

Subsequent work [MCO+15] proposed strategies to avoid leaking the MOPE secret offset when the distribution of range query inputs is known, through the use of "dummy queries." However, if the distribution is far from uniform, the number of necessary dummy queries can be prohibitive. In addition, Boldyreva et al. [BCO11] show that MOPE schemes do not help to hide the distances between plaintext values. Like an OPE scheme, an MOPE scheme reveals roughly half of the bits of the distance between two plaintext values. Thus, for a uniform message distribution, we conclude that our ORE scheme achieves better hiding properties on the distances between plaintexts, while retaining direct support for efficiently answering range queries.

5 Conclusions

In this work, we introduced a new notion of security for order-preserving, and more generally, order-revealing encryption. Our simulation-based security notion is defined with respect to a leakage function which precisely characterizes what the ciphertexts in the scheme leak about the underlying messages. We then give a practical order-revealing encryption scheme which achieves this security notion for a simple leakage function. By composing our ORE construction with existing OPE schemes, we obtain an ORE scheme with increased security. It is our hope that having a concrete leakage model will enable practitioners to make better-informed decisions on whether an ORE scheme is appropriate for their particular application. We conclude with several open problems:

1. Can we construct a practical ORE scheme with stronger security guarantees?
2. Can we reduce the ciphertext length of our ORE scheme while still maintaining a similar level of security?

3. Is it possible to build a practical ORE scheme with best-possible security from standard assumptions?

Acknowledgments

We would like to thank Sam Kim for helpful discussions about ORE. This work was partially supported by an NSF Graduate Research Fellowship. Opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Facebook.

References

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In *TCC*, pages 528–556, 2015.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *CRYPTO*, pages 308–326, 2015.

[AKSX04] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Order-preserving encryption for numeric data. In *SIGMOD*, pages 563–574, 2004.

[BCLO09] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. Order-preserving symmetric encryption. In *EUROCRYPT*, pages 224–241, 2009.

[BCO11] Alexandra Boldyreva, Nathan Chenette, and Adam O'Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *CRYPTO*, pages 578–595, 2011.

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *EUROCRYPT*, pages 221–238, 2014.

[BHF09] Carsten Binnig, Stefan Hildenbrand, and Franz Färber. Dictionary-based order-preserving string compression for main memory column stores. In *ACM SIGMOD*, pages 283–296, 2009.

[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 *EUROCRYPT*, pages 563–594, 2015.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

[BWZ14] Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. *IACR Cryptology ePrint Archive*, 2014:930, 2014.

[BZ15] Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. *IACR Cryptology ePrint Archive*, 2015:417, 2015.

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In *CRYPTO*, pages 247–266, 2015.

[CGKO06] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *ACM CCS*, pages 79–88, 2006.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *EUROCRYPT*, pages 3–12, 2015.

[CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the new CLT multilinear maps. *IACR Cryptology ePrint Archive*, 2015:934, 2015.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO*, pages 476–493, 2013.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In *CRYPTO*, pages 267–286, 2015.

[CM05] Yan-Cheng Chang and Michael Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *ACNS*, pages 442–455, 2005.

[Cor15] Jean-Sébastien Coron. Cryptanalysis of GGH15 multilinear maps, 2015.

[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 *EUROCRYPT*, pages 578–602, 2014.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *TCC*, pages 498–527, 2015.

[GGHZ14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Fully secure functional encryption without obfuscation. *IACR Cryptology ePrint Archive*, 2014:666, 2014.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *FOCS*, pages 464–479, 1984.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO*, pages 162–179, 2012.

[HJ15] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. *IACR Cryptology ePrint Archive*, 2015:301, 2015.

[KAK10] Hasan Kadhem, Toshiyuki Amagasa, and Hiroyuki Kitagawa. A secure and efficient order preserving encryption scheme for relational databases. In *KMIS*, pages 25–35, 2010.

[Ker15] Florian Kerschbaum. Frequency-hiding order-preserving encryption. In *ACM CCS*, pages 656–667, 2015.

[KS14] Florian Kerschbaum and Axel Schröpfer. Optimal average-complexity ideal-security order-preserving encryption. In *ACM CCS*, pages 275–286, 2014.

[MCO+15] Charalampos Mavroforakis, Nathan Chenette, Adam O'Neill, George Kollios, and Ran Canetti. Modular order-preserving encryption, revisited. In *ACM SIGMOD*, pages 763–777, 2015.

[MF15] Brice Minaud and Pierre-Alain Fouque. Cryptanalysis of the new multilinear map over the integers. *IACR Cryptology ePrint Archive*, 2015:941, 2015.

[PLZ13] Raluca A. Popa, Frank H. Li, and Nickolai Zeldovich. An ideal-security protocol for order-preserving encoding. In *IEEE Symposium on Security and Privacy*, pages 463–477, 2013.

[PR12] Omkant Pandey and Yannis Rouselakis. Property preserving symmetric encryption. In *EUROCRYPT*, pages 375–391, 2012.

[PRZB11] Raluca A. Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *SOSP*, pages 85–100, 2011.

[RACY15] Daniel Roche, Daniel Apon, Seung Geol Choi, and Arkady Yerukhimovich. POPE: Partial order-preserving encoding. Cryptology ePrint Archive, Report 2015/1106, 2015.

[Sky] Skyhigh Networks Inc. `https://www.skyhighnetworks.com/`. Accessed: 2015-11-12.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In *ACM CCS*, pages 463–472, 2010.

[TYM14] Isamu Teranishi, Moti Yung, and Tal Malkin. Order-preserving encryption secure beyond one-wayness. In *ASIACRYPT*, pages 42–61, 2014.

[XYH12] Liangliang Xiao, I-Ling Yen, and Dung T. Huynh. Extending order preserving encryption for multi-user systems. *IACR Cryptology ePrint Archive*, 2012:192, 2012.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In *EUROCRYPT*, pages 439–467, 2015.

A Proof of Theorem 3.2

Fix a security parameter λ and let $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_q)$ where $q = \text{poly}(\lambda)$ be an efficient adversary for the ORE security game (Definition 2.4). To prove security, we give an efficient simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ for which the outputs of the distributions $\mathsf{REAL}^{\text{ORE}}_{\mathcal{A}}(\lambda)$ and $\mathsf{SIM}^{\text{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}_\mathsf{f}}(\lambda)$ are computationally indistinguishable.

We use a hybrid argument similar to the argument used in the proof of Theorem 3.1. We begin by defining the hybrid experiments:

- **Hybrid** H_0: This is the real experiment $\mathsf{REAL}^{\text{ORE}}_{\mathcal{A}}(\lambda)$.
- **Hybrid** H_1: Same as H_0, except during $\mathsf{ORE.Setup}$, a random function $f_1 \xleftarrow{\text{R}} \mathsf{Funs}[([n] \times \{0,1\}^n), \{0,1\}^\lambda]$ is chosen. In all invocations of $\mathsf{ORE.Encrypt}$, the function $F(k_1, \cdot)$ is replaced by $f_1(\cdot)$.
- **Hybrid** H_2: Same as H_1, except during $\mathsf{ORE.Setup}$, two random functions $f_1, f_2 \xleftarrow{\text{R}} \mathsf{Funs}[([n] \times \{0,1\}^n), \{0,1\}^\lambda]$ are chosen. During $\mathsf{ORE.Encrypt}$, $f_1(\cdot)$ is used in place of $F(k_1, \cdot)$ and $f_2(\cdot)$ is used in place of $F(k_2, \cdot)$.

By the same argument as in the proof of Theorem 3.1, each consecutive pair of hybrids $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2$ is computationally indistinguishable under the PRF security of F. It suffices to show that there exists a simulator $\mathcal{S}$ such that the distribution of outputs in H_2 is computationally indistinguishable from $\mathsf{SIM}^{\text{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}_\mathsf{f}}(\lambda)$.

Description of the simulator. We now describe the simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$. First, $\mathcal{S}_0$ initializes two empty lookup tables $\mathsf{L}_b : [q] \times [n] \to \{0,1\}^\lambda$, for $b \in \{1,2\}$. It then outputs $\mathsf{st}_\mathcal{S} = (\mathsf{L}_1, \mathsf{L}_2)$. Then, for each $t \in [q]$, after the adversary outputs a query m_t, the simulation algorithm $\mathcal{S}_t$ is invoked on input $\mathsf{st}_\mathcal{S} = (\mathsf{L}_1, \mathsf{L}_2)$ and $\mathcal{L}_\mathsf{f}(m_1, \ldots, m_t)$. In particular, $\mathcal{L}_\mathsf{f}(m_1, \ldots, m_t)$ contains the values $\mathbf{1}(m_j < m_t)$ and $\mathsf{ind}_\mathsf{diff}(m_j, m_t)$ for all $j \in [t-1]$, where $\mathsf{ind}_\mathsf{diff}(m_j, m_t)$ is the index of the first bit in m_j and m_t that differ. For each $s \in [n]$, there are three cases to consider:

- **Case 1:** There exists a $j \in [t-1]$ such that $\mathsf{ind}_\mathsf{diff}(m_j, m_t) > s$. If there are multiple j for which $\mathsf{ind}_\mathsf{diff}(m_j, m_t) > s$, let j be the smallest one. Then, the simulator sets
$$\overline{u}_s = \mathsf{L}_1(j, s) \qquad \text{and} \qquad \overline{v}_s = \mathsf{L}_2(j, s).$$
- **Case 2:** For each $\ell \in [t-1]$, $\mathsf{ind}_\mathsf{diff}(m_\ell, m_t) \leq s$ and there exists a $j \in [t-1]$ for which $\mathsf{ind}_\mathsf{diff}(m_j, m_t) = s$. If there are multiple j for which $\mathsf{ind}_\mathsf{diff}(m_j, m_t) = s$, let j be the smallest one. Then, the simulator samples $y_1 \xleftarrow{\text{R}} \{0,1\}^\lambda$ and sets
$$\overline{u}_s = y_1 \qquad \text{and} \qquad \overline{v}_s = \mathsf{L}_2(j, s) - (1 - 2 \cdot \mathbf{1}(m_j < m_t)) \pmod 3.$$
- **Case 3:** For each $\ell \in [t-1]$, $\mathsf{ind}_\mathsf{diff}(m_\ell, m_t) < s$. In this case, the simulator samples $y_1, y_2 \xleftarrow{\text{R}} \{0,1\}^\lambda$ and sets
$$\overline{u}_s = y_1 \qquad \text{and} \qquad \overline{v}_s = y_2.$$

For each $s \in [n]$, the simulator adds the mapping $(t, s) \mapsto \overline{u}_s$ to L_1 and $(t, s) \mapsto \overline{v}_s$ to L_2. Finally, the simulator $\mathcal{S}_t$ outputs the ciphertext $\overline{\mathsf{ct}}_t = (\overline{u}_1 \| \overline{v}_1, \ldots, \overline{u}_n \| \overline{v}_n)$ and the updated state $\mathsf{st}_{\mathcal{S}} = (\mathsf{L}_1, \mathsf{L}_2)$. This completes the description of the simulator $\mathcal{S}$.

Correctness of the simulation. We show that the simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ perfectly simulates the distribution in hybrid H_2. Let $(\mathsf{ct}_1, \ldots, \mathsf{ct}_q)$ be the joint distribution of the ciphertexts output in hybrid H_2, and let $(\overline{\mathsf{ct}}_1, \ldots, \overline{\mathsf{ct}}_q)$ be the joint distribution of the ciphertexts output by the simulator. We proceed inductively in the number of queries q. The base case ($q = 0$) follows trivially.

Suppose now that $(\mathsf{ct}_1, \ldots, \mathsf{ct}_{t-1}) \equiv (\overline{\mathsf{ct}}_1, \ldots, \overline{\mathsf{ct}}_{t-1})$ for some $t \in [q]$. We show that the statement holds for $t + 1$. Consider the distributions of ct_t and $\overline{\mathsf{ct}}_t$. First, for any $j \in [t]$, write ciphertext ct_j as $(u_{j,1} \| v_{j,1}, \ldots, u_{j,n} \| v_{j,n})$ and $\overline{\mathsf{ct}}_j$ as $(\overline{u}_{j,1} \| \overline{v}_{j,1}, \ldots, \overline{u}_{j,n} \| \overline{v}_{j,n})$. For each $s \in [n]$, we again consider three cases:

- **Case 1:** There exists a $j \in [t-1]$ such that $\mathsf{ind}_{\mathsf{diff}}(m_j, m_t) > s$. If there are multiple j for which $\mathsf{ind}_{\mathsf{diff}}(m_j, m_t) > s$, let j be the smallest one. This means that m_j and m_t share a prefix of length at least s. Let $p \in \{0,1\}^s$ be the first s bits and $p' \in \{0,1\}^{s-1}$ be the first $s-1$ bits of this common prefix. Then, in hybrid H_2, we have

  $$u_{t,s} = f_1(s, p \| 0^{n-s}) = u_{j,s} \qquad \text{and} \qquad v_{t,s} = f_2(s, p' \| 0^{n-s+1}) = v_{j,s}.$$

  In the simulation,

  $$\overline{u}_{t,s} = \mathsf{L}_1(j, s) = \overline{u}_{j,s} \qquad \text{and} \qquad \overline{v}_{t,s} = \mathsf{L}_1(j, s) = \overline{v}_{j,s}.$$

  Since $j < t$, we conclude from the induction hypothesis that $(u_{t,s}, v_{t,s})$ and $(\overline{u}_{t,s}, \overline{v}_{t,s})$ are identically distributed.

- **Case 2:** For each $\ell \in [t-1]$, $\mathsf{ind}_{\mathsf{diff}}(m_\ell, m_t) \leq s$ and there exists a $j \in [t-1]$ such that $\mathsf{ind}_{\mathsf{diff}}(m_j, m_t) = s$. If there are multiple j for which $\mathsf{ind}_{\mathsf{diff}}(m_j, m_t) = s$, let j be the smallest one. This means that m_j and m_t share a prefix $p \in \{0,1\}^{s-1}$ of length $s-1$. Let $b_{j,s}$ be the s^{th} bit of m_j and let $b_{t,s}$ be the s^{th} bit of m_t. Then, in hybrid H_2, we have

  $$u_{t,s} = f_1(s, p \| b_{t,s} \| 0^{n-s}) \qquad \text{and} \qquad v_{t,s} = f_2(s, p \| 0^{n-s+1}) + b_{t,s} \pmod 3.$$

  In the simulation, $\overline{u}_{t,s}$ is a uniformly random string, and

  $$\overline{v}_{t,s} = \mathsf{L}_2(j, s) - (1 - 2 \cdot \mathbf{1}(m_j < m_t)) = \overline{v}_{j,s} - (1 - 2 \cdot \mathbf{1}(m_j < m_t)) \pmod 3.$$

  By assumption, none of the messages $m_1, \ldots, m_{t-1}$ begin with the prefix $p \| b_{t,s}$. Since $f_1(\cdot)$ is a truly random function, the value of $f_1(s, p \| b_{t,s} \| 0^{n-s})$ is uniform in $\{0,1\}^\lambda$ and independent of all other ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_{t-1}$. Thus, $u_{t,s}$ and $\overline{u}_{t,s}$ are identically distributed. Next, in hybrid H_2, $v_{j,s} = f_2(s, p \| 0^{n-s+1}) + b_{j,s}$. By assumption, $b_{j,s} \neq b_{t,s}$, so we can write $b_{t,s} = b_{j,s} - (1 - 2 \cdot \mathbf{1}(m_j < m_t))$. Thus, in hybrid H_2, we have

  $$v_{t,s} = f_2(s, p \| 0^{n-s+1}) + b_{t,s} = v_{j,s} - (1 - 2 \cdot \mathbf{1}(m_j < m_t)) \pmod 3.$$

  By the inductive hypothesis, $v_{j,s}$ and $\overline{v}_{j,s}$ are identically distributed, so we conclude that $(u_{t,s}, v_{t,s})$ and $(\overline{u}_{t,s}, \overline{v}_{t,s})$ are identically distributed.

- **Case 3:** For each $\ell \in [t-1]$, $\mathsf{ind}_{\mathsf{diff}}(m_\ell, m_t) < s$. Let $p \in \{0,1\}^{s-1}$ be the first $s-1$ bits of m_t. Let $b_{t,s}$ be the s^{th} bit of m_t. Then, in hybrid H_2, we have

$$u_{t,s} = f_1(s, p \| b_{t,s} \| 0^{n-s}) \qquad \text{and} \qquad v_{t,s} = f_2(s, p \| 0^{n-s+1}) + b_{t,s} \pmod 3,$$

  while in the simulation $\overline{u}_{t,s}$ and $\overline{v}_{t,s}$ are uniformly random strings. By assumption, none of the messages $m_1, \ldots, m_{t-1}$ begin with the prefix p. Since $f_1(\cdot)$ and $f_2(\cdot)$ are truly random functions, the values of $f_1(s, p \| b_{t,s} \| 0^{n-s})$ and $f_2(s, p \| 0^{n-s+1})$ are uniform in $\{0,1\}^\lambda$ and independent of all other ciphertexts. Thus, $(u_{t,s}, v_{t,s})$ and $(\overline{u}_{t,s}, \overline{v}_{t,s})$ are identically distributed.

We conclude that for all $s \in [n]$, $(u_{t,s} \| v_{t,s}) \equiv (\overline{u}_{t,s} \| \overline{v}_{t,s})$. Since the components of each ciphertext are constructed independently in both hybrid H_2 and in the simulation, this suffices to show that ct_t and $\overline{\mathsf{ct}}_t$ are identically distributed. The claim then follows by induction on t. □

B Proof of Theorem 3.3

This proof is very similar to that of Theorem 3.2 from Appendix A. Fix a security parameter λ and let $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_q)$ where $q = \text{poly}(\lambda)$ be an efficient adversary for the ORE security game (Definition 2.4). To prove security, we give an efficient simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ for which the outputs of the distributions $\mathsf{REAL}^{\text{ORE}}_{\mathcal{A}}(\lambda)$ and $\mathsf{SIM}^{\text{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}^d_{\mathsf{f}}}(\lambda)$ are computationally indistinguishable.

We use a hybrid argument similar to the argument used in the proof of Theorem 3.1. We begin by defining the hybrid experiments:

- **Hybrid** H_0: This is the real experiment $\mathsf{REAL}^{\text{ORE}}_{\mathcal{A}}(\lambda)$.
- **Hybrid** H_1: Same as H_0, except during $\mathsf{ORE.Setup}$, a random function $f_1 \xleftarrow{\text{R}} \mathsf{Funs}[([n] \times \{0,1\}^n), \{0,1\}^\lambda]$ is chosen. In all invocations of $\mathsf{ORE.Encrypt}$, the function $F(k_1, \cdot)$ is replaced by $f_1(\cdot)$.
- **Hybrid** H_2: Same as H_1, except during $\mathsf{ORE.Setup}$, two random functions $f_1, f_2 \xleftarrow{\text{R}} \mathsf{Funs}[([n] \times \{0,1\}^n), \{0,1\}^\lambda]$ are chosen. During $\mathsf{ORE.Encrypt}$, $f_1(\cdot)$ is used in place of $F(k_1, \cdot)$ and $f_2(\cdot)$ is used in place of $F(k_2, \cdot)$.

By the same argument as in the proof of Theorem 3.1, each consecutive pair of hybrids $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2$ is computationally indistinguishable under the PRF security of F. It suffices to show that there exists a simulator $\mathcal{S}$ such that the distribution of outputs in H_2 is computationally indistinguishable from $\mathsf{SIM}^{\text{ORE}}_{\mathcal{A},\mathcal{S},\mathcal{L}^d_{\mathsf{f}}}(\lambda)$.

Description of the simulator. We now describe the simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$. First, $\mathcal{S}_0$ initializes two empty lookup tables $\mathsf{L}_b : [q] \times [n] \to \{0,1\}^\lambda$, for $b \in \{1,2\}$. It then outputs $\mathsf{st}_{\mathcal{S}} = (\mathsf{L}_1, \mathsf{L}_2)$. Then, for each $t \in [q]$, after the adversary outputs a query m_t, the simulation algorithm $\mathcal{S}_t$ is invoked on input $\mathsf{st}_{\mathcal{S}} = (\mathsf{L}_1, \mathsf{L}_2)$ and $\mathcal{L}^d_{\mathsf{f}}(m_1, \ldots, m_t)$. In particular, $\mathcal{L}^d_{\mathsf{f}}(m_1, \ldots, m_t)$ contains the values $\mathbf{1}(m_j < m_t)$ and $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t)$ for all $j \in [t-1]$, where $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t)$ consists of a pair (k, w) where $k \in [n]$ is the index of the first d-ary chunk in m_j and m_t that differ, $w = m_{j,k} - m_{t,k}$, and $m_{j,k}$ and $m_{t,k}$ are the k^{th} d-ary chunks of m_j and m_t, respectively. For each $s \in [n]$, there are three cases to consider:

- **Case 1:** There exists a $j \in [t-1]$ and $(k, w) \in [n] \times [-d, d]$ such that $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t) = (k, w)$ and $k > s$. If there are multiple j for which the property holds, let j be the smallest one. Then, the simulator sets

$$\overline{u}_s = \mathsf{L}_1(j, s) \qquad \text{and} \qquad \overline{v}_s = \mathsf{L}_2(j, s).$$

- **Case 2:** For each $\ell \in [t-1]$, there exists a $(k, z) \in [n] \times [-d, d]$ for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_\ell, m_t) = (k, z)$ and $k \leq s$, and there exists a $j \in [t-1]$ and $w \in [-d, d]$ for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t) = (s, w)$. If there are multiple j for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t) = (s, w)$ for some w, let j be the smallest one. Then, the simulator samples $y_1 \xleftarrow{\mathrm{R}} \{0,1\}^\lambda$ and sets

$$\overline{u}_s = y_1 \qquad \text{and} \qquad \overline{v}_s = \mathsf{L}_2(j, s) - w \pmod{2d-1}.$$

- **Case 3:** For each $\ell \in [t-1]$, there exist a $(k, z) \in [n] \times [-d, d]$ for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_\ell, m_t) = (k, z)$ and $k < s$. In this case, the simulator samples $y_1, y_2 \xleftarrow{\mathrm{R}} \{0,1\}^\lambda$ and sets

$$\overline{u}_s = y_1 \qquad \text{and} \qquad \overline{v}_s = y_2.$$

For each $s \in [n]$, the simulator adds the mapping $(t, s) \mapsto \overline{u}_s$ to L_1 and $(t, s) \mapsto \overline{v}_s$ to L_2. Finally, the simulator $\mathcal{S}_t$ outputs the ciphertext $\overline{\mathsf{ct}}_t = (\overline{u}_1 \| \overline{v}_1, \ldots, \overline{u}_n \| \overline{v}_n)$ and the updated state $\mathsf{st}_{\mathcal{S}} = (\mathsf{L}_1, \mathsf{L}_2)$. This completes the description of the simulator $\mathcal{S}$.

Correctness of the simulation. We show that the simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ perfectly simulates the distribution in hybrid H_2. Let $(\mathsf{ct}_1, \ldots, \mathsf{ct}_q)$ be the joint distribution of the ciphertexts output in hybrid H_2, and let $(\overline{\mathsf{ct}}_1, \ldots, \overline{\mathsf{ct}}_q)$ be the joint distribution of the ciphertexts output by the simulator. We proceed inductively in the number of queries q. The base case ($q = 0$) follows trivially.

Suppose now that $(\mathsf{ct}_1, \ldots, \mathsf{ct}_{t-1}) \equiv (\overline{\mathsf{ct}}_1, \ldots, \overline{\mathsf{ct}}_{t-1})$ for some $t \in [q]$. We show that the statement holds for $t+1$. Consider the distributions of ct_t and $\overline{\mathsf{ct}}_t$. First, for any $j \in [t]$, write ciphertext ct_j as $(u_{j,1} \| v_{j,1}, \ldots, u_{j,n} \| v_{j,n})$ and $\overline{\mathsf{ct}}_j$ as $(\overline{u}_{j,1} \| \overline{v}_{j,1}, \ldots, \overline{u}_{j,n} \| \overline{v}_{j,n})$. For each $s \in [n]$, we again consider three cases:

- **Case 1:** There exists a $j \in [t-1]$ and $(k, w) \in [n] \times [-d, d]$ such that $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t) = (k, w)$ and $k > s$. If there are multiple j for which this property holds, let j be the smallest one. This means that m_j and m_t share a prefix of length at least s. Let $p \in \{0, \ldots, d-1\}^s$ be the first s d-ary chunks and $p' \in \{0,1\}^{s-1}$ be the first $s-1$ d-ary chunks of this common prefix. Then, in hybrid H_2, we have

$$u_{t,s} = f_1(s, p \| 0^{n-s}) = u_{j,s} \qquad \text{and} \qquad v_{t,s} = f_2(s, p' \| 0^{n-s+1}) = v_{j,s}.$$

  In the simulation,

$$\overline{u}_{t,s} = \mathsf{L}_1(j, s) = \overline{u}_{j,s} \qquad \text{and} \qquad \overline{v}_{t,s} = \mathsf{L}_1(j, s) = \overline{v}_{j,s}.$$

  Since $j < t$, we conclude from the induction hypothesis that $(u_{t,s}, v_{t,s})$ and $(\overline{u}_{t,s}, \overline{v}_{t,s})$ are identically distributed.

- **Case 2:** For each $\ell \in [t-1]$, there exists a $(k, z) \in [n] \times [-d, d]$ for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_\ell, m_t) = (k, z)$ and $k \leq s$, and there exists a $j \in [t-1]$ and $w \in [-d, d]$ for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t) = (s, w)$. If there are multiple j for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_j, m_t) = (s, w)$ for some w, let j be the smallest one. This means that m_j and m_t share a prefix $p \in \{0,1\}^{s-1}$ of length $s-1$. Let $b_{j,s}$ be the s^{th} d-ary chunk of m_j and let $b_{t,s}$ be the s^{th} d-ary chunk of m_t. Then, in hybrid H_2, we have

  $$u_{t,s} = f_1(s, p \| b_{t,s} \| 0^{n-s})$$

  and

  $$v_{t,s} = f_2(s, p \| 0^{n-s+1}) + b_{t,s} \pmod{2d-1}.$$

  In the simulation, $\overline{u}_{t,s}$ is a uniformly random string, and

  $$\overline{v}_{t,s} = \mathsf{L}_2(j, s) - w = \overline{v}_{j,s} - w \pmod{2d-1}.$$

  By assumption, none of the messages $m_1, \ldots, m_{t-1}$ begin with the prefix $p \| b_{t,s}$. Since $f_1(\cdot)$ is a truly random function, the value of $f_1(s, p \| b_{t,s} \| 0^{n-s})$ is uniform in $\{0,1\}^\lambda$ and independent of all other ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_{t-1}$. Thus, $u_{t,s}$ and $\overline{u}_{t,s}$ are identically distributed. Next, in hybrid H_2, $v_{j,s} = f_2(s, p \| 0^{n-s+1}) + b_{j,s}$. By definition of $\mathsf{ind}^{(d)}_{\mathsf{diff}}$, we have that $w = b_{j,s} - b_{t,s}$, so we can write $b_{t,s} = b_{j,s} - w$. Thus, in hybrid H_2, we have

  $$v_{t,s} = f_2(s, p \| 0^{n-s+1}) + b_{t,s} = v_{j,s} - w \pmod{2d-1}.$$

  By the inductive hypothesis, $v_{j,s}$ and $\overline{v}_{j,s}$ are identically distributed, so we conclude that $(u_{t,s}, v_{t,s})$ and $(\overline{u}_{t,s}, \overline{v}_{t,s})$ are identically distributed.

- **Case 3:** For each $\ell \in [t-1]$, there exist a $(k, w) \in [n] \times [-d, d]$ for which $\mathsf{ind}^{(d)}_{\mathsf{diff}}(m_\ell, m_t) = (k, w)$ and $k < s$. Let $p \in \{0,1\}^{s-1}$ be the first $s-1$ d-ary chunks of m_t. Let $b_{t,s}$ be the s^{th} d-ary chunk of m_t. Then, in hybrid H_2, we have

  $$u_{t,s} = f_1(s, p \| b_{t,s} \| 0^{n-s})$$

  and

  $$v_{t,s} = f_2(s, p \| 0^{n-s+1}) + b_{t,s} \pmod{2d-1},$$

  while in the simulation $\overline{u}_{t,s}$ and $\overline{v}_{t,s}$ are uniformly random strings. By assumption, none of the messages $m_1, \ldots, m_{t-1}$ begin with the prefix p. Since $f_1(\cdot)$ and $f_2(\cdot)$ are truly random functions, the values of $f_1(s, p \| b_{t,s} \| 0^{n-s})$ and $f_2(s, p \| 0^{n-s+1})$ are uniform in $\{0,1\}^\lambda$ and independent of all other ciphertexts. Thus, $(u_{t,s}, v_{t,s})$ and $(\overline{u}_{t,s}, \overline{v}_{t,s})$ are identically distributed.

We conclude that for all $s \in [n]$, $(u_{t,s} \| v_{t,s}) \equiv (\overline{u}_{t,s} \| \overline{v}_{t,s})$. Since the components of each ciphertext are constructed independently in both hybrid H_2 and in the simulation, this suffices to show that ct_t and $\overline{\mathsf{ct}}_t$ are identically distributed. The claim then follows by induction on t.

C Proof of Theorem 4.3

We first define a sequence of hybrid experiments:

- Hybrid H_0: This corresponds to experiment $\mathsf{Expt}^{\mathsf{gow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda)$.

- Hybrid H_1: Same as H_0, except $m_1, \ldots, m_z \xleftarrow{R} \{0,1\}^n$ (rather than sampled without replacement).
- Hybrid H_2: Same as H_1, except the ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_z$ are constructed by invoking the simulator (Theorem 3.2) on $\mathcal{L}_{\mathsf{f}}(m_1, \ldots, m_z)$ where $\mathcal{L}_{\mathsf{f}}$ is the leakage function from Equation (3.1).
- Hybrid H_3: Fix a parameter $k = \omega(\log \lambda)$ where $k \leq n$. In hybrid H_3, after sampling the messages $m_1, \ldots, m_n \xleftarrow{R} \{0,1\}^n$, the experiment also samples messages $m'_i \xleftarrow{R} \{0,1\}^n$ subject to the restriction that the first k bits of m_i and m'_i are equal. The ciphertexts are constructed by invoking the simulator on $\mathcal{L}_{\mathsf{f}}(m'_1, \ldots, m'_z)$, but the success criterion at the end of the experiment is still checked using $m_1, \ldots, m_n$.

Lemma C.1. *Hybrids H_0 and H_1 are statistically indistinguishable.*

Proof. In H_0, the messages are sampled without replacement from $\{0,1\}^n$. In particular, this means that for $i \in [z]$, the message m_i is sampled from $\mathsf{Unif}(\{0,1\}^n \setminus \{m_1, \ldots, m_{i-1}\})$. But since $z = \mathrm{poly}(\lambda)$ and $n = \omega(\log \lambda)$ (so the size of the message space $\{0,1\}^n$ is super-polynomial in λ), the distribution $\mathsf{Unif}(\{0,1\}^n)$ is statistically close to $\mathsf{Unif}(\{0,1\}^n \setminus \{m_1, \ldots, m_{i-1}\})$ for all $i \in [z]$. Thus, H_0 and H_1 are statistically indistinguishable. □

Lemma C.2. *Hybrids H_1 and H_2 are computationally indistinguishable.*

Proof. Follows directly by simulation security of Π_{ore} (Theorem 3.2). □

Lemma C.3. *Hybrids H_2 and H_3 are statistically indistinguishable.*

Proof. It suffices to argue that $\mathcal{L}_{\mathsf{f}}(m_1, \ldots, m_n) = \mathcal{L}_{\mathsf{f}}(m'_1, \ldots, m'_n)$ with overwhelming probability. Take any distinct pair of indices i, j. With overwhelming probability $m_i \neq m_j$. Consider the probability $\Pr[\mathsf{ind}_{\mathsf{diff}}(m_i, m_j) = k]$. Since $\mathsf{ind}_{\mathsf{diff}}(m_i, m_j)$ gives the first bit on which messages m_i and m_j differ, $\mathsf{ind}_{\mathsf{diff}}(m_i, m_j) = k$ only if the first $k-1$ bits of m_i and m_j match and the k^{th} bit of m_i and m_j differ. Since m_i and m_j are both sampled uniformly and independently from $\{0,1\}^n$, we conclude that $\Pr[\mathsf{ind}_{\mathsf{diff}}(m_i, m_j) = k] = 2^{-k}$. Moreover, $\Pr[\mathsf{ind}_{\mathsf{diff}}(m_i, m_j) \geq k] = \sum_{s=k}^{n} 2^{-s} \leq 2^{-k+1}$. By a union bound, we have that for each $i \in [z]$,

$$\Pr[\forall j \neq i : \mathsf{ind}_{\mathsf{diff}}(m_i, m_j) < k] \geq 1 - \frac{z}{2^{k-1}}.$$

For $k = \omega(\log \lambda)$ we have that $\Pr[\forall j \neq i : \mathsf{ind}_{\mathsf{diff}}(m_i, m_j) < k] = 1 - \mathrm{negl}(\lambda)$. By another union bound, we conclude that $\mathsf{ind}_{\mathsf{diff}}(m_i, m_j) < k$ for all distinct i, j with probability $1 - \mathrm{negl}(\lambda)$. This means that all pairs of messages m_i, m_j differ on at least one of the first k bits. In H_3, the messages m_i and m'_i all agree on the first k bits for all $i \in [q]$, and so the first bit on which each pair of messages m'_i, m'_j differ is unaffected. We conclude that $\mathcal{L}_{\mathsf{f}}(m_1, \ldots, m_z) = \mathcal{L}_{\mathsf{f}}(m'_1, \ldots, m'_z)$, and the claim follows. □

To complete the proof we bound the adversary's advantage in hybrid H_3. Let C be the circuit output by the adversary in H_3, and let $S = \{x \in \{0,1\}^n : C(x) = 1\}$. Without loss of generality, assume that $|S| \leq r$ (otherwise, the adversary's advantage is 0). Take any message m_i and consider the probability $\Pr[m_i \in S]$. In hybrid H_3, the view of the adversary depends only on the first k bits of each message $m_1, \ldots, m_z$, and in fact, it is equivalent to sample the remaining $n - k$ bits of the test messages $m_1, \ldots, m_z$ *after* the adversary has submitted its circuit.[10] If the latter $n - k$ bits

[10] This is equivalent since $m \xleftarrow{R} \{0,1\}^n$, so each bit of m can be viewed as an independent and uniform value in $\{0,1\}$.

of m_i are chosen after the adversary has committed to the circuit C, then $\Pr[m_i \in S] \leq |S|/2^{n-k}$. By a union bound over each message, we have

$$\mathsf{Adv}^{\mathsf{gow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda) = \Pr[\exists i \in [z] : m_i \in S] \leq \frac{z\,|S|}{2^{n-k}} = \frac{z \cdot r}{2^{n-k}}.$$

We can set $k = \varepsilon n/2 = \omega(\log \lambda)$ since $n = \omega(\log \lambda)$. Finally, since $r = 2^{n(1-\varepsilon)}$, the advantage is bounded by

$$\mathsf{Adv}^{\mathsf{gow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda) \leq \frac{z \cdot 2^{n(1-\varepsilon)}}{2^{n(1-\varepsilon/2)}} = \frac{z}{2^{n\varepsilon/2}} = \mathrm{negl}(\lambda),$$

since $z = \mathrm{poly}(\lambda)$. □

D Proof of Theorem 4.6

The proof of this statement proceeds very similarly to that of Theorem 4.3. For completeness, we describe the set of hybrid arguments we use:

- Hybrid H_0: This corresponds to experiment $\mathsf{Expt}^{\mathsf{gdow}}_{r,z,\Pi_{\mathsf{ore}},\mathcal{A}}(1^\lambda)$.
- Hybrid H_1: Same as H_0, except $m_1, \ldots, m_z \xleftarrow{\mathrm{R}} \{0,1\}^n$.
- Hybrid H_2: Same as H_1, except the ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_z$ are constructed by invoking the simulator (Theorem 3.2) on $\mathcal{L}_{\mathsf{f}}(m_1, \ldots, m_z)$.
- Hybrid H_3: Fix a parameter $k = \omega(\log \lambda)$ where $k \leq n$. In hybrid H_3, after sampling the messages $m_1, \ldots, m_n \xleftarrow{\mathrm{R}} \{0,1\}^n$, the experiment also samples messages $m'_i \xleftarrow{\mathrm{R}} \{0,1\}^n$ subject to the restriction that the first k bits of m_i and m'_i are equal. The ciphertexts are constructed by invoking the simulator on $\mathcal{L}_{\mathsf{f}}(m'_1, \ldots, m'_z)$, but the success criterion at the end of the experiment is still checked using $m_1, \ldots, m_n$.

Invoking Lemmas C.1 through C.3 from the proof of Theorem 4.3, we have that hybrids H_0 and H_3 are computationally indistinguishable. Consider the probability that the experiment H_3 outputs 1. Let C be the circuit output by the adversary and let $S = \{x \in \{0,1\}^n : C(x) = 1\}$. It suffices to assume that $|S| \leq r$ since otherwise, the output of H_3 is 0. Take any distinct pairs of indices i, j and let $d_{ij} = m_j - m_i \pmod{2^n}$. Take any $z \in \{0,1\}^n$. Since the view of the adversary is *independent* of the last $n-k$ bits of m_j and m_j is sampled uniformly and independently from $\{0,1\}^n$, we conclude that

$$\Pr[d_{ij} = z] = \Pr[m_j = z + m_i \pmod{2^n}] \leq \frac{1}{2^{n-k}}.$$

By a union bound over the elements of S, we conclude that $\Pr[d_{ij} \in S] \leq r/2^{n-k}$. Applying another union bound to all pairs i, j, we conclude that in H_3,

$$\Pr[\forall i \neq j : d_{ij} \notin S] \geq 1 - \frac{r \cdot z^2}{2^{n-k}}.$$

As in the proof of Theorem 4.3, the theorem now follows by setting $k = \varepsilon n/2 = \omega(\log \lambda)$. □