Impossibility of Order-Revealing Encryption in Idealized Models

Mark Zhandry[1] and Cong Zhang[2]

[1] Department of Computer Science, Princeton University
[2] Department of Computer Science, Rutgers University

Abstract. An Order-Revealing Encryption (ORE) scheme gives a public procedure by which two ciphertexts can be compared to reveal the order of their underlying plaintexts. The ideal security notion for ORE is that *only* the order is revealed — anything else, such as the distance between plaintexts, is hidden. The only known constructions of ORE achieving such ideal security are based on cryptographic multilinear maps and are currently too impractical for real-world applications.

In this work, we give evidence that building ORE from weaker tools may be hard. Indeed, we show black-box separations between ORE and most symmetric-key primitives, as well as public key encryption and anything else implied by generic groups in a black-box way. Thus, any construction of ORE must either (1) achieve weaker notions of security, (2) be based on more complicated cryptographic tools, or (3) require non-black-box techniques. This suggests that any ORE achieving ideal security will likely be somewhat inefficient.

Central to our proof is a proof of impossibility for something we call *information theoretic ORE*, which has connections to tournament graphs and a theorem by Erdös. This impossibility proof will be useful for proving other black box separations for ORE.

Keywords. Black-box separations, Order-revealing encryption, Random oracle model, Generic group model

1 Introduction

Order preserving encryption (OPE) [1, 3, 4] and order revealing encryption (ORE) [5] have been proposed as useful tools to facilitate fast operations on encrypted databases, such as lookup and range queries.

Order Preserving Encryption (OPE). In OPE, plaintexts and ciphertexts are both integers, and encryption is monotonic: if $m_0 < m_1$, then $\mathsf{Enc}(k, m_0) < \mathsf{Enc}(k, m_1)$. Such a scheme allows, e.g., for binary search and range queries to be easily performed over encrypted data by replacing the plaintext comparisons with ciphertext comparisons. Boldyreva et al. [3] give an efficient construction using pseudorandom functions.

While clearly, such a scheme will reveal the order of the underlying plaintexts, one may hope that nothing else is revealed; for example, the distance

between plaintexts should not be learnable from the ciphertexts without the secret key. However, Boldyreva et al. also show that some additional leakage is necessary in OPE: any such scheme with polynomially-large ciphertexts will reveal some information beyond just the order of the plaintexts; in essence, their proof shows that the approximate distance of two plaintexts will be revealed. For their scheme, they instead prove a different notion of security, namely that encryption is indistinguishable from a random monotone function. Characterizing the kind of information revealed by such a scheme is non-trivial, and has only been analyzed in certain cases such as uniformly random plaintexts [4]. Despite being limited to non-ideal security notions, OPE has been deployed in real products[3] and been studied in applied research [21, 27, 25].

Order Revealing Encryption (ORE). In order to circumvent Boldyreva et al.'s [3] impossibility result, Boneh et al. [5] define a relaxation called order *revealing* encryption. Here, ciphertexts are no longer necessarily integers. Instead, integer comparison for ciphertexts is replaced by a more general comparison procedure Comp. The correctness requirement is, roughly, that

$$\mathsf{Comp}(\ \mathsf{Enc}(k, m_0),\ \mathsf{Enc}(k, m_1)\ ) = \begin{cases} \text{“}<\text{”} & \text{if } m_0 < m_1 \\ \text{“}=\text{”} & \text{if } m_0 = m_1 \\ \text{“}>\text{”} & \text{if } m_0 < m_1 \end{cases}$$

Boneh et al. give a construction using multilinear maps [7, 16, 13], and argue that their scheme reveals no information beyond the ordering of the plaintexts. We will call such an ORE scheme *ideal*. Alternate constructions achieving ideal leakage have since been proposed using multi-input functional encryption [5] or even single input functional encryption [9]. Unfortunately, as all known instantiations of functional encryption rely on multilinear maps anyway, all known constructions of ORE require multilinear maps as well.

Attacks on ORE. By considering more general comparison procedures for ciphertexts, ideal ORE provably leaks less information than OPE. Nevertheless, a series of works starting with Naveed et al. [23, 18, 14] have shown that when the adversary has a good estimate of the distribution of the data, even ideal ORE provides little protection. The problem is that the definition of ideal ORE, while precise, does not immediately provide any "semantically meaningful" guarantees for the privacy of the underlying data.

Despite these attacks, we still believe ORE is an interesting object to study for several reasons:

- ORE can still provide meaningful notions of security in some settings. For one example, suppose that each data point is sampled i.i.d. from some underlying secret distribution D with large min-entropy (so all samples are distinct),

[3] e.g. `https://www.skyhighnetworks.com`, `https://www.ciphercloud.com/`, `https://www.bluecoat.com/` and Cipherbase [2].

and suppose the adversary has no side information about the data. Then ideal ORE provably hides the distribution D, since all the adversary will see is a random ordering. Note that in contrast, since OPE reveals approximate differences, it will also reveal the approximate scale of D.
- ORE represents one of the simplest functionalities for functional encryption that we do not know how to construct from traditional tools. As such, ORE represents a potential stepping stone toward more advanced functionalities
- Finally, the comparison structure of ORE is shared with several other concepts in cryptography. For example, most collusion-resistant traitor tracing systems are built on top of *private linear broadcast* encryption [6], which is a form of encryption where there are N secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$, and messages are encrypted to numbers j. Any sk_i for $i \geq j$ can decrypt, but any sk_i for $i < j$ cannot. For another example, positional witness encryption [17] also has a similar comparison structure, and is currently the best way to prove security of witness encryption under "instance-independent" assumptions.

However, all known ideal ORE schemes are built on heavy tools, such as multilinear maps, and current multilinear map candidates are quite inefficient, meaning the resulting constructions of order-revealing encryption are far from practical use. Therefore, a natural question is:

Is it possible to build ideal ORE from efficient tools so that it can be practical?

1.1 Our Work

We make a first attempt toward answering the above question by showing that natural constructions of ORE from several simple tools are impossible. Specifically, we give black box impossibility results for building ORE from symmetric key cryptography or public key encryption.

Theorem 1 (Informal). *There is no fully black box construction of an ORE scheme for a super-polynomial plaintext space from random oracles, or any object that can be constructed from random oracles in a black box way, including one-way functions, collision resistant hashing, PRGs, PRFs, and block ciphers.*

Theorem 2 (Informal). *There is no fully black box construction of an ORE scheme for a super-polynomial plaintext space from generic groups, or any object that can be constructed from cryptographic groups in a black box way, including public key encryption and non-interactive key agreement.*[4]

[4] There is some overlap in the implications of Theorems 1 and 2, as generic groups can also be used to build much of symmetric key cryptography. However, we still separate our black-box separations into these two theorems for a couple reasons. First, Theorem 1 is simpler, and serves to highlight the ideas that will be needed for Theorem 2. Second, the random oracle model is a very natural way to model hash functions, and may capture many security properties desired of hash functions in addition to one-wayness and collision resistance (such as universal computational extractors). Our random oracle proof shows that *any* property that follows from

Thus, any black-box construction of order-revealing encryption will require tools with more involved structure, such as bilinear maps, multilinear maps, or lattice assumptions. Such tools tend to be less efficient than those needed to build symmetric cryptography or public key encryption. While we do not rule out non-black-box constructions, such constructions tend to be very inefficient. We, therefore, take our separations as evidence that some inefficiency is required to achieve order revealing encryption with ideal leakage.

In addition to proving Theorems 1 and 2, we also give a framework for proving black box separations for ORE from other cryptographic tools, which may be useful for extending our results.

1.2 Our Techniques

To prove our separation results, we start with an idealized model $\mathcal{M}$ capturing the primitive that we want to separate ORE from: in this work, we take $\mathcal{M}$ to be a random oracle or the generic group model [26].

We now imagine a very relaxed notion of order-revealing encryption using the model (relaxing the notion of ORE we consider only makes our separations stronger):

- There is no explicit decryption procedure[5]
- The scheme is only partially correct, in that Comp may result in an incorrect answer, but is noticeably biased towards the correct answer.[6]
- The scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ may make queries to the model $\mathcal{M}$
- The algorithms are allowed to run arbitrary computations; the only restrictions are that (1) the number of queries to $\mathcal{M}$ is polynomially bounded, and (2) that the length of ciphertexts is polynomially bounded. Running times and key sizes can be unbounded.
- For simplicity in the following discussion, we will also assume the algorithms are deterministic, although our analysis readily applies to randomized schemes as well.
- The adversary can only make polynomially-many queries to $\mathcal{M}$ and can only see a polynomial number of ciphertexts, but we do not consider its computational power.

We next give a general recipe for proving that such a relaxed order-revealing encryption scheme does not exist. To prove impossibility, we proceed in three steps:

---

a random oracle is insufficient for constructing ORE in a black-box way. In addition, to the best of our knowledge the random oracle and generic group models are incomparable, so providing both proofs gives the most complete separations.

[5] Though note that this is actually without loss of generality, since decryption can be derived from encryption and comparison by using a binary search.

[6] This is also essentially without loss of generality, as correctness can be boosted by running multiple instances of the scheme in parallel.

1. Compile any scheme satisfying the above requirements into one where Comp does not make any queries to $\mathcal{M}$.
2. Compile the resulting scheme into one where the entire scheme completely ignores $\mathcal{M}$. We call such ORE scheme information-theoretic ORE. This step may lose some level of correctness, so even starting from a perfectly correct scheme, the information-theoretic scheme will no longer be perfectly correct.
3. Finally, show that (even partially correct) information-theoretic ORE does not exist.

We now expand on the three steps above in reverse order:

Impossibility of information-theoretic ORE. In information-theoretic ORE, the public/secret key are allowed to be arbitrarily (e.g. exponentially) large, the running times of $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp}$ are allowed to be arbitrary, while security must hold for arbitrary adversaries. There is no mention of a model $\mathcal{M}$; the only constraints are that ciphertexts must be polynomially bounded, and that the adversary sees only a polynomial number of ciphertexts.

First, since the scheme is deterministic, we can assume that $\mathsf{Comp}(u, v)$ only outputs " $=$ " if u and v are actually the same. Indeed, if $\mathsf{Comp}(u, v) =$ " $=$ " for $u \neq v$, it means that u, v could not simultaneously be valid encryptions of two messages under the same secret key (since then Comp would report " $=$ " when the plaintexts are in fact not equal). Therefore, for $u \neq v$, if $\mathsf{Comp}(u, v) =$ " $=$ ", we can simply change the answer arbitrarily without affecting correctness. Hence, we will choose arbitrarily $\mathsf{Comp}(u, v) =$ " $<$ " or $\mathsf{Comp}(u, v) =$ " $>$ ". By a similar argument, we can also assume that $\mathsf{Comp}(u, v) =$ " $<$ " if and only if $\mathsf{Comp}(v, u) =$ " $>$ ".

Now, for such a scheme, we can construct an (exponentially large) graph $\mathcal{G}$ associated with the public key where nodes are all possible ciphertexts. There is a directed edge from node u to node v if $\mathsf{Comp}(u, v) =$ " $<$ ". Notice that any two distinct nodes have exactly one edge between them. $\mathcal{G}$ is therefore what is known as a tournament graph.

Let s be the number of nodes in $\mathcal{G}$, equivalently the number of ciphertexts. Let $[1, t]$ be plaintext space, which is assumed to be superpolynomial[7]. We show that $\log s$ — the bit length of ciphertexts — must be superpolynomial, a contradiction.

This graph must have a significant amount of structure. In our setting, every key k corresponds to a set S of t nodes in $\mathcal{G}$, the encryptions of each of the plaintext elements. Assuming the scheme is perfectly correct, these nodes form a complete DAG, with the encryption of 1 at the beginning and the encryption of t at the end. Therefore, $\mathcal{G}$ must contain many complete DAGs on t nodes.

Moreover, security imparts additional structure on $\mathcal{G}$. Security says, roughly, that the encryptions of any two polynomial-length sequences of ordered messages

[7] In reality, we would want the number of plaintexts to be exponential, but our impossibility rules out even superpolynomial message spaces.

must be indistinguishable. If we insist on *perfect* security, we have the following. For a given key k, consider the set T of encryptions of $1, \ldots, p$ for some polynomial p. Then by security, there must be some key k' such that T are the encryptions under k' of $2, \ldots, p+1$. Therefore, the encryption of 1 under k' will have an edge to each of the nodes in T. Notice that this property must hold *for any* set T that can be represented as the encryptions of $1, \ldots, p$ for some key k.

The situation above is reminiscent of a problem studied by Erdös [15]. He asked the question: suppose every set of p nodes is *dominated* by another node; that is, for every set T of p nodes, there is a node u such that u has an edge to each node in T. He showed that the number of nodes in any tournament graph satisfying this property must be exponential in p. The proof is by induction: for any graph $\mathcal{G}$ satisfying the property for p, there is a graph on half as many nodes that satisfies the property for $p-1$. Continuing until the base case $p = 1$, we see that there must be a graph $\mathcal{G}'$ that is exponentially smaller than $\mathcal{G}$, meaning $\mathcal{G}$ must be exponentially-large.

We prove an analog of Erdös's proof in our setting. Namely, we show that for any polynomial p, the number of nodes s in $\mathcal{G}$ must be exponential in p. Since s is exponential in any polynomial, then $\log s$ must larger than any polynomial, a contradiction. Our proof is inspired by Erdös's proof, except complicated in several ways:

- Our structure, while superficially similar, has several key differences. For example, there will be sets T that do not correspond to encryptions of $1, \ldots, p$ under one key. For example, T may be formed by encrypting $1, \ldots, p/2$ under k_1 and $1, \ldots, p/2$ under k_2.
- We do not insist on perfect security, but instead on statistical security. This means, for example, that the dominating property may not hold for all sets T that are encryptions of $1, \ldots, p$.

Nonetheless, we show an inductive argument that resolves these difficulties, and proves that s must be exponential in p for any polynomial p. Hence, $\log s$ must be larger than any polynomial, as desired.

The above discussion assumed that the scheme was perfectly correct. However, looking ahead, we would like to prove the impossibility for even partially correct schemes, where the output of Comp may be incorrect, but is biased toward the right answer. We show how to compile such a partially correct scheme into one that is perfectly correct. Then invoking the impossibility above, we see that even a partially correct scheme is impossible. The compilation is simple: first we run multiple instances of the scheme in parallel to boost correctness arbitrarily high, but still not necessarily perfect. However, we argue that we can boost correctness high enough so that, with high probability over the key, Comp will produce the right answer for all ciphertexts. Then we just change the scheme so that the key is chosen randomly from the set of "good" keys. This only negligibly affects security (since the key is "good" with high probability anyway). Verifying that a key is "good" will of course take exponential time since one must verify that it outputs the right answer for any possible pair of messages; however, this is fine since we do not place any computational restrictions.

Comparison to Boldyreva et al. [3]. Order *preserving* encryption is the special case of ORE where the entire ciphertext graph is actually one large DAG. Boldyreva et al.'s impossibility can be interpreted as a special case of our proof above where the graph is restricted to DAG. Our proof is much stronger, as it applies to much less structured graphs — any structure we use is solely a function of the correctness and security requirements, and no additional structure is assumed.

Compiling schemes where Comp does not make queries to $\mathcal{M}$. We show that if Comp does not make queries to $\mathcal{M}$, then it can be compiled into an information-theoretic scheme, and then we can apply the above impossibility to rule out the original scheme. Our compilation process works even if the starting scheme was only partially correct; since the impossibility above works with partially-correct schemes, we can still rule out partially correct schemes where Comp makes no oracle queries.

The process is simple. Since Comp does not make any queries to $\mathcal{M}$, the model is not needed outside of encryption. This means, in particular, that it makes sense to restrict the adversary from querying $\mathcal{M}$. Doing so only enhances security.

Next, we can simply have the secret key holder construct the oracle $\mathcal{M}$ for himself, and include it as part of the secret key. The description of the oracle might be exponential in size, but this is acceptable since we do not place any bounds on the key size or running time of the honest users. The result is a scheme which makes no reference to an idealized model.

Removing oracle queries from Comp. The final step is to remove oracle queries from Comp. This is the only part that is specific to the model $\mathcal{M}$ being considered. This step can be seen as an ORE analog of several recent results showing black box impossibilities for constructing obfuscation from simple objects. We note however, as expanded on below, that there are some crucial differences from obfuscation that make our proofs significantly different.

The Random Oracle Model. This first model $\mathcal{M}$ we consider is the random oracle model. Here, $\mathcal{M}$ just implements a random function $\mathcal{O}$. At a very high level, our compilation is conceptually similar to Canetti et al.'s [10] analogous compilation for program obfuscation. They show how to compile out a random oracle from the evaluation of an obfuscation scheme. Roughly, the idea is that evaluation of the obfuscated program will be "sensitive" only to the query points that were queried during the obfuscation; all other points will be independent of the obfuscated code, and hence can be answered randomly. Therefore, the obfuscator can just give the (polynomially-many) sensitive query answers out as part of the obfuscated code, and now the evaluator can answer any oracle query without actually making a call to the oracle.

In more detail, the sensitive queries can be split into two classes: "heavy" queries that are somewhat likely to be queried when evaluating the program on a

random input, and "light" queries that are unlikely to be queried. Canetti et al. first run the obfuscated code on a handful of random "test" points, and collect the random oracle queries and responses. By setting the number of test queries to be sufficiently large, they guarantee that all heavy queries will make it into the list of query/response pairs. Then they just output this list as part of the obfuscated code. Since an adversary could always run the code on random inputs and make the oracle queries, this cannot impact the security of the obfuscator. However now the evaluator, on a random input, will usually not need to make any oracle queries. Indeed, on a random input, the evaluator will likely only need to query on heavy inputs (or non-sensitive inputs, which can be answered randomly), which it already has included as a part of the obfuscated code.

The straightforward attempt at translating this approach to our setting is to first encrypt a handful of random test plaintexts, run the comparison procedure between each pair of test ciphertexts, and collect all of the oracle queries made. Then hand out the list of query/response pairs as part of the public key.

Unfortuantely, this strategy does not work, for at least three reasons:

- First, the test ciphertexts will allow one to learn the approximate difference between points, violating ORE security. In particular, using the ORE comparison procedure, one can compute the fraction of test ciphertexts lying between any two given ciphertexts. This fraction, scaled up by the size of the plaintext space, will approximately equal the difference between the plaintexts.
- Second, the notion of "sensitive" and "heavy" queries are specific to each individual plaintext, and not a global property of the encryption scheme. For example, it could be that to encrypt a message m, the oracle is queried on m. m will be a sensitive and heavy query point only for the message m. Therefore, as we increase the number of test ciphertexts, we also increase the number of sensitive and heavy queries, making it more difficult to ensure that we eventually capture all heavy queries for each ciphertext in question.
- Third, correctness will only hold for a plaintext drawn from the same distribution as the test points — namely random plaintexts — whereas our steps above require correctness to hold for *any* plaintexts

To overcome the first limitation, we will simply set our test ciphertexts to be the smallest and largest several elements of the plaintext space. Now for any two ciphertexts not at the extremes of the domain, there will be no test ciphertexts between; we can therefore restrict the domain of actual ciphertexts to a smaller interval so as to not collide with the test ciphertexts. This change unfortunately makes the third limitation even worse: the test elements now are the extreme elements in the plaintext space, but we need correctness to hold for all possible points in between.

To remedy the second limitation, we further modify the compiled scheme so that in addition to comparing all pairs of test ciphertexts, any new ciphertext is also compared to all of the test ciphertexts. If we set the number of test ciphertexts to be much larger than the number of heavy queries for a ciphertext, then hopefully these comparisons will generate all heavy queries. Indeed, each

comparison will generate heavy queries for one of the two ciphertexts being compared. Note, however, that at this point in the discussion, it could be the case that the comparisons only generate heavy queries for the test ciphertexts, which would be useless for establishing the correctness of the scheme.

To overcome this issue, as well as the third limitation above, we will invoke ORE security to switch back and forth between points in the middle of the plaintext space and the extreme points at the ends of the plaintext space. Using security (as opposed to an information-theoretic argument) means that the proof has to be phrased as a reduction, which requires a delicate analysis. For example, an adversary cannot necessarily test whether a query is sensitive or heavy, so our reduction cannot know if it learned all of the important queries for a particular ciphertext. We give the full details in Section 5.

The Generic Group Model. Next, we consider the generic group model. Here, there is a cyclic group $\mathbb{G}$. We will consider the group represented additively. Each group element is associated with a handle (that is, a bit string), and only the model $\mathcal{M}$ has access to the mapping. Everyone can query $\mathcal{M}$ on a group element g to get a handle h, and can also query $\mathcal{M}$ on two handles h_1, h_2, receiving the handle for the sum of corresponding group elements. However, it is not possible to query $\mathcal{M}$ on a handle h and recover the original group element g.

An equivalent formulation is the following. Instead of being able to query on two handles h_1, h_2 to get the handle for the sum, only the following is possible: query on a vector $\mathbf{h} = (h_1, \ldots, h_i)$ of handles corresponding to group elements $\mathbf{g} = (g_1, \ldots, g_i)$, and a vector $\mathbf{v} = (v_1, \ldots, v_i)$ of integers. The response will be a single bit: 0 if $\sum_j v_j g_j = 0$, and 1 otherwise. We call these queries *zero test* queries.

Our high-level proof strategy will be conceptually similar to Pass and Shelat [24], which show how to remove generic groups from obfuscation constructions[8]. However, our setting faces similar complications as to the random oracle setting above, requiring a much more delicate proof.

During encryption of a message m, Enc will query the generic group on several new group elements $g_1^{(m)}, \ldots, g_t^{(m)}$, obtaining handles. Now, when comparing two ciphertexts, Comp will make several zero test queries on various handles coming from m_0, m_1. Whenever Comp gets a 0 in response, it learns a linear constraint on the unknown g elements. Suppose the probability of getting a 0 in comparison is μ. We will assume that μ is noticeably large, since otherwise the zero test queries would be useless, as one could simulate them reasonably accurately just by always answering 1.

If the adversary sees q ciphertexts, the total number of constraints she can find will be $O(\mu q^2)$. And yet, the total number of unknown variables is only qt. For large enough q, this is much smaller than the number of constraints. The constraints are then necessarily linearly dependent. This means that, analogous

[8] We note that Mahmoody et al. [22] extend the Pass and Shelat result to any (even non-commutative) finite ring; we leave extending our impossibility to the non-commutative setting as an interesting open problem.

to the random oracle case above, the adversary will be able to answer zero test queries for herself based on the results of previous queries. We show using a similar strategy to the random oracle setting how to compile the ORE scheme in a way that preserves security and correctness, while removing the generic group oracle queries from Comp. Of course, formalizing this intuition is non-trivial, and we give the details in Section 6.

Difficulties for extending to bilinear and multilinear maps. Pass and Shelat's [24] proof naturally extends to bilinear maps and more generally constant-degree multilinear maps. A natural question is whether or not our techniques can be extended to these settings as well. Roughly, a bilinear map allows for zero-test queries that are degree 2 polynomials, and a multilinear map allows for even higher degree.

Pass and Shelat's proof, as well as ours, inherently relies on linear algebra, so does not immediately extend to non-linear settings. Indeed, their proofs and ours cannot possibly work for general multilinear maps, as there do exist black box constructions of obfuscation [8] and ORE [5] from polynomial-degree multilinear maps.

Nonetheless, Pass and Shelat show how to extend their result to *constant degree* multilinear maps. Essentially, the idea is to linearize the constant-degree polynomials by describing them as linear combinations of monomials. Then using similar arguments as in the generic group case, they show how to remove oracle queries from obfuscation.

Unfortunately, such linearization will not work in our setting, even in the bilinear map case. Once we linearize, the total number of variables grows $O((qt)^2)$, while the number of constraints is still only $O(\mu q^2)$. Since both grow with q^2, the number of variables always remains large than the number of constraints, so there is no linear dependence amongst the constraints. Without this linear dependence the proof falls apart. Another perspective for why the linearization does not work: in the bilinear group model, $\mathsf{Enc}(m)$ will query the generic group on new group elements $g_1^{(m)}, \ldots, g_t^{(m)}$, while the comparison on $\mathsf{Enc}(m_0), \mathsf{Enc}(m_1)$ learns a degree-2 constraint on the variables, containing monomials such as $g_1^{(m_0)} \cdot g_1^{(m_1)}$. However, note that this monomial *only* appears in constraints obtained when compairing encryptions of m_0 and m_1; any other pair of messages will give different monomials. Hence, the constraints for different pairs of ciphertexts are linearly independent, making it difficult (if not impossible) to argue that the results of certain comparisons will help us answer other comparisons. We leave it as an interesting open question whether our impossibility can be extended to, say, the bilinear map setting, and if not, giving a black-box construction of ORE from bilinear maps.

1.3 Discussion

In light of our impossibility, it is natural to ask: *now what?* Here, we briefly discuss possible other directions.

Weaker notions of security. One possibility is to consider weaker notions of security, where more than just the order is revealed. For example, [12] give a construction of ORE where the position of the most significant differing bit of two plaintexts is revealed, but nothing else (put another way, the difference is revealed, rounded to a power of two). Their construction is efficient, using only PRFs (which can in turn be built from one-way functions). [11] give a still-practical construction using bilinear maps which reveals even less, though still more than the ideal security notion. [19] give a notion of functional revealing encryption and build an efficient ORE under the standard DLIN assumption, while it leaks no less than [11].

An interesting direction is to extend our impossibility result to other leakage profiles, perhaps showing that the leakage profile of [12] is optimal for constructions based on one-way functions. Such an impossibility would require reworking several parts of our proof, since we use the ideal ORE leakage in several parts, including the impossibility of information-theoretic ORE, as well as the step removing random oracle queries from Comp.

Non-black-box constructions. Another option is to resort to non-black-box construction. We do not know if such a construction is possible. However, non-black-box techniques tend to result in inefficient schemes, as such a non-black-box construction is likely to be inefficient.

Other cryptographic tools. We only rule out black-box constructions from certain building blocks; other building blocks are still possible. For example, it may be possible to build ORE from the Learning With Errors (LWE) assumption, RSA or integer factorization, or bilinear/multilinear maps. Indeed, using multilinear maps of polynomial degree, it is possible to build ORE with ideal leakage, as shown by Boneh et al. [5]. However, many of the tools not covered by our impossibility, including polynomial-degree multilinear maps or learning with errors, involve large parameter sizes, likely resulting in somewhat impractical schemes. Nonetheless, we believe that constructing ideal ORE from weaker tools including LWE or bilinear maps, or providing black-box separations for these tools by building on our techniques, are fascinating open questions.

2 Background

NOTATION. For $n, n_1, n_2 \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}, [n_1, n_2] := \{n_1, \ldots, n_2\}$. Throughout this paper, $\lambda \in \mathbb{N}$ denote the security parameter. For a finite set $\mathcal{S}$, we denote $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a probabilistic algorithm A, we denote $y \leftarrow A(x; R)$ the process of running A on input x and randomness R, and assigning y the result. We let $\mathcal{R}_A$ denote the randomness space of A; we require $\mathcal{R}_A$ to be the form $\mathcal{R}_A = \{0,1\}^r$. We write $y \leftarrow A(x)$ for $y \leftarrow A(x, R)$ with uniformly chosen $R \in \mathcal{R}_A$, and we write $y_1, \ldots, y_m \leftarrow A(x)$ for $y_1 \leftarrow A(x), \ldots, y_m \leftarrow A(x)$ with fresh randomness in each execution. If A's

running time is polynomial in λ, then A is called probabilistic polynomial-time (PPT).

We say a function $\mu(n)$ is negligible if $\mu \in o(n^{-\omega(1)})$, and is non-negligible otherwise. We let $\mathsf{negl}(n)$ denote an arbitrary negligible function. If we say some $p(n)$ is poly, we mean that there is some polynomial q such that for all sufficiently large n, $p(n) \leq q(n)$. We say a function $\rho(n)$ is noticeable if the inverse $1/\rho(n)$ is poly. We use boldface to denote vector, i.e. $\boldsymbol{m}$; we denote $\boldsymbol{m}_i$ as the i-th component of $\boldsymbol{m}$ and $|\boldsymbol{m}|$ as the length of $\boldsymbol{m}$. The statistical distance of two random variables X and Y over some countable domain S is defined as $\mathsf{SD}(X;Y) = \frac{1}{2}\sum_{s\in S} |\Pr[X = s] - \Pr[Y = s]|$. We write $X \overset{d}{\approx} Y$ for $\mathsf{SD}(X;Y) \leq d$, and $X \overset{\mathsf{stat}}{\approx} Y$ for $\mathsf{SD}(X;Y) \leq 2^{-\lambda}$.

ORE. The following definition of syntax for order-revealing encryption makes explicit that comparison may use helper information (e.g. a description of a particular group) by incorporating a *public key*, denote pk.

Definition 3. *([ORE]). An ORE scheme with message space $[N]$ is a tuple of algorithms $\Pi = \mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp}$ with the following syntax.*

- *The key generation algorithm* Gen *is randomized, takes inputs* $(1^\lambda, N)$*, and always emits two outputs* $(\mathsf{pk}, \mathsf{sk})$*. We refer to the first output* pk *as the* public key *and the second output* sk *as the* secret key*.*
- *The encryption algorithm* Enc *takes inputs* (sk, m) *where* $m \in [N]$*, and always emits a single output* c*, that we refer to as a* ciphertext*.*
- *The comparison algorithm* Comp *takes inputs* (pk, c_1, c_2)*, and emits "$<$", "$=$" or "$>$", which indicates the order of the underlying plaintexts.*

If Comp *is simple integer comparison (i.e., if* $\mathsf{Comp}(\mathsf{pk}, c_1, c_2)$ *is a canonical algorithm that treats its the ciphertexts and binary representations of integers and tests which is greater) then the scheme is said to be an* order-preserving encryption (OPE) *scheme.*

Correctness for ORE. Intuitively, an ORE scheme is correct if the comparison algorithm can output the order of the underlying plaintexts. For any two message pair (m_0, m_1), let $\mathsf{Comp}(m_0, m_1)$ be the order of (m_0, m_1), where:

$$\mathsf{Comp}(m_0, m_1) = \begin{cases} \text{“}<\text{”} & m_0 < m_1 \\ \text{“}=\text{”} & m_0 = m_1 \\ \text{“}>\text{”} & m_0 > m_1 \end{cases}$$

we consider four notions of correctness:

- **Perfect Correctness.** For any message pair (m_0, m_1), we have

$$\Pr[\mathsf{Comp}(\mathsf{pk}, C_0, C_1) = \mathsf{Comp}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(), C_b = \mathsf{Enc}(\mathsf{sk}, m_b)] = 1$$

- **Almost Perfect Correctness.** There is a negligible function $\mu = \mathsf{negl}(\lambda)$ such that

$$\Pr[\exists(m_0, m_1), \mathsf{Comp}(\mathsf{pk}, C_{m_0}, C_{m_1}) \neq \mathsf{Comp}(m_0, m_1) : C_b = \mathsf{Enc}(\mathsf{sk}, m_b)] \leq \mu$$

  where the probability is taken over the choice of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}()$.
- **Statistical Correctness.** There is a negligible function $\mu = \mathsf{negl}(\lambda)$ such that for any (m_1, m_2)

$$\Pr[\mathsf{Comp}(\mathsf{pk}, C_0, C_1) = \mathsf{Comp}(m_0, m_1) : C_b = \mathsf{Enc}(\mathsf{sk}, m_b)] \geq 1 - \mu$$

  where the probability is taken over the choice of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}()$.
- **Partial Correctness.** There is a noticeable function $\rho(\lambda)$ such that, for any (m_1, m_2),

$$\Pr[\mathsf{Comp}(\mathsf{pk}, C_0, C_1) = \mathsf{Comp}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}()] \geq \frac{1}{2} + \rho$$

In this work, we also consider ORE in idealized models, where the scheme's algorithms have access to an oracle.

Definition 4. *([Idealized Model].) An idealized model is a deterministic function $\mathcal{M}$. $\mathcal{M}$ takes two inputs: a string k which is the seed for the model, and a query q. Unless otherwise stated, we allow all players — the honest parties, the protocol algorithms, and the adversary — to query $\mathcal{M}$. In a query to $\mathcal{M}$:*

- *Any player sends q to $\mathcal{M}$;*
- *The player receives $\mathcal{M}(k, q)$ in return.*

We will denote an ORE scheme Π in an idealized model $\mathcal{M}$ as $\Pi^{\mathcal{M}} = (\mathsf{Gen}^{\mathcal{M}}, \mathsf{Enc}^{\mathcal{M}}, \mathsf{Comp}^{\mathcal{M}})$. This notation means that key generation, encryption, and comparison have access to $\mathcal{M}$ and the outputs also depend on $\mathcal{M}$'s response. Our definitions of security and correctness for ORE easily extend to the idealized model, where the probabilities are over the random seed k that generates $\mathcal{M}$.

Efficiency for ORE. Typically in the literature, ORE is defined as having computationally efficient algorithms:

Definition 5. *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ be an ORE scheme with respect to the message space $[N]$. We say Π is computationally efficient if $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp}$ run in time polynomial in $(\log N, \lambda)$. If Π is a scheme in an idealized model $\mathcal{M}$, we additionally require that the algorithms only make a polynomial number of queries to $\mathcal{M}$.*

Here, we will generally not impose any such restrictions, and allow for computationally inefficient algorithms. We only impose two efficiency constraints. First, if the scheme is an ideal-model scheme, we still require the number of queries to be polynomial.

Definition 6. *Let $\Pi^{\mathcal{M}} = (\mathsf{Gen}^{\mathcal{M}}, \mathsf{Enc}^{\mathcal{M}}, \mathsf{Comp}^{\mathcal{M}})$ be an ORE scheme in an idealized model $\mathcal{M}$. We say Π is query efficient if $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp}$ only make a number of queries that is polynomial in $(\log N, \lambda)$.*

The second efficiency requirement (for both idealized model schemes and standard model schemes) is that the ciphertexts produced by the scheme are polynomial sized.

Definition 7. *Let Π (resp. $\Pi^{\mathcal{M}}$) be an ORE with respect to the message space $[N]$ (resp. in idealized model). We say Π ($\Pi^{\mathcal{M}}$) has succinct ciphertexts if the ciphertext length is polynomial in $(\log N, \lambda)$.*

We call a scheme for which there is no idealized model but which still has succinct ciphertexts an *information-theoretic* scheme.

Security for ORE. An ORE scheme leaks the order of the underlying plaintexts, so the ideal security notion for ORE is that *only* the order is revealed. Roughly speaking, given two sequences of message $\boldsymbol{m}, \boldsymbol{m}'$ such that $\mathsf{Comp}(\boldsymbol{m}_i, \boldsymbol{m}_j) = \mathsf{Comp}(\boldsymbol{m}'_i, \boldsymbol{m}'_j), \forall i, j \in |\boldsymbol{m}|$, the distribution of $\mathsf{Enc}(\boldsymbol{m})$ and $\mathsf{Enc}(\boldsymbol{m}')$ are statistically indistinguishable. We firstly consider a weak version, which we call t-time secure, with the restriction that $|\boldsymbol{m}| = |\boldsymbol{m}'| \leq t$, then we define an interactive game with an unbounded adversary in the following:

$\underline{\mathsf{t\text{-}SIND}(\mathcal{A})}$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(N, 1^\lambda)$; $m_1 < \ldots < m_t, m'_1 < \ldots < m'_t \leftarrow \mathcal{A}(\mathsf{pk}, N, 1^\lambda)$;
$\boldsymbol{C}_0 = (\mathsf{pk}, \mathsf{Enc}(\mathsf{sk}, m_1), \ldots, \mathsf{Enc}(\mathsf{sk}, m_t))$; $\boldsymbol{C}_1 = (\mathsf{pk}, \mathsf{Enc}(\mathsf{sk}, m'_1), \ldots, \mathsf{Enc}(\mathsf{sk}, m'_t))$;
$b' \leftarrow \mathcal{A}(\boldsymbol{C}_b)$; Return $(b \stackrel{?}{=} b')$

Fig. 1: t-time Static Indistinguishable Game

Definition 8. *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ be an ORE scheme with respect to the message space $[N]$. For any PPT (resp. unbounded) adversary $\mathcal{A}$ we define the game $\mathsf{t\text{-}SIND}(\mathcal{A})$ in figure 1. The advantage of $\mathcal{A}$ for the t-time static indistinguishable game is defined to be:*

$$\mathsf{Adv}^{\mathsf{t\text{-}SIND}}_{\mathcal{A}}(1^\lambda) = 2Pr[\mathsf{t\text{-}SIND}(\mathcal{A})] - 1$$

We say that Π is t-time computationally (resp. statistically) secure if for any PPT (resp. unbounded) adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{t\text{-}SIND}}_{\mathcal{A}}(1^\lambda)$ is negligible. And we say Π is fully (computationally/statistically) secure if Π is t-time (computationally/statistically) secure for any polynomial $t = poly(logN, \lambda)$.

If Π is an ORE scheme in the idealized model $\mathcal{M}$, we extend the security notions above by allowing $\mathcal{A}$ to make a polynomial number of queries to $\mathcal{M}$, and all probabilities are taken over the seed for $\mathcal{M}$.

3 Impossibility of information-theoretic ORE

In this section, we show that for information-theoretic ORE, full statistical security is impossible if the message space is super-polynomial. Note that this is

qualitatively tight, as [20] [9]shows how to construct information-theoretic ORE where the ciphertext size is polynomial in the size of the message space.

Note that our impossibility applies to schemes where the public/secret key are allowed to be arbitrarily (e.g., exponentially) large, the running time of $\mathsf{Enc}, \mathsf{Comp}$ are allowed to be arbitrary. However, the following restrictions must hold: 1) the size of ciphertexts must be polynomially bounded, 2) the security must hold for arbitrary adversaries (even for unbound adversary), 3) the adversary sees only a polynomial number of ciphertexts. Now, we prove our theorem.

Theorem 9. *In standard model, there does not exist a fully statistically secure ORE Π such that*

- *Π is partially correct;*
- *Π's message space is super-polynomial;*
- *Π has succinct ciphertexts.*

Roughly speaking, our proof strategy is: (1) prove the result in the simpler setting where we insist on *perfect* correctness, and then (2) show how to convert any partially correct information-theoretic ORE into a perfectly correct one.

3.1 Impossibility for perfect correct ORE

In this part, we consider the ORE scheme in the perfectly correct setting.

Theorem 10. *In standard model, there does not exist a statistically secure ORE Π such that*

- *Π is perfectly correct;*
- *Π's message space is super-polynomial;*
- *Π has succinct ciphertexts.*

Firstly, we give a brief description of our proof strategy. Let Π be an ORE scheme on message space $[t+1]$, where $t = \mathsf{poly}(\lambda)$, such that Π is perfectly correct and statistically secure. We immediately observe that Π is t-time secure, next we show, for any such an ORE, there exists an exponential lower bound on the size of the ciphertext space (roughly $O(2^{t/2})$), which means the size of ciphertext is at least $\mathsf{poly}(t)$. Based on that, it's trivial to note that, for any ORE with super-polynomial message space, the ciphertext size is at least $\mathsf{poly}(t)$(for arbitrary $t = \mathsf{poly}(\lambda)$). Then we set t to be sufficiently large to contradict the theorem statement.

The core technique we use is inspired by Erdös [15]. Roughly, for any Π with plaintext space $[t+1]$, we interpret its ciphertext space as a graph G_{t+1}, which has a similar structure to the graphs studied in [15].Then we sample a sequence of sub-graphs such that $G_{t+1} \supseteq G_{t-1} \supseteq \dots G_1$[10] in a specific way (based on our ORE). After that, we prove for any adjacent pair, we have $\mathbb{E}[\log|G_i|] \geq \mathbb{E}[\log|G_{i-2}|] + \log(1.6), \forall i \in \{t+1, t-1, \dots, 3\}$, which means $\mathbb{E}[\log|G_{t+1}|] \geq \lfloor \frac{t-1}{2} \rfloor \log 1.6$. More precisely:

[9] here we treat the PRFs and PRPs in [20] as real random functions and permutations, which achieving statistical security, rather than only computational security.

[10] here we assume t is even.

Lemma 11. *In standard model, let Π be a perfectly correct t-time secure ORE on message space $[t+1]$, then Π requires ciphertexts of size at least $\lfloor \frac{t-1}{2} \rfloor \log 1.6$*

Proof. This proof applies a similar spirit to a proof technique used by Erdös [15].

Let $\Pi_{t+1} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ be a perfect correct t-time secure ORE, with respect to message space $[t+1]$ and ciphertext space $\mathcal{C}$. We construct a new ORE Π^*_{t+1} as follows. The public key for Π_{t+1} defines a graph G_{t+1}, where the nodes of G_{t+1} represent the ciphertexts in $\mathcal{C}$. We set the edges for G_{t+1} as:

- If $\mathsf{Comp}(C_0, C_1) = \text{“}<\text{”}$, then there is a directed edge from C_0 to C_1
- Otherwise, we arbitrarily assign a single directed edge between the two nodes.

By perfect correctness of Π_{t+1}, we note that there is at most one directed edge between any two nodes, and if C_0 and C_1 are not simultaneously valid ciphertexts under the same secret key (we can view this as $C_0 = \mathsf{Enc}(\mathsf{sk}_0, i), C_1 = \mathsf{Enc}(\mathsf{sk}_1, j)$, and in such a case they are the ciphertexts encrypted under distinct secret keys), we set an arbitrary edge for these two nodes. Hence G_{t+1} is a “tournament” graph. Now we define $\Pi^*_{t+1} = (\mathsf{Gen}^*_{t+1}, \mathsf{Enc}^*_{t+1}, \mathsf{Comp}^*_{t+1})$:

- $\mathsf{Gen}^*()$ Runs $(pk, sk) \leftarrow \mathsf{Gen}()$, computes G_{t+1} as above, and outputs $\mathsf{pk}^* = (\mathsf{pk}, G_{t+1}), \mathsf{sk}^* = \mathsf{sk}$;
- $\mathsf{Enc}^*(\mathsf{sk}^*, m)$ It runs $C = \mathsf{Enc}(\mathsf{sk}^*, m)$, and outputs $C^* = C$;
- $\mathsf{Comp}^*(\mathsf{pk}^*, C^*_0, C^*_1)$ If $C^*_0 = C^*_1$, outputs “=”, else outputs “$<$” if there is directed edge from C^*_0 to C^*_1 in G_{t+1} , and “$>$” otherwise.

The only difference between Π_{t+1} and Π^*_{t+1} is adding G_{t+1} to the public key, which only affects the efficiency of Gen and Comp, while perfect correctness and t-time security are preserved.

Then, we sample the sub-graphs $G_{t-1} \supseteq \ldots \supseteq G_1$(assume t is even). For any $j \in \{2, 4, \ldots, t\}$, graph G_{t+1-j} is sampled as:

- Run $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}^*_{t+1}$, compute $C^i_L = \mathsf{Enc}(\mathsf{sk}^*, i), C^i_R = \mathsf{Enc}(\mathsf{sk}^*, t+1-i)$ for $i \in [j/2]$;
- Set G_{t+1-j} be the sub-graph of G_{t+1} consisting of all nodes v dominated by $\{C^1_L, \ldots, C^{j/2}_L\}$ (that is, there is an edge from C^i_L to v for all i) and dominate $\{C^1_R, \ldots, C^{j/2}_R\}$ (that is, there is an edge from v to C^i_R for all i)

Clearly, $|G_1| \geq 1$, therefore it's sufficient to prove that for $j \in \{2, 4 \ldots, t\}$,

$$\mathbb{E}(\log |G_{t+3-j}|) \geq \mathbb{E}(\log |G_{t+1-j}|) + \log 1.6$$

First, recall that Π^* is t-time secure, implying the distribution of the encryptions for $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$ are statistically close, over the probability $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}^*_{t+1}$, where,

$$\boldsymbol{M}_0 = (1, 2, \ldots, j/2, j/2+1, t+1-j/2, \ldots, t+1)$$
$$\boldsymbol{M}_1 = (1, 2, \ldots, j/2, t-j/2, t+1-j/2, \ldots, t+1)$$

Then, let f_L, f_R be the expected fraction of nodes in G_{t+3-j} that are dominated by $\mathsf{Enc}(j/2+1), \mathsf{Enc}(t-j/2)$, respectively. Due to security, we have

$$(\mathsf{pk}^*, \mathsf{Enc}(\boldsymbol{M}_0)) \overset{\mathsf{stat}}{\approx} (\mathsf{pk}^*, \mathsf{Enc}(\boldsymbol{M}_1)) \Rightarrow |f_L - f_R| \leq \mathsf{negl} \leq 1/4$$

Besides, G_{t+3-j} is also tournament, which indicates the expected fraction of nodes in G_{t+3-j} that dominate $\mathsf{Enc}(t-j/2)$ is

$$1 - f_R \leq 1 - f_L + 1/4$$

Moreover, G_{t+1-j} is the intersection of the nodes in G_{t+3-j} which dominate $\mathsf{Enc}(t-j/2)$ and which are dominated by $\mathsf{Enc}(j/2+1)$, the ratio $|G_{t+1-j}|/|G_{t+3-j}|$ is at most the minimum of:

- The fraction of nodes in G_{t+3-j} which dominate $\mathsf{Enc}(t-j/2)$
- The fraction of nodes in G_{t+3-j} dominated by $\mathsf{Enc}(j/2+1)$

Now, we can upper bound $\mathbb{E}[\log |G_{t+1-j}|]$ as:

$$\begin{aligned}
\mathbb{E}[\log |G_{t+1-j}|] &= \mathbb{E}[\log |G_{t+3-j}|] + \mathbb{E}[\log \frac{|G_{t+1-j}|}{|G_{t+3-j}|}] \\
&\leq \mathbb{E}[\log |G_{t+3-j}|] + \log \mathbb{E}[\frac{|G_{t+1-j}|}{|G_{t+3-j}|}] \qquad \text{Jensen's inequality} \\
&\leq \mathbb{E}[\log |G_{t+3-j}|] + \log \min(f_L, 1 - f_L + 1/4) \\
&\leq \mathbb{E}[\log |G_{t+3-j}|] + \log \frac{1+1/4}{2} = \mathbb{E}[\log |G_{t+3-j}|] - \log 1.6
\end{aligned}$$

For the last line, we used the fact that for any f_L, $\min(f_L, c - f_L) \leq \frac{c}{2}$. Putting everything together, we have

$$\mathbb{E}[\log |G_{t+1}|] \geq \mathbb{E}[\log |G_1|] + \lfloor \frac{t-1}{2} \rfloor \log 1.6$$

In addition, applying exactly the same technique, the theorem also holds when t is odd. □

Now, we complete the entire proof for Theorem 10. Suppose Π is an ORE such that: 1) Π is perfect correct and statistically secure; 2) Π's message space is $[N]$, where N is super-polynomial; 3) Π has succinct ciphertexts, which is bounded by $r = \mathsf{poly}(\lambda, \log N)$. Then, let $t = 4r$ (t is still polynomial here), we know that Π is t-time secure. According to Lemma 11, $r \geq \lfloor \frac{t-1}{2} \rfloor \cdot \log 1.6 > r$, a contradiction. □

3.2 Boosting to perfect correctness

To strengthen our result, we also consider ORE scheme that is only partially correct, and in this part, we show how to boost any partially correct scheme to a perfectly correct one.

Theorem 12. *If there exists partially correct and statistically secure ORE in the standard model that has succinct ciphertexts and super-polynomial message space, then statistically secure ORE in standard model with succinct ciphertexts and perfect correctness on the same message space exists.*

Proof. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ be an ORE in the standard model such that

1. Π is $\frac{1}{2} + \rho$ correct, where ρ is noticeable;
2. Π's message space is $[N]$, where N is super-polynomial;
3. Π^* has succinct ciphertexts, which is bounded by $r = \mathsf{poly}(\lambda, \log N)$

Then we construct a new ORE $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Comp}')$ that is statistically correct. More precisely, let $s = \frac{2}{\rho^2} \log N^2 \lambda$, we define Π' as

- $\mathsf{Gen}'(\rho, \log N, \lambda)$ runs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i=1}^s \leftarrow \mathsf{Gen}()$, and outputs $\mathsf{pk}' = (\mathsf{pk}_i)_{i=1}^s; \mathsf{sk}' = (\mathsf{sk}_i)_{i=1}^s$;
- $\mathsf{Enc}'(\mathsf{sk}', m)$ runs $C_i = \mathsf{Enc}(\mathsf{sk}_i, m), i \in [s]$ Outputs $\boldsymbol{C} = (C_1, \ldots, C_s)$;
- $\mathsf{Comp}'(\mathsf{pk}', \boldsymbol{C}_0, \boldsymbol{C_1})$ let $\boldsymbol{C}_0 = (C_1^0, \ldots, C_s^0), \boldsymbol{C}_1 = (C_1^1, \ldots, C_s^1)$, outputs the majority of $(\mathsf{Comp}(pk_i, C_i^0, C_i^1))_{i=1}^s$

We immediately observe that Π' also has succinct ciphertexts, and by hybrid argument, it's easy to have that Π' is statistically secure. Now, applying Chernoff Bound, we have

$$\Pr[\Pi' \text{ is correct}] \geq 1 - e^{-\frac{1}{1+2\rho} s\rho^2} \geq 1 - \frac{1}{N^2} e^{-\lambda}$$

We note Π' is statistically correct such that: within overwhelming probability over the choice of $(\mathsf{pk}', \mathsf{sk}')$, the comparison is correct for all message pairs. Then we construct the perfectly correct ORE $\Pi^* = (\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Comp}^*)$, same as Π' except we modify Gen^*: it draws $(\mathsf{pk}^*, \mathsf{sk}^*)$, conditioned on correctness holding for all message pairs. As $\Pi' \overset{\mathsf{stat}}{\approx} \Pi^*$, this only negligibly changes the distribution of keys, Π^* is also statistically secure. Notice that Gen^* is no longer efficient even if Gen was. Fortunately, our notion in standard model allows us to have inefficient Gen. Thus, statistically secure ORE in standard model with succinct ciphertexts and perfect correctness on the same message space exists.

□

Combing Theorem 10 and 12, we establish Theorem 9.

4 Impossibility of statistically secure ORE In idealized models

In this section, we begin our investigation of ORE in idealized models, where the algorithms of ORE have access to the model $\mathcal{M}$ ($\mathcal{M}$ is deterministic and computable). We give a unified strategy to help answer prove statements of the form:

For some particular idealized model $\mathcal{M}$, there does not exist randomized, partially

correct and statistically secure ORE that has succinct ciphertexts with super-poly message space

Roughly speaking, our strategy is consist of four steps:

- Convert a randomized, partially correct and statistically secure ORE in an idealized model into a deterministic, partially correct and statistically secure ORE in the same model;
- Compile the scheme to remove the oracle queries from the comparison procedures;
- Remove the model from ORE completely.
- Invoke Theorem 9 to finish the impossibility

In this section, we show that step 1 and 3 is achievable for any deterministic and computable model $\mathcal{M}$, and we note that when achieving step 3, it indicates the existence of partially correct and statistically secure ORE in standard model, which conflicts our result in Theorem 9. Hence the only step that depends on the exact model in question is step 2, removing the oracle query access from the comparison while still preserving the partial correctness and statistical security. In later sections, we will show how to do this for the random oracle model and generic group model.

Theorem 13. *If there exists a randomized partially correct and statistically secure ORE in idealized model $\mathcal{M}$ that has succinct ciphertexts and super-polynomial message space, then deterministic, partially correct and statistically secure ORE in the same model $\mathcal{M}$ with succinct ciphertexts on the same message space exists.*

Proof. ORE typically allows for randomized encryption. We may even allow for randomized comparison. However, we will show how to convert such a scheme into a deterministic one.

To handle a randomized comparison, we simply add a sequence of random coins to the secret key and every individual ciphertext. These random coins will be used for any run of Comp. While in the original scheme, each run of Comp uses independent randomness, here we use the same randomness every time. However, since the experiment defining correctness only considers a single run of Comp, the correctness probability is not affected by this change.

To handle a randomized encryption, we just generate the random coins r_m for every message m, and include r_m in the secret key. When encrypting a message m, encrypt using the random coins r_m. Notice that this blows up the secret key size. However, note that for this work we do not care about the size of the secret key; it can be exponential in size, and still our impossibility will hold. We note that another approach is to have r_m be the output of a PRF evaluated on m; suitable PRFs can be built from most interesting models, including the random oracle and generic group models we consider. This prevents the secret key length from exploding. However, this is unnecessary for our purposes.

Suppose $\Pi = (\mathsf{Gen}^{\mathcal{M}}, \mathsf{Enc}^{\mathcal{M}}, \mathsf{Comp}^{\mathcal{M}})$ be a randomized ORE where encryption and comparison procedures are both randomized, then we construct Π^* as:

- Gen^* runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$, samples $N+1$ randomness $(r, r_1, \ldots, r_N)$, and outputs $\mathsf{pk}^* = \mathsf{pk}, \mathsf{sk}^* = (\mathsf{sk}, r, r_1, \ldots, r_N)$;
- $\mathsf{Enc}^*(\mathsf{sk}^*, m)$ runs $C = \mathsf{Enc}^{\mathcal{M}}(\mathsf{sk}, m, r_m)$ and outputs $C^* = (C||r)$;
- $\mathsf{Comp}^*(\mathsf{pk}^*, C_0^*, C_1^*)$ outputs $\mathsf{Comp}^{\mathcal{M}}(\mathsf{pk}, C_0, C_1, r)$

We note that Π^* is a deterministic ORE now, both in encryption and comparison. Moreover, ignoring r for ciphertexts, as long as we do not encrypt the same message twice, the distribution of the ciphertext in Π^* is exactly the same as Π's. We note that the correctness is well preserved. In fact, according to the partial correctness definition, the randomness used in Comp is uniform just as in the original scheme.

For statistical security, we see that the adversary only additionally learns a random string (r, used for Comp) after it submits the message sequence, and the random string is independent of the message sequence, hence the adversary does not gain more information than in Π. Thus, statistical security is also preserved.

From now on, we treat ORE scheme as deterministic encryption and the message space is super-polynomial, unless otherwise specified.

Theorem 14. *If there exists partially correct and statistically secure ORE in idealized model that makes no query to $\mathcal{M}$ in comparison procedure and has succinct ciphertext, then partially correct and statistically secure ORE in standard model exists that has succinct ciphertexts.*

Proof. This proof is very straightforward. Since there is no access to $\mathcal{M}$ during the comparison procedure, there is no need for the idealized model to be public. Instead, we set $\mathcal{M}$ as part of the secret key and only the encrypter has access to it. Not giving the adversary access to $\mathcal{M}$ only helps security. Of course, in such a setting, the secret key is now exponentially large, and encryption is no longer efficient. However, our notion of ORE in standard model allows such large key and inefficiencies of encryption, which completes the proof.

The only remaining part is step 2, which is model-specific and non-trivial. We need to remove $\mathcal{M}$ from comparison procedures, while the input of Comp only includes the public key and ciphertext, and we cannot just absorb the model to the public key as we did in Theorem 14. Otherwise, the adversary would have the complete access to the oracle, indicating that it gains more information than it has in t-time statistical security game, and might break the game. Hence, we need to find ways to simulate the model while still preserving the statistical security. In the next two sections, we present our methods on two specific models: random oracle model and generic group model.

5 The Random Oracle Model

In this section, we finish the separation result in the case that $\mathcal{M}$ is a random oracle, which we denote by $\mathcal{O}$. Using the results of Sections 3 and 4, it remains to show that the random oracle model can be removed from the comparison

procedure of an ORE scheme. Our proof is inspired by [10], which shows how to remove random oracles from obfuscation schemes. However, for reason's outlined in the introduction, the technical details of our proof will be substantially different.

We first observe the following. Consider running $\mathsf{Comp}^{\mathcal{O}}(C_0, C_1)$ where C_0, C_1 encrypt m_0, m_1 respectively. Consider an oracle query x made by Comp. If x was not a query made during encryption $(\mathsf{Enc}^{\mathcal{O}}(m_0), \mathsf{Enc}^{\mathcal{O}}(m_1))$, then we claim Comp must output the right answer, even if it is given the incorrect query response. Indeed, for any possible response y', there is an oracle $\mathcal{O}'$ that is consistent with $\mathcal{O}$ on the points queried during encryption of m_0, m_1, but where $\mathcal{O}'(x) = y'$. Therefore, any potentially incorrect query answer can be "explained" by an oracle $\mathcal{O}'$, and correctness of the scheme says that Comp must still output the right value in this case.

For a particular run of Comp on encryptions of m_0, m_1, we therefore call the oracle queries made during encryption "sensitive" queries. Comp only needs access to $\mathcal{O}$ on sensitive queries; for all others, it can answer randomly. The difficulty, then, is (1) allowing Comp to figure out the sensitive queries, and (2) giving it the right oracle answers in this case.

For simplicity, consider two extremes. On the one end, suppose none of Comp's queries are ever sensitive. In this case, Comp can just ignore its oracle entirely, simulating the responses with random answers. In this case, we are already done. In the other extreme, suppose all of Comp's queries are always sensitive. In this case, if the adversary sees ℓ ciphertexts, she expects to make at least $\Omega(\ell^2)$ oracle queries on sensitive queries. However, there are only $q\ell$ possible query values, where q is the number of queries made during each encryption. Therefore, heuristically, we may expect to eventually pick of *all* of the sensitive queries made during encryption by setting ℓ large enough (namely, bigger than q). Even so, security must hold. Therefore, we can construct a modified scheme where Enc simply outputs all the queries it makes and the corresponding answers along with the ciphertext. Then all the sensitive queries Comp needs are provided as input, and it does not need to make any oracle queries.

To formalize the above sketch, we must show how to handle cases between the two extremes, where some of Comp's queries are sensitive, and others are not, and we cannot necessarily tell which is the case. Moreover, we need to deal with the fact that we may not actually get all of the sensitive queries if there are sufficiently many collisions. In this case, handing out all of the queries made during encryption could actually hurt security (for example, if a query is made on the message itself). Nonetheless, we now prove the following theorem:

Theorem 15. *If there exists partially correct and statistically secure ORE in random oracle model that has succinct ciphertexts, then there exists partially correct and statistically secure ORE with succinct ciphertexts such that the comparison procedures makes no queries to the random oracle.*

Proof. Let $\Pi^0 = (\mathsf{Gen}_0, \mathsf{Enc}_0^{\mathcal{O}}, \mathsf{Comp}_0^{\mathcal{O}})$ be a statistically secure ORE in the random oracle model with plaintext space $[N]$. Here, we assume Gen_0 makes no

queries to $\mathcal{O}$. This is actually without loss of generality: since $\mathcal{O}$ is a deterministic oracle, we can always treat sk as the random coins inputted to Gen_0, and run Gen_0 every time we encrypt a message.

For convenience, we denote $\Pr[\Pi^0]$ as the lower bound on the correctness probability:

$$\Pr[\Pi^0] = \min_{m_0,m_1} \Pr[\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}, C_0, C_1) = \mathsf{Comp}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_0(); C_b \leftarrow \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}, m_b)]$$

We assume that $\mathsf{Comp}_0(\mathsf{pk}_0, C_0, C_1)$ does not query the same point twice; since $\mathcal{O}$ is deterministic, Comp_0 can always store a table of query/response pairs already seen, and use this table to answer subsequent queries on the same point. Here we specify some parameters:

1. $\Pr[\Pi^0] \geq \frac{1}{2} + 2\rho$, where ρ is noticeable; $q, u = \mathsf{poly}(\lambda)$ by query efficiency; $s := \frac{110u^4 \cdot q^2}{\rho^3}$; $s_i := \frac{110u^3 \cdot q^2 \cdot i}{\rho^3}, i \in [u]$;
2. $\mathsf{Enc}_0^{\mathcal{O}}$ makes q queries to the oracle $\mathcal{O}$. Let $Q_{\mathsf{sk},m}$ be the set of query-answer pairs made when encrypting m under key sk. Notice that the set $Q_{\mathsf{sk},m}$ is fully determined by sk and m since Enc and $\mathcal{O}$ are deterministic.
3. $\mathsf{Comp}_0^{\mathcal{O}}$ makes u queries to the oracle $\mathcal{O}$. Let S_{pk,m_0,m_1} be the set of query-answer pairs made when comparing the encryptions of (m_0, m_1) under key pk. Again, S_{pk,m_0,m_1} is fully determined by $\mathsf{pk}, \mathsf{sk}, m_0, m_1$.
4. $D := [s] \cup [N - s + 1, N]$; $D_i := [s_i] \cup [N - s_i + 1, N], i \in [u]$
5. $T_i = [i] \cup [N - i + 1, N], i \in [N]$;

Next we construct a new ORE $\Pi^* = (\mathsf{Gen}, \mathsf{Enc}^{\mathcal{O}}, \mathsf{Comp})$ with plaintext space $[s + 1, N - s]$ as:

- $\mathsf{Gen}()$ runs $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0()$, computes $C_i = \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}_0, i), i \in D$ and outputs $\mathsf{pk} = \mathsf{pk}_0, \mathsf{sk} = (\mathsf{sk}_0, \{C_i\}_{i \in D})$;
- $\mathsf{Enc}^{\mathcal{O}}(\mathsf{sk}, m)$ runs $C \leftarrow \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}_0, m)$. Then it runs $\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}_0, C_i, C)$ for all $i \in D$, recording all query-answer pairs $S_{\mathsf{pk},m} = \cup_{i \in D} S_{\mathsf{pk},m,i}$. Then it outputs $C^* = (C, S_{\mathsf{pk},m})$;
- $\mathsf{Comp}(\mathsf{pk}, C_0^*, C_1^*)$: let $C_0^* = (C_0, S_0), C_1^* = (C_1, S_1)$. Run $\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}_0, C_0, C_1)$, except that when querying the oracle with input x, do the following:
  1. If there is a pair (x, y) in $S_0 \cup S_1$, Comp responds to the query with y;
  2. Otherwise, returns a random string.

We note that in the comparison procedure of Π^*, we remove the oracle access, so it remains to show that Π^* is statistically secure and partially correct.

Lemma 16. *If Π^0 is $t+2s$ statically secure, then Π^* is t-time statically secure.*

Proof. The entire view of the the adversary $\mathcal{A}$ in the t-time experiment for Π^* can be simulated by a $t + 2s$-time adversary $\mathcal{B}$ for Π^0: the lists of messages are those produced by $\mathcal{A}$, plus all the messages in D. Then, the lists S associated with ciphertext C can be constructed by comparing C to each of the C_i for $i \in D$. □

It's obvious that Lemma 16 holds for any $t = \mathsf{poly}(\log N, \lambda)$, which means Π^* is statistically secure. And what's more interesting is that Π^*'s partial correctness. In the following, we prove that Π^* also preserves partial correctness, though there is some loss in the concrete correctness parameter.

Lemma 17. $\Pr[\Pi^*] \geq \frac{1}{2} + \rho$

Proof. We establish our proof by hybrid argument, and define u alternative ORE schemes $\Pi_j = (\mathsf{Gen}_j, \mathsf{Enc}_j^{\mathcal{O}}, \mathsf{Comp}_j^{\mathcal{O}}), j \in [u]$ on message space $[s_j + 1, N - s_j]$:

- $\mathsf{Gen}_j()$ runs $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0()$, computes $C_i = \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}_0, i)$ for $i \in D_j$ and outputs $\mathsf{pk}_j = \mathsf{pk}_0, \mathsf{sk}_j = (\mathsf{sk}_0, \{C_i\}_{i \in D_j})$;
- $\mathsf{Enc}_j^{\mathcal{O}}(\mathsf{sk}_j, m)$ runs $C \leftarrow \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}_0, m)$ and $\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}_0, C_i, C)$ for $i \in D_j$, records all query-answer pairs $S_{\mathsf{pk},m} = \cup_{i \in D_j} S_{pk,m,i}$ and outputs $C^* = (C, S_{\mathsf{pk},m})$;
- $\mathsf{Comp}_j^{\mathcal{O}}(\mathsf{pk}_j, C_0^*, C_1^*)$: let $C_0^* = (C_0, S_0), C_1^* = (C_1, S_1)$. It runs $\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}_j, C_0, C_1)$, except that when querying $\mathcal{O}$ with input x, it does the following:
  1. If x is one of the first $u - j$ queries, make a query to $\mathcal{O}$ as usual.
  2. If x is one of the final j queries and there is a pair $(x, y) \in S_0 \cup S_1$, then respond with y.
  3. Otherwise, returns a random string.

We observe that $\Pi_u = \Pi^*$, hence it suffices to prove the following lemma,

Lemma 18.

$$\Pr[\Pi_j] \geq \Pr[\Pi_{j-1}] - \frac{\rho}{u}, \forall j \in [u]$$

We here only prove the case $j = 1$, the rest can be handled analogously. Specifically, we show $\Pr[\Pi_1] \geq \frac{1}{2} + 2\rho - \frac{\rho}{u}$.

According to the definition, we see that Comp_1 works the same as Comp_0, except for the final query x to $\mathcal{O}$ in which we use the list of oracle outputs provided with the ciphertext to answer the oracle query. We prove that the response made by Π_1 for x does not significantly harm the ability of Comp_1 to output the correct answer. To do so, we introduce yet another sequence of s_1 ORE schemes $\Pi_{1,j}, j \in [s_1]$ on message space $[j + 1, N - j]$. The only difference between $\Pi_{1,j}$ and Π_1 is the number of test ciphertexts that are generated.

- $\mathsf{Gen}_{1,j}()$ runs $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0()$, computes $C_i = \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}_0, i)$ for $i \in T_j$ and outputs $\mathsf{pk}_{1,j} = \mathsf{pk}_0, \mathsf{sk} = (\mathsf{sk}_0, \{C_i\}_{i \in T_j})$;
- $\mathsf{Enc}_{1,j}^{\mathcal{O}}(\mathsf{sk}_{1,j}, m)$ runs $C \leftarrow \mathsf{Enc}_0^{\mathcal{O}}(\mathsf{sk}_0, m)$ and $\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}_0, C_i, C)$ for $i \in T_j$, records all query-answer pairs $S_{\mathsf{pk},m}^{(j)} = \cup_i S_{\mathsf{pk},m,i}$ and outputs $C^* = (C, S_{\mathsf{pk},m}^{(j)})$;
- $\mathsf{Comp}_{1,j}^{\mathcal{O}}(\mathsf{pk}_{1,j}, C_0^*, C_1^*)$: let $C_0^* = (C_0, S_0), C_1^* = (C_1, S_1)$. It runs $\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}_{1,j}, C_0, C_1)$, except that when querying $\mathcal{O}$ with input x, it does the following:
  1. If x is one of the first $u - 1$ queries, make a query to $\mathcal{O}$ as usual.
  2. If x is the final query and there is a pair $(x, y) \in S_0 \cup S_1$, then respond with y.

3. Otherwise, returns a random string.

We note that $\Pi_1 = \Pi_{1,s_1}$. We now claim that increasing j must improve the correctness of the scheme:

Claim. If $\Pr[\Pi_{1,j}] < \frac{1}{2} + 2\rho - \frac{\rho}{u}$, then $\Pr[\Pi_{1,j+1}] \geq \Pr[\Pi_{1,j}] + \frac{\rho^3}{110u^3 \cdot q^2}$

Notice that this means as j increases, $\Pr[\Pi_{1,j}]$ must increase by increments of at least $\frac{1}{s_1} = \frac{\rho^3}{110u^3 \cdot q^2}$ until $\Pr[\Pi_{1,j}] \geq \frac{1}{2} + 2\rho - \frac{\rho}{u}$. Therefore, by setting $j = s_1$, we get that $\Pr[\Pi_1] = \Pr[\Pi_{1,j}] \geq \frac{1}{2} + 2\rho - \frac{\rho}{u}$ as desired. It remains to prove the claim.

Assuming $\Pr[\Pi_{1,j}] < \frac{1}{2} + 2\rho - \frac{\rho}{u}$, there are two messages m_0^*, m_1^* minimizing the correctness probability; that is, the comparison procedure on encryptions of m_0^*, m_1^* outputs the correct answer with probability less than $\frac{1}{2} + 2\rho - \frac{\rho}{u}$. Since comparison succeeding is a detectable event, we can invoke the security of ORE to conclude that, for *any* m_0, m_1, comparison must output the correct answer with probability at most $\frac{1}{2} + 2\rho - \frac{\rho}{u} + \mathsf{negl} < \frac{1}{2} + 2\rho - \frac{2\rho}{3u}$.

Fix two messages $m_0, m_1 \in [s_1 + 1, N - s_1]$. We denote $S^{(j)} := S^{(j)}_{\mathsf{pk},m_0} \cup S^{(j)}_{\mathsf{pk},m_1}; Q := Q_{\mathsf{sk},m_0} \cup Q_{\mathsf{sk},m_1}$. Let x be the final query made when comparing the encryptions of m_0, m_1.

Define the event Bad_j where the following happens:

- $x \in Q \setminus S^{(j)}$, so that x was queried during the encryption of m_0 or m_1, but not during any of the comparisons to the test ciphertexts.
- $\mathsf{Comp}_0^{\mathcal{O}}$ outputs the correct answer on encryptions of m_0, m_1.
- $\mathsf{Comp}_{1,j}^{\mathcal{O}}$ outputs the incorrect answer on encryptions of m_0, m_1.

We consider four cases:

- $x \in S^{(j)}$ In this case, Π_1 answers the same as $\Pi_{1,j}$ since it has access to $\mathcal{O}(x)$
- $x \notin Q$ Then the ciphertexts components C_0, C_1 under Π_0 are independent of $\mathcal{O}(x)$, meaning that during the correctness experiment, $\mathcal{O}(x)$ in Π_0 is a random string. Hence Π_1 answers the query with the correct distribution.
- $x \in Q \setminus S^{(j)}$, but Bad_j does not occur. Here, we must have that Comp_0 either produced the incorrect answer, or $\mathsf{Comp}_{1,j}$ produced the correct answer.
- Bad_j occurs In this case, C_0, C_1 will depend on $\mathcal{O}(x)$, while $\Pi_{1,j}$ cannot find it in $S^{(j)}$. Hence, $\Pi_{1,j}$ will answer randomly, but Comp may expect an answer correlated with C_0, C_1. Moreover, we know that by answering randomly, $\mathsf{Comp}_{1,j}$ goes from outputting the correct answer to the incorrect answer.

We note in the first three cases above, the expected correctness probability does not decrease relative to $\Pi_{1,j}$. Indeed, in the first and third cases, $\Pi_{1,j}$ is at least as correct as Π_0, and in the second case, $\Pi_{1,j}$ in expectation has the same correctness as Π_0. Only in the final case might answering randomly decrease the probability of correctness. Therefore, since comparison in $\Pi_{1,j}$ outputs the

correct answer with probability less than $\frac{1}{2} + 2\rho - \frac{2\rho}{3u}$, we must have $\Pr[\mathsf{Bad}_j] > \frac{2\rho}{3u}$.

We consider two sub-events of Bad_j, denoted $\mathsf{Bad}_j^{(b)}$, corresponding to $x \in Q_{\mathsf{sk},m_b}/S$. Notice that $\Pr[\mathsf{Bad}_j] \leq \Pr[\mathsf{Bad}_j^{(0)}] + \Pr[\mathsf{Bad}_j^{(1)}]$. By our assumption above, we have $\max\{\Pr[\mathsf{Bad}_j^{(0)}], \Pr[\mathsf{Bad}_j^{(1)}]\} > \frac{\rho}{3u}$. We will assume that $\Pr[\mathsf{Bad}_j^{(0)}] > \frac{\rho}{3u}$, the other case handled analogously

Next we split the message space into two parts: $[j+1, \frac{N}{2}]$ and $[\frac{N}{2}+1, N-j]$, and sample $w \leftarrow [j+1, \frac{N}{2}]$ and $z_1, \ldots, z_\ell \leftarrow [\frac{N}{2}+1, N-j]$, where $\ell = \frac{6u \cdot q}{\rho}$. Let t_i be the indicator as:

$$t_i = \begin{cases} 1 \text{ if } \mathsf{Bad}_j^{(0)} \text{ occurs for message pair } (w, z_i) \\ 0 \text{ Otherwise} \end{cases}$$

and T be the event that $\sum_{i=1}^{\ell} t_i > q$, we must have that:

$$\Pr[T] \cdot \ell + q \cdot (1 - \Pr[T]) \geq \mathbb{E}(\sum_{i=1}^{\ell} t_i) > 2q \Rightarrow \Pr[T] > \frac{\rho}{6u}$$

as $\Pr[t_i = 1] > \frac{\rho}{3u}$, which refers $\mathbb{E}\left[\sum_{i=1}^{\ell} t_i\right] > \ell \cdot \frac{\rho}{3u} > 2q$.

For three messages m_0, m_1, m_2, $m_0 < m_1 < m_2$, we define the event $\mathsf{Collision}$ as the following: the final queries x_1, x_2 when comparing encryptions of m_0 to m_1 and respectively m_0 to m_2 satisfy: (1) $\mathsf{Bad}_j^{(0)}$ occurs simultaneously for both (m_0, m_1) and (m_0, m_2), and (2) $x_1 = x_2$.

We observe that if T occurs, there are at least $q+1$ index such that $t_i = 1$. Moreover, in $\mathsf{Enc}_{1,j}^{\mathcal{O}}(w)$, there are at most q distinct queries. This means there is some $z_{i_1} < z_{i_2}$ such that $\mathsf{Bad}_j^{(0)}$ occurs for both (w, z_{i_1}) and (w, z_{i_2}) and moreover the final query in both comparisons is identical. This in particular means that $\mathsf{Collision}$ happens for (w, z_{i_1}, z_{i_2}).

Now we bound the probability of $\mathsf{Collision}$ for a random message w in $[j+1, \frac{N}{2}]$ and random distinct z_1^*, z_2^* in $[\frac{N}{2}+1, N-j]$. One way to sample random w, z_1^*, z_2^* is to sample w at random in $[j+1, \frac{N}{2}]$, and sample ℓ random distinct z_i in $[\frac{N}{2}+1, N-j]$. Then we choose two random indices i_1, i_2, and set $z_b^* = z_{i_b}$. The above analysis shows that with probability at least $\rho/6u$, there some $\mathsf{Collision}$ among the z_i. Since z_b^* are chosen as a random pair from this set, there is a collision in z_1^*, z_2^* with probability at least

$$\Pr[\text{ } \mathsf{Collision} \text{ for random } (w, z_1^*, z_2^*)]\} \geq \frac{1}{\binom{\ell}{2}} \cdot \Pr[T] > \frac{\rho^3}{108u^3 \cdot q^2}$$

Now, we would like to use security of ORE to show that $\mathsf{Collision}$ happens for arbitrary fixed triples m_0, m_1, m_2. Unfortunately, $\mathsf{Collision}$ is not necessarily detectable by an adversary, since an adversary does not know Q. Instead, we define a slightly different event $\mathsf{Collision}'$. $\mathsf{Collision}'$ is the same as $\mathsf{Collision}$ except that

it removes the requirement that the common query x is in Q for either w, z_1^* or w, z_2^*. Since $\mathsf{Collision}$ implies $\mathsf{Colision}'$, we must have that $\mathsf{Collision}'$ happens with probability at least $\frac{\rho^3}{108u^3 \cdot q^2}$ for a random w, z_1^*, z_2^*.

Now, $\mathsf{Collision}'$ *is* an event that can be detected by an adversary, thus by statistical security, we have that for *arbitrary* $(m_0, m_1, m_2) \in [j+1, N-j]$,

$$\Pr[\ \mathsf{Collision}' \text{ for } (m_0, m_1, m_2)] \geq \frac{\rho^3}{108u^3 \cdot q^2} - \mathsf{negl} > \frac{\rho^3}{110u^3 \cdot q^2}$$

Specifically, let $m_2 = N - j$, we see that for any $(m_0, m_1) \in [j+2, N-j-1]$, if we move to $\Pi_{1,j+1}$, m_2 is included in the test queries for the scheme. Notice that $\mathsf{Collision}'$ means that in $\Pi_{1,j}$, comparing m_0, m_1 would have been incorrect (since the final query is answered randomly), but in $\Pi_{1,j+1}$ comparing m_0, m_1 would be correct due to the additional queries provided from comparing m_0, m_2 (since comparing m_0, m_2 would add the missing query x to the list of queries included in the encryption of m_0). Thus:

$$\Pr[\Pi_{1,j+1}] \geq \Pr[\Pi_{1,j}] + \frac{\rho^3}{110u^3 \cdot q^2} \Rightarrow \Pr[\Pi_1] \geq \ \Pr\ [\Pi_0] - \frac{\rho}{u}$$

Now we have shown that $\Pr[\Pi_1] \geq \ \Pr\ [\Pi_0] - \frac{\rho}{u}$. This handles the case of Π_1. However, note that at this point, what use to be the second-to-last query is now the last query (since the last query is no longer made). Therefore, we can apply the exact same techniques as above to handle the general case of Π_j, giving

$$\Pr[\Pi_{j+1}] \geq \ \Pr\ [\Pi_j] - \frac{\rho}{u}$$

Combing together, we get

$$\Pr[\Pi^*] \geq \frac{1}{2} + \rho$$

which completes the entire proof. □

6 The Generic Group Model

In this section, we finish the separation result in generic group model, which we denote by $\mathcal{G}$. It remains to show that the generic group oracle model can be removed from the comparison procedure of any ORE scheme. Our strategy is inspired by [24], which shows how to remove constant graded encoding from obfuscation schemes. Before we illustrate the main idea of our proof, we recall a simple variant of the generic group model, which is equivalent to the usual generic group model [26]:

Definition 19. *(Variant Generic Group Model) Let $(G, \odot)$ be any group of size N and let S be any set of size at least N. The generic group oracle $\mathcal{G} : G \mapsto S$. At first an injective random function $\sigma : G \mapsto S$ is chosen, and two type of queries are answered as:*

- ***Type 1: Labeling queries.*** *Given $g \in G$, oracle returns handle $h = \sigma(g)$;*
- ***Type 2: Zero-test queries.*** *Given $\boldsymbol{h} = (h_1, \ldots h_n) \in S$, a vector $\boldsymbol{v} = (v_0, \ldots, v_n)$ of integers, oracle returns a single bit: 0 if there exists $g_1, \ldots, g_n \in G$ such that $h_i = \sigma(g_i)$ and $v_0 + \odot_j v_j g_j = 0$; 1 otherwise.*

WLOG, we can assume that the ORE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}^{\mathcal{G}}, \mathsf{Comp}^{\mathcal{G}})$ satisfies the following:

- Gen makes no queries to $\mathcal{G}$.
- Enc has the access of both labeling and zero-test query, while Comp only makes zero-test queries. This is because Comp gains no advantage by making labeling queries; it can always keep track of any group element it would have made a labeling query on, and adjust the v_0 term in a zero-test query to compensate.
- Let $\boldsymbol{h}_m$ be the vector of handles returned by the labeling queries during the encryption of m. We will assume the comparison procedure, when comparing encryptions of m_0, m_1, only makes zero-test queries using handles derived during the encryption. In other words, it will always have the form $(\boldsymbol{h}_{m_0}, \boldsymbol{h}_{m_1}, \boldsymbol{v})$. We can assume this as $\mathsf{Comp}'s$ view only depends on those labels; if it queried the zero-test on other labels, then it would somehow be guessing labels it never saw before, which is statistically unlikely.
- For any m, $|\boldsymbol{h}_m| = |\boldsymbol{g}_m| = q$, where $q = \mathsf{poly}(\lambda)$ is a fixed integer.

Then we present a brief description of our strategy. Similar to our random oracle proof, given an ORE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}^{\mathcal{G}}, \mathsf{Comp}^{\mathcal{G}})$ on message space $[N]$ with partial correctness $\frac{1}{2} + 2\rho$, we construct an new ORE $\Pi^* = (\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Comp}^*)$ on message space $[s+1, N-d](s, d = \mathsf{poly}(\log N, \lambda))$ with correctness $\frac{1}{2} + \rho$, where we remove $\mathcal{G}$ from Comp^*. In the key generation procedure, Π^* additionally outputs the encryption of $i, i \in [s] \cup [N-d+1, N]$.

Next, $\mathsf{Enc}(k, m)$ runs $\mathsf{Enc}(k, m), \mathsf{Comp}(\mathsf{Enc}(k, m), \mathsf{Enc}(k, i)), \mathsf{Comp}(\mathsf{Enc}(k, i), \mathsf{Enc}(k, j)), i, j \in [s] \cup [N-d+1, N]$. It collects all of the zero test queries and responses produced during the comparisons. It deletes all queries that outputted 1. It is left with a set of linear constraints on the $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s, \boldsymbol{g}_m, \boldsymbol{g}_{N-d+1}, \ldots, \boldsymbol{g}_N$ terms. It therefore produces a set S_m of linearly independent constrains over these variables. It finally outputs $(\mathsf{Enc}(m), S_m)$.

Meanwhile, $\mathsf{Comp}^*(C_{m_0}, C_{m_1})$, runs Comp on the two Π-ciphertexts contained in C_{m_0}, C_{m_1}. Whenever $\mathsf{Comp}_{1,j}$ tries to make a zero-test query, $\mathsf{Comp}^*_{1,j}$ intercepts, and answers using the sets S_{m_0}, S_{m_1} as follows. It determines if the zero test query is linearly dependent on the constraints in $S_{m_0} \cup S_{m_1}$. If so, it knows that the answer to the zero test query is 0. Otherwise, it guesses that the zero test query answer is non-zero.

We claim that this modified comparison procedure answers all zero test queries right except with small probability. Roughly, the idea is that Comp only needs to learn the constraint space when restricted to $\boldsymbol{g}_{m_0}, \boldsymbol{g}_{m_1}$, and does so using the constraints it obtains through the test ciphertexts. Notice that the number of constraints we obtain grows quadratically with the number of test ciphertexts computed, while the dimension of the space of constraints only grows

linearly. Therefore, by using enough test elements, we "should" exhaust all linear constraints and recover the entire constraints space. Indeed, we show that with sufficiently large s, d, $S_{m_0} \cup S_{m_1}$ has either recovered the full basis of the space (which allows one to correctly answer all remaining zero-test queries), or it's very unlikely that a new constraint appears, which in turn means that Comp^* simulates the oracle itself properly except with a small probability. We now prove the following theorem:

Theorem 20. *If there exists partially correct and statistically secure ORE in generic group model that has succinct ciphertexts, then partially correct and statistically secure ORE with succinct ciphertexts that makes no query to generic group oracle in comparison procedures exists.*

Due to the space limit, we skip the rigorous proof here, and refer the whole proof in our full version [28].

Acknowledgments

Mark Zhandry is supported by NSF. Cong Zhang is partially supported by DARPA and SSC Pacific under contract N66001-15-C-4070. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF, DARPA or SSC Pacific.

References

1. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 563–574. ACM, 2004.
2. A. Arasu, S. Blanas, K. Eguro, R. Kaushik, D. Kossmann, R. Ramamurthy, and R. Venkatesan. Orthogonal security with cipherbase. In *6th Biennial Conference on Innovative Data Systems Research (CIDR'13)*, January 2013.
3. A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 224–241. Springer, 2009.
4. A. Boldyreva, N. Chenette, and A. O'Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *Annual Cryptology Conference*, pages 578–595. Springer, 2011.
5. D. Boneh, K. Lewi, M. Raykova, A. Sahai, M. Zhandry, and J. Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In *Proceedings of EUROCRYPT 2015*, 2015.
6. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 573–592, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
7. D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

8. Z. Brakerski and G. N. Rothblum. *Virtual Black-Box Obfuscation for All Circuits via Generic Graded Encoding*, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
9. Z. Brakerski and G. Segev. Function-private functional encryption in the private-key setting. In *Theory of Cryptography Conference*, pages 306–324. Springer, 2015.
10. R. Canetti, Y. T. Kalai, and O. Paneth. On obfuscation with random oracles. In *Theory of Cryptography Conference*, pages 456–467. Springer, 2015.
11. D. Cash, F.-H. Liu, A. O'Neill, and C. Zhang. Reducing the leakage in practical order-revealing encryption. Technical report, Cryptology ePrint Archive, Report 2016/661, 2016.
12. N. Chenette, K. Lewi, S. A. Weis, and D. J. Wu. Practical order-revealing encryption with limited leakage. In *International Conference on Fast Software Encryption*, pages 474–493. Springer, 2016.
13. J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology–CRYPTO 2013*, pages 476–493. Springer, 2013.
14. F. B. Durak, T. M. DuBuisson, and D. Cash. What else is revealed by order-revealing encryption? In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1155–1166. ACM, 2016.
15. P. Erdős and V. Sós. On a problem of graph theory.
16. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–17. Springer, 2013.
17. C. Gentry, A. Lewko, and B. Waters. Witness encryption from instance independent assumptions. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 426–443, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
18. P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart. Leakage-abuse attacks against order-revealing encryption. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 655–672. IEEE, 2017.
19. M. Joye and A. Passelègue. Function-revealing encryption. 2016.
20. K. Lewi and D. J. Wu. Order-revealing encryption: New constructions, applications, and lower bounds. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1167–1178, New York, NY, USA, 2016. ACM.
21. W. Lu, A. L. Varna, and M. Wu. Security analysis for privacy preserving search of multimedia. In *2010 IEEE International Conference on Image Processing*, pages 2093–2096, Sept 2010.
22. M. Mahmoody, A. Mohammed, and S. Nematihaji. On the impossibility of virtual black-box obfuscation in idealized models. In *Theory of Cryptography Conference*, pages 18–48. Springer, 2016.
23. M. Naveed, S. Kamara, and C. V. Wright. Inference attacks on property-preserving encrypted databases. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 644–655. ACM, 2015.
24. R. Pass and A. Shelat. Impossibility of vbb obfuscation with ideal constant-degree graded encodings. In *Theory of Cryptography Conference*, pages 3–17. Springer, 2016.
25. R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles 2011, SOSP 2011, Cascais, Portugal, October 23-26, 2011*, pages 85–100, 2011.

26. V. Shoup. Lower bounds for discrete logarithms and related problems. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 256–266. Springer, 1997.
27. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou. Secure ranked keyword search over encrypted cloud data. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 253–262, June 2010.
28. M. Zhandry and C. Zhang. Impossibility of order-revealing encryption in idealized models. Cryptology ePrint Archive, Report 2017/1001, 2017. `https://eprint.iacr.org/2017/1001`.