# ORDER-REVEALING ENCRYPTION: NEW CONSTRUCTIONS AND BARRIERS

**by**

**CONG ZHANG**

**A dissertation submitted to the**

**School of Graduate Studies**

**Rutgers, The State University of New Jersey**

**In partial fulfillment of the requirements**

**For the degree of**

**Doctor of Philosophy**

**Graduate Program in Computer Science**

**Written under the direction of**

**David Cash and Rebecca Wright**

**And approved by**

_____

_____

_____

_____

**New Brunswick, New Jersey**

**October, 2020**

## ABSTRACT OF THE DISSERTATION

# Order-Revealing Encryption: New Constructions and Barriers

**By Cong Zhang**

**Dissertation Director:**
**David Cash and Rebecca Wright**

An order-revealing encryption (ORE) is a symmetric encryption scheme that gives a public procedure by which two ciphertexts can be compared to reveal the order of their underlying plaintexts. ORE is a very popular primitive for outsourcing databases and has seen deployments in products and usage in applied research, as it allows for efficiently performing range queries over encrypted data. However, a series of works, starting with Naveed *et al.* (CCS 2015), have shown that when the adversary has a good estimate of the distribution of the data, ORE provides little protection.

In this dissertation, we present our works on order-revealing encryption, which include novel security notions, new constructions, and barriers. First we consider the best-possible security notion for ORE (ideal ORE), which means that, given the ciphertexts, *only* the order is revealed — anything else, such as the distance between plaintexts, is hidden. Despite the fact that this notion provides the best security for ORE, the only known constructions of ideal ORE are based on cryptographic multilinear maps and are currently too impractical for real-world applications. In this work,

we give evidence that building ideal ORE from weaker tools is hard. Essentially, we show black-box separations between ideal ORE and most symmetric-key primitives, as well as public key encryption and anything else implied by generic group model in a black-box way. This result tells us that 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 thus it suggests that any ORE achieving ideal security will likely be at least somewhat inefficient.

Then we propose a new meaningful security notion— *parameter-hiding*. In our definition, we consider the case that the database entries are drawn identically and independently from the distribution of known shape, but for which the mean and variance are not (and thus the attacks of Naveed et al. do not apply). We say an ORE is parameter-hiding, if for any probabilistic and polynomial-time adversary, given any sequence (polynomial-size) of ciphertexts, the mean and variance of the message distribution are hidden. Based on this notion, we build the corresponding construction of ORE that satisfying it from bi-linear maps.

Next, we study a particular case of ORE, which is called order-preserving encryption. OPE schemes are the subset of ORE schemes for which the ciphertexts themselves are numerical values which can be compared naturally. For OPE, we study its ciphertext length under the security notion proposed by Chenette et al. for OPE (FSE 2016); their notion says that the comparison of two ciphertexts should only leak the index of the most significant bit on which they differ (MSDB-secure). In their work, they propose two constructions, both ORE and OPE; the ORE scheme has very short ciphertexts that only expand the plaintext by a factor $\approx 1.58$, while the ciphertext-size of the OPE scheme expand the plaintext by a security-parameter factor. We give evidence that this gap between ORE and OPE is inherent, by proving that *any* OPE meeting the information-theoretic version of their security definition (for instance, in the random oracle model) must have the ciphertext length close to that of their constructions.

# Acknowledgements

First and foremost, I would like to express my endless gratitude and appreciation to my academic advisors Prof. David Cash and Prof. Rebecca Wright. David gave his extraordinary guidance, support, and encouragement throughout the first three years of my PhD study. In particular, he guided me into the research on ORE, where we have two joint works together and helped me how to present our work properly. When he left Rutgers, Rebecca became my advisor. She supported me for the rest three years and helped me to be a good researcher.

Second, I would like to express my deepest gratitude to Prof. Mark Zhandry, who lifted my career to a better stage. He enlightened me with beautiful theoretical researches and guide me how to identify and resolve problems myself. We already have four papers together and it is still counting on.

I am also extremely grateful to my collaborators, Prof. Adam O'Neill and Prof. Feng-Hao Liu and my committee members, Prof. Eric Allender, Prof. Shiqing Ma, and Prof. Qiang Tang. This dissertation greatly benefits from their careful reading, insightful comments, and high standards. Also, I have learned invaluable lessons from interacting with them.

Finally, I would like to thank my family and friends for their incredible support in my career. Without their support, I cannot reach this point. I hope I have graduated to be a better husband, a better father, a better son and a better friend.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

An emerging area of cryptography concerns the design and analysis of "leaky" protocols (see *e.g.* Mohassel and Franklin [2006], Popa et al. [2011], Cash et al. [2013] and additional references below), which are protocols that deliberately give up some level of security in order to achieve better efficiency or additional functionality. One important tool in this area is *order-revealing encryption* Boldyreva et al. [2011], Boneh et al. [2015][1]. Order-revealing encryption (ORE) is a special type of symmetric encryption which leaks the order of the underlying plaintexts through a *public* procedure $\mathsf{Comp}$, and a paticular case of ORE is called *order preserving encryption*(OPE), where $\mathsf{Comp}$ is just numerical comparison. Specifically, in OPE, plaintexts and ciphertexts are both integers and encryption is monotonic: if $m_0 < m_1$, then $\mathsf{Enc}(k, m_0) \leq \mathsf{Enc}(k, m_1)$. In contrast, ciphertexts of ORE are no longer necessarily to be integers and the comparison for ciphertexts is replaced by a more general algorithm $\mathsf{Comp}$. The correctness requirement is, roughly, that

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

Why ORE/OPE is interesting? We immediately note that ORE/OPE allows for a client to store a database on an untrusted server in encrypted form, while still permitting the server to efficiently perform various operations such as range queries on the encrypted data without the secret decryption key. Due to this property, ORE

[1] In Boldyreva et al. [2011], it was called efficiently-orderable encryption.

has been implemented and used in real-world encrypted database systems, including CryptDB Popa et al. [2011]. Since then, various security notions of ORE have been proposed. The strongest, called "ideal" ORE Boneh et al. [2015], insists that everything about the plaintexts is hidden, except for their order. For example, it should be impossible to distinguish between encryptions of $1, 2, 3$ and $1, 4, 9$. Such ideal ORE can be constructed from multilinear maps, showing that in principle ideal ORE is achievable.

Unfortunately, a series of works starting with Naveed et al. [2015] have shown that, even hypothetical ideal ORE is insecure for various use cases Hore et al. [2012], Islam et al. [2012], Arasu et al. [2013b], Dautrich Jr and Ravishankar [2013], Liu et al. [2014], Islam et al. [2014], Cash et al. [2015], Naveed et al. [2015], Durak et al. [2016], Grubbs et al. [2017]. This is even if the scheme itself reveals nothing but the order of the plaintexts. The problem is that just the order of plaintexts alone can already reveal a significant amount of information about the data. For example, if the data is chosen uniformly from the entire domain, then even *ideal* ORE will leak the most significant bits. Moreover, the most significant bits are often the most important ones, which makes ORE in trouble. The deep reason of those attacks on ORE is that the security notions, while precise and provable, do not immediately provide any "semantically meaningful" guarantees for the privacy of the underlying data. Indeed, the above attacks show that when the adversary has a good estimate of the prior distribution the data is drawn from, essentially no security is possible.

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), 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.

- 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 Boneh et al. [2006], 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 $\mathsf{sk}_i$ for $i \geq j$ can decrypt, but any $\mathsf{sk}_i$ for $i < j$ cannot. For another example, positional witness encryption Gentry et al. [2014] also has a similar comparison structure, and is currently the best way to prove security of witness encryption under "instance-independent" assumptions.

Hence, in this dissertation, we present three of our results on ORE/OPE, which include new security notions, new constructions, and black box separations.

## 1.1 Overview of our work

This dissertation is threefold. Firstly, we prove that building ideal ORE from weaker tools is hard, where we show black box separations between ideal ORE and random oracle model, as well as generic group model. Secondly, we propose a new security notion, called parameter-hiding and build the corresponding scheme from bi-linear map. Thirdly, we show the ciphertext-length expansion of any MSDB-secure OPE (comparing to MSDB-secure ORE) is inherent, which indicates the construction in Chenette et al. [2016] (below we denote it as CLWW for ease) is optimal. In the following, we illustrate our results one by one concretely.

### 1.1.1 Black box separations between ideal ORE and weaker tools

**Motivation.** In Boldyreva et al. [2009], Boldyreva et al. give an efficient OPE 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 serect 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 plainltexts will be revealed. In order to circumvernt Boldyreva et al.'s impossibility result, Boneh et al. Boneh et al. [2015] give an ideal ORE construction from multilinear maps Boneh and Silverberg [2003], Garg et al. [2013], Coron et al. [2013], and argue that their scheme reveals no information beyond the ordering of the plaintexts. Alternate constructions achieving ideal leakage have since been proposed using multi-input functional encryption Boneh et al. [2015] or even single input functional encryption Brakerski and Segev [2015]. Unfortunately, as all known instantiations of functional encryption rely on multilinear maps anyway, all known constructions of ideal ORE require multilinear maps as well. However, current multilinear map candidates are quite inefficient and moreover have been subject to numerous attacks (e.g. Cheon et al. [2015], Miles et al. [2016], Coron et al. [2016]), meaning the resulting constructions of ideal ORE are far from practical use. Therefore, a natrual question is:

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

**Our result.** In Zhandry and Zhang [2018], we make a first attempt toward answering the above question by showing that natural constructions of ideal ORE from several simple tools are impossible. Specifically, we give black box impossibility results for building ideal ORE from symmetric key cryptography or public key encryption.

**Theorem 1.1.1 (Informal)** *There is no fully black box construction of an ideal 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 1.1.2 (Informal)** *There is no fully black box construction of an ideal 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.*[2]

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.

### 1.1.2 Parameter-hiding ORE

**Motivation.** By our result above, we know that, ideal ORE has to suffer some inefficiency. In order to develop an efficient scheme, one can relaxe the security requirements, allowing for some additional leakage. OPE is, of course, such a example and very efficient constructions of OPE are known Boldyreva et al. [2009]. However, OPE necessarily leaks much more information about the plaintexts than ideal ORE; essentially the difference between ciphertexts can be used to approximate the difference between the plaintexts. More recently, there have been efforts to achieve more precise security without sacrificing too much efficiency: CLWW Chenette et al. [2016] recently gives an ORE construction which leaks only the position of the most significant differing bits of the plaintexts (we call such a leakage profile MSDB-secure).

Unfortunately, recent works Naveed et al. [2015], Durak et al. [2016], Grubbs et al. [2017,?] have shown that, even ideal ORE can be insecure. Indeed, the attacks tell us that when the adversary has a good estimate of the prior distribution the data is draw from, essentially no security is guaranteed. However, we contend that there are

[2]The is some overlap in the implications of Theorems 1.1.1 and 1.1.2, 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 1.1.1 is simpler, and serves to highlight the ideas that will be needed for Theorem 1.1.2. 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 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.

scenarios (see below) where the adversary lacks this knowledge. A core problem in such scenarios is that the privacy of one message is inherently dependent on what other ciphertexts the adversary sees. Analyzing these correlations under arbitrary sources of data, even for ideal ORE, can be quite difficult. Only very mild results are known, for example the fact that either CLWW leakage or ideal leakage provably hides the *least* significant bits of uniformly chosen data (those bits are probably of less importance). Thus, from the construction perspective, it's natural to ask:

*Is it possible to devise a semantically meaningful notion of security for the underlying data in the case that the adversary does not have a strong estimate of the prior distribution, and develop a construction attaining this notion not based on multilinear maps?*

We stress that we are not trying to devise a scheme that is secure in the use cases of the attacks above, as many of the attacks above would apply to *any* ORE scheme; we are instead aiming to identify settings where the attacks do not apply, and then provide a scheme satisfying a given notion of security in this setting.

**Our result.** In Cash et al. [2018], we give one possible answer to the question above by introducing a novel security notion called *parameter-hiding*, and constructing the corresponding scheme *only* based on bi-linear map. In our notion, rather than focusing on the individual data records, we instead consider about the privacy of the distribution they came from, and we show how to protect some information about the underlying data distribution.

To motivate our notion, we first give an intuitive exmaple: a large university wants to outsource its database of student GPAs. For simplicity, we will assume each student's academic ability is independent of other students, and that this is reflected in the GPA. Thus, we will assume that each GPA is sampled independently and identically according to some underlying distribution. The university clearly wants to keep each individual's GPA hidden. It also may want aggregate statistics such as mean and variance to be hidden, perhaps to avoid getting a reputation for handing out very high or very low grades.

**Distribution-Hiding.** This example motivates a notion of distribution-hiding ORE, where all data is sampled independently and identically from some underlying distribution $D$, and we wish to hide as much as possible about $D$. We would ideally like to handle arbitrary distributions $D$, but in many cases will accept handling certain special classes of distributions. Notice that if the distribution itself is completely hidden, then so too is every individual record, since any information about a record is also information about $D$.

We begin with the following trivial observation: if $D$ has high min-entropy (namely, super-logarithmic), then the ideal ORE leakage is just a random ordering with no equality, since there are no collisions with overwhelming probability. In particular, this leakage is independent of the distribution $D$; as such, ideal ORE leakage hides everything about the underlying distribution, except for the super-logarithmic lower bound on min-entropy. Thus, we can use the multilinear map-based scheme of Boneh et al. [2015] to achieve distribution-hiding ORE for any distribution with high min-entropy.

We note the min-entropy requirement is critical, since for smaller min-entropies, the leakage allows for determining the frequency of the most common elements, hence learning non-trivial information about $D$[3].

Unfortunately, the only way we know to build distribution-hiding ORE is using ideal ORE, as such, we do not know of a construction not based on multilinear maps. Note that, building an ideal ORE always has to sacrifice efficiency. Thus in hopes of building such a scheme, we will relax the security notion, by allowing some information about the distribution to leak.

**Parameter-Hiding.** We recall that in many settings, data follows a known type of distribution. For example, the central limit theorem implies that many quantities such

[3]This min-entropy requirement may be somewhat problematic in some settings. GPAs for example, probably have fewer than 10 bits of entropy. However, adding small random noise to the data before encrypting (much smaller than the precision of the data) will force the data to have high min-entropy without changing the order of data, with the exception that identical data will appear different when comparing. In many cases (such as answering range queries) it is totally acceptable to fail to identify identical data.

as various physical, biological, and financial quantities are (approximately) normally distributed. It is also common practice to assign grades on an approximately normal distribution, so GPAs might reasonably be conjectured to be normal. For a different example, insurance claims are often modeled according to the Gamma distribution.

Therefore, since the general shape of the distribution is typically known, a reasonable relaxation of distribution-hiding is what we will call *parameter-hiding*. Here, we will assume the distribution has a *known, public* "shape" (*e.g.* normal, uniform, Laplace *etc.*) but it may be shifted or scaled. We will allow the overall shape to be revealed; our goal instead is to completely hide the shifting and scaling information. More precisely, we consider a distribution $D$ over $[0, 1]$ which will describe the general shape of the family of distributions in question. For example, if the shape in consideration is the set of uniform distributions over an interval, we may take $D$ to be uniform distribution over $[0, 1]$; if the shape is the normal distribution, we will take $D$ be be the normal distribution with mean $1/2$, and standard deviation small enough so that the vast majority of the mass is in $[0, 1]$. Let $D_{\alpha,\beta}$ be the distribution defined as: first sample $x \leftarrow D$, and then output $\lfloor \alpha x + \beta \rfloor$. We will call $\alpha$ the scaling term and $\beta$ the shift. The adversary receives a polynomial number of encryptions of plaintext sampled iid from $D_{\alpha,\beta}$ for some $\alpha, \beta$. We will call an ORE scheme *parameter hiding* if the scale and shift are hidden from any computationally bounded adversary given *any* polynomial bounded ciphertexts. Our main theorem is that it is possible to construct such parameter-hiding ORE from bilinear maps:

**Theorem 1.1.3 (Informal)** *Assuming bi-linear map, it is possible to construct parameter-hiding ORE for any "smooth" distribution $D$, provided the scaling term is "large enough."*

We note the restrictions to large scaling are inherent: any small scaling will lead to a distribution with low min-entropy. As discussed above, even with ideal ORE, it is possible to estimate the min-entropy of low min-entropy distributions, and hence it would be possible to recover the scaling term if the scaling term is small. Some restrictions on the shape of $D$ are also necessary, as certain shapes can yield low min-entropy even for large scaling(when we transfer the data distribution to the discrete version).

"Smoothness" (which we will define as having a bounded derivative) guarantees high min-entropy at large scales, and is also important technically for our analysis.

**Discussion.** The original ORE scheme in Boldyreva et al. [2009] leaks "whatever a random order-preserving function leaks." Unfortunately, this notion does not say anything about what such leakage actually looks like, and due to the analysis in Boldyreva et al. [2011], we know that such an ORE scheme leaks, at least, the most significant half of the bits. The situation has been improved in recent works on ORE Chenette et al. [2016], Lewi and Wu [2016], Cash et al. [2016], Joye and Passelègue [2016], where more precise "leakage profiles" are proposed. However, such leakage profiles are still of limited use, since they do not obviously say anything about the actual security of the underlying data.

We instead study ORE with a well-defined security notion for the underlying data. A key contribution of our result is that: we build a bridge to show how to connect leakage profile and our notion. Concretely, we show how to build a parameter-hiding ORE from the scheme in Cash et al. [2016], Joye and Passelègue [2016]. Nonetheless, we do not claim that parameter-hiding is sufficiently safe to use in general higher-level protocols; it only claims security in the setting where all sensitive information are the scale and shift of the underlying plaintext distribution. For instance, if the shape of the distribution is highly sensitive, or there are correlations to other data available to the attacker, then our notion would be insufficient.

However, our construction provably has better leakage than existing efficient schemes, and it at least shows some meaningful security for specific situations. Moreover we conjecture that the scheme can be shown to be useful in many other settings by extending our techniques.

### 1.1.3 Ciphertext-size lower bound for MSDB-secure OPE

**Motivation.** Order-preserving encryption is a particular case of ORE, with the restriction that the ciphertexts are numerical. Comparing to ORE, OPE indeed has its own advantage. Note that a typical application of ORE is in databases, where one party encrypts numeric columns of a database table. Later, to issue a range query on

the column, that party encrypts the endpoints of the range and requests all ciphertexts between them, an operation that can be processed by anyone who holds the encrypted column. In these settings, OPE is preferable because it can more easily be added to a database application, as the server can be oblivious to the fact that encryption is used at all. With more general ORE schemes, one needs to implement the specialized comparison operation in the database, which can be inconvenient (e.g. in a slow SQL implementation) or impossible, for instance when adding encryption to legacy systems. In fact, OPE has seen deployments in products[4] and usage in applied research Lu et al. [2010], Wang et al. [2010], Popa et al. [2011].

However, OPE does not perform as good as ORE in the realm of security. To our best of knowledge, OPE can not achieve ideal-secure Boldyreva et al. [2009], equality-pattern MSDB-secure Cash et al. [2016] and parameter-hiding Cash et al. [2018]. Hence, here we focus on the MSDB-secure OPE Chenette et al. [2016], which is the best-known security notion that OPE can achieve. In Chenette et al. [2016], CLWW propose both MSDB-secure ORE and OPE, where the ciphertext size of the ORE scheme is only $1.58m$, comparing to $\lambda m$ for the OPE scheme. Hence, a natural question arises:

*Could we build an MSDB-secure OPE associated with shorter ciphertext-size?*

**Our Result.** In Cash and Zhang [2018], we make a first attempt toward answering the above question by showing a (almost-tight) lower bound of the ciphertext size for MSDB-secure OPE against unbounded adversary. Specifically, we prove that any OPE scheme meeting the information-theoretic MSDB-secure notion must have ciphertexts of length

$$\lambda m - m \log m + m \log e,$$

where again $m$ is the message length, logarithms are base 2, and $e$ is the base of the natural logarithm. This bound tells that CLWW has almost optimal ciphertext size, as it has leading term $\lambda m$ instead of $(\lambda - \log m)m$.

[4] `https://www.skyhighnetworks.com`, `https://www.ciphercloud.com/`, SAP's SEEED `https://www.sics.se/sites/default/files/pub/andreasschaad.pdf`, `https://www.bluecoat.com/` and Cipherbase Arasu et al. [2013a].

**Theorem 1.1.4 (Informal)** *Any MSDB-secure OPE scheme associated with message space* $\{0,1\}^m$ *must have ciphertext of length* $(\lambda - \log m)m$.

**A following-up work.** In Cash and Zhang [2018], we *only* consider an information theoretic version of MSDB-secure notion, which requires the same security but against unbounded adversaries. Surprisingly, a following-up work by Segev and Shahaf Segev and Shahaf [2018] extend our result to computational security level, without losing any advantage for the attacks. Concretely, they illustrate a (almost-tight) lower bound for MSDB-secure OPE against non-uniform adversary, while the lower bound against the uniform adversary is still an open problem.

### 1.1.4 Discussion and Perspective.

In light of our impossibility results and the known attacks, a natural question is: *what's the future of ORE?* In this part, we give one positive thought on this question.

Easy to note that, the essential power of the most known attacks is the sorting, for instance, the binomial attack. And due to the functionality, the attack seem inevitable when only using ORE. Hence, to prevent those attacks, ORE itself is insufficient and new technique is required. In this part, we illustrate one potential solution.

The new technique we here propose is adding fake points during the encryption procedure, specifically, let $m$ be the message and $\mathsf{OEnc}$ as the ORE encryption, our new encryption scheme would be:

$$\mathsf{Enc}(m) = \pi(\mathsf{OEnc}(m), \mathsf{OEnc}(m_1), \ldots, \mathsf{OEnc}(m_q)),$$

where $\pi$ is a random permutation, and $m_1, \ldots, m_q$ are the fake points that are sampled from some distribution. Of course, each ORE ciphertext would be associated with message authentication code to indicate the underlying message is real or fake.

Why does it help? The basic intuition is that, if the adversary can not tell which one is the real encryption, then it cannot do the sorting anymore. Moreover, our new encryption is based on ORE, and it's trivial that our scheme still supports whatever queries that ORE supports, with additional cost of ciphertext length and bandwidth.

This potential solution seems quite promising, but yet the security behind it is quite unknown. Say, what kind of distribution should the fake points draw from? Hence, we treat the analysis of our new scheme as an open problem and one of the possible research directions for ORE.

## 1.2 Related work on other tools

In this part, we briefly go over the other tools on leaky protocols. Specifically, those tools done on "leaky cryptography" includes work on multiparty computation Mohassel and Franklin [2006], searchable symmetric and structured encryption Song et al. [2000], Goh et al. [2003], Chang and Mitzenmacher [2005], Curtmola et al. [2006], Chase and Kamara [2010], Cash et al. [2013], Kamara and Moataz [2016], and property-preserving encryption Bellare et al. [2007], Boldyreva et al. [2009], Pandey and Rouselakis [2012]. In the database community, the problem of querying an encrypted database was introduced by Hacigümüş, Iyer, Li and Mehrotra Hacigümüş et al. [2002], leading to a variety of proposals there but mostly lacking formal security analysis. Proposals of specific outsourced database systems based on property-preserving encryption like ORE include CryptDB Popa et al. [2011], Cipherbase Arasu et al. [2013a], and TrustedDB Bajaj and Sion [2014].

## 1.3 Organization

This dissertation will present our results on ORE, following the same order listed above.

- Chapter 2 covers the notations and the formal definition of security/efficiency/-correctness for ORE.
- Chapter 3 resolves the first question by showing a black-box separation between ideal ORE and random oracle model, as well as generic group model.
- Chapter 4 resolves the second question, by first proposing a meaningful security notion called parameter-hiding and constructing the corresponding ORE scheme from bi-linear map.

- Chapter 5 resolves the last question by giving a lower bound which indicates that the scheme in Chenette et al. [2016] is almost optimal.
- Chapter 6 completes the dissertation with conclusion.

# Chapter 2

# Preliminaries

## 2.1 Notation and basic results

All algorithms are assumed to be polynomial-time in the security parameter (though we will sometimes refer to efficient algorithms explicitly). We will denote the security parameter by $\lambda$. For a random variable $Y$, we write $y \xleftarrow{\$} Y$ to denote that $y$ is sampled according to $Y$'s distribution, moreover, let $D$ be $Y$'s distribution, we abuse notation $y \xleftarrow{\$} D$ to mean that $y$ is sampled according to $D$. For an algorithm $A$, by $y \xleftarrow{\$} A(x)$ we mean that $A$ is executed on input $x$ and the output is assigned to $y$, furthermore, if $A$ is randomized, then we write $y \xleftarrow{\$} \mathcal{A}(x)$ to denote running $\mathcal{A}$ on input $x$ with a fresh random tape and letting $y$ be the random variable induced by its output. We denote by $\Pr[A(x) = y : x \xleftarrow{\$} X]$ the probability that $A$ outputs $y$ on input $x$ when $x$ is sampled according to $X$. We say that an adversary $\mathcal{A}$ has advantage $\epsilon$ in distinguishing $X$ from $Y$ if $\Pr[A(x) = 1 : x \xleftarrow{\$} X]$ and $\Pr[A(y) = 1 : y \xleftarrow{\$} Y]$ differ by at least $\epsilon$. If $A$'s running time is polynomial in $\lambda$, then $A$ is called probabilistic polynomial-time(PPT).

When more convenient, we use the following probability-theoretic notation instead. We write $P_X(x)$ to denote the probability that $X$ places on $x$, i.e. $P_X(x) = \Pr[X = x]$, and we say $P_X(x)$ is the probability density function (PDF) of $X$'s distribution. The *statistical distance* between $X$ and $Y$ is given by $\Delta = \frac{1}{2}\sum_x |P_X(x) - P_Y(x)|$. If $\Delta(X, Y)$ is at most $\epsilon$ then we say $X, Y$ are $\epsilon$-close. It is well-known that if $X, Y$ are $\epsilon$-close then any (even computationally unbounded) adversary $A$ has advantage at most $\epsilon$ in distinguishing $X$ from $Y$.

The *min-entropy* of a random variable $X$ is $H_\infty(X) = -\log(\max_x P_X(x))$. 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 $\mathsf{poly}$, 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 $\mathsf{poly}$. For $M, N \in \mathbb{N}$, we let $[M] = \{1, \ldots, M\}$, $[M]' = \{0, \ldots, M-1\}$ and $[M, N] = \{M, \ldots, N\}$. We write $\vec{m}$ as a vector of plaintexts and $|\vec{m}|$ as the vector's length, namely $\vec{m} = (m_1, \ldots, m_s)$ and $|\vec{m}| = s$. For a vector $\vec{m}$, by $a\vec{m}$ we mean $(am_1, \ldots, am_s)$ and we write $\vec{m} + b$ to denote $(m_1 + b, \ldots, m_s + b)$. Let $x$ be a real number, we write $\lfloor x \rfloor$ as the largest integer s.t. $\lfloor x \rfloor \leq x$, and $\lceil x \rceil$ as the smallest integer s.t. $\lceil x \rceil \geq x$. By $\lfloor x \rceil$, we mean rounding $x$ to the nearest integer, namely $-1/2 \leq \lfloor x \rceil - x < 1/2$. If $P$ is a predicate, we write $\mathbf{1}(P)$ for the function that takes the inputs to $P$ and returns 1 if $P$ holds and 0 otherwise.

Next we give a well-known data processing lemma (c.f. Cover and Thomas [2006]), which we will use in our proof.

**Lemma 2.1.1** *Let $X$ and $Y$ be r.v.s, and $f$ be any function that includes the support of $X$ and $Y$ in its domain. Then $\Delta(f(X), f(Y)) \leq \Delta(X, Y)$.*

**Definition 2.1.2 (Pseudorandom Function.)** *A function $F : \{0,1\}^\lambda \times D \to \{0,1\}^\lambda$ is said to be a pseudorandom function with domain $D$ if for all efficient $\mathcal{A}$ we have that*

$$Adv_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}^{F(K,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{g(\cdot)}(1^\lambda) = 1]|$$

*is a negligible function of $\lambda$, where $K$ is uniform over $\{0,1\}^\lambda$ and $g$ is uniform over all functions from $D$ to $\{0,1\}^\lambda$.*

**Definition 2.1.3 (Bi-linear Map.)** *Let $\mathbb{G}$, $\hat{\mathbb{G}}$, $\mathbb{G}_T$ be three groups of order $p$ for some large prime $p$, and $g$ be generator of $\mathbb{G}$ and $\hat{g}$ be a generator of $\hat{\mathbb{G}}$. We say a map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ is a bilinear map, if $e$ satisfies the following properties:*

1. *Bilinear: We say a map $e$ is bilinear if for all $a, b \in \mathbb{Z}_p$, we have $e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$.*

2. *Non-degenerate: The map does not send all pairs in $\mathbb{G} \times \hat{\mathbb{G}}$ to the identity in $\mathbb{G}_T$. Note that since $\mathbb{G}, \hat{\mathbb{G}}$ and $\mathbb{G}_T$ are groups of prime order, which implies that if $g$(resp. $\hat{g}$) is the generator of $\mathbb{G}$(resp. $\hat{\mathbb{G}}$), then $e(g, \hat{g})$ is the generator of $\mathbb{G}_T$.*

3. *Computable: For any $a, b \in \mathbb{Z}_p$, there exists an efficient algorithm to compute $e(g^a, \hat{g}^b)$.*

*We say a bilinear map is symmetric if $\mathbb{G} = \hat{\mathbb{G}}$, else it is asymmetric.*

**Definition 2.1.4 (SXDH assumption.)** *Let $\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T$ be prime-order $p$ groups, $g$ be generator of $\mathbb{G}$ and $\hat{g}$ be a generator of $\hat{\mathbb{G}}$, and $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ be a bilinear pairing. We say the* symmetric external Diffie-Hellman assumption holds *with respect to these groups and pairing if for all efficient $\mathcal{A}$,*

$$|\Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, R) = 1]|$$

*and*

$$|\Pr[\mathcal{A}(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^{ab}) = 1] - \Pr[\mathcal{A}(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{R}) = 1]|$$

*are negligible functions of $\lambda$, where $a, b, c$ are uniform over $\mathbb{Z}_p$ and $R$ (resp. $\hat{R}$) is uniform over $\mathbb{G}$ (resp. $\hat{\mathbb{G}}$).*

**Definition 2.1.5 (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 $\Pi$ 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 $\mathcal{M}$.

**Definition 2.1.6 (Random Oracle Model.)** *Random oracle model Bellare and Rogaway [1993] is an idealized model (a theoretical black box) which responds to any unique*

*query with a truly random string, and if the query is repeated, the response would be consistent. More concretely, a random oracle model has a publicly accessible hash function $H : \{0,1\}^* \to \{0,1\}^n$ such that : 1) for any $x$, every bit of $H(x)$ is truly random; 2) for any $x \neq y$, $H(x)$ and $H(y)$ are independent.*

For ease of exposition, here we consider of a simple variant of the generic group model, which is equivalent to the usual generic group model Shoup [1997]:

**Definition 2.1.7 (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 a handle vector $\vec{h} = (h_1, \ldots h_n) \in S$, and a vector $\vec{v} = (v_0, \ldots, v_n)$ 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.*

## 2.2 Definition, Correctness and Efficiency for ORE

The following definition of syntax for order revealing encryption makes explicit that the comparison procedure may use helper information (e.g. a description of a particular group) by incorporating a public key, denote $\mathsf{pk}$.

**Definition 2.2.1 (ORE.)** *An ORE scheme $\Pi$ with message space $[N]$ is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ with the following syntax.*

- *The key generation algorithm $\mathsf{Gen}$ is randomized, takes inputs $(1^\lambda, N)$, and always emits two outputs $(\mathsf{pk}, \mathsf{sk})$. We refer to the first output $\mathsf{pk}$ as the* public key *and the second output $\mathsf{sk}$ as the* secret key*.*
- *The encryption algorithm $\mathsf{Enc}$ takes inputs $(\mathsf{sk}, m)$ where $m \in [N]$, and always emits a single output $c$, that we refer to as a* ciphertext*.*

- *The comparison algorithm* $\mathsf{Comp}$ *takes inputs* $(\mathsf{pk}, c_1, c_2)$*, and emits "<", "=" or ">", which indicates the order of the underlying plaintexts.*

*If* $\mathsf{Comp}$ *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.*

Next, we give the definitions of correctness for order-revealing encryption. 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 $\mathsf{Order}(m_0, m_1)$ be the order of $(m_0, m_1)$, where:

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

In this dissertation, we will consider five notions of correctness. Concretely:

**Definition 2.2.2 (Perfect Correctness.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with respect to the message space* $[N]$*. We say* $\Pi$ *has perfect correctness, if for any message pair* $(m_0, m_1) \in [N]$*, we have*

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

*where the probability is taken over the choice of* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*.*

**Definition 2.2.3 (Almost Perfect Correctness.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with respect to the message space* $[N]$*. We say* $\Pi$ *has almost perfect correctness, if there is a negligible function* $\mu = \mathsf{negl}(\lambda)$ *such that*

$$\Pr[\exists (m_0, m_1), \mathsf{Comp}(\mathsf{pk}, C_{m_0}, C_{m_1}) \neq \mathsf{Order}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)] \leq \mu,$$

*where the probability is taken over the choice of* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*.*

$\underline{\mathsf{CCor}(\mathcal{A})}$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda); (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Enc}(\mathsf{sk},\cdot)}(1^\lambda);$
Return 1 if $\mathsf{Comp}(\mathsf{pk}, \mathsf{Enc}(\mathsf{sk}, \mathsf{m_0}, \mathsf{Enc}(\mathsf{sk}, \mathsf{m_1}) \neq \mathsf{Order}(m_0, m_1)$, and 0 otherwise.

Figure 2.1: Computational Correctness

**Definition 2.2.4 (Statistical Correctness.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with respect to the message space* $[N]$*. We say* $\Pi$ *has statistical correctness, if for any message pair* $(m_0, m_1)$*, there is a negligible function* $\mu = \mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Comp}(\mathsf{pk}, C_0, C_1) = \mathsf{Order}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)] \geq 1 - \mu,$$

*where the probability is taken over the choice of* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\lambda)$*.*

**Definition 2.2.5 (Computational Correctness.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with respect to the message space* $[N]$*. For any PPT adversary* $\mathcal{A}$ *we define the game* CCor *in figure 2.1. The advantage of* $\mathcal{A}$ *for the game* CCor *is defined to be:*

$$Adv_{\mathcal{A}}^{\mathsf{CCor}}(1^\lambda) = \Pr[\mathsf{CCor}(\mathcal{A}) = 1].$$

*We say that* $\Pi$ *has computational correctness if for any PPT adversary,* $Adv_{\mathcal{A}}^{\mathsf{CCor}}(1^\lambda)$ *is negligible.*

**Definition 2.2.6 (Partial Correctness.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with respect to the message space* $[N]$*. We say* $\Pi$ *has statistical correctness, if for any message pair* $(m_0, m_1)$*, there is a noticeable function* $\rho(\lambda)$ *such that*

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

*where the probability is taken over the choice of* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*.*

Easy to note that, the definitions of correctness is illustrated in the decreasing order, for instance, statistical correctness is stronger than computational one and also computational correctness is stronger than partial one. Next, we present the definition

of efficiency for order-revealing encryption. Typically in the literature, ORE is defined as having computationally efficient algorithms and in this work, we also consider the ORE scheme in idealized model, where algorithms have access to an oracle.

**Definition 2.2.7 (Efficiency.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with respect to the message space* $[N]$*. We say* $\Pi$ *is computationally efficient if the three algorithms run in time polynomial in* $(\log N, \lambda)$*. If* $\Pi$ *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 in the idealized model, we still require the number of queries to be polynomial.

**Definition 2.2.8 (Query Efficiency.)** *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* $\Pi$ *is query efficient if the three algorithms 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 2.2.9 (Succinct Ciphertext.)** *Let* $\Pi$ *(resp.* $\Pi^{\mathcal{M}}$*) be an ORE with respect to the message space* $[N]$*(resp. in idealized model). We say* $\Pi$ *(resp.* $\Pi^{\mathcal{M}}$*) has succinct ciphertext 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.

## 2.3 Security for ORE

In this part, we first give a simulation-based security definition, due to Chenette et. al. Chenette et al. [2016]. Here, a *leakage profile* is any randomized algorithm

The definition refers to games given in Figure 2.1, which we review now. In the real game, key generation is run and the adversary is given the comparison key and oracle

| Game $\text{REAL}^{\mathsf{ore}}_{\Pi}(\mathcal{A})$: | Game $\text{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}}(\mathcal{A},\mathcal{S})$: |
|---|---|
| $(\mathsf{sk},\mathsf{ck}) \xleftarrow{\$} \mathcal{K}(1^\lambda, M)$; $b \xleftarrow{\$} \mathcal{A}^{\text{ENC}}(\mathsf{ck})$ | $\mathsf{st}_\ell \xleftarrow{\$} \perp$; $(\mathsf{ck},\mathsf{st}_s) \xleftarrow{\$} S(1^\lambda)$; $b \xleftarrow{\$} \mathcal{A}^{\text{ENC}}(\mathsf{ck})$ |
| Return $b$ | Return $b$ |
| ENC($m$): | ENC($m$): |
| Return $\mathsf{E}(\mathsf{sk}, m)$ | $(L,\mathsf{st}_\ell) \xleftarrow{\$} \mathcal{L}(\mathsf{st}_\ell, m)$; $(c,\mathsf{st}_s) \xleftarrow{\$} \mathcal{S}(L,\mathsf{st}_s)$ |
| | Return $c$ |

Table 2.1: Games $\text{REAL}^{\mathsf{ore}}\Pi(\mathcal{A})$ (left) and $\text{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}}(\mathcal{A},\mathcal{S})$ (right).

access to the encryption algorithm with the corresponding secret key. The adversary eventually outputs a bit that the game uses as its own output. In the ideal simulation game, the adversary is interacting with the same oracle, but the comparison key is generated by a stateful simulator, and the oracle responses are generated by the simulator which receives leakage from the stateful leakage algorithm $\mathcal{L}$.

**Definition 2.3.1 ($\mathcal{L}$-simulation-security for ORE)** *For an ORE scheme $\Pi$, an adversary $\mathcal{A}$, a simulator $\mathcal{S}$, and leakage profile $\mathcal{L}$, we define the games $\text{REAL}^{\mathsf{ore}}_{\Pi}(\mathcal{A})$ and $\text{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}}(\mathcal{A})$ in Figure 2.1. The advantage of $\mathcal{A}$ with respect to $\mathcal{S}$ is defined as*

$$\mathsf{Adv}^{\mathsf{ore}}_{\Pi,\mathcal{L},\mathcal{A},\mathcal{S}}(\lambda) = \left|\Pr[\text{REAL}^{\mathsf{ore}}_{\Pi}(\mathcal{A}) = 1] - \Pr[\text{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}}(\mathcal{A},\mathcal{S}) = 1]\right|.$$

*We say that $\Pi$ is $\mathcal{L}$-simulation-secure if for every efficient adversary $\mathcal{A}$ there exists an efficient simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\mathsf{ore}}_{\Pi,\mathcal{L},\mathcal{A},\mathcal{S}}(\lambda)$ is a negligible function.*

*We also define* non-adaptive *variants of the games where $\mathcal{A}$ gets a single query to an oracle that accepts a vector of messages of unbounded size. In the real game $\text{REAL}^{\mathsf{ore\text{-}na}}_{\Pi}(\mathcal{A})$, the oracle returns the encryptions applied independently to each message. In the ideal game $\text{SIM}^{\mathsf{ore\text{-}na}}_{\Pi}(\mathcal{A})$, the leakage function gets the entire vector of messages as input and produces an output $L$ that is then given to $\mathcal{S}$ which produces a vector of ciphertexts, which are returned by the oracle.*

*We define the* non-adaptive advantage *of $\mathcal{A}$ with respect to $\mathcal{S}$ analogously, and denote it $\mathsf{Adv}^{\mathsf{ore\text{-}na}}_{\Pi,\mathcal{L},\mathcal{A},\mathcal{S}}(\lambda)$.* Non-adaptive $\mathcal{L}$-simulation security *is defined analogously.*

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

Table 2.2: t-time Static Indistinguishable Game

**Ideal-secure ORE.** Ideal ORE is the case where the leakage profile $\mathcal{L}$ is simply the list of results of comparisons between the plaintexts. Formally, the leakage profile is:

$$\mathcal{L}_{\text{ideal}}(m_1, \ldots, m_q) = \{(i, j, \mathbf{1}(m_i < m_j)) \ : \ 1 \leq i < j \leq q\}.$$

We note that such a $\mathcal{L}$ is *always* revealed by the comparison algorithm, so ideal ORE is the best one can hope for.

**MSDE-secure ORE.** As an example of a non-ideal leakage profile, consider the leakage profile $\mathcal{L}_{\mathsf{clww}}$ in Chenette et al. [2016]. For $m_0, m_1 \in \{0,1\}^n$, we define the most significant differing bit of $m_0$ and $m_1$, denoted $\mathsf{msdb}(m_0, m_1)$, as the index of first bit where $m_0, m_1$ differ, or $n+1$ if $m_0 = m_1$.

The CLWW leakage profile $\mathcal{L}_{\mathsf{clww}}$ takes in input a vector of plaintext $\vec{m} = (m_1, \ldots, m_q)$ and produce the following:

$$\mathcal{L}_{\mathsf{clww}}(m_1, \ldots, m_q) := (\forall 1 \leq i, j \leq n, \mathbf{1}(m_i < m_j), \mathsf{msdb}(m_i, m_j))$$

Next, to strengthen our impossibility results, we also introduce two indistinguishability-based notions, namely $t$-time secure game and $\mathsf{Ind\text{-}clww}$ game. Note that, for ideal security, only the order is revealed, roughly, given two sequences of message $\vec{m}, \vec{m}'$ such that $\forall i, j \in |\vec{m}|, \mathsf{Order}(m_i, m_j) = \mathsf{Order}(m'_i, m'_j)$, the distribution of $\mathsf{Enc}(\vec{m})$ and $\mathsf{Enc}(\vec{m}')$ are indistinguishable. Now, for the first notion, we add a restriction that $|\vec{m}| = |\vec{m}'| = t$, define game in figure 2.2, and for the second one, comparing to CLWW security in Chenette et al. [2016], we only deal with a non-adaptive adversary and define it in figure 2.2.

**Definition 2.3.2 ($t$-time secure.)** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ *be an ORE scheme with*

$\underline{\mathsf{Ind\text{-}clww}(\mathcal{A})}$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(N, 1^\lambda); (\vec{m_0}, \vec{m_1}) \leftarrow \mathcal{A}(\mathsf{pk}, N, 1^\lambda)$ *s.t.* $\mathcal{L}_{\mathsf{clww}}(\vec{m_0}) = \mathcal{L}_{\mathsf{clww}}(\vec{m_1})$;
$\vec{C_0} = (\mathsf{pk}, \mathsf{Enc}(\mathsf{sk}, \vec{m_0}); \vec{C_1} = (\mathsf{pk}, \mathsf{Enc}(\mathsf{sk}, \vec{m_1})$;
$b' \leftarrow \mathcal{A}(\vec{C_b})$; Return $(b \stackrel{?}{=} b')$

Figure 2.2: IND-MSDB-secure

*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 2.2. The advantage of* $\mathcal{A}$ *for the t-time static indistinguishable game is defined to be:*

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

*We say that* $\Pi$ *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* $\Pi$ *is fully (computationally/statistically) secure if* $\Pi$ *is t-time (computationally/statistically) secure for any polynomial* $t = poly(logN, \lambda)$.

*If* $\Pi$ *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* $\mathcal{M}$.

**Definition 2.3.3 (IND-MSDB-secure.)** *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* $\mathsf{ind} - \mathsf{clww}(\mathcal{A})$ *in figure 2.2. The advantage of* $\mathcal{A}$ *is defined to be:*

$$Adv_{\Pi,\mathcal{A}}(1^\lambda) = 2\Pr[\mathsf{Ind\text{-}clww}(\mathcal{A}) = 1] - 1.$$

*We say that* $\Pi$ *is IND-MSDB-computationally secure if for all efficient* $\mathcal{A}$, $Adv_{\Pi,\mathcal{A}}$ *is a negligible. We say that* $\Pi$ *is IND-MSDB-statistically-secure if the same condition holds for all (unbounded, wlog deterministic) adversaries* $\mathcal{A}$; *more specifically, we say* $\Pi$ *is* $2^\lambda$-ind-clww-statistically-secure *if for all unbounded adversaries, the advantage is at least* $2^\lambda$.

# Chapter 3

# Impossibility of Ideal Order-Revealing Encryption in Idealized Models

In this section, we present our impossibility result for ideal ORE in the idealized model Zhandry and Zhang [2018]. First, we give a technique overview, then we show impossibility for information-theoretic ideal ORE, after that, we extend our result to random oracle model and generic group model.

## 3.1 Technique overview

First, we start with the idealized model $\mathcal{M}$ capturing the primitive that we want to separate ORE from: in this dissertation, we take $\mathcal{M}$ to be a random oracle or the generic group model Shoup [1997].

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[1]
- The scheme is only partially correct, in that $\mathsf{Comp}$ may result in an incorrect answer, but is noticeably biased towards the correct answer.[2]
- 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

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

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

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:

1. Compile any scheme satisfying the above requirements into one where $\mathsf{Comp}$ 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 $\mathsf{Comp}$ 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[3]. 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 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 Erdős and Sós. He asked the question: suppose every set of $p$ nodes is *dominated* by another node;

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

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 set $T$ 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 $\mathsf{Comp}$ 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, $\mathsf{Comp}$ 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 very 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. Boldyreva et al. [2009].** 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 $\mathsf{Comp}$ does not make queries to $\mathcal{M}$.**

We show that if $\mathsf{Comp}$ 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 $\mathsf{Comp}$ makes no oracle queries.

The process is simple. Since $\mathsf{Comp}$ 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 Canetti et al.'s Canetti et al. [2015] 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 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 3.4.

**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 Pass

and Shelat [2016], which show how to remove generic groups from obfuscation constructions[4]. 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 $\mu$. We will assume that $\mu$ 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 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 3.5.

**Difficulties for extending to bilinear and multilinear maps.** Pass and Shelat's Pass and Shelat [2016] 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

[4]We note that Mahmoody et al. Mahmoody et al. [2016] extend the Pass and Shelat results result to any (even non-commutative) finite ring; we leave extending our impossibility to the non-commutative setting as an interesting open problem

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 Brakerski and Rothblum [2014] and ORE Boneh et al. [2015] 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.

## 3.2 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 Lewi and Wu [2016] [5]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 $\mathsf{Enc}, \mathcal{C}$ 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 3.2.1** *In standard model, there does not exist fully statistically secure ORE* $\Pi$ *such that*

- $\Pi$ *is partially correct;*
- $\Pi$*'s message space is super-polynomial;*
- $\Pi$ *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.

### 3.2.1 Impossibility for perfect correct ORE

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

**Theorem 3.2.2** *In standard model, there does not exist statistically secure ORE* $\Pi$ *such that*

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

[5]here we treat the PRFs and PRPs in Lewi and Wu [2016] as real random functions and permutations, which achieving statistical security, rather than only computational security

Firstly, we give a brief description of our proof strategy. Let $\Pi$ be an ORE scheme on message space $[t+1]$, where $t = \mathsf{poly}(\lambda)$, such that $\Pi$ is perfectly correct and statistically secure. We immediately observe that $\Pi$ 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 Erdős and Sós. Roughly, for any $\Pi$ 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 Erdős and Sós.Then we sample a sequence of sub-graphs such that $G_{t+1} \supseteq G_{t-1} \supseteq \ldots G_1$[6] in a specific way(based on our ORE). After that, we prove for any adjacent pair, we have $\mathbb{E}[\log|G_i|] \geq \mathbb{E}[\log|G_{i-2}|] + \log(1.6), \forall i \in \{t+1, t-1, \ldots, 3\}$, which means $\mathbb{E}[\log|G_{t+1}|] \geq \lfloor \frac{t-1}{2} \rfloor \log 1.6$. More precisely:

**Lemma 3.2.3** *In standard model, let* $\Pi$ *be an perfectly correct* $t$*-time secure ORE on message space* $[t+1]$*, then* $\Pi$ *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 Erdős and Sós.

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 $\Pi^*_{t+1}$ as follows. The public key for $\Pi_{t+1}$ 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 $\Pi_{t+1}$, 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

[6]here we assume $t$ is even

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 an 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 $\Pi_{t+1}$ and $\Pi^*_{t+1}$ is adding $G_{t+1}$ to the public key, which only affects the efficiency of $\mathsf{Gen}$ and $\mathsf{Comp}$, 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 $\Pi^*$ is $t$-time secure, implying the distribution of the encryptions for

$\vec{M}_0$ and $\vec{M}_1$ are statistically close, over the probability $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}^*_{t+1}$, where,

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

Then, let $f_L, f_R$ be the expected fraction of 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

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

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 3.2.2. Suppose $\Pi$ is an ORE such that: 1) $\Pi$ is perfect correct and statistically secure; 2) $\Pi$'s message space is $[N]$, where $N$ is super-polynomial; 3) $\Pi$ 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 $\Pi$ is $t$-time secure. According to Lemma 3.2.3, $r \geq \lfloor \frac{t-1}{2} \rfloor \cdot \log 1.6 > r$, a contradiction. ∎

### 3.2.2 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 3.2.4** *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. $\Pi$ is $\frac{1}{2} + \rho$ correct, where $\rho$ is noticeable;
2. $\Pi$'s message space is $[N]$, where $N$ is super-polynomial;
3. $\Pi^*$ 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 $\Pi'$ 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 $\vec{C} = (C_1, \ldots, C_s)$;
- $\mathsf{Comp}'(\mathsf{pk}', \vec{C}_0, \vec{C}_1)$ let $\vec{C}_0 = (C_1^0, \ldots, C_s^0), \vec{C}_1 = (C_1^1, \ldots, C_s^1)$, outputs the majority of $(\mathsf{Comp}(pk_i, C_i^0, C_i^1))_{i=1}^s$.

We immediately observe that $\Pi'$ also has succinct ciphertexts, and by hybrid argument, it's easy to have that $\Pi'$ 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 $\Pi'$ 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 $\Pi'$ except we modify $\mathsf{Gen}^*$: 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, $\Pi^*$ is also statistically secure. Notice that $\mathsf{Gen}^*$ is no longer efficient even if $\mathsf{Gen}$ was. Fortunately, our notion in standard model allows us to have inefficient $\mathsf{Gen}$. Thus, statistically secure ORE in standard model with succinct ciphertexts and perfect correctness on the same message space exists. ∎

Combing Theorem 3.2.2 and 3.2.4, we establish Theorem 3.2.1.

## 3.3 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 3.2.1 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 3.2.1. 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 3.3.1** *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 $\mathcal{C}$. While in the original scheme, each run of $\mathcal{C}$ uses independent randomness, here we use the same randomness every time. However, since the experiment defining correctness only considers a single run of $\mathcal{C}$, 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 $\Pi^*$ as:

- $\mathsf{Gen}^*$ 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 $\Pi^*$ 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 $\Pi^*$ is exactly the same as $\Pi$'s. We note that the correctness is well preserved. In fact, according to the partial correctness definition, the randomness used in $\mathsf{Comp}$ 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 $\mathcal{C}$) 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 $\Pi$. 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 3.3.2** *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 $\mathsf{Comp}$ only includes the public key and ciphertext, and we cannot just absorb the model to the public key as we did in Theorem 3.3.2. 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.

## 3.4 Impossibility for ORE in 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 3.2 and 3.3, 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 Canetti et al. [2015], 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 $\mathsf{Comp}$. If $x$ was not a

query made during encryption $(\mathsf{Enc}^{\mathcal{O}}(m_0), \mathsf{Enc}^{\mathcal{O}}(m_1))$, then we claim $\mathsf{Comp}$ 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 $\mathsf{Comp}$ must still output the right value in this case.

For a particular run of $\mathsf{Comp}$ on encryptions of $m_0, m_1$, we therefore call the oracle queries made during encryption "sensitive" queries. $\mathsf{Comp}$ only needs access to $\mathcal{O}$ on sensitive queries; for all others, it can answer randomly. The difficulty, then, is (1) allowing $\mathsf{Comp}$ 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 $\mathsf{Comp}$'s queries are ever sensitive. In this case, $\mathsf{Comp}$ 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 $\mathsf{Comp}$'s queries are always sensitive. In this case, if the adversary sees $\ell$ 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 $\ell$ large enough (namely, bigger than $q$). Even so, security must hold. Therefore, we can construct a modified scheme where $\mathsf{Enc}$ simply outputs all the queries it makes and the corresponding answers along with the ciphertext. Then all the sensitive queries $\mathsf{Comp}$ 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 $\mathsf{Comp}$'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 3.4.1** *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 $\mathsf{Gen}_0$ makes no queries to $\mathcal{O}$. This is actually without loss of generality: since $\mathcal{O}$ is a deterministic oracle, we can always treat $\mathsf{sk}$ as the random coins inputted to $\mathsf{Gen}_0$, and run $\mathsf{Gen}_0$ every time we encrypt a message.

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

$$\Pr[\Pi^0] = \min_{m_0,m_1} \Pr[\mathsf{Comp}_0^{\mathcal{O}}(\mathsf{pk}, C_0, C_1) = \mathsf{Order}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_0(1^\lambda)],$$

where $C_b = \mathsf{Enc}^{\mathcal{O}}(\mathsf{sk}, m_b)$. 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, $\mathsf{Comp}_0$ 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 $\rho$ 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 $\mathsf{sk}$. Notice that the set $Q_{\mathsf{sk},m}$ is fully determined by $\mathsf{sk}$ and $m$ since $\mathsf{Enc}$ and $\mathcal{O}$ are deterministic,

3. $\mathsf{Comp}_0^{\mathcal{O}}$ makes $u$ queries to the oracle $\mathcal{O}$. Let $S_{\mathsf{pk},m_0,m_1}$ be the set of query-answer pairs made when comparing the encryptions of $(m_0, m_1)$ under key $\mathsf{pk}$. Again, $S_{\mathsf{pk},m_0,m_1}$ 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 $S_0 \cup S_1$, $\mathsf{Comp}$ responds to the query with $y$;
  2. Otherwise, returns a random string.

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

**Lemma 3.4.2** *If $\Pi^0$ is $t + 2s$ statically secure, then $\Pi^*$ is $t$-time statically secure.*

The entire view of the the adversary $\mathcal{A}$ in the $t$-time experiment for $\Pi^*$ can be simulated by a $t + 2s$-time adversary $\mathcal{B}$ for $\Pi^0$: 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 3.4.2 holds for any $t = \mathsf{poly}(\log N, \lambda)$, which means $\Pi^*$ is statistically secure. And what's more interesting is that $\Pi^*$'s partial correctness. In the following, we prove that $\Pi^*$ also preserves partial correctness, though there is some loss in the concrete correctness parameter.

**Lemma 3.4.3** $\Pr[\Pi^*] \geq \frac{1}{2} + \rho$.

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 3.4.4**

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

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 $\mathsf{Comp}_1$ works the same as $\mathsf{Comp}_0$, 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 $\Pi_1$ for $x$ does not significantly harm the ability of $\mathsf{Comp}_1$ 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 $\Pi_1$ 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 $C^* = (C, S_{\mathsf{pk},m}^{(j)})$;

- $\mathsf{Comp}^{\mathcal{O}}_{1,j}(\mathsf{pk}_{1,j}, C^*_0, C^*_1)$ : let $C^*_0 = (C_0, S_0), C^*_1 = (C_1, S_1)$. It runs $\mathsf{Comp}^{\mathcal{O}}_0(\mathsf{pk}_{1,j}, C_0, C_1)$, 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 3.4.5** *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 $\mathsf{Bad}_j$ 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.
- $\mathsf{Comp}^{\mathcal{O}}_0$ 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, $\Pi_1$ 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 $\Pi_0$ are independent of $\mathcal{O}(x)$, meaning that during the correctness experiment, $\mathcal{O}(x)$ in $\Pi_0$ is a random string. Hence $\Pi_1$ answers the query with the correct distribution.
- $x \in Q \setminus S^{(j)}$, but $\mathsf{Bad}_j$ does not occur. Here, we must have that $\mathsf{Comp}_0$ either produced the incorrect answer, or $\mathsf{Comp}_{1,j}$ produced the correct answer.
- $\mathsf{Bad}_j$ 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 $\mathsf{Comp}$ 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 $\Pi_0$, and in the second case, $\Pi_{1,j}$ in expectation has the same correctness as $\Pi_0$. 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 $\mathsf{Bad}_j$, 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 $\ell$ 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 $\Pi_1$. 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 $\Pi_j$, 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. ∎

### 3.5 Impossibility for ORE in 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 Pass

and Shelat [2016], which shows how to remove constant graded encoding from obfuscation schemes.Without loss of generality, we can assume that the ORE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}^{\mathcal{G}}, \mathsf{Comp}^{\mathcal{G}})$ satisfies the following:

- $\mathsf{Gen}$ makes no queries to $\mathcal{G}$.
- $\mathsf{Enc}$ has the access of both labeling and zero-test query, while $\mathsf{Comp}$ only makes zero-test queries. This is because $\mathsf{Comp}$ 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 $\vec{h}_m$ 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 $(\vec{h}_{m_0}, \vec{h}_{m_1}, \vec{v})$. 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 $m$, $|\vec{h}_m| = |\vec{g}_m| = q$, 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 $\mathsf{Comp}^*$. In the key generation procedure, $\Pi^*$ additionally outputs the encryption of $i$ where $i \in [s] \cup [N-d+1, N]$.

After that, the encryption procedure runs $\mathsf{Enc}(k, m), \mathsf{Comp}(\mathsf{Enc}(k, m), \mathsf{Enc}(k, i))$ and $\mathsf{Comp}(\mathsf{Enc}(k, i), \mathsf{Enc}(k, j)), i, j \in [s] \cup [N-d+1, N]$. 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 $\vec{g}_1, \ldots, \vec{g}_s, \vec{g}_m, \vec{g}_{N-d+1}, \ldots, \vec{g}_N$ 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 $\mathsf{Comp}$ on the two $\Pi$-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 $\mathsf{Comp}$ only needs to learn the constraint space when restricted to $\vec{g}_{m_0}, \vec{g}_{m_1}$, 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 $\mathsf{Comp}^*$ simulates the oracle itself properly except with a small probability. We now prove the following theorem:

**Theorem 3.5.1** *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.*

**Proof:** In our proof, for simplicity we will assume all queries to the zero testing oracle are homogeneous (there is no constant term $v_0$); it is straightforward to extend our proof the full inhomogeneous setting. Let $\Pi_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0^{\mathcal{G}}, \mathsf{Comp}_0^{\mathcal{G}})$ be a statistical secure ORE in generic group model (we view $\mathsf{sk}$ is simply the randomness fed into $\mathsf{Gen}$, thus we do not have oracle access for $\mathsf{Gen}$). For convenience, we denote $\Pr[\Pi_0]$ as the lower bound on the correctness probability:

$$\Pr[\Pi_0] = \min_{m_0, m_1} \Pr[\mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}, C_0, C_1) = \mathsf{Comp}(m_0, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_0(1^\lambda)].$$

Similar to the random oracle case, we specify some parameters:

1. $\Pr[\Pi_0] \geq \frac{1}{2} + 2\rho$, where $\rho$ is noticeable,
2. $q, u = \mathsf{poly}(\lambda)$ by query efficiency, $\ell_1 = \frac{20q \cdot u^2}{\rho^2}, \ell_2 = \frac{42q \cdot \ell_1^2 \cdot u^2}{\rho^2}$,
3. $\mathsf{Enc}_0^{\mathcal{G}}(m)$ makes $q$ labeling queries to oracle when encrypting $m$ under $\mathsf{sk}$,
4. $\mathsf{Comp}_0^{\mathcal{G}}$ makes $u$ queries to the oracle, and let $S_{\mathsf{pk},m_0,m_1}$ be the set of the constraints in value $\vec{g}_{m_0}, \vec{g}_{m_1}$(with form of $p = (\vec{v}, \vec{h}_{m_0}, \vec{h}_{m_1})$) that it stores,
5. $s = u\ell_2 \cdot 2\ell_2 \cdot \frac{42u^2 \cdot \ell_1}{\rho^2}, s_i = i\ell_2 \cdot 2\ell_2 \cdot \frac{42u^2 \cdot \ell_1}{\rho^2}, s^* = \frac{s_1}{\ell_2}$,
6. $d = \frac{\ell_1 s}{\ell_2}, d_i = \frac{\ell_1 s_i}{\ell_2}, i \in [u]$,
7. $D = [s] \cup [N - d + 1, N]; D_i = [s_i] \cup [N - d_i + 1, N]$,
8. $T_i = [i\ell_2] \cup [N - i\ell_1 + 1, N]$.

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

- $\mathsf{Gen}(1^\lambda)$ runs $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0(1^\lambda)$, computes $C_i = \mathsf{Enc}_0^{\mathcal{G}}(\mathsf{sk}_0, i), i \in D$. And outputs $\mathsf{pk} = \mathsf{pk}_0, \mathsf{sk} = (\mathsf{sk}_0, \{C_i\})$
- $\mathsf{Enc}^{\mathcal{G}}(\mathsf{sk}, m)$ runs $C \leftarrow \mathsf{Enc}_0^{\mathcal{G}}(\mathsf{sk}_0, m)$ and $\mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}_0, C_i, C)$ and $\mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}_0, C_i, C_j)$ where$i, j \in D$, stores the set of constraints on $(\vec{g}_m, \vec{g}_i)_{i \in D}$ as $S_{\mathsf{sk},m}$ and outputs $C^* = (C, S_{\mathsf{sk},m})$;
- $\mathsf{Comp}(\mathsf{pk}, C_0^*, C_1^*)$ : let $C_0^* = (C_0, S_0), C_1^* = (C_1, S_1)$, runs $\mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}_0, C_0, C_1)$, except that when querying a linear zero test $p$ on unknown value $(\vec{g}_{m_0}, \vec{g}_{m_1})$ , it responds as follows:
  1. $p$ is a linear combination of constraints in $S_0 \cup S_1$, then returns "0";
  2. Otherwise, returns "1".

We note that in the comparison procedure of $\Pi^*$, we remove the oracle access, hence it remains to show that $\Pi^*$ is statistically secure and partially correct. It's trivial that $\Pi^*$ is statistically secure, due to the almost identical argument as in random oracle case, and in the following we prove $\Pi^*$ is partially correct.

**Lemma 3.5.2** $\Pr[\Pi^*] \geq \frac{1}{2} + \rho$.

We establish our proof by hybrid argument, and we define $u$ ORE schemes $\Pi_j = (\mathsf{Gen}_j, \mathsf{Enc}_j^{\mathcal{G}}, \mathsf{Comp}_j^{\mathcal{G}}), j \in [u]$ with message space $[s_j+1, N-d_j]$. There are two difference between $\Pi^*$ and $\Pi_j$: 1) numbers of tested ciphertexts that are generated; 2) $\Pi_j$ *only* uses the constraint set to answer the last $j$ queries (for the first $u-j$ queries, $\Pi_j$ answers as usual by accessing $\mathcal{G}$).

- $\mathsf{Gen}_j(1^\lambda)$ runs $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0(1^\lambda)$, computes $C_i = \mathsf{Enc}_0^{\mathcal{G}}(\mathsf{sk}_0, i), i \in D_j$ and outputs $\mathsf{pk} = \mathsf{pk}_0, \mathsf{sk} = (\mathsf{sk}_0, \{C_i\}_{i \in D_j})$;
- $\mathsf{Enc}_j^{\mathcal{G}}(\mathsf{sk}, m)$ runs $C \leftarrow \mathsf{Enc}_0^{\mathcal{G}}(\mathsf{sk}_0, m)$ and $\mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}_0, C_i, C), \mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}_0, C_i, C_k)$ where $i, k \in D_j$, stores the set of constraints on $(\vec{g}_m, \vec{g}_i)_{i \in D_j}$ as $S_{\mathsf{sk},m}$ and outputs $C^* = (C, S_{\mathsf{sk},m})$;
- $\mathsf{Comp}_j(\mathsf{pk}, C_0^*, C_1^*)$ : let $C_0^* = (C_0, S_0), C_1^* = (C_1, S_1)$, runs $\mathsf{Comp}_0^{\mathcal{G}}(\mathsf{pk}_0, C_0, C_1)$, except that when querying the oracle on zero test $p$, it does the following:
  1. If $p$ is one of the first $u - i$ zero test queries, make a query to $\mathcal{G}$ as usual,
  2. If $p$ is one of the last $i$ zero test queries and $p$ is a linear combination of the constraints stored in $S_0 \cup S_1$, then it returns "0",
  3. Otherwise, returns "1".

Similar to the proof in ROM, we here prove $\Pr[\Pi_1] \geq \frac{1}{2} + 2\rho - \frac{\rho}{u}$. According to the definition, we see that $\mathsf{Comp}_1$ works as the same as $\mathsf{Comp}_0$, except for the final query in which we test whether $p$ is a linear combination of the constraints provided with the ciphertext to answer the oracle query. We prove that the response made by $\Pi_1$ for $p$ does not significantly harm the ability of $\mathsf{Comp}_1$ to output the correct answer. To do so, we

introduce yet another sequence of $s^*$ ORE schemes $\Pi_{1,j} = (\mathsf{Gen}_{1,j}, \mathsf{Enc}^{\mathcal{G}}_{1,j}, \mathsf{Comp}^{\mathcal{G}}_{1,j}), j \in [s^*]$ on message space $[j\ell_2 + 1, N - j\ell_1]$. The only difference between $\Pi_{1,j}$ and $\Pi_1$ is the number of test ciphertexts that are generated.

- $\mathsf{Gen}_{1,j}(1^\lambda)$ runs $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0(1^\lambda)$, computes $C_i = \mathsf{Enc}^{\mathcal{G}}_0(\mathsf{sk}_0, i), i \in T_j$, and outputs $\mathsf{pk} = \mathsf{pk}_0, \mathsf{sk} = (\mathsf{sk}_0, \{C_i\}_{i \in T_j})$;
- $\mathsf{Enc}^{\mathcal{G}}_{1,j}(\mathsf{sk}, m)$ runs $C \leftarrow \mathsf{Enc}^{\mathcal{G}}_0(\mathsf{sk}_0, m)$ and $\mathsf{Comp}^{\mathcal{G}}_0(\mathsf{pk}_0, C_i, C), \mathsf{Comp}^{\mathcal{G}}_0(\mathsf{pk}_0, C_i, C_k)$ where $i, k \in T_j$, stores the set of constraints on $(\vec{g}_m, \vec{g}_i)_{i \in T_j}$ as $S^{(j)}_{\mathsf{sk},m}$ and outputs $C^* = (C, S_{\mathsf{sk},m})$;
- $\mathsf{Comp}^{\mathcal{G}}_{1,j}(\mathsf{pk}, C^*_0, C^*_1)$ : let $C^*_0 = (C_0, S_0), C^*_1 = (C_1, S_1)$, runs $\mathsf{Comp}^{\mathcal{G}}_0(\mathsf{pk}_0, C_0, C_1)$, except that when querying the oracle on with the zero test $p$, it does the following:
  1. If $p$ is one of the first $u - 1$ zero test queries, make a query to $\mathcal{G}$ as usual.
  2. If $p$ is the last test query and is linearly dependent of the constraints stored in $S_0 \cup S_1$, then it returns "0".
  3. Otherwise returns "1".

Therefore, it remains to prove the following claim.

**Claim 3.5.3** *If* $\Pr[\Pi_{1,j}] < \frac{1}{2} + 2\rho - \frac{\rho}{u}$, *then* $\Pr[\Pi_{1,j+1}] \geq \Pr[\Pi_{1,j}] + \frac{1}{s^*}$.

Assuming $\Pr[\Pi_{1,j}] < \frac{1}{2} + 2\rho - \frac{\rho}{u}$, there is a message pair $(m^*_0, m^*_1)$ minimizing the correctness probability; that is the comparison procedure on encryptions of $(m^*_0, m^*_1)$ outputs the correct answer with probability less than $\frac{1}{2} + 2\rho - \frac{\rho}{u}$. Due to the statistical security, we have that for any $(m_0, m_1)$, comparison must output the correct answer with probability at most $\frac{1}{2} + 2\rho - \frac{\rho}{u} + \mathsf{negl} < \frac{1}{2} + 2\rho - \frac{\rho}{2u}$.

Fix two message $m_0, m_1 \in [j\ell_2 + 1, N - j\ell_1]$. We let $S^{(j)} = S^{(j)}_{\mathsf{sk},m_0} \cup S^{(j)}_{\mathsf{sk},m_1}$ (the constraint set in $\mathsf{Enc}_{1,j}(m_b)$). Let $p$ be the final zero test made when comparing the encryptions of $m_0, m_1$. Define the event $\mathsf{Bad}^{(j)}$ where the following happen:

- $p$ is a constraint satisfied by $\vec{g}_{m_0}, \vec{g}_{m_1}$, but $p$ is linearly independent of the constraints stored $S^{(j)}$,

- $\mathsf{Comp}_0^{\mathcal{G}}$ outputs the correct answer on encryption of $m_0, m_1$,
- $\mathsf{Comp}_{1,j}^{\mathcal{G}}$ outputs the incorrect answer on encryption of $m_0, m_1$.

We consider four cases:

- $p$ is linearly dependent of the constraints of $S^{(j)}$. In this case, $\Pi_{1,j}$ answers the same as $\Pi_0$ since it knows $p$ is a valid constraint,
- $p$ is not satisfied by $\vec{g}_{m_0}, \vec{g}_{m_1}$. In this case, $p$ must be linearly independent of $S^{(j)}$, hence $\Pi_0, \Pi_{1,j}$ answer the same,
- $p$ is a constraint satisfied by $\vec{g}_{m_0}, \vec{g}_{m_1}$, and independent of $S^{(j)}$, but $\mathsf{Bad}^{(j)}$ does not occur. Here we must have $\mathsf{Comp}_0$ either outputs the incorrect answer, or $\mathsf{Comp}_{1,j}$ outputs the correct answer,
- $\mathsf{Bad}^{(j)}$ occurs. We know that by answering "1", $\mathsf{Comp}_{1,j}$ goes from outputting the correct answer to the incorrect one.

Similar to the random oracle setting, only the last case decreases the probability of correctness, therefore, $\Pr[\mathsf{Bad}^{(j)}] > \frac{\rho}{2u}$.

Next we split the message space into two parts $[j \cdot \ell_2 + 1, \frac{N}{2}]$ and $[\frac{N}{2} + 1, N - j \cdot \ell_1]$, and sample $w \leftarrow [j \cdot \ell_2 + 1, \frac{N}{2}], z_1 < \ldots < z_{\ell_1} \leftarrow [\frac{N}{2} + 1, N - j \cdot \ell_1]$. Let $\mathsf{Bad}_i^{(j)}$ be the event that "$\mathsf{Bad}^{(j)}$" occurs for $(w, z_i), i \in [\ell_1]$. Then we claim that $\Pr[\mathsf{Bad}_1^{(j)} \cap \mathsf{Bad}_i^{(j)}] > \frac{\rho^2}{10u^2}$, and prove it by contradiction. In fact, we note that unlike the case in ROM, the bad event here is detectable for unbounded adversary, then invoking the security we must have that for any $j, k \in [t]$ where $t = \frac{4u}{\rho} < \ell_1$,

$$\Pr[\mathsf{Bad}_j \cap \mathsf{Bad}_k] \leq \Pr[\mathsf{Bad}_1 \cap \mathsf{Bad}_i] + \mathsf{negl}$$

If assuming that $\Pr[\mathsf{Bad}_1 \cap \mathsf{Bad}_i] \leq \frac{\rho^2}{10u^2}$, then $\Pr[\mathsf{Bad}_j \cap \mathsf{Bad}_k] < \frac{\rho^2}{9u^2}$, which means

$$\begin{aligned}\Pr[\mathsf{Bad}_1 \cup \ldots \cup \mathsf{Bad}_t] &\geq \sum_{i=1}^{t} \Pr[\mathsf{Bad}_i] - \sum_{i<j\in[t]} \Pr[\mathsf{Bad}_i \cap \mathsf{Bad}_j] \\ &\geq \frac{\rho \cdot t}{2u} - \frac{t^2}{2}\frac{\rho^2}{9u^2} > 2 - 8/9 > 1\end{aligned}$$

a contradiction.

Now we define $t_i$ to be the indicator,

$$t_i = \begin{cases} 1 & \text{if} \quad \mathsf{Bad}_1 \cap \mathsf{Bad}_i = 1 \\ 0 & \text{Otherwise} \end{cases}$$

and it's apparent that

$$\Pr[t_i = 1] \geq \frac{\rho^2}{10u^2} \Rightarrow \mathbb{E}(\sum_{i=2}^{\ell_1} t_i) > (\ell_1 - 1) \cdot \frac{\rho^2}{10u^2} > 2q - \frac{\rho^2}{10u^2}.$$

Let $T$ be the event that $\sum_{i=2}^{\ell_1} t_i > q$, then

$$\Pr[T] \cdot \ell_1 + q \cdot (1 - \Pr[T]) \geq \mathbb{E}(\sum_{i=1}^{\ell} t_i) \Rightarrow \Pr[T] > \frac{\rho^2}{20u^2}.$$

We immediately observe that when $T$ occurs, there must be a constraint $p^*$ on ( $\vec{g}_{z_1}, \ldots, \vec{g}_{z_{\ell_1}}$) such that

$$p^* = a_1 p_1 + \ldots + a_{\ell_1} p_{\ell_1},$$

where $p_i$ is the last query in $\mathsf{Comp}(\mathsf{Enc}(w), \mathsf{Enc}(z_i))$(if $p_i$ is not a constraint, we set $a_i = 0$). Parallelly, we define a hybrid event $T^*$, and we say $T^*$ happens if $T$ happens and $a_1 \neq 0$. Invoking the security, it's trivial that

$$\Pr[T^*] \geq \frac{1}{\ell_1} \cdot \Pr[T] - \mathsf{negl} \geq \frac{\rho^2}{21u^2 \cdot \ell_1},$$

and in such a case, we can write $p_1$ as:

$$p_1 = \frac{1}{a_1}(a_2 p_2 + \ldots + a_{\ell_1} p_{\ell_1} - p^*)$$

where $p^*$ is a constraint on $(\vec{g}_{z_1}, \ldots, \vec{g}_{z_{\ell_1}})$. We observe that, if $p^*$ is known, then we can just move $z_2, \ldots, z_{\ell_1}$ to the right end, and applying exactly the same technique as in ROM, we complete the proof. Hence it's rest to show how to extract $p^*$.

We note that $p^*$ is a constraint on $(\vec{g}_{z_1}, \ldots, \vec{g}_{z_{\ell_1}})$, and the cardinality of the entire space is $q\ell_1$, hence one trivial way to extract $p^*$ is to recover the whole basis. To do so, we sample $\vec{w} = (w_1 < \ldots < w_{\ell_2}) \in [j \cdot \ell_2 + 1, \frac{N}{2}]$ and use every element in $\vec{w}$ to recover it. Now we define a new indicator $e_i$ such that $e_i = 1$ if $T^*$ occurs for $(w_i, \vec{z}), i \in [\ell_2]$. Based on $e_i$, we introduce a new event $G$ and say $G$ occurs if $\sum_{i=1}^{\ell_2} e_i > q\ell_1$,

$$\Pr[G] \cdot \ell_2 + q\ell_1 \cdot (1 - \Pr[G]) \geq \mathbb{E}(\sum_{i=1}^{\ell_2} g_i) > 2q\ell_1 \Rightarrow \Pr[G] \geq \frac{\rho^2}{42u^2\ell_1}.$$

It's obvious that when $G$ occurs, either the whole basis is recovered or there exists $w_i$ such that the corresponding $p^*$ can be extracted by $(w_1, \ldots, w_{i-1}, z_1, \ldots, z_{\ell_1})$, and in either case we are done. Here we define a new event, called $\mathsf{Collision}$, to characterize it; for any message sequence $(w_1 < \ldots < w_k < z_1 < \ldots < z_{\ell_1})$, we say $\mathsf{Collision}$ happens if,

- the last query $p$ in $\mathsf{Comp}(\mathsf{Enc}(w_k), \mathsf{Enc}(z_1))$ is a constraint in value $(\vec{g}_{w_k}, \vec{g}_{z_1})$, and $\mathsf{Bad}^{(j)}$ occurs for $(w_k, z_1)$,
- $p$ is a linear combination of the constraints collected in $\mathsf{Comp}(\mathsf{Enc}(w_i, z_j)), i \in [k], j \in [\ell_1], (i,j) \neq (k,1)$[7].

By definition, it's trivial that adding more samples would not decrease the probability of $\mathsf{Collision}$, namely, if $\mathsf{Collision}$ happens for $(w_1 < \ldots < w_k < z_1 < \ldots < z_{\ell_1})$, then it also happens for $(x_1 < \ldots < x_{k'}, w_1 < \ldots < w_k < z_1 < \ldots < z_{\ell_1})$ as long as $x_{k'} < w_k$.

Note that when $G$ occurs, there must exist a sub-sequence $(w_{i_1} < \ldots < w_{i_k}, \vec{z})$ such

[7] $(i,j) \neq (k,1)$ means $i \neq k \cap j \neq 1$

that Collision happens, which means for $(w_1 < w_2 \ldots < w_{i_k}, \vec{z})$ Collision also happens, so we have

$$\max_{i \in [\ell_2]} \Pr[\text{Collision for } (w_1 < \ldots < w_i < z_1 < \ldots < z_{\ell_1})] \geq \frac{1}{\ell_2} \cdot \Pr[G].$$

Then, due to security, we claim that for *arbitrary* $(w_1 < \ldots < w_{\ell_2} < z_1 < \ldots < z_{\ell_1}) \in [j \cdot \ell_2 + 1, N - j \cdot \ell_1]$

$$\Pr[\text{Collision}] \geq \frac{1}{\ell_2} \cdot \Pr[G] - \text{negl} \geq \frac{1}{2\ell_2} \cdot \Pr[G] = \frac{1}{s^*}.$$

Specifically, let $w_i = j\ell_2 + i, i \in [\ell_2 - 1]$,$z_i = N - j\ell_1 - i, i \in [\ell_1 - 1]$, we see that for any $(w_{\ell_2}, z_1)$, if we move to $\Pi_{1,j+1}$, $(w_1, \ldots, w_{\ell_2 - 1}, z_2, \ldots, z_{\ell_1})$ are included in the test queries for the scheme, hence,

$$\Pr[\Pi_{1,j+1}] \geq \Pr[\Pi_{1,j}] + \frac{1}{s^*} \Rightarrow \Pr[\Pi_1] \geq \Pr[\Pi_0] - \frac{\rho}{u}.$$

Now we have shown that $\Pr[\Pi_1] \geq \Pr[\Pi_0] - \frac{\rho}{u}$. This handles the case of $\Pi_1$. 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 $\Pi_j$, give $\Pr[\Pi_{j+1}] \geq \Pr[\Pi_j] - \frac{\rho}{u}$ Combing together, we get

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

which establishes the entire proof. ∎

# Chapter 4

# Parameter-Hiding Order Revealing Encryption

According to last chapter, we note that somewhat inefficiency is inevitable for ideal ORE. To "possibly" make the scheme practical, one direction is to, somehow, relax the security notion. In this chapter, we present our construction — *parameter-hiding ORE*. First, we give the formal definition of our new security notion, parameter-hiding, then we build a scheme, only based on bilinear map, that achieves our notion. In our construction, we propose a new primitive, called property-preserving hash(PPH), and show how to build a parameter-hiding ORE from it, then we illustrate how to build a PPH from bilinear map, under some computational assumptions.

## 4.1 Technical Overview

As a starting point, we consider the CLWW leakage Chenette et al. [2016], which reveals the position of the most significant differing bit between any two plaintexts. This is quite a lot of information: for example, it can be used to get rough bounds on the difference between two plaintexts. Thus, CLWW cannot be parameter hiding, since the scaling term is not hidden. However, CLWW will be a useful starting point, as it will allow us to construct *shift-hiding* ORE, where we only care about hiding the shift term. To help illustrate our approach, we will therefore first describe an equivalent formulation of CLWW leakage, which we will then explain how to extend to get full parameter-hiding ORE.

### 4.1.1 An alternative view of CLWW leakage

Consider the plaintext space $\{0, 1, 2, \ldots, 2^\ell - 1\}$. We will think of the plaintexts as leaves in a full binary tree of depth $\ell$. In this tree, the position of the most significant

differing bit between two plaintexts corresponds to the depth of their nearest ancestor. The leakage of CLWW can therefore can be seen as revealing the tree consisting of all given plaintexts, their ancestors in the tree up to the lowest common ancestor, and the order of the leaves, with all other information removed. See Figure 4.1 for an illustration.

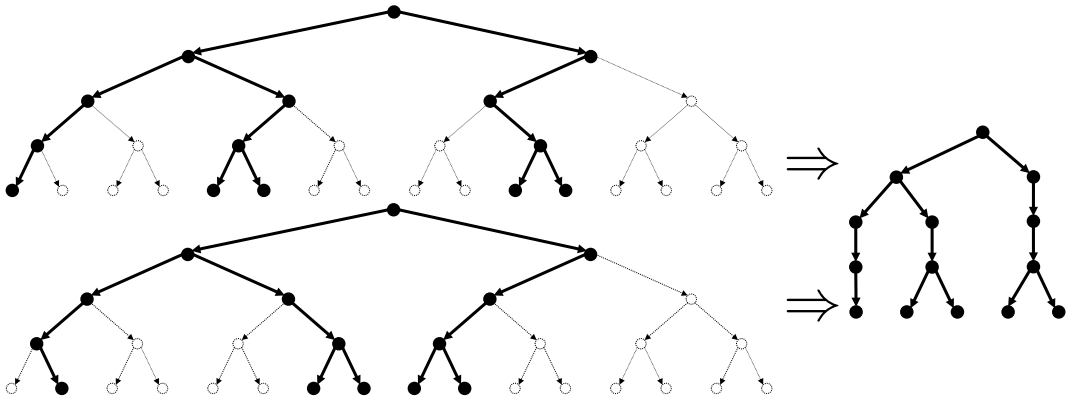

Figure 4.1: CLWW Leakage.

Now, suppose all plaintext elements are in the range $[0, 2^i)$ for some $i$. This means they all belong in the same subtree at height $i$; in particular, the CLWW leakage will only have depth at most $i$. Now, suppose we add a multiple of $2^i$ to every plaintext. This will simply shift all the plaintexts to being in a different subtree, but otherwise keep the same structure. Therefore, the CLWW leakage will remain the same.

Therefore, while CLWW is not shift hiding, it is *shift periodic*. In particular, if imagine a distribution $D$ whose support is on $[0, 2^i)$, and consider shifting $D$ by $\beta$. Consider an adversary $A$, which is given the CLWW leakage from $q$ plaintexts sampled from the shifted $D$, and outputs a bit. If we plot the probability $p(\beta)$ that $A$ outputs 1 as a function of $\beta$, we will see that the function is periodic with period $2^i$.

**Shift-Hiding ORE/OPE.** With this periodicity, it is simple to construct a scheme that is shift hiding. To get a shift-hiding scheme for message space $[0, 2^\ell)$, we instantiate CLWW with message space $[0, 2^{\ell+1})$. We also include as part of the secret key a random shift $\gamma$ chosen uniformly in $[0, 2^\ell)$. We then encrypt a message $m$ as $\mathsf{Enc}(m+\gamma)$. Adding

a random shift can be seen as convolving the signal $p(\beta)$ with the rectangular function

$$q(\beta) = \begin{cases} 2^{-\ell} & \text{if } \beta \in [0, 2^{\ell}) \\ 0 & \text{otherwise} \end{cases}$$

Since the rectangular function's support matches the period of $p$, the result is that the convolved signal $\hat{p}$ is *constant*. In other words, the adversary always has the same output distribution, regardless of the shift $\beta$. Thus, we achieve shift hiding.

When the comparison algorithm of an ORE scheme is simple integer comparison, we say the scheme is an *order-preserving encryption* (OPE) scheme. OPE is preferable because it can be used without any modification to a database server. We recall that CLWW can be made into an OPE scheme — where ciphertexts are integers and comparision is integer comparison — while maintaining the CLWW leakage profile. Our conversion to shift-hiding preserves the OPE property, so we similarly achieve a shift-hiding OPE scheme.

**Scale-Hiding ORE/OPE.** We note that we can also turn any shift-hiding ORE into a scale-hiding ORE. Simply take the logarithm of the input before encrypting; now multiplying by a constant corresponds to shifting by a constant. Of course, taking the logarithm will result in non-integers; this can easily be fixed by rounding to the appropriate level of precision (enough precision to guarantee no collisions over the domain) and scaling up to make the plaintexts integral. Similarly, we can also obtain scale-hiding OPE if we start with an OPE scheme.

**Impossibility of parameter-hiding OPE.** One may hope to achieve both shift-hiding and scale-hiding by some combination of the two above schemes. For example, since order *preserving* encryption schemes can be composed, one can imagine composing a shift-hiding scheme with a scale-hiding scheme. Interestingly, this does not give a parameter-hiding scheme. The reason is that shifts/scalings of the plaintext do not correspond to shifts/scalings of the ciphertexts. Therefore, while the outer OPE may provide, say, shift-hiding for its inputs, this will not translate to shift-hiding of the inner

OPE's inputs.

Nonetheless, one may hope that tweaks to the above may give a scheme that is simultaneously scale and shift hiding. Perhaps surprisingly, we show that this is actually impossible. Namely, we show that *OPE cannot possibly be parameter-hiding.* We prove this in section 4.9. We first provide a simple proof of the impossibility of ideal security for OPE, re-proving Boldyreva et al. [2009], Chenette et al. [2016] but in a much simpler way. Our proof is flexible to the exact choice of plaintexts, and we then show how to extend it to parameter-hiding OPE, even when the adversary sees just two ciphertexts.

This impossibility shows that strategies leveraging CLWW leakage are unlikely to yield parameter-hiding ORE schemes. Interestingly, all ORE schemes we are aware of that can be constructed from symmetric crypto can also be made into OPE schemes. Thus, this suggests we need stronger tools than those used by previous efficient schemes.

### 4.1.2 Parameter Hiding via Smoothed CLWW Leakage

Motivated by the above, we must seek a different leakage profile if we are to have any hope of achieving parameter-hiding ORE. We therefore first describe a "dream" leakage that will allow us to perform similar tricks as in the shift hiding case in order to achieve both scale and shift hiding simultaneously. Our dream leakage will be a "smoothed" CLWW leakage, where all nodes of degree exactly 2 are replaced with an edge between the two neighbors. In other words, the dream leakage is the smallest graph that is "homeomorphic" to the CLWW leakage. See Figure 4.2 for an illustration.

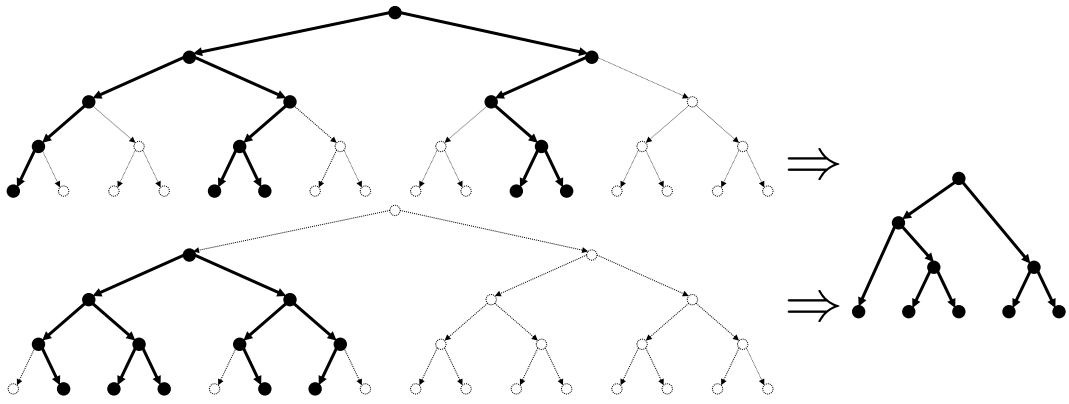

Figure 4.2: Smoothed CLWW Leakage.

Our key observation is that this smoothed CLWW leakage now exhibits additional periodicity. Namely, if we multiply every plaintext by 2, every edge in the bottom layer of the CLWW leakage will get subdivided into a path of length 2, but smoothing out the leakage will result in the same exact graph. This means that smoothed CLWW leakage is periodic in the *log* domain.

In particular, if imagine a distribution $D$ whose support is on $[0, 2^i)$, and consider multiplying by $\alpha$. Consider an adversary $A$, which is given the smoothed CLWW leakage from $q$ plaintexts sampled from a scaled $D$, and outputs a bit. If we plot the probability $p(\log_2 \alpha)$ that $A$ outputs 1 as a function of $\alpha$, we will see that the function is periodic with period 1.

Therefore, we can perform a similar trick as above. Namely, we convolve $p$ with the uniform distribution over the period of $p$ in the log domain. We accomplish this by including a random scalar $\alpha$ as part of the secret key, and multiplying by $\alpha$ before encrypting. However, this time several things are different:

- Since we are working in the log domain, the logarithm of the random scalar $\alpha$ has to be uniform. In other words, $\alpha$ is log-uniform
- Since we are working over integers instead of real numbers, many issues arise.
  - First, $\alpha$ needs to be an integer to guarantee that the scaled plaintexts are still integers. This means we cannot choose $\alpha$ at log-uniformly over a single log period, since then $\alpha$ only has support on $\{1, 2\}$. Instead, we need to choose $\alpha$ log-uniformly over a sufficiently large multiple of the period that $\alpha$ approximates the continuous log-uniform distribution sufficiently well.
  - Second, unlike the shift case, sampling at random from $D$ and then scaling is not the same as sampling from a scaled version of $D$, since the rounding step does not commute with scaling. For example, for concreteness consider the normal distribution. If we sample from a normal distribution (and round) and then scale, the resulting plaintexts will all be multiples of $\alpha$. However, if we sample directly from a scaled normal distribution (and then round),

the support of the distribution will include integers which are not multiples of $\alpha$.

To remedy this issue, we observe that if the plaintexts are sampled from a wide enough distribution, their differing bits will not be amongst the lowest significant bits. Hence, the leakage will actually be independent of the lower order bits. For example, this means that while the rounding does not commute with the scaling, the leakage actually does not depend on the order in which the two operations are carried out.

– The above arguments can be made to work for, say, the normal distribution. However, we would like to have a proof that works for any distribution. Unfortunately, for distributions that oscillate rapidly, we may run into trouble with the above arguments, since rounding such distributions can cause odd behaviors at all scales. This problem is actually unavoidable, as quickly oscillating distributions may have actually have low min-entropy even at large scales. Therefore, we must restrict to "smooth" functions that have a bounded derivative.

Using a careful analysis, we are able to show for smooth distributions that we achieve the desired scale hiding.

- Finally, we want to have a scheme that is both scale and shift hiding. This is slightly non-trivial, since once we introduce, say, a random shift, we have modified the leakage of the scheme, and cannot directly appeal to the arguments above to obtain scale hiding as well. Instead, we distill a set of specific requirements on the leakage that will work for both shift hiding and scale hiding. We show that our shift hiding scheme above satisfies the requirements needed in order for us to introduce a random scale and additionally prove scale hiding.

### 4.1.3 Achieving Smoothed CLWW Leakage.

Next we turn to actually constructing ORE with smoothed CLWW leakage. Of course, ideal ORE has better than (smoothed) CLWW leakage, so we can construct such ORE

based on multilinear maps. However, we want a construction that uses standard tools.

We therefore provide a new construction of ORE using pairings that achieves smoothed CLWW leakage. We believe this construction is of interest on its own, as it is achieves the to-date smallest leakage of any non-multilinear map based scheme.

**CLWW ORE and how to reduce its leakage.** Our construction builds on the ideas of CLWW, so we first briefly recall the ORE scheme of CLWW. In their (basic) scheme, the encryption key is just a PRF key $K$. To encrypt a plaintext $x \in \{0,1\}^n$, for each prefix $p_i = x[1,\ldots,i]$, the scheme computes

$$y_i = \mathsf{PRF}_K(p_i) + x_{i+1},$$

where $x_{i+1}$ is the $(i+1)$-st bit of $x$, and the output of $\mathsf{PRF} \in \{0,1\}^\lambda$ is treated as an integer (we will take $\lambda$ to be the security parameter). The ORE ciphertext is then $(y_1 \ldots, y_n)$. To compare two ciphertexts $(y_1 \ldots, y_n)$ and $(y'_1 \ldots, y'_n)$, one finds the smallest index $i$ such that $y_i \neq y'_i$, and outputs 1 if $y'_i - y_i = 1$. This naturally reveals the index of the bit where the plaintexts differ.

Our approach to reducing the leakage is to attempt to hide the index $i$ where the plaintexts differ. As a naive attempt at this, first consider what happens if we modify the scheme to simply randomly permute the outputs $(y_1 \ldots, y_n)$ (with a fresh permutation chosen for each encryption). We can still compare ciphertexts by appropriately modifying the comparison algorithm: now given $c = (y_1 \ldots, y_n)$ and $c' = (y'_1 \ldots, y'_n)$ (permuted as above), it will look for indices $i, j$ such that either $y'_i - y_j = 1$, in which case it outputs 1, or $y_j - y'_i = 1$, in which case it outputs 0. (If we choose the output length of the PRF to be long enough then this check will be correct with overwhelming probability.)

This modification, however, does not actually reduce leakage: an adversary can still determine the most significant differing bit by counting how many elements $c$ and $c'$ have in common.

We can however recover this approach by preventing an adversary from detecting

how many elements $c$ and $c'$ have in common. To do so, we employ a new notion of *property-preserving hashing* (PPH) we introduce. Intuitively, a PPH is a randomized hashing scheme that is designed to publicly reveal a particular predicate $P$ on pairs of inputs.

PPH can be seen as the hashing (meaning, no decryption) analogue of the notion of property-preserving encryption, a generalization of order-revealing encryption to arbitrary properties due to Pandey and Rouselakis Pandey and Rouselakis [2012]. (This can also be seen as a symmetric-key version of the notion of "relational hash" due to Mandal and Roy Mandal and Roy [2015].)

Specifically, we construct and employ a PPH for the property

$$P_1(x, x') = \begin{cases} 1 & \text{if } x = x' + 1, \\ 0 & \text{otherwise.} \end{cases}$$

(Here $x, x'$ are not plaintexts of the ORE scheme, think of them as other inputs determined below.) Security requires that this is *all* that is leaked; in particular, input equality is *not* leaked by the hash values (which requires a randomized hashing algorithm).

Now, the idea is to modify the scheme to include a key $K_H$ for such a PPH $\mathcal{H}$, and the encryption algorithm to not only randomly permute the $y_i$'s but hash them as well, i.e., output $(h_1, \ldots, h_n)$ where $h_i = \mathcal{H}_{K_H}(y_i)$ for the permuted $y_i$'s.[1] The comparison algorithm can again be modified appropriately, namely to not to check if $y'_i - y_j = 1$ but rather if their $h'_i$ and $h'_j$ hash values satisfy $P_1$ via the PPH (and similarly for the check $y_j - y'_i = 1$).

For any two messages, the resulting ORE scheme is actually ideal: it only reveals the order of the underlying plaintexts, but nothing else. However, for three messages $m, m', m''$ we see that some additional information is leaked. Namely, if we find that $y'_i - y_j = 1$ $y''_k - y_j = 1$, then we know that $y'_j = y''_k$. We choose the range of the PRF

[1] A minor issue here is that we now lose decryptability for the resulting ORE scheme; however, this can easily be added back in a generic way by also encrypting the plaintext separately under a semantically secure scheme.

large enough so that this can only happen if $y'_j$ and $y'_k$ are both $\mathsf{PRF}_K(p_\ell) + x_{\ell+1}$ for the same prefix $p_\ell$ and same bit $x_{\ell+1}$, and $y'_j$ corresponds to the most significant bit where $m'$ differs from $m$, $y''_k$ corresponds to the most significant bit where $m''$ differs from $m$, and moreover these positions are the same. Therefore, the adversary learns whether these most-significant differing bits are the same. It is straightforward to show that this leakage is exactly equivalent to the smoothed CLWW leakage we need. Proving this ORE scheme secure wrt. this leakage based on an achievable notion of security for the PPH turns out to be technically challenging. Nevertheless, we manage to prove it "non-adaptively secure," meaning the adversary is required to non-adaptively choose the dataset, which is realistic for a passive adversary in the outsourced database setting.

**Property-preserving hash from bilinear maps.** Next we turn to constructing a property-preserving hash (PPH) for the property $P_1(x, x') = x = x' + 1$. For this, we adapt techniques from perfectly one-way hash functions Canetti [1997], Mandal and Roy [2015] to the symmetric-key setting and use asymmetric bilinear groups. Roughly, in our construction the key for the hash function is a key $K$ for a pseudorandom function $\mathsf{PRF}$ and, letting $e\colon G_1 \times G_2 \to G_T$ be an asymmetric bilinear map on prime order cyclic groups $G_1, G_2$ with generators $g_1, g_2$, the hash of $x$ is

$$\mathcal{H}_K(x) = (g_1^{r_1}, g_1^{r_1 \mathsf{PRF}_K(x)}, g_2^{r_2}, g_2^{r_2 \mathsf{PRF}_K(x+1)}),$$

for fresh random $r_1, r_2 \in \mathbb{Z}_p$. (Thus, the PRF is also pushed to our PPH construction and can be dropped from from the higher-level ORE scheme when our hash function is plugged-in.) The bilinear map allows testing whether $P_1(x, x')$ from $\mathcal{H}_K(x), \mathcal{H}_K(x')$, and intuitively our use of asymmetric bilinear groups prevents testing other relations such as equality (formally we use the XSDH assumption). We prove the construction secure under an indistinguishability-based notion in which the adversary has to distinguish between the hash of a random challenge $x^*$ and a random hash value, and can query for hash values of inputs $x$ of its choice as long as $P_1(x, x^*)$ and $P_1(x^*, x)$ are

both 0. Despite being restricted,[2], this notion suffices in our ORE scheme above.

When our PPH is plugged-in to our ORE scheme, ciphertexts consist of $4n$ group elements, and order comparison requires $n(n-1)$ pairing computations on average. We also note that CLWW gave an improved version of their scheme where ciphertexts are size $O(n)$ rather than $O(n\lambda)$ for security parameter $\lambda$, however, we have reason to believe this may be difficult for schemes with our improved leakage profile, see below.

Piecing everything together, we obtain a parameter-hiding ORE from bilinear maps. We note that, as parameter-hiding OPE is impossible, we achieve the first construction of ORE without multilinear maps secure with a security notion that is impossible for OPE.

**Generalizing our ORE scheme.** In our work, we also show several extensions to our smoothed CLWW ORE scheme. In one direction, we show an improved leakage by considering blocks of bits at a time(encrypting message block by block, rather than bit by bit). And interestingly, we show that if the block size is only 2, then we improve security and efficiency simultaneously, while for larger block, the leakage continues to reduce but efficiency compared to the basic scheme (in terms of both ciphertext size and pairings required for comparison) decreases.

On the other hand, we also show how to improve efficiency while sacrificing some security. Interestingly, we are able to show a more efficient version of the scheme than above(only need $O(n)$ pairings for each comparison), that is still sufficient for achieving parameter-hiding ORE using our conversion.

In addition, we also show how our ORE scheme easily gives a *left/right ORE* as defined by Lewi and Wu [2016] that also improves on their leakage. In left/right ORE, ciphertexts can be generated in either the left mode or right mode, and the comparison algorithm only compares a left and a right ciphertext. Security requires that no information is leaked amongst left and right ciphertexts in isolation.

---

[2]More generally, following Pandey and Rouselakis [2012] one could allow the adversary to choose two challenge inputs and make queries that do not allow it to trivially distinguish them, but we are unable to prove our construction secure under this stronger notion.

### 4.1.4 Discussion and Perspective

The original OPE scheme of Boldyreva et al. [2009] leaks "whatever a random order-preserving function leaks." Unfortunately, this notion does not say anything about what such leakage actually looks like. The situation has been improved in recent works on ORE/OPE such as CLWW, LW Lewi and Wu [2016] and JP Joye and Passelègue [2016], which define a precise "leakage profile" for their schemes. However, such leakage profiles are still of limited use, since they do not obviously say anything about the actual privacy of the underlying data. The same situation also

We instead study ORE with a well-defined privacy notion for the underlying plaintexts. A key part of our results is showing how to translate sufficiently strong leakage profiles into such privacy notions. Nonetheless, we do not claim that our new ORE scheme is safe to use in general higher-level protocols. We only claim security as long all that is sensitive is the scale and shift of the underlying plaintext distributions. If, for example, if the shape of the distribution is highly sensitive, or if there are correlations to other data available to the attacker, our notion is insufficient.

However, our construction provably has better leakage than existing efficient schemes, and it at least shows some meaningful security for specific situations. Moreover we suspect that the scheme can be shown to be useful in many other settings by extending our techniques.

## 4.2 Definition for Parameter-Hiding

In this section, we present the formal definition of the notion for parameter-hiding. As we showed in the introduction, parameter-hiding is motivated by a stronger notion called distribution-hiding, we would first illustrate what is distribution-hiding, then turn to parameter-hiding.

In the notions, we are considering the privacy of the underlying *distribution* of data records, rather than the individual data records, and we assume that all database entries are independently and identically distributed according to some distribution D[3].

[3]By D, here we mean a sampling algorithm, such that the outputs of this algorithm obey the

Game $\mathsf{DH}_{\Pi,q}(\mathcal{A},\lambda)$:
$(\mathsf{sk},\mathsf{ck}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, M)$; $D_0, D_1 \leftarrow \mathcal{A}(1^\lambda, \mathsf{ck})$ s.t. $H_\infty(D_b) \geq \omega(\log\lambda)$
$b \xleftarrow{\$} \{0,1\}, \vec{m} \xleftarrow{\$} D_b$ *s.t.* $|m| = q; \max D_b \leq M; \vec{c} = \mathsf{Enc}(\mathsf{sk}, \vec{m})$
$b' = \mathcal{A}(\mathsf{ck}, \vec{c})$; Return $(b \stackrel{?}{=} b')$

Table 4.1: Games $\mathsf{DH}_{\Pi,q}(\mathcal{A},\lambda)$.

Here we define distribution-hiding in Figure 4.1. Intuitively, in the interactive game, after receiving the public parameter and comparison key, adversary $\mathcal{A}$ picks two distributions $D_0, D_1$ and sends to challenger $\mathcal{C}$, $\mathcal{C}$ then flips a coin $b$, samples a sequence of entries from $D_b$, and sends back the encrypted entries. Eventually $\mathcal{A}$ outputs a bit, and we say adversary wins if it guesses $b$ correctly. We note that if either of $D_b$ has low min-entropy, it is possible for an adversary to estimate the min-entropy by looking for collisions in its ciphertexts. Therefore, we must restrict $D_b$ to have high min-entropy.

**Definition 4.2.1 (Distribution-Hiding for ORE)** *For an ORE scheme* $\Pi$*, an adversary* $\mathcal{A}$*, function* $q = q(\lambda)$ *we define the games* $\mathsf{DH}_{\Pi,q}(\mathcal{A},\lambda)$ *in Figure 4.1. The advantage of* $\mathcal{A}$ *is defined as* $\mathbf{Adv}^{\mathsf{DH}}_{\Pi,q}(\mathcal{A},\lambda) = |\Pr[\mathsf{DH}_{\Pi,q}(\mathcal{A},\lambda) - \frac{1}{2}]|$*. We say that* $\Pi$ *is distribution-hiding if for every efficient adversary* $\mathcal{A}$*, and any polynomial* $q = \mathsf{poly}(\lambda)$*,* $\mathbf{Adv}^{\mathsf{DH}}_{\Pi,q}(\mathcal{A},\lambda)$ *is a negligible function.*

We immediately observe that ideal ORE achieves distribution-hiding, while for other known leakier ORE schemes, it's seems unfeasible to achieve this privacy guarantee. However, in many settings, the general shape of the distribution is often known (that is, if the distribution is normal, uniform, Laplace, etc), and it is reasonable to allow the overall shape to be reveal but hide its mean and/or variance completely, subject to certain restrictions. Before formalize these notion, we firstly introduce some notations.

For a continuous random variable $X$, where $D$ is $X$'s distribution, we abuse notation $p_D(x) = p_X(x)$. Now we introduce three alternative distributions: $D^{\delta}_{\mathsf{scale}}, D^{\ell}_{\mathsf{shift}}, D^{\delta,\ell}_{\mathsf{aff}}$ with parameter $\delta, \ell$, where the corresponding probability density function is defined as:

$$p_{D_{\mathsf{scale}}} = \frac{p_D(\frac{x}{\delta})}{\delta};\ p_{D_{\mathsf{shift}}}(x) = p_D(x-\ell);\ p_{D_{\mathsf{aff}}} = \frac{p_D(\frac{x-\ell}{\delta})}{\delta}.$$

distribution D, for ease we denote $\max D$ as the maximum item in $D$'s support.

Game $(\gamma, D)$-$\mathsf{para\text{-}hid}_{\Pi,q}(\mathcal{A}, \lambda)$:

$(\mathsf{sk}, \mathsf{ck}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, M)$; $\delta_0, \ell_0, \delta_1, \ell_1 \leftarrow \mathcal{A}(\mathsf{ck}, D)$
If $\delta_0 < \gamma$ or $\delta_1 < \gamma$, output a random bit and abort,
else, $b \xleftarrow{\$} \{0,1\}, \vec{m} \xleftarrow{\$} \lfloor D_{\mathsf{aff}}^{\delta_b,\ell_b} \rceil, s.t. |m| = q; \max \lfloor D_{\mathsf{aff}}^{\delta_b,\ell_b} \rceil \leq M; \vec{c} = \mathsf{Enc}(\mathsf{sk}, \vec{m})$
$b' = \mathcal{A}(\mathsf{ck}, \vec{c})$ Return $(b \stackrel{?}{=} b')$

Table 4.2: Games $\mathsf{para\text{-}hid}_{\Pi,q}(\mathcal{A}, \lambda)$.

In other words, $D_{\mathsf{scale}}^{\delta}$ scales the shape of $D$ by a factor of $\delta$; $D_{\mathsf{shift}}$ shifts $D$ by $\ell$ and $D_{\mathsf{aff}}$ does both.

**Rounded distribution.** As our plaintexts are integers, we need map real number to its rounded integer, namely $x \to \lfloor x \rceil$. More precisely, let $D$ be a distribution over real numbers between $\alpha$ and $\beta$; we induce a rounded distribution $R_D^{\alpha,\beta}$ on $[\lceil \alpha \rceil, \lfloor \beta \rfloor]$which samples from $D$ and then rounds. Its probability density function is:

$$p_{R_D^{\alpha,\beta}}(k) = \begin{cases} \frac{\int_\alpha^{\lceil \alpha \rceil + 1/2} p_D(x)dx}{\int_\alpha^\beta p_D(x)dx} & k = \alpha, \\ \frac{\int_{k-1/2}^{k+1/2} p_D(x)dx}{\int_\alpha^\beta p_D(x)dx} & k \in [\lceil \alpha + 1 \rceil, \lfloor \beta - 1 \rfloor], \\ \frac{\int_{\lfloor \beta \rfloor - 1/2}^{\beta} p_D(x)dx}{\int_\alpha^\beta p_D(x)dx} & k = \beta, \\ 0 & \textit{Otherwise.} \end{cases}$$

In the case of $D_{\mathsf{scale}}^{\delta}$, $D_{\mathsf{shift}}^{\ell}$, or $D_{\mathsf{aff}}^{\delta,\ell}$, we will use the notation $\lfloor D_{\mathsf{scale}}^{\delta} \rceil$, $\lfloor D_{\mathsf{shift}}^{\ell} \rceil$, and $\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil$ to denote the respective rounded distributions.

Now, we present the notion "$(\gamma, D)$-parameter-hiding" ORE, referring to the game defined in Figure 4.2. Here, $D$ is a distribution over $[0, 1]$, which represents the description of the known shape of the distribution of plaintexts. $\gamma$ is a lower-bound on the scaling that is allowed. Then key generation is run and adversary is given the public parameter, $(\gamma, D)$, and the comparison key. Then, the adversary $\mathcal{A}$ sends two pairs of parameters $(\delta_0, \ell_0), (\delta_1, \ell_1)$ to challenger $\mathcal{C}$. Next, $\mathcal{C}$ flips a coin $b$, checks whether the parameter is proper($\mathbf{1}(\delta_0 \geq \gamma \cap \delta_1 \geq \gamma)$ ), then samples a sequence of data entries from the rounded distribution $\lfloor D_{\mathsf{aff}}^{\delta_b,\ell_b} \rceil$ and sends back encrypted data. Eventually $\mathcal{A}$ outputs a bit, and we say adversary wins if it guesses $b$ correctly.

**Definition 4.2.2 ($(\gamma, D)$-parameter hiding for ORE)** *For an ORE scheme $\Pi$, an adversary $\mathcal{A}$, a distribution $D$, and function $q = q(\lambda)$, we define the interactive game $(\gamma, D)$-$\mathsf{para\text{-}hid}_{\Pi,q}(\mathcal{A}, \lambda)$ in Figure 4.2. The advantage of $\mathcal{A}$ is defined as*

$$\mathbf{Adv}^{\mathsf{para\text{-}hid}}_{\Pi,q,\gamma,D}(\mathcal{A}, \lambda) = |\Pr[(\gamma, D)\text{-}\mathsf{para\text{-}hid}_{\Pi,q}(\mathcal{A}, \lambda) - \frac{1}{2}]|$$

*We say that $\Pi$ is $(\gamma, D)$-parameter hiding if for every efficient adversary $\mathcal{A}$ and polynomial $q$ $\mathbf{Adv}^{\mathsf{para\text{-}hid}}_{\Pi,q,\gamma,D}(\mathcal{A}, \lambda)$ is a negligible function.*

Similarly, we define $(\gamma, D)$-scale hiding and $(\gamma, D)$-shift hiding with little change as above. More precisely, in the game of $(\gamma, D)$-scale hiding, we add the restriction $\ell_0 = \ell_1 = 0$ and in the game of $(\gamma, D)$-shift hiding, we add the restriction $\delta_0 = \delta_1$. Due to the space limit, we skip the formal definitions here.

We note that these three notions are distribution dependent, and we would like they work for any distribution. Unfortunately, quickly oscillating distributions do not fit into our case, as they may have actually low min-entropy for their discretized distributions on integers, even at large scales. Hence, we place additional restrictions. We place the following restriction, which is sufficient, but potentially stronger than necessary:

**$(\eta, \mu)$-smooth distribution.** We let $D$ be a distribution where its support mainly on $[0, 1]$ ($\Pr[x \notin [0, 1] : x \leftarrow D] \leq \mathsf{negl}(\lambda)$), we denote $p'_D(x)$ as its derivative, and we say that $D$ is $(\eta, \mu)$-smooth if 1) $\forall x \in [0, 1], p_D(x) \leq \eta$; 2) $|p'_D(x)| \leq \eta$ for all $x \in [0, 1]$ except for $\mu$ points.

**Definition 4.2.3 ($(\gamma, \eta, \mu)$-parameter hiding for ORE)** *For an ORE scheme $\Pi$, we say $\Pi$ is $(\gamma, \eta, \mu)$-parameter hiding if for every efficient adversary $\mathcal{A}$, polynomial $q$, and any $(\eta, \mu)$-smooth distribution $D$, $\mathbf{Adv}^{\mathsf{para\text{-}hid}}_{\Pi,q,\gamma,D}(\mathcal{A}, \lambda)$ is a negligible function.*

## 4.3 Parameter Hiding ORE

In this section, we will assume we are given an ORE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ with a "smoothed" version of CLWW leakage, defined below. Later, in Section 4.4, we will show how to instantiate such a scheme from bilinear maps.

We show how to convert a scheme with smoothed CLWW leakage into a parameter-hiding ORE scheme by simply composing with a linear function: namely, for any plaintext $m$, the ciphertext has form $\mathsf{Enc}(\alpha m+\beta)$, where $\alpha, \beta$ are the same across all messages and are sampled as part of the secret key. Intuitively, $\alpha$ helps to hide the scale parameter and $\beta$ hides the shift. We need to be careful about the distributions of $\alpha$ and $\beta$; $\alpha$ needs to be drawn from a "discrete log uniform" distribution of appropriate domain, and $\beta$ needs to be chosen from a uniform distribution of appropriate domain.

The discrete log uniform distribution $D$ on $[A, B]$ ($\mathsf{logU}(A, B)$) has probability density function:

$$p_D(k) = \begin{cases} \frac{1/k}{\sum_{i=A}^{B} 1/i} & i \in [A, B] \\ 0 & \textit{Otherwise} \end{cases}$$

We say a leakage function $\mathcal{L}$ is smoothed CLWW if:

1. For any two plaintext sequences $\vec{m}_0, \vec{m}_1$, if $\mathcal{L}_{\mathsf{clww}}(\vec{m}_0) = \mathcal{L}_{\mathsf{clww}}(\vec{m}_1)$, then $\mathcal{L}(\vec{m}_0) = \mathcal{L}(\vec{m}_1)$ (in other words, it leaks no more information that CLWW);

2. For any plaintext sequence $\vec{m}$, $\mathcal{L}(\vec{m}) = \mathcal{L}(2\vec{m})$

### 4.3.1 Description of our scheme

In this part, we give the formal description of parameter-hiding ORE. To simplify our exposition, we first specify some parameters:

- $M = 2^{\mathsf{poly}(\lambda)}, \gamma = 2^{\omega(\log \lambda)}, \eta, \mu \leq O(1), q = \mathsf{poly}(\lambda)$;

- $\tau = \gamma, \xi = \gamma^2, U = 4\xi M, T = \gamma^2 \times U, K = 2 \times T$.

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ be an ORE scheme on message space $[K]$ with smoothed CLWW leakage $\mathcal{L}$. We define our new ORE $\Pi_{\mathsf{aff}} = (\mathsf{Gen}_{\mathsf{aff}}, \mathsf{Enc}_{\mathsf{aff}}, \mathsf{Comp}_{\mathsf{aff}})$ on message space $[M]$ as follows:

- $\mathsf{Gen}_{\mathsf{aff}}(1^\lambda, M, \Pi)$: On input the security parameter $\lambda$, message space $[M]$ and $\Pi$, the algorithm picks a super-polynomial $\gamma = 2^{\omega(\log \lambda)}$ as a global parameter, and computes parameters above. Then it runs $(\mathsf{ck}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, K)$, draws

$\alpha \xleftarrow{\$} \mathsf{logU}(\xi, 2\xi - 1)$ and $\beta$ from discrete uniform on $[T]'$ and outputs $\mathsf{sk}_{\mathsf{aff}} = (\mathsf{sk}, \alpha, \beta), \mathsf{ck}_{\mathsf{aff}} = \mathsf{ck}$.

- $\mathsf{Enc}_{\mathsf{aff}}(\mathsf{sk}_{\mathsf{aff}}, m)$. On input the secret key $\mathsf{sk}_{\mathsf{aff}}$ and a message $m \in [M]$, it outputs

$$\mathsf{CT}_{\mathsf{aff}} = \mathsf{Enc}(\alpha m + \beta).$$

  By our choice of message space $[K]$ for $\Pi$, the input to $\mathsf{Enc}$ is guaranteed to be in the message space.

- $\mathsf{Comp}_{\mathsf{aff}}(\mathsf{ck}_{\mathsf{aff}}, \mathsf{CT}^0_{\mathsf{aff}}, \mathsf{CT}^1_{\mathsf{aff}})$: On inputs the comparison key $\mathsf{ck}_{\mathsf{aff}}$, two ciphertexts $\mathsf{CT}^0_{\mathsf{aff}}, \mathsf{CT}^1_{\mathsf{aff}}$, it outputs $\mathsf{Comp}(\mathsf{ck}_{\mathsf{aff}}, \mathsf{CT}^0_{\mathsf{aff}}, \mathsf{CT}^1_{\mathsf{aff}})$.

Here we also give the description of composted schemes that only achieve "scale-hiding" or "shift-hiding". Formally, we define $\Pi_{\mathsf{scale}} = (\mathsf{Gen}_{\mathsf{scale}}, \mathsf{Enc}_{\mathsf{scale}}, \mathsf{Comp}_{\mathsf{scale}})$ and $\Pi_{\mathsf{shift}} = (\mathsf{Gen}_{\mathsf{shift}}, \mathsf{Enc}_{\mathsf{shift}}, \mathsf{Comp}_{\mathsf{shift}})$, respectively:

- $\mathsf{Gen}_{\mathsf{scale}}(1^\lambda, M, \Pi)$: On input the security parameter $\lambda$, the message space $[M]$ and $\Pi$, the algorithm picks a super-polynomial $\gamma = 2^{\omega(\log \lambda)}$ as a global parameter, and computes parameters above. Then it runs $(\mathsf{ck}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, K)$, draws $\alpha \xleftarrow{\$} \mathsf{logU}(\xi, 2\xi - 1)$ and outputs $\mathsf{sk}_{\mathsf{scale}} = (\mathsf{sk}, \alpha), \mathsf{ck}_{\mathsf{scale}} = \mathsf{ck}$.

- $\mathsf{Enc}_{\mathsf{scale}}(\mathsf{sk}_{\mathsf{scale}}, m)$. On input the secret key $\mathsf{sk}_{\mathsf{scale}}$ and a message $m \in [M]$, it outputs

$$\mathsf{CT}_{\mathsf{scale}} = \mathsf{Enc}(\alpha m).$$

- $\mathsf{Comp}_{\mathsf{scale}}(\mathsf{ck}_{\mathsf{scale}}, \mathsf{CT}^0_{\mathsf{scale}}, \mathsf{CT}^1_{\mathsf{scale}})$: On inputs the comparison key $\mathsf{ck}_{\mathsf{scale}}$, two ciphertexts $\mathsf{CT}^0_{\mathsf{scale}}, \mathsf{CT}^1_{\mathsf{scale}}$, it outputs $\mathsf{Comp}(\mathsf{ck}_{\mathsf{scale}}, \mathsf{CT}^0_{\mathsf{scale}}, \mathsf{CT}^1_{\mathsf{scale}})$.

- $\mathsf{Gen}_{\mathsf{shift}}(1^\lambda, M, \Pi)$: On input the security parameter $\lambda$, the message space $[M]$ and $\Pi$, the algorithm picks a super-polynomial $\gamma = 2^{\omega(\log \lambda)}$ as a global parameter, and computes parameters above. Then it runs $(\mathsf{ck}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, draws $\beta$ from discrete uniform on $[T]'$ and outputs $\mathsf{sk}_{\mathsf{shift}} = (\mathsf{sk}, \alpha), \mathsf{ck}_{\mathsf{shift}} = \mathsf{ck}$.

- $\mathsf{Enc}_{\mathsf{shift}}(\mathsf{sk}_{\mathsf{shift}}, m)$. On input the secret key $\mathsf{sk}_{\mathsf{shift}}$ and a message $m \in [M]$, it outputs

$$\mathsf{CT}_{\mathsf{shift}} = \mathsf{Enc}(m + b).$$

- $\mathsf{Comp}_{\mathsf{shift}}(\mathsf{ck}_{\mathsf{shift}}, \mathsf{CT}^0_{\mathsf{shift}}, \mathsf{CT}^1_{\mathsf{shift}})$: On inputs the comparison key $\mathsf{ck}_{\mathsf{shift}}$, two ciphertexts $\mathsf{CT}^0_{\mathsf{shift}}, \mathsf{CT}^1_{\mathsf{shift}}$, it outputs $\mathsf{Comp}(\mathsf{ck}_{\mathsf{shift}}, \mathsf{CT}^0_{\mathsf{shift}}, \mathsf{CT}^1_{\mathsf{shift}})$.

The correctness easily follows and what is more interesting is the privacy that those scheme can guarantee.

### 4.3.2 Security

In the part, we prove $\Pi_{\mathsf{aff}}$ is parameter hiding, formally:

**Theorem 4.3.1 (Parameter-Hiding)** *Assuming* $\Pi$ *has* $\mathcal{L}$*-simulation-security where* $\mathcal{L}$ *is smoothed CLWW, then for any* $\gamma = 2^{\omega(\log \lambda)}$, $\Pi_{\mathsf{aff}}$ *is* $(\gamma, \eta, \mu)$*-parameter hiding.*

**Proof:** According to the security notions, it is straightforward that if an ORE scheme is $(\gamma, \eta, \mu)$-parameter hiding, then it is also $(\gamma, \eta, \mu)$-scale hiding and $(\gamma, \eta, \mu)$-shift hiding. Next we claim the converse proposition holds.

**Claim.** If an ORE scheme $\Pi$ achieves $(\gamma, \eta, \mu)$-scale hiding and $(\gamma, \eta, \mu)$-shift hiding simultaneously, then $\Pi$ is $(\gamma, \eta, \mu)$-parameter hiding.

We sketch the proof by hybrid argument. For any $\gamma = 2^{\omega(\log \lambda)}$ and $(\eta, \mu)$-smooth distribution $D$, firstly, by shift-hiding, there is no efficient adversary that distinguish $(\delta_0, \ell_0)$ from $(\delta_0, 0)$ with non-negligible probability. Then due to scale-hiding, no efficient adversary can differ $(\delta_0, 0)$ from $(\delta_1, 0)$ with non-negligible probability. Thirdly, same as the first argument, any efficient adversary can distinguish $(\delta_1, 0)$ from $(\delta_1, \ell_1)$ with only negligible advantage. Combining together, $\Pi$ achieves $(\gamma, \eta, \mu)$-parameter hiding.

Thus, it suffices to show $\Pi_{\mathsf{aff}}$ is both $(\gamma, \eta, \mu)$-scale hiding and $(\gamma, \eta, \mu)$-shift hiding.

Before showing these two properties, we first prove a lemma about $(\eta, \mu)$-smooth distributions.

**Lemma 4.3.2** *If $D$ is $(\eta, \mu)$-smooth, we have the following properties: for any $\delta \geq \gamma, \ell > 0$,*

- *Property 1.* $\Delta(\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil, \lfloor D_{\mathsf{aff}}^{\delta,\lfloor \ell \rceil} \rceil) \leq \mathsf{negl}(\lambda)$,
- *Property 2. for any $s > 0$,* $\Delta(\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil, \lfloor D_{\mathsf{aff}}^{\delta(1+s),\ell} \rceil) \leq O(s) + \mathsf{negl}(\lambda)$,
- *Property 3. for any integer $z, b > 0$,* $\Delta(\mathcal{L}_{\mathsf{clww}}(z\vec{m} + b), \mathcal{L}_{\mathsf{clww}}(\vec{m}') + b) \leq \mathsf{negl}(\lambda)$, *where* $\vec{m} \xleftarrow{\$} \lfloor D_{\mathsf{scale}}^{\delta,0} \rceil, \vec{m}' \xleftarrow{\$} \lfloor D_{\mathsf{scale}}^{z\delta,0} \rceil$.

Intuitively, property 1 means that when shifting a smooth distribution small amount, the rounded distribution will negligibly change; similarly property 2 indicates that when scaling a smooth distribution by two close factors, the rounded distribution will change negligibly; property 3 considers scaling and then rounding or rounding and then scaling, and shows that the leakage will be almost identical in both cases (this follows from the fact that the lower-order bits of the plaintext would not affect the leakage profile). Next we prove these three properties one by one.

**Property 1.** Recalling the definitions, we note that the support of $\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil, \lfloor D_{\mathsf{aff}}^{\delta,\lfloor \ell \rceil} \rceil$ is $[\lceil \ell \rceil, \lfloor \delta + \ell \rfloor], [\lfloor \ell \rceil, \lfloor \delta \rfloor + \lfloor \ell \rceil]$, respectively, so:

$$\Delta(\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil, \lfloor D_{\mathsf{aff}}^{\delta,\lfloor \ell \rceil} \rceil) = \frac{1}{2} \sum_{k=\lfloor \ell \rceil}^{\lfloor \delta+\ell \rfloor+1} |p_{\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil}(k) - p_{\lfloor D_{\mathsf{aff}}^{\delta,\lfloor \ell \rceil} \rceil}(k)|.$$

Moreover, by the definition of $D_{\mathsf{aff}}^{\delta,\ell}$, we have the implication

$$p_{D_{\mathsf{aff}}^{\delta,\ell}}(x) = \frac{p_D(\frac{x-\ell}{\delta})}{\delta} \Rightarrow \int_{\ell}^{\delta+\ell} p_{D_{\mathsf{aff}}^{\delta,\ell}}(x)dx = \int_0^1 p_D(x)dx.$$

Besides, $D$ is $(\eta, \mu)$-smooth, which means $\int_0^1 p_D(x)dx \geq 1\text{-}\mathsf{negl} = O(1)$(for ease, we denote the integration as $C$). Therefore, for any $k, \delta > \gamma$, and $\forall \ell$,

$$p_{\lfloor D_{\mathsf{aff}}^{\delta,\ell} \rceil}(k) \leq \frac{\int_{k-1}^{k+1/2} p_{D_{\mathsf{aff}}^{\delta,\ell}}(x)dx}{C} \leq \frac{1.5\eta}{\delta C} = O(\frac{1}{\delta}).$$

Now, we make use of the smooth derivative property. We observe that for any $k \in$

$[\lceil \ell \rceil + 1, \lfloor \delta + \ell \rfloor - 1]$, if $\forall x \in [k - 1/2, k + 1/2]$ satisfying $|p'_{\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil}(x)|, |p'_{\lfloor D^{\delta,\lfloor \ell \rceil}_{\mathsf{aff}} \rceil}(x)| \leq \eta$, then,

$$\begin{aligned}
|p_{\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil}(k) - p_{\lfloor D^{\delta,\lfloor \ell \rceil}_{\mathsf{aff}} \rceil}(k)| &= \frac{1}{C} \int_{k-1/2}^{k+1/2} |p_{D^{\delta,\ell}_{\mathsf{aff}}}(x) - p_{D^{\delta,\lfloor \ell \rceil}_{\mathsf{aff}}}(x)| dx \\
&= \frac{1}{C} \int_{k-1/2}^{k+1/2} |\frac{p_D(\frac{x-\ell}{\delta})}{\delta} - \frac{p_D(\frac{x-\lfloor \ell \rceil}{\delta})}{\delta}| dx \\
&\leq \frac{1}{C\delta} \int_{k-1/2}^{k+1/2} 2 \max(p'_D(x))(\frac{x-\ell}{\delta} - \frac{x - \lfloor \ell \rceil}{\delta}) \\
&\leq \frac{1}{C\delta} \cdot \frac{\eta}{\delta} = O(\frac{1}{\delta^2}).
\end{aligned}$$

In addition, $D$ is $(\eta, \mu)$-smooth, there are at most $\mu$ bad points such that the derivative is not bounded by $\eta$, thus:

$$\begin{aligned}
\Delta(\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil, \lfloor D^{\delta,\lfloor \ell \rceil}_{\mathsf{aff}} \rceil) &\leq \frac{1}{2} \sum_{k=\lceil \ell \rceil + 1}^{\lfloor \delta + \ell \rfloor - 1} |p_{\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil}(k) - p_{\lfloor D^{\delta,\lfloor \ell \rceil}_{\mathsf{aff}} \rceil}(k)| + 4 \frac{1.5\eta}{\delta C} \\
&\leq \delta \cdot \frac{1}{C\delta} \cdot \frac{\eta}{\delta} + 4\mu \frac{1.5\eta}{\delta C} + 4 \frac{1.5\eta}{\delta C} \\
&= (\frac{7\eta + 6\eta\mu}{C} \cdot \frac{1}{\delta}) = O(\frac{1}{\delta}) \leq \mathsf{negl}.
\end{aligned}$$

**Property 2.** Similarly, the support of $\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil$ and $\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}} \rceil$ is $[\lceil \ell \rceil, \lfloor \delta + \ell \rfloor], [\lceil \ell \rceil, \lfloor \delta(1+s) + \ell \rfloor]$, respectively, so:

$$\Delta(\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil, \lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}} \rceil) = \frac{1}{2} \sum_{k=\lceil \ell \rceil}^{\lfloor \delta(1+s) + \ell \rfloor} |p_{\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil}(k) - p_{\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}} \rceil}(k)|.$$

Same as the proof in property 1, we have that for any $k$, $p_{\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil}(k), p_{\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}} \rceil}(k) \leq O(\frac{1}{\delta})$. Moreover, for any $k \in [\lceil \ell \rceil + 1, \lfloor \delta + \ell \rfloor - 1]$, if $\forall x \in [k - 1/2, k + 1/2]$ satisfying

$|p'_{\lfloor D^{\delta,\ell}_{\mathsf{aff}}\rceil}(x)|, |p'_{\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}}\rceil}(x)| \leq \eta$, then,

$$
\begin{aligned}
&|p_{\lfloor D^{\delta,\ell}_{\mathsf{aff}}\rceil}(k) - p_{\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}}\rceil}(k)| = \frac{1}{C}\int_{k-1/2}^{k+1/2} |p_{D^{\delta,\ell}_{\mathsf{aff}}}(x) - p_{D^{\delta(1+s),\ell}_{\mathsf{aff}}}(x)|dx \\
&= \frac{1}{C}\int_{k-1/2}^{k+1/2} |\frac{p_D(\frac{x-\ell}{\delta})}{\delta} - \frac{p_D(\frac{x-\ell}{\delta(1+s)})}{\delta(1+s)}|dx \\
&= \frac{1}{C\delta(1+s)}\int_{k-1/2}^{k+1/2} p_D(\frac{x-\ell}{\delta})(1+s) - p_D(\frac{x-\ell}{\delta(1+s)})dx \\
&= \frac{1}{C\delta(1+s)}\int_{k-1/2}^{k+1/2} p_D(\frac{x-\ell}{\delta}) - p_D(\frac{x-\ell}{\delta(1+s)})dx + \frac{s}{C\delta(1+s)}\int_{k-1/2}^{k+1/2} p_D(\frac{x-\ell}{\delta})dx \\
&\leq \frac{1}{C\delta(1+s)}\int_{k-1/2}^{k+1/2} 2\max(p'_D(x))\frac{x-\ell}{\delta}(1-\frac{1}{1+s})dx + \frac{s\eta}{C\delta(1+s)} \\
&\leq \frac{1}{C\delta(1+s)}\frac{2\eta s}{1+s} + \frac{s\eta}{C\delta(1+s)} \leq \frac{3\eta s}{C\delta(1+s)}.
\end{aligned}
$$

Besides, $D$ is $(\eta, \mu)$-smooth, there are at most $\mu$ bad points s.t. the derivative is not bounded by $\eta$, thus we have $\Delta(\lfloor D^{\delta,\ell}_{\mathsf{aff}}\rceil, \lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}}\rceil)$:

$$
\begin{aligned}
&\leq \frac{1}{2}\sum_{k=\lceil\ell\rceil+1}^{\lfloor\delta+\ell\rfloor-1} |p_{\lfloor D^{\delta,\ell}_{\mathsf{aff}}\rceil}(k) - p_{\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}}\rceil}(k)| + \sum_{k=\lceil\delta+\ell\rceil}^{\lfloor\delta(1+s)+\ell\rfloor} p_{\lfloor D^{\delta(1+s),\ell}_{\mathsf{aff}}\rceil}(k) + O(\frac{1}{\delta}) \\
&\leq \delta\frac{3\eta s}{C\delta(1+s)} + 4\mu O(\frac{1}{\delta}) + \lceil\delta s\rceil O(\frac{1}{\delta}) + O(\frac{1}{\delta}) \\
&= s\frac{3\eta}{C(1+s)} + (4\mu+1)O(\frac{1}{\delta}) + O(s) \\
&= O(s) + O(\frac{1}{\delta}) \leq O(s) + \mathsf{negl}.
\end{aligned}
$$

**Property 3.** The statement of property 3 considers scaling and then rounding or rounding and then scaling, and shows that the leakage will be almost identical in both case, and we prove it in two steps. Roughly, in the first step we show that adding any small noise to the plaintext sequence would not change the leakage profile whp; in step 2, we construct a new distribution $\lfloor D^{z,\delta,0}_{\mathsf{noise}}\rceil$ based on a noise distribution, such that $\lfloor D^{z,\delta,0}_{\mathsf{noise}}\rceil, \lfloor D_{\mathsf{z}\delta,0}\rceil) \leq \mathsf{negl}$ and $\lfloor D^{z,\delta,0}_{\mathsf{noise}}\rceil$ samples the noised plaintexts in step 1.

**Step 1.** For any integer $z, b > 0$, for any noise vector $\vec{e}$ such that the maximum of $\vec{e}$ is

less than $z$, then we claim

$$\Delta(\mathcal{L}_{\mathsf{clww}}(z\vec{m}+b), \mathcal{L}_{\mathsf{clww}}(z\vec{m}+\vec{e}+b)) \leq \mathsf{negl}, \vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil.$$

By definition, we know that $\mathcal{L}_{\mathsf{clww}}(\vec{m})$ gives the position of the most significant differ bit for all plaintext pairs, namely $\mathsf{msdb}(m_i, m_j)$. Let $\nu = \gamma^{1/4}$($\nu$ is also a super-polynomial in $\lambda$), we have

$$\Pr[\exists m_i, m_j, s.t. |m_i - m_j| \leq \nu^2] \leq \frac{1}{\nu} \leq \mathsf{negl}.$$

which referring to whp the most significant different bit of any pair would not lay in the last $2\log\nu$ bits ($\mathsf{msdb}(m_i, m_j) \leq K - 2\log\nu$). Therefore, for any fixed $z, b > 0$, $\mathsf{msdb}(zm_i + b, zm_j + b) \leq K - \lfloor \log z \rfloor - 2\log\nu$ whp, and it seems sufficient to prove that for any fixed $z, b$, for all plaintexts adding a small noise would not change the first $K - \lfloor \log z \rfloor - 2\log\nu$ bits, because $\mathcal{L}_{\mathsf{clww}}$ only leaks the position of the most significant differ bit.

We say a plaintext $m$ is "*bad*" if there is a noise $0 < e \leq z$ s.t. the first $K - \lfloor \log z \rfloor - 2\log\nu$ bits of $zm + b$ and $zm + b + e$ are distinct, and we show that with high probability, none of the plaintext in $\vec{m}$ is bad.

For ease, we write $S \subset \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil$ to denote the set of bad plaintext in the support of $\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil$ and we define the event $\mathsf{Bad}$ as when sampling a plaintext vector $\vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil$, there exists $m^* \in \vec{m}$ such that $m \in S$. As plaintext is drawn i.i.d., we have

$$\Pr[\mathsf{Bad}] = \Pr[\vec{m} \cap S \neq \emptyset : \vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil] \leq q \Pr[m \in S : m \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil].$$

One key observation is that the bad message appears periodically, namely if $m$ is bad, then $m + 1, \ldots, m + \nu^2/2$ are not bad(if $m$ is bad, the first $2\log\nu - 1$ bits of the last $2\log\nu + \lfloor \log z \rfloor$ bits of $zm + b$ must be $\underbrace{1 \ldots 1}_{2\log\nu - 1}$, otherwise adding an small noise would not affect the prefix, and we immediately observe that for $m + i, i \in [1, \nu^2/2]$, the first $2\log\nu - 1$ bits of the last $2\log\nu + \lfloor \log z \rfloor$ bits of $zm_i + b$ are not all 1). Based on this observation, we define a sequence of disjoint set $S_i \in \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil$, and we say $m \in S_i$ if

$m - i \in S$, then we claim

$$\Pr[m \in S_i] \geq \Pr[m \in S] - O(\frac{i}{\gamma}) - O(\frac{1}{\gamma}), \forall i \in [1, \nu^2/2].$$

Firstly, we note that any $k \in [1, \lfloor \delta \rfloor - 1]$,

$$|p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k+1) - p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k)| = |p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k+1) - p_{\lfloor D^{\delta,1}_{\mathsf{scale}} \rceil}(k+1)|.$$

Applying property 1, for any $k \in [\lceil \ell \rceil + 1, \lfloor \delta + \ell \rfloor - 1]$, if $\forall x \in [k - 1/2, k + 1/2]$ such that $|p'_{\lfloor D^{\delta,\ell}_{\mathsf{aff}} \rceil}(x)|, |p'_{\lfloor D^{\delta,\lfloor \ell \rceil}_{\mathsf{aff}} \rceil}(x)| \leq \eta$, then

$$|p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k) - p_{\lfloor D^{\delta,1}_{\mathsf{scale}} \rceil}(k)| \leq O(\frac{1}{\gamma^2}) \Rightarrow |p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k) - p_{\lfloor D^{\delta,i}_{\mathsf{scale}} \rceil}(k)| \leq O(\frac{i}{\gamma^2}),$$

which means $|p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k+i) - p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k)| \leq O(\frac{i}{\gamma^2})$. By the periodicity of the appearance for bad plaintext and there are at most $\mu$ points s.t. the derivative is not bounded by $\eta$, we have

$$\Pr[S_i] \geq \Pr[S] - \delta O(\frac{i}{\gamma}) - O(\frac{4\mu}{\gamma}) - O(\frac{1}{\gamma}) = \Pr[S] - O(\frac{i}{\gamma}).$$

Then we prove $\Pr[\mathsf{Bad}] \leq \mathsf{negl}$. If not, say $\Pr[\mathsf{Bad}] = t$ is noticeable, then $\Pr[S] > t/q$, which indicates that $\Pr[S_i] \geq t/q - O(\frac{i}{\gamma})$. Moreover, by definition $\nu^{1.5} > 3q/t$, we have:

$$\begin{aligned}
1 &\geq \Pr[m \in S] + \Pr[m \in S_1] + \ldots + \Pr[m \in S_{\nu^{1.5}}] \\
&\geq t/q + t/q - O(\frac{1}{\gamma}) + \ldots + t/q - O(\frac{\nu^{1.5}}{\gamma}) \\
&\geq \nu^{1.5} \cdot t/q - \sum_{i=1}^{\nu^{1.5}} \frac{i}{\gamma} > 3 - \frac{\nu^3}{\gamma} = 3 - O(\frac{1}{\nu}) > 2.
\end{aligned}$$

**Step 2.** In Step 1, we see that the leakage profile is resilient to adding any small noise($|e| < z$), and in the following we prove that there exists a distribution $E_{\mathsf{noise}}$ such that we can induce a new distribution $\lfloor D^{z,\delta,0}_{\mathsf{noise}} \rceil$ from $\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil$ and $E_{\mathsf{noise}}$. More specifically, let $M, M', E$ be random variables drawn from $\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil$, $\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil$ and $E_{\mathsf{noise}}$

respectively, and we define a new variable as $M_E = zM + E$. Then we complete our proof by showing $\Delta(M_E, M') \leq \mathsf{negl}$. As $E_{\mathsf{noise}}$ can be any distribution as long as its variable is small, we define $M_E$ as following:

$$\Pr[M_E = zk + i] = \underline{p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k)} \times \frac{p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk+i)}{p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk) + \ldots + p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk+z-1)}.$$

On the other side, we have that

$$\begin{aligned}\Pr[M' = zk + i] &= p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk+i) \\ &= \underline{p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(zk) + \ldots + p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(zk+z-1)} \times \frac{p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk+i)}{p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk) + \ldots + p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk+z-1)}.\end{aligned}$$

Therefore, we can calculate the the statistical distance:

$$\Delta(\lfloor D^{z,\delta,0}_{\mathsf{noise}} \rceil, \lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil) = \Delta(M_E, M') = \sum_{k=0}^{\lfloor \delta \rfloor} |p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k) - (p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk) + \ldots + p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(zk+z-1))|.$$

By definition, easy to note that

$$\begin{aligned}p_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(k) &= \frac{1}{C} \int_{k-1/2}^{k+1-1/2} \frac{p_D(\frac{x}{\delta})}{\delta} dx, \\ \sum_{i=zk}^{zk+z-1} p_{\lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil}(i) &= \frac{1}{C} \int_{zk-1/2}^{z(k+1)-1/2} \frac{p_D(\frac{x}{z\delta})}{z\delta} dx = \frac{1}{C} \int_{k-\frac{1}{2z}}^{k+1-\frac{1}{2z}} \frac{p_D(\frac{x}{\delta})}{\delta} dx.\end{aligned}$$

Therefore, for any $k \in [0, \lfloor \delta \rfloor]$, if $\forall x \in [k - \frac{1}{2}, k + 1 - \frac{1}{2z}]$ such that $|p'_{\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil}(x)| \leq \eta$, then,

$$\begin{aligned}&|\frac{1}{C} \int_{k-1/2}^{k+1-1/2} \frac{p_D(\frac{x}{\delta})}{\delta} dx - \frac{1}{C} \int_{k-\frac{1}{2z}}^{k+1-\frac{1}{2z}} \frac{p_D(\frac{x}{\delta})}{\delta} dx| \\ =&\frac{1}{C} |\frac{p_D(\frac{x}{\delta})}{\delta} dx - \int_{k-1/2}^{k+1-1/2} \frac{p_D(\frac{x+\frac{1}{2}-\frac{1}{2z}}{\delta})}{\delta} dx| \\ \leq&\frac{1}{C} \int_{k-1/2}^{k+1/2} |\frac{p_D(\frac{x}{\delta})}{\delta} - \frac{p_D(\frac{x+\frac{1}{2}-\frac{1}{2z}}{\delta})}{\delta}| dx \\ \leq&\frac{1}{C\delta} \int_{k-1/2}^{k+1/2} \max|p'| \frac{1}{\delta} dx \leq \frac{\eta}{C\delta^2}.\end{aligned}$$

In addition, $D$ is $(\eta, \mu)$-smooth, there are at most $\mu$ points such that the derivative is not bounded by $\eta$, thus:

$$\Delta(\lfloor D^{z,\delta,0}_{\mathsf{noise}} \rceil, \lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil) \leq \delta \frac{\eta}{C\delta^2} + 4\mu O(\frac{1}{\delta}) \leq O(\frac{1}{\delta}) \leq \mathsf{negl}$$

In the following we prove that $\Pi_{\mathsf{aff}}$ is both $(\gamma, \eta, \mu)$-scale hiding and $(\gamma, \eta, \mu)$-shift hiding.

**Lemma 4.3.3** *Assuming $\Pi$ has $\mathcal{L}$-simulation-security where $\mathcal{L}$ is smoothed CLWW, then for any $\gamma = 2^{\omega(\log \lambda)}$, $\Pi_{\mathsf{aff}}$ is $(\gamma, \eta, \mu)$-shift hiding.*

As $\Pi$ is $\mathcal{L}$-simulation-secure, it suffices to consider an adversary $\mathcal{A}$ that only gets the leakage function $\mathcal{L}(\vec{m})$. It outputs a single bit. Hence it suffices to show that, for any fixed $\alpha \in [\xi, 2\xi)$, $\Delta(\mathcal{L}(\alpha\vec{m}_0 + \beta), \mathcal{L}(\alpha\vec{m}_1 + \beta)) \leq \mathsf{negl}(\lambda)$, where $\vec{m_0} \xleftarrow{\$} \lfloor D^{\delta,\ell_0}_{\mathsf{aff}} \rceil; \vec{m_1} \xleftarrow{\$} \lfloor D^{\delta,\ell_1}_{\mathsf{aff}} \rceil; \beta \xleftarrow{\$} [T]'$. Moreover, we observe that for any $\alpha$, the maximum element in $a\vec{m}_b$ is less than $T/2$. Thus, if we view the message space as a binary tree, each element lies in the left-most subtree depth 2. Adding $T$ just moves the messages to a different subtree; according to CLWW leakage this indicates $\mathcal{L}(\vec{m}) = \mathcal{L}(\vec{m} + T)$.

Let $M_0, M_1$ be the random variables drawn from $\lfloor D^{\delta,\lfloor \ell_0 \rceil}_{\mathsf{aff}} \rceil, \lfloor D^{\delta,\lfloor \ell_1 \rceil}_{\mathsf{aff}} \rceil$. We have that $M_0 = M_1 - (\lfloor \ell_1 \rceil - \lfloor \ell_0 \rceil)$ as random variables (we assume $\ell_1 > \ell_0$ wlog).

Now we define a bijective map $f : [T]' \to [T]'$ as $f(x) = \alpha(\lfloor \ell_1 \rceil - \lfloor \ell_0 \rceil) + x \bmod T$. Let $B \xleftarrow{\$} [0, T)$. Since the leakage is invariant to shifts by $T$, we have that $\mathcal{L}(\alpha\vec{m}'_0 + B) = \mathcal{L}(\alpha\vec{m}'_1 + f(B))$, where $\vec{m_0}' \xleftarrow{\$} \lfloor D^{\delta,\lfloor \ell_0 \rceil}_{\mathsf{aff}} \rceil; \vec{m_1}' \xleftarrow{\$} \lfloor D^{\delta,\lfloor \ell_1 \rceil}_{\mathsf{aff}} \rceil$. As $\beta$ is sampled uniformly, thus, $\mathcal{L}(\alpha\vec{m_0}' + \beta) = \mathcal{L}(\alpha\vec{m_1}' + \beta)$.

Now, applying Property 1 in lemma 4.3.2, we have $\Delta(\lfloor D^{\delta,\ell_b} \rceil, \lfloor D^{\delta,\lfloor \ell_b \rceil}_{\mathsf{aff}} \rceil) \leq \mathsf{negl}(\lambda)$. Combing together, we get

$$\Delta(\mathcal{L}(\alpha\vec{m}_0 + \beta), \mathcal{L}(\alpha\vec{m}_1 + \beta)) \leq \mathsf{negl}(\lambda).$$

Next we show that $\Pi_{\mathsf{aff}}$ is also $(\gamma, \eta, \mu)$-scale hiding. However, The proof of the scale hiding part is much more involved than the one in Lemma 4.3.3, more specifically, for

CLWW smoothed leakage profile, we only have $\mathcal{L}(\vec{m}) = \mathcal{L}(2\vec{m})$ and it's very unlikely $\mathcal{L}(\vec{m}+\beta) = \mathcal{L}(2\vec{m}+\beta)$ for any fixed $\beta$. Thus we do not have the periodicity of multiplying 2 directly and a more tricky technique is needed.

A core observation is that we can decompose $\Pi_{\mathsf{aff}}$ to $(\Pi_{\mathsf{shift}})_{\mathsf{scale}}$, namely for any $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comp})$ (with smoothed CLWW leakage), $\mathsf{Enc}_{\mathsf{aff}}(m) = \mathsf{Enc}(\alpha m + \beta)$, in contrast, for $(\Pi_{\mathsf{shift}})_{\mathsf{scale}} = (\mathsf{Gen}_{\mathsf{ss}}, \mathsf{Enc}_{\mathsf{ss}}, \mathsf{Enc}_{\mathsf{ss}})$ we have

$$\mathsf{Enc}_{\mathsf{ss}}(m) = \mathsf{Enc}_{\mathsf{shift}}(\alpha m) = \mathsf{Enc}(\alpha m + \beta) = \mathsf{Enc}_{\mathsf{aff}}(m).$$

Hence, it seems sufficient to prove the following two statements: "*assuming* $\Pi$ *is* $\mathcal{L}$*-simulation-secure where* $\mathcal{L}$ *is smoothed CLWW, then* $\Pi_{\mathsf{scale}}$ *is* $(\gamma, \eta, \mu)$*-scale hiding*" and "$\Pi_{\mathsf{shift}}$ *is* $\mathcal{L}$*-simulation-secure and* $\mathcal{L}$ *is smoothed CLWW* ".

Unfortunately, this attempt does not work neither. By definition, the leakage profile of $\Pi_{\mathsf{shift}}$ is $\mathcal{L}(\vec{m}+\beta)$ (we will denote the leakage as $\mathcal{L}_{\mathsf{shift}}$ below for short), and for any plaintext sequences $\vec{m}_0, \vec{m}_1$ such that $\mathcal{L}_{\mathsf{clww}}(\vec{m}_0) = \mathcal{L}_{\mathsf{clww}}(\vec{m}_1)$, it's unlikely that $\mathcal{L}_{\mathsf{shift}}(\vec{m}_0 + \beta) = \mathcal{L}_{\mathsf{shift}}(\vec{m}_1 + \beta)$ (even over the probability of $\beta$), because the smoothed CLWW leakage profile might be very sensitive by adding a large noise. Hence the leakage profile of $\Pi_{\mathsf{shift}}$ is not likely to be smoothed CLWW and we need to analyze $\mathcal{L}_{\mathsf{shift}}$ carefully. Formally:

**Lemma 4.3.4** *Assuming* $\Pi$ *is* $\mathcal{L}$*-simulation-secure where* $\mathcal{L}$ *is CLWW-smoothed, then for any* $\gamma = 2^{\omega(\log\lambda)}$, $\Pi_{\mathsf{aff}}$ *is* $(\gamma, \eta, \mu)$*-scale hiding.*

Firstly we show $\mathcal{L}_{\mathsf{shift}}$ satisfies the following two properties:

1. for any integer $z > 0$, $\Delta(\mathcal{L}_{\mathsf{shift}}(z\vec{m}), \mathcal{L}_{\mathsf{shift}}(\vec{m}')) \leq \mathsf{negl}(\lambda)$ where $\vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil, \vec{m}' \xleftarrow{\$} \lfloor D^{z\delta,0}_{\mathsf{scale}} \rceil$;
2. $\Delta(\mathcal{L}_{\mathsf{shift}}(\vec{m}), \mathcal{L}_{\mathsf{shift}}(2\vec{m})) \leq \mathsf{negl}(\lambda)$

The first property is followed directly by the Property 3 in Lemma 4.3.2, since $\mathcal{L}_{\mathsf{shift}}(\vec{m}) =$

$\mathcal{L}(\vec{m}+\beta)$ and $\mathcal{L}$ is smoothed CLWW. Now we show the second property. It is straightforward to show that for a given $\beta$, $\mathcal{L}_{\mathsf{shift}}(\vec{m})$ is likely not equal to $\mathcal{L}_{\mathsf{shift}}(2\vec{m})$. However, $\beta$ is part of secret key, and it's sufficient to show that, over the probability of $\beta$, the distributions of $\mathcal{L}_{\mathsf{shift}}(\vec{m})$ and $\mathcal{L}_{\mathsf{shift}}(2\vec{m})$ are close.

In fact, for any plaintext pair $(m_i, m_j) \in \vec{m}$ (assuming all the plaintexts are distinct), we write $z_{i,j}$ to denote the position of the most significant differing bit of $(2m_i+\beta, 2m_j+\beta)$, and since $2m_i, 2m_j$ are both have the last bit 0, we know that $z_{i,j}$ is not the last bit. Since $\mathcal{L}$ is smoothed CLWW, we have the implication

$$\mathcal{L}_{\mathsf{clww}}(2\vec{m}+\beta) = \mathcal{L}_{\mathsf{clww}}(2\vec{m}+2\lfloor\frac{\beta}{2}\rfloor) \Rightarrow \mathcal{L}(2\vec{m}+\beta) = \mathcal{L}(2\vec{m}+2\lfloor\frac{\beta}{2}\rfloor) = \mathcal{L}(\vec{m}+\lfloor\frac{\beta}{2}\rfloor).$$

We note that $\lfloor\frac{\beta}{2}\rfloor$ is a random variable distributed on $[T/2]'$ uniformly, and we need extend it to $[T]'$. Here we define a bijective map $f : [T]' \to [T]'$ as $f(\beta) = \lfloor\frac{\beta}{2}\rfloor + (\beta \ mod\ 2)\cdot(T/2)$, and want to show that with overwhelming probability $\mathcal{L}(\vec{m}+\lfloor\frac{\beta}{2}\rfloor) = \mathcal{L}(\vec{m}+f(\beta))$.

In the case that $\beta \ mod\ 2 = 0$, this holds trivially, as $f(\beta) = \lfloor\frac{\beta}{2}\rfloor$. Else, suppose $\beta \ mod\ 2 = 1$. According to the definition of CLWW leakage, we have that $\mathcal{L}_{\mathsf{clww}}(\vec{m}+\lfloor\frac{\beta}{2}\rfloor) \neq \mathcal{L}_{\mathsf{clww}}(\vec{m}+\lfloor\frac{\beta}{2}\rfloor+(T/2))$ only if there exists an $m_i$ such that $m_i + \lfloor\frac{\beta}{2}\rfloor \geq (T/2)$ (if not, then $\forall i, j, \mathsf{msdb}(m_i, m_j) \geq 3$, which means adding $(T/2)$ would preserve the leakage profile). Moreover, $|m_i| \leq \frac{T}{2\gamma^2}$, which means $\Pr_{\beta \xleftarrow{\$} [T]'}[m_i + \lfloor\frac{\beta}{2}\rfloor \geq (T/2)] \leq \frac{1}{\gamma^2}$. Thus, we have

$$\Delta(\mathcal{L}(\vec{m}+\beta), \mathcal{L}(2\vec{m}+\beta)) = \Delta(\mathcal{L}_{\mathsf{shift}}(\vec{m}), \mathcal{L}_{\mathsf{shift}}(2\vec{m})) \leq q\frac{1}{\gamma^2} \leq \mathsf{negl}(\lambda).$$

Hence, it suffices to prove that $\Pi_{\mathsf{scale}}$ is scale hiding if the leakage profile of $\Pi$ satisfies the two properties above. We call such a leakage profile *scale-smoothed*.

**Lemma 4.3.5** *Assuming $\Pi$ is $\mathcal{L}$-simulation-secure where $\mathcal{L}$ is scale-smoothed, then for any $\gamma = 2^{\omega(\log\lambda)}$, $\Pi_{\mathsf{scale}}$ is $(\gamma, \eta, \mu)$-scale hiding.*

Similarly to Lemma 4.3.3, we consider adversary $\mathcal{A}$ that takes the leakage profile as input, and we define

$$P(\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil) := \Pr[\mathcal{A}(\mathcal{L}(\alpha\vec{m})) : \vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil, \alpha \xleftarrow{\$} \mathsf{logU}(\xi, 2\xi-1)],$$
$$Q(\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil) := \Pr[\mathcal{A}(\mathcal{L}(\vec{m})) : \vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil].$$

We note that it is sufficient to show that $\Pi_{\mathsf{scale}}$ is scale hiding that for any valid $\delta_0, \delta_1$ ( meaning $\gamma \leq \delta_0, \delta_1 \leq M$), we have $|P(\lfloor D^{\delta_0,0}_{\mathsf{scale}} \rceil) - P(\lfloor D^{\delta_1,0}_{\mathsf{scale}} \rceil)| \leq \mathsf{negl}(\lambda)$). Since $\mathcal{L}$ is scale-smoothed, we have that for any $\alpha \in [\xi, 2\xi-1]$, $\Delta(\mathcal{L}(\alpha\vec{m}), \mathcal{L}(\vec{m}')) \leq \mathsf{negl}(\lambda)$, where $\vec{m} \xleftarrow{\$} \lfloor D^{\delta,0}_{\mathsf{scale}} \rceil, \vec{m}' \xleftarrow{\$} \lfloor D^{\alpha\delta,0}_{\mathsf{scale}} \rceil$. Hence we can represent $P(\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil)$ as:

$$\begin{aligned} P(\lfloor D^{\delta,0}_{\mathsf{scale}} \rceil) &= \sum_{k=\xi}^{2\xi-1} \Pr[\alpha = k]\mathcal{A}(\mathcal{L}(k\vec{m})) \\ &\overset{\mathsf{negl}}{\approx} \sum_{k=\xi}^{2\xi-1} \Pr[\alpha = k]\mathcal{A}(\mathcal{L}(\vec{m}')) \\ &= \sum_{k=\xi}^{2\xi-1} \Pr[\alpha = k] Q(\lfloor D^{k\delta,0}_{\mathsf{scale}} \rceil). \end{aligned}$$

(here we abuse the notation $\vec{m}'$ to denote the plaintext sequence drawn from $\lfloor D^{k\delta,0}_{\mathsf{scale}} \rceil$).

In the next step, we make use of the periodicity of scaling 2, intuitively, we show that for any integer $z_0, z_1$, if there is an integer $s$ such that $z_1 \approx 2^s z_0$, then $Q(\lfloor D^{z_0,0}_{\mathsf{scale}} \rceil) \approx Q(\lfloor D^{z_1,0}_{\mathsf{scale}} \rceil)$. To make it concrete, we first introduce $\tau$ disjoint buckets $B_0, \ldots, B_{\tau-1}$ such that:

$$B_i = \{(\tau{+}i)\ ,\ (2(\tau{+}i), 2(\tau{+}i){+}1)\ ,\ \ldots,\ (2^d(\tau{+}i), 2^d(\tau{+}i){+}1, \ldots, 2^d(\tau{+}i){+}2^d{-}1) \ldots\}.$$

and claim that for any distinct integer $\delta_0, \delta_1$, if they fall into the same bucket, then $Q(\lfloor D^{\delta_0,0}_{\mathsf{scale}} \rceil) \overset{\mathsf{negl}}{\approx} Q(\lfloor D^{\delta_1,0}_{\mathsf{scale}} \rceil)$.

**Claim.** $\forall \gamma \leq \delta_0, \delta_1 \leq T$, if there is $i \in [\tau]'$ such that $\delta_0, \delta_1 \in B_i$, then $|Q(\lfloor D^{\delta_0,0}_{\mathsf{scale}} \rceil) - Q(\lfloor D^{\delta_1,0}_{\mathsf{scale}} \rceil)| \leq \mathsf{negl}(\lambda)$.

We assume $\delta_0, \delta_1 \in B_0$ wlog, more precisely, we denote $\delta_0 = 2^{d_0}\tau{+}t_0; \delta_1 = 2^{d_1}\tau{+}t_1; d_0 <$

$d_1$ and $0 \leq t_b < 2^{d_b}$. Apparently, $\delta_b \leq (1+\frac{1}{\tau})2^{d_b}\tau$, so applying the Property 2 in Lemma 4.3.2, we know that

$$\begin{aligned}&\Delta(\lfloor D_{\mathsf{scale}}^{\delta_b,0}\rceil, \lfloor D_{\mathsf{scale}}^{2^{d_b}\tau,0}\rceil) \leq O(\frac{1}{\tau}) + \mathsf{negl}(\lambda) \leq O(\frac{1}{\tau})\\ \Rightarrow &|Q(\lfloor D_{\mathsf{scale}}^{\delta_b,0}\rceil) - Q(\lfloor D_{\mathsf{scale}}^{2^{d_b}\tau,0}\rceil)| \leq \mathsf{negl}(\lambda).\end{aligned}$$

Moreover, let $\vec{m}_0, \vec{m}_1$ be the plaintext sequence drawn from $\lfloor D_{\mathsf{scale}}^{2^{d_0}\tau,0}\rceil, \lfloor D_{\mathsf{scale}}^{2^{d_1}\tau,0}\rceil$ respectively, due to the second property of $\mathcal{L}$, we have

$$|Q(\lfloor D_{\mathsf{scale}}^{2^{d_0}\tau,0}\rceil - Q(\lfloor D_{\mathsf{scale}}^{2^{d_1}\tau,0}\rceil)| \leq \Delta(\mathcal{L}(\vec{m}_0), \mathcal{L}(2^{d_1-d_0}\vec{m}_0)) + \mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda).$$

Combing together, $|Q(\lfloor D_{\mathsf{scale}}^{\delta_0,0}\rceil) - Q(\lfloor D_{\mathsf{scale}}^{\delta_1,0}\rceil)| \leq \mathsf{negl}$. Therefore, we can extract a representative for each bucket and for ease, we write $Q(B_i)$ as the value such that for any $\delta \in B_i$, $|Q(\lfloor D_{\mathsf{scale}}^{\delta,0}\rceil) - Q(B_i)| \leq \mathsf{negl}$.

By this bucket technique, we can re-organize the representation of $P(\lfloor D_{\mathsf{scale}}^{\delta_b,0}\rceil)$ *bucket by bucket*, namely, let $p_{i,\delta_b} := \Pr[\alpha\delta_b \in B_i : \alpha \xleftarrow{\$} \mathsf{logU}(\xi, 2\xi-1)]$, we have:

$$P(\lfloor D_{\mathsf{scale}}^{\delta_b,0}\rceil) \overset{\mathsf{negl}}{\approx} \sum_{k=\xi}^{2\xi-1} \Pr[\alpha = k] Q(\lfloor D_{\mathsf{scale}}^{k\delta,0}\rceil) = \sum_{i=0}^{\tau-1} \sum_{\alpha\delta_b \in B_i} Q(\lfloor D_{\mathsf{scale}}^{\alpha\delta_b,0}\rceil) \overset{\mathsf{negl}}{\approx} \sum_{i=0}^{\tau-1} p_{i,\delta_b} Q(B_i).$$

By definition, $\mathcal{A}$ is a predicate algorithm, which means $\forall i \in [\tau]', Q(B_i) \leq 1$, then

$$|P(\lfloor D_{\mathsf{scale}}^{\delta_0,0}\rceil) - P(\lfloor D_{\mathsf{scale}}^{\delta_1,0}\rceil)| \leq |\sum_{i=0}^{\tau-1} p_{i,\delta_0} Q(B_i) - \sum_{i=0}^{\tau-1} p_{i,\delta_1} Q(B_i)| + \mathsf{negl} \leq \sum_{i=0}^{\tau-1} |p_{i,\delta_0} - p_{i,\delta_1}| + \mathsf{negl}.$$

Therefore, it remains to prove $\sum_{i=0}^{\tau-1} |p_{i,\delta_0} - p_{i,\delta_1}| \leq \mathsf{negl}$.

In fact, the range of $\alpha$ is $[\xi, 2\xi)$, then for any fixed $\delta$, there exists $d_\delta$ such that $\alpha\delta \in [2^{d_\delta}\tau, 2^{d_\delta+2}\tau) : \alpha \xleftarrow{\$} \mathsf{logU}(\xi, 2\xi-1)$. Here we split this interval into 2 parts: $C_0 := [2^{d_\delta}\tau, 2^{d_\delta+1}\tau); C_1 := [2^{d_\delta+1}\tau, 2^{d_\delta+2}\tau)$, and define 3 disjoint subsets $S_{C_0}^{\delta}, S_{C_1}^{\delta}, S_{C_0,C_1}^{\delta} \subset [\tau]'$. We say $i \in S_{C_0}^{\delta}$, if for all $\alpha \in [\xi, 2\xi)$ such that $\alpha\delta \in B_i$, we have $\alpha\delta \in C_0$; similarly $i \in S_{C_1}^{\delta}$, if $\forall\alpha$ such that $\alpha\delta \in B_i$, we have $\alpha\delta \in C_1$; for the last one, we define it as there exist $\alpha_0, \alpha_0 \in [\xi, 2\xi)$ satisfying $\alpha_0\delta, \alpha_1\delta \in B_i$ and $\alpha_0\delta \in C_0, \alpha_1\delta \in C_1$.

Now we calculate $p_{i,\delta}$(for ease, we denote $W = \sum_{j=\xi}^{2\xi-1} 1/j$, and it is straightforward that $W \approx \ln 2 = O(1)$), if $i \in S^{\delta}_{C_0}$, then:

$$\begin{aligned}
p_{i,\delta} &= \Pr[\alpha\delta \in B_i : \alpha \xleftarrow{\$} \mathsf{logU}(\xi, 2\xi-1)] \\
&= \sum_{j=\lfloor \frac{2^{d_\delta}\tau + 2^{d_\delta}i}{\delta} \rfloor}^{\lceil \frac{2^{d_\delta}\tau + 2^{d_\delta}(i+1)}{\delta} \rceil} \frac{1}{W}\frac{1}{j} + O(\frac{\delta}{2^{d_\delta}\tau}) \\
&= \frac{1}{W}(\ln\lceil \frac{2^{d_\delta}\tau + 2^{d_\delta}(i+1)}{\delta} \rceil - \ln\lfloor \frac{2^{d_\delta}\tau + 2^{d_\delta}i}{\delta} \rfloor) + O(\frac{\delta}{2^{d_\delta}\tau}) \\
&= \frac{1}{W}(\ln \frac{2^{d_\delta}\tau + 2^{d_\delta}(i+1)}{\delta} - \ln \frac{2^{d_\delta}\tau + 2^{d_\delta}i}{\delta}) + O(\frac{\delta}{2^{d_\delta}\tau}) \\
&= \frac{1}{W}(\ln \frac{\tau+i+1}{\tau+i}) + O(\frac{\delta}{2^{d_\delta}\tau}) = \frac{1}{W}(\ln \frac{\tau+i+1}{\tau+i}) + O(\frac{1}{\xi}).
\end{aligned}$$

Being of little change, we have that for $i \in S^{\delta}_{C_1}$,

$$p_{i,\delta} = \sum_{j=\lfloor \frac{2^{d_\delta+1}\tau + 2^{d_\delta+1}i}{\delta} \rfloor}^{\lceil \frac{2^{d_\delta+1}\tau + 2^{d_\delta+1}(i+1)}{\delta} \rceil} \frac{1}{W}\frac{1}{j} = \frac{1}{W}(\ln \frac{\tau+i+1}{\tau+i}) + O(\frac{1}{\xi}).$$

and similarly, for $i \in S^{\delta}_{C_0,C_1}$,

$$p_{i,\delta} = \sum_{\xi}^{\lceil \frac{2^{d_\delta}\tau + 2^{d_\delta}(i+1)}{\delta} \rceil} \frac{1}{W}\frac{1}{j} + \sum_{\lfloor \frac{2^{d_\delta+1}\tau + 2^{d_\delta+1}(i+1)}{\delta} \rfloor}^{2\xi-1} \frac{1}{W}\frac{1}{j} = \frac{1}{W}(\ln \frac{\tau+i+1}{\tau+i}) + O(\frac{1}{\xi}).$$

Now we replace $\delta$ with $\delta_0, \delta_1$ and we have that, for any $i \in [\tau]'$,

$$|p_{i,\delta_0} - p_{i,\delta_1}| \leq O(\frac{1}{\xi}) \Rightarrow \sum_{i=0}^{\tau-1} |p_{i,\delta_0} - p_{i,\delta_1}| \leq \tau O(\frac{1}{\xi}) \leq \mathsf{negl}.$$

which completes the entire proof. ∎

## 4.4 ORE with smoothed CLWW Leakage

We start by defining the security we target via a smoothed CLWW leakage function. Then we recall a primitive for our construction called a *property-preserving hash (PPH)*

*function*, and state and analyze our ORE construction using a PPH. In a later section we instantiate the PPH to complete the construction. Next, we give variant constructions with trade-offs between efficiency and leakage.

Now We define the non-adaptive version of the leakage profile for our construction. The leakage profile takes in input a vector of messages $\vec{m} = (m_1, \ldots, m_q)$ and produces the following:

$$\mathcal{L}_f(m_1, \ldots, m_q) := (\forall 1 \leq i, j, k \leq q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathsf{msdb}(m_i, m_j) = \mathsf{msdb}(m_i, m_k)).)$$

By definition, it's easy to note that $\mathcal{L}_f$ leaks strictly less than CLWW. Except for the order of underlying plaintexts, it only leaks whether the position of $\mathsf{msdb}(m_i, m_j)$ and $\mathsf{msdb}(m_i, m_j)$ are the same, therefore the leakage profile preserve consistent if we left-shift all the plaintexts by one bit, which referring to $\mathcal{L}_f(\vec{m}) = \mathcal{L}_f(2\vec{m})$. Thus, $\mathcal{L}_f$ is smoothed CLWW.

### 4.4.1 Property Preserving Hash

Our construction will depend on a tool – *property preserving hash (PPH)*, which is essentially a property-preserving encryption scheme Pandey and Rouselakis [2012] without the decryption algorithm. In this section we recall the syntax and security of a PPH.

**Definition 4.4.1** *A* property-preserving hash (PPH) scheme *is a tuple of algorithms* $\Gamma = (\mathcal{K}_h, \mathcal{H}, \mathcal{T})$ *with the following syntax:*

- *The key generation algorithm* $\mathcal{K}_h$ *is randomized, takes as input* $1^\lambda$ *and emits two outputs* $(\mathsf{hk}, \mathsf{tk})$ *that we refer to as the* hash key $\mathsf{hk}$ *and* test key $\mathsf{tk}$*. These implicitly define a domain* $D$ *and range* $R$ *for the hash.*

- *The evaluation algorithm* $\mathcal{H}$ *is randomized, takes as input the hash key* $\mathsf{hk}$*, an input* $x \in D$*, and emits a single output* $h \in R$ *that we refer to as the* hash of $x$*.*

- *The test algorithm* $\mathcal{T}$ *is deterministic, takes as input the test key* $\mathsf{tk}$ *and two hashes* $h_1, h_2$*, and emits a bit.*

| |
|---|
| Game $\text{IND}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A})$: |
| $(\mathsf{hk}, \mathsf{tk}) \xleftarrow{\$} \mathcal{K}_h(1^\lambda)$; $x^* \xleftarrow{\$} \mathcal{A}(\mathsf{tk})$ |
| $h_0 \xleftarrow{\$} \mathcal{H}(\mathsf{hk}, x^*)$; $h_1 \xleftarrow{\$} R$; $b \xleftarrow{\$} \{0,1\}$; $b' \xleftarrow{\$} \mathcal{A}^{\text{HASH}}(\mathsf{tk}, x^*, h_b)$ |
| Return $(b \stackrel{?}{=} b')$ |
| HASH$(x)$: |
| If $P(x^*, x) = 1$ or $P(x, x^*) = 1$, then $h \leftarrow \perp$, Else $h \xleftarrow{\$} \mathcal{H}(\mathsf{hk}, x)$ |
| Return $h$ |

Table 4.3: Game $\text{IND}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A})$.

**Correctness of PPH schemes.** Let $P$ be a predicate on pairs of inputs. We define correctness of a PPH $\Gamma$ with respect to $P$ via the game $\text{COR}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A})$, which is as follows: It starts by running $(\mathsf{hk}, \mathsf{tk}) \xleftarrow{\$} \mathcal{K}_h(1^\lambda)$ and gives $\mathsf{tk}$ to $\mathcal{A}$. Then $\mathcal{A}$ outputs $x, y$. The game computes $h \xleftarrow{\$} \mathcal{H}(\mathsf{hk}, x), h' \xleftarrow{\$} \mathcal{H}(\mathsf{hk}, y)$ and outputs 1 if $\mathcal{T}(\mathsf{tk}, h, h') \neq P(x, y)$. We say that $\Gamma$ is *computationally correct with respect to* $P$ if for all efficient $\mathcal{A}$, $\Pr[\text{COR}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A}) = 1]$ is a negligible function of $\lambda$.

**Security of PPH schemes.** We recall a simplified version of the security definition for PPH that is a weaker version of PPE security defined by Pandey and Rouselakis Pandey and Rouselakis [2012]. The definition is a sort of semantic security for random messages under chosen-plaintext attacks, except that the adversary is restricted from making certain queries.

**Definition 4.4.2** *Let $P$ be some predicate and $\Gamma = (\mathcal{K}_h, \mathcal{H}, \mathcal{T})$ be a PPH scheme with respect to $P$. For an adversary $\mathcal{A}$ we define the game* $\text{IND}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A})$ *in Figure 4.3. The* restricted-chosen-input advantage of $\mathcal{A}$ *is defined to be* $\mathsf{Adv}^{\mathsf{pph}}_{\Gamma,P,\mathcal{A}}(\lambda) = 2\Pr[\text{IND}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A}) = 1] - 1$. *We say that* $\Gamma$ *is* restricted-chosen-input secure *if for all efficient adversaries $\mathcal{A}$,* $\mathsf{Adv}^{\mathsf{pph}}_{\Gamma,P,\mathcal{A}}(\lambda)$ *is negligible.*

### 4.4.2 ORE from PPH

**Construction.** Let $F : K \times ([n] \times \{0,1\}^n) \to \{0,1\}^\lambda$ be a secure PRF. Let $P(x, y) = \mathbf{1}(x = y + 1)$ be the predicate that outputs 1 if and only if $x = y + 1$, and let $\Gamma = (\mathcal{K}_h, \mathcal{H}, \mathcal{T})$ be a PPH scheme with respect to $P$. In our construction, we interpret the output of $F$ as a $\lambda$-bit integer, which is also the input domain of the PPH $\Gamma$. We define

our ORE scheme $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ as follows:

- $\mathcal{K}(1^\lambda, M)$: On input the security parameter and message space $[M]$, the algorithm chooses a key $k$ uniformly at random for $F$, and runs the key generation algorithm of the property preserving hash function $\Gamma.\mathcal{K}_h$ to obtain the hash and test keys $(\mathsf{hk}, \mathsf{tk})$. It sets $\mathsf{ck} \leftarrow \mathsf{tk}$, $\mathsf{sk} \leftarrow (k, \mathsf{hk})$ and outputs $(\mathsf{ck}, \mathsf{sk})$.

- $\mathsf{E}(\mathsf{sk}, m)$: On input the secret key $\mathsf{sk}$ and a message $m$, the algorithm writes the binary representation as $m$ as $(b_1, \ldots, b_n)$, and then for $i = 1, \ldots, n$, it computes:

$$u_i = F(k, (i, b_1 b_2 \cdots b_{i-1} || 0^{n-i+1})) + b_i \mod 2^\lambda, \quad t_i = \Gamma.\mathcal{H}(\mathsf{hk}, u_i).$$

  We note that $u_i$ is computed by treating the PRF output as a member of $\{0, \ldots, 2^\lambda - 1\}$. Then it chooses a random permutation $\pi : [n] \to [n]$, and sets $v_i = t_{\pi(i)}$. The algorithm outputs $\mathsf{CT} = (v_1, \ldots, v_n)$.

- $\mathcal{C}(\mathsf{ck}, \mathsf{CT}_1, \mathsf{CT}_2)$: on input the public parameter, two ciphertexts $\mathsf{CT}_1, \mathsf{CT}_2$ where $\mathsf{CT}_1 = (v_1, \ldots, v_n), \mathsf{CT}_2 = (v'_1, \ldots, v'_n)$, the algorithm runs $\Gamma.\mathcal{T}(\mathsf{tk}, v_i, v'_j)$ and $\Gamma.\mathcal{T}(\mathsf{tk}, v'_i, v_j)$ for every $i, j \in [n]$. If there exists a pair $(i^*, j^*)$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v_{i^*}, v'_{j^*}) = 1$, then the algorithm outputs 1, meaning $m_1 > m_2$; else if there exists a pair $(i^*, j^*)$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v'_{i^*}, v_{j^*}) = 1$, then the algorithm outputs 0, meaning $m_1 < m_2$; otherwise it outputs $\perp$, meaning $m_1 = m_2$.

**Correctness.** For two messages $m_1, m_2$, let $(b_1, \ldots b_n)$ and $(b'_1, \ldots, b'_n)$ be their binary representations. Assuming $m_1 > m_2$, there must exists a unique index $i^* \in [n]$ such that $u_i = u'_i + 1$. Therefore correctness of $\Pi$ is followed by correctness of PPH. We can use the same argument for the case $m_1 = m_2$ and $m_1 < m_2$. What is more interesting is its simulation based security, as it is the foundation for parameter hiding ORE, formally:

**Theorem 4.4.3** *Assuming $F$ is a secure PRF and $\Gamma$ is restricted-chosen-input secure, $\Pi$ is $\mathcal{L}_\mathsf{f}$-non-adaptively-simulation secure.*

**Proof:** We use a hybrid argument, and define a sequence of hybrid games as follows:

- $\mathsf{H}_{-1}$: Real game $\mathrm{REAL}^{\mathsf{ore}}_{\Pi}(\mathcal{A})$;
- $\mathsf{H}_0$: Same as $\mathsf{H}_{-1}$, except replacing PRF $F_k(\cdot)$ by a truely random function $F^*$ in the encryption oracle;
- $\mathsf{H}_{i\cdot q+j}$ Depend on a predicate $\mathsf{Switch}_{(i,j)}$ which is define below. If $\mathsf{Switch}_{(i,j)} = 0$, then $\mathsf{H}_{i\cdot q+j} = \mathsf{H}_{i\cdot q+j-1}$, else in procedure of $\mathsf{E}(m_j)$, $u_i^j$ is replaced by a random string.

From the high level, we establish the proof by showing show that any adjacent hybrids are indistinguishable, and then we construct an efficient simulator $S$ such that the output of $\mathsf{H}_{\mathsf{qn}}$ and $\mathrm{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}_{\mathsf{f}}}(\mathcal{A},\mathcal{S})$ are statistically identical. For the predicate, we say $\mathsf{Switch}_{i,j} = 1$ if $\forall k \in [q], \mathsf{msdb}(m_j, m_k) \neq i$, and 0 otherwise. We note that when $\mathsf{Switch}_{i,j} = 0$, there exists $u_i^k$ such that $u_i^j = u_i^k \pm 1$, the relation which can be detected by the test algorithm of PPH(for the $i$-th bit of $m_j$, we call such a bit a leaky bit), which means we cannot replace it with random string, otherwise adversary can trivially distinguish it. In the following we firstly prove any adjacent objects are computational indistinguishable.

**Lemma 4.4.4** *Assuming $\Gamma$ is restricted-chosen-input secure, then for any $k \in [qn]$ $\mathsf{H}_{k-1} \overset{\mathsf{comp}}{\approx} \mathsf{H}_k$.*

Due to the security of PRF, it's trivial that $\mathsf{H}_{-1} \overset{\mathsf{comp}}{\approx} \mathsf{H}_0$, and for any $k > 0$(for ease, $k = i^* \cdot q + j^*$ where $i^* \in [n-1], j^* \in [q]$ ), it suffices to show $\mathsf{H}_{k-1} \overset{\mathsf{comp}}{\approx} \mathsf{H}_k$ under the condition $\mathsf{Switch}_{i^*,j^*} = 1$($\mathsf{Switch}_{i^*,j^*} = 0$ implies $\mathsf{H}_{k-1} = \mathsf{H}_k$). We prove that if there exists adversary $\mathcal{A}$ that distinguish $\mathsf{H_k}$ from $\mathsf{H_{k-1}}$ with noticeable advantage $\epsilon$, then we can construct a simulator $\mathcal{B}$ wins the restricted-chosen-input game with $\epsilon$-$\mathsf{negl}$. Here is the description of $\mathcal{B}$. Firstly it runs $\mathrm{IND}^{\mathsf{pph}}_{\Gamma}$, and sends $\mathsf{tk}$ as the comparison key $\mathsf{ck}$ to $\mathcal{A}$. After receiving a sequence of plaintext $m_1, \ldots, m_q$, it picks a random function $F^*$(using the lazy sampling technique for instance), sets $X^* = F^*(i^*, b_1^{j^*} b_2^{j^*} \cdots b_{i^*-1}^{j^*} || 0^{n-i^*+1}) + b_{i^*}^{j^*}$ where $b_i^j$ is the $i$-th bit of $m_j$. Then it sends $X^*$ to its challenger in restricted-chosen-input game and gets back $T$ as the challenge term. To simulate the encryption oracle, $\mathcal{B}$ works as follows:

1. $(i', j') > (i^*, j^*)$(here using a natural order for tuples, $(i, j) > (i', j')$ iff $iq + j > i'q + j'$ ), $\mathcal{B}$ computes:

$$u_{i'}^{j'} = F^*(i^*, b_1^{j'} b_2^{j'} \cdots b_{i'-1}^{j'} || 0^{n-i'+1}) + b_{i'}^{j'}; t_{i'}^{j'} = \Gamma.\mathcal{H}(\mathsf{hk}, u_{i'}^{j'}).$$

2. $(i', j') < (i^*, j^*) \cap \mathsf{Switch}_{i',j'} = 0$, then same as above, else $u_{i'}^{j'} \xleftarrow{\$} \{0,1\}^\lambda, t_{i'}^{j'} = \Gamma.\mathcal{H}(\mathsf{hk}, u_{i'}^{j'})$.

3. sets $t_{i^*}^{j^*} = T$, and $\forall j \in [q]$, picks a random permutation $\pi_j$ and outputs the ciphertexts $\mathsf{CT}_j = (t^j_{\pi_j(1)}, \ldots, t^j_{\pi_j(n)})$.

Finally, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs[4].

Since $F^*$ is a random function, $\Pr[u_{i'}^{j'} = X^* \pm 1]$ is negligible for all $(i', j') \neq (i^*, j^*)$, which means $\mathcal{B}$ fails to simulate the encryption oracle with only negligible probability. Besides, when $T = \Gamma.\mathcal{H}(\mathsf{hk}, X^*)$, $\mathcal{B}$ properly simulates $\mathsf{H}_{k-1}$, and if $T$ is random, then $\mathcal{B}$ simulates $\mathsf{H}_k$(due to the PRF security, the distribution of $\Gamma.\mathcal{H}(\mathsf{hk}, r) : r \xleftarrow{\$} \{0,1\}^\lambda$ is computationally close to a random variable that uniformly sampled from the range of $\Gamma$). Hence, if $\mathbf{Adv}(\mathcal{A})$ is noticeable, then $\mathcal{B}$'s advantage is also noticeable.

In the following, we describe an efficient simulator $S$ such that the output of $\mathsf{H}_{qn}$ and $\mathrm{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}_\mathsf{f}}(\mathcal{A}, \mathcal{S})$ are statistically identical. Roughly speaking, we note that $\mathsf{Switch}_{i,j} = 1$ means that $i$-th bit of $m_j$ is not a leaky bit, indicating that its value would not affect the leakage profile whp. Hence, it suffices to only simulate the leaky bit of each individual message, which can be extracted by $\mathcal{L}_f$, and sets the rest just as random string. Due to the final random permutations, $\mathsf{H}_{qn}$ and $\mathrm{SIM}^{\mathsf{ore}}_{\Pi,\mathcal{L}_\mathsf{f}}(\mathcal{A}, \mathcal{S})$ are statistically identical. Formally:

**Description of the simulator**. For fixed a message set $\mathcal{M} = \{m_1, \ldots, m_q\}$ (without loss of generality, we assume $m_1 > \ldots > m_q$), the simulator $\mathcal{S}$ is given the leakage information $\mathcal{L}_f(m_1, \ldots, m_q)$. $\mathcal{S}$ firstly keeps a $q \times n$ matrix $\mathcal{B}$ and runs a recursive algorithm $\mathsf{FillMatrix}(1, 1, q)$ to fill in the entries, as follows:

[4]We note that $\mathcal{B}$ does not have $\mathsf{hk}$, what it does is to call the hash oracle

- If $j = k$, then $\forall i' \in [i, n]$, $\mathcal{B}[j][i'] = r$ where $r \xleftarrow{\$} \{0,1\}^\lambda$;
- Else, it proceeds as follows:
  - searches the smallest $j^* \in [j, k]$ s.t. $P(m_j, m_{j^*}) = P(m_j, m_k)$;
  - sets $\mathcal{B}[j'][i] = r', \forall j' \in [j, j^* - 1]; \mathcal{B}[j'][i] = r' - 1, \forall j' \in [j^*, k]$, where $r' \xleftarrow{\$} \{0,1\}^\lambda$;
  - runs $\mathsf{FillMatrix}(i+1, j, j'-1)$ and $\mathsf{FillMatrix}(i+1, j', k)$ recursively.

More concretely, our recursive algorithm is to fill in the entries by

$$\mathsf{FillMatrix}(i, j, k),\ \forall i \in [n], j \leq k \in [q].$$

Then $\mathcal{S}$ runs $\Gamma.\mathcal{K}_h(1^\lambda)$ and gets the keys $\mathsf{tk}, \mathsf{hk}$, and sets $t_{i,j} = \Gamma.\mathcal{H}(\mathsf{hk}, \mathcal{B}[j][i])$, $\forall i \in [n], j \in [q]$. Finally, $\mathcal{S}$ samples random permutations $\pi_j$, outputs $\mathsf{CT}_j$ as $\mathsf{CT}_j = (t^j_{\pi_j(1)}, \ldots, t^j_{\pi_j(n)})$ We note that the FillMatrix algorithm terminates after at most $qn$ steps as each cell will not be written twice, hence $\mathcal{S}$ is an efficient simulator.

Finally we claim that $\mathcal{S}$ properly simulates the relevant games. We first observe that the simulator identifies how many leaked bits (prefixes) there are for the messages $m_1, \ldots, m_q$. Recall that if messages $m_1, \ldots, m_q$ share the same prefix up to the $\ell - 1$-th bit, and if there exists (the first ) $i^*$ such that $\mathsf{msdb}(m_1, m_{i^*}) = \mathsf{msdb}(m_1, m_q)$, then we can conclude that $\{m_1, \ldots, m_{i^*-1}\}$ has 1 on their $\ell$-th bit, and and $\{m_{i^*}, \ldots, m_q\}$ has 0 on their $\ell$-th bit. This way the $\ell$-th bit of these messages are leaked. The simulator recursively identifies other leaked bits for these two sets. At the end, for each message, how many prefixes whose next bits are leaked will be identified. As this information will also be identified in the hybrid $\mathsf{H}_{qn}$. So a random permutation (for $\mathsf{H}_{qn}$ and the simulation) will hide these leaked prefixes, except the total number. Thus, our simulation is identical to $\mathsf{H}_{qn}$, and we establish the entire proof. ∎

## 4.5 PPH from Bilinear Maps

We construct a PPH scheme for the predicate $P$ required in our ORE construction. That is, $P(x, y) = 1$ if and only if $x = y + 1$.

We let $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \mathbb{Z}_p$ be a PRF, where $p$ is a prime to be determined at key generation.

**Construction.** We now define our PPH $\Gamma = (\mathcal{K}_h, \mathcal{H}, \mathcal{T})$.

- $\mathcal{K}_h(1^\lambda)$ This algorithm takes the security parameter as input. It samples descriptions of prime-order $p$ groups $\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T$, generators $g \in \mathbb{G}, \hat{g} \in \hat{\mathbb{G}}$, a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$. It then chooses $k \xleftarrow{\$} \{0,1\}^\lambda$. It sets the hash key $\mathsf{hk} \leftarrow (k, g, \hat{g})$, the test key $\mathsf{tk} \leftarrow (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e)$, a description of the bilinear map and groups, and outputs $(\mathsf{hk}, \mathsf{tk})$.

- $\mathcal{H}(\mathsf{hk}, x)$ This algorithm takes as input the hash key $\mathsf{hk}$, an input $x$, picks two random non-zero $r_1, r_2 \in \mathbb{Z}_p$ and outputs

$$\mathcal{H}(\mathsf{hk}, x) = (g^{r_1}, g^{r_1 \cdot F(k,x)}, \hat{g}^{r_2}, \hat{g}^{r_2 \cdot F(k,x+1)}).$$

- $\mathcal{T}(\mathsf{tk}, h_1, h_2)$ To test two hash values $(A_1, A_2, B_1, B_2)$ and $(C_1, C_2, D_1, D_2)$, $\mathcal{T}$ outputs 1 if

$$e(A_1, D_2) = e(A_2, D_1),$$

  and otherwise it outputs 0.

Hence the domain $D$ is $\{0,1\}^\lambda$ and the range $R$ is $(\mathbb{G}^2, \hat{\mathbb{G}}^2)$.

**Correctness.** Correctness reduces to testing if $F(k, y+1) = F(k, x)$. If $x = y+1$ then this always holds. If not, then it is easily shown that finding $x, y$ with this property (and without knowing the key) with non-negligible probability leads to an adversary that contradicts the assumption that $F$ is a PRF.

**Security.** We prove that PPH is restricted-chosen-input secure, assuming that $F$ is a PRF and the SXDH assumption holds. We can now state and prove our security theorem.

**Theorem 4.5.1** *Our PPH $\Gamma$ is restricted-chosen-input secure, assuming $F$ is a PRF and the SXDH assumption hold with respect to the appropriate groups and pairing.*

**Proof:** We use a hybrid argument. Let $(A_1, A_2, B_1, B_2) \in \mathbb{G}^2 \times \hat{\mathbb{G}}^2$ denote the challenge hash value given to the adversary during the real game $\mathsf{H}_0 = \mathrm{IND}^{\mathsf{pph}}_{\Gamma,P}(\mathcal{A})$. Additionally, let $R$ be a random element of $\mathbb{G}$, $\hat{R}$ be a random element of $\hat{\mathbb{G}}$, both independent of the rest of the random variables under consideration. Then we define the following hybrid experiments:

- $\mathsf{H}_1$: At the start of the game, a uniformly random function $F^* \xleftarrow{R} \mathsf{Funs}[\{0,1\}^\lambda, \{0,1\}^\lambda]$ is sampled instead of the PRF key $K$, the rest remain unchanged.
- $\mathsf{H}_2$: The challenge hash value is $(A_1, R, B_1, B_2)$, where $R \xleftarrow{\$} \mathbb{G}$.
- $\mathsf{H}_3$: The challenge hash value is $(A_1, R, B_1, \hat{R})$, where $R \xleftarrow{\$} \hat{\mathbb{G}}$.

In $\mathsf{H}_3$, the adversary is given a random element from the range $\mathcal{R}$. Therefore,

$$\mathsf{Adv}^{\mathsf{pph}}_{\Gamma,P,\mathcal{A}}(\lambda) = |Pr[\mathsf{H}_0 = 1] - Pr[\mathsf{H}_3 = 1]|.$$

To prove $\mathsf{H}_0$ is indistinguishable from $\mathsf{H}_3$, we show that each step of the hybrid is indistinguishable from the next. First, it is apparent that $\mathsf{H}_0$ and $\mathsf{H}_1$ are computational indistinguishable by the PRF security, then:

**Lemma 4.5.2** $\mathsf{H}_1 \approx \mathsf{H}_2$ *under the SXDH assumption.*

Let $\mathcal{A}$ be an adversary playing the PPH security game, and let

$$\epsilon = |Pr[\mathsf{H}_1 = 1] - Pr[\mathsf{H}_2 = 1]|.$$

Then we can build adversary $\mathcal{B}$ that solves SXDH with advantage $\epsilon$. $\mathcal{B}$ is given as input $(g, \hat{g}, B, C)$ and the challenge term $T$. $\mathcal{B}$ works as follows:

- $\mathcal{B}$ sets $\mathsf{tk} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e)$ and sends it to $\mathcal{A}$. After receiving $x^* \xleftarrow{\$} \mathcal{A}(\mathsf{tk})$ it simulates a random function $F^*$ via lazy sampling, and it will implicitly set $F^*(x^*) = b$,

the discrete logarithm of $B$. It prepares the challenge as by selecting $r^* \xleftarrow{\$} \mathbb{Z}_p$ and computing

$$A_1 = g^c, A_2 = T, B_1 = \hat{g}^{r^*}, B_2 = \hat{g}^{r^* F^*(x^*+1)},$$

and runs $\mathcal{A}$ on input $\mathsf{tk}, x^*, (A_1, A_2, B_1, B_2)$.

- To answer hash query for $x \neq x^*$ from $\mathcal{A}$, $\mathcal{B}$ calculates $F^*(x)$ and $F^*(x+1)$ (note that $x, x+1 \neq x^*$). Then $\mathcal{B}$ picks $r_1, r_2$ randomly and computes:

$$\mathcal{H}(x) = g^{r_1}, g^{r_1 \cdot F^*(x)}, \hat{g}^{r_2}, \hat{g}^{r_2 \cdot F^*(x+1)};$$

If $\mathcal{A}$ queries $x = x^*$, $\mathcal{B}$ calculates $F^*(x^*+1)$, picks $r'_1, r'_2 \xleftarrow{\$} \mathbb{Z}_p$, and computes

$$\mathcal{H}(x^*) = g^{r'_1}, B^{r'_1}, \hat{g}^{r'_2}, \hat{g}^{r'_2 \cdot F^*(x^*+1)};$$

- Finally $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

We note that in $\mathcal{A}$'s view, without querying $\mathcal{A}(x^* - 1)$, $\mathcal{B}$ simulates the game properly. If $T = g^{bc}$, then $\mathcal{B}$ simulates $\mathsf{H}_1$, and if $T$ s random then it simulates $\mathsf{H}_2$. Hence if $\mathcal{A}$ has an advantage $\epsilon$ in distinguishing $\mathsf{H}_1$ and $\mathsf{H}_2$, then $\mathcal{B}$ has the same advantage to break SXDH assumption.

We also have the following lemma:

**Lemma 4.5.3** $\mathsf{H}_2 \approx \mathsf{H}_3$ *under the SXDH assumption.*

The proof is exactly the same as the prior hybrid step, except in the group $\hat{\mathbb{G}}$ part of the hash instead of $\mathbb{G}$. We omit the details.

Collecting the steps completes the proof of Theorem 4.5.1. ▌

## 4.6 Further reducing leakage

We now give a generalized construction that results in strictly less leakage, and for some parameter settings, a more efficient comparison algorithm and shorter ciphertexts. At a high level, we modify our main constructions to work with blocks of $d$ bits rather than bit-by-bit, and design a generalized type of PPH for our construction. We note that curiously when $d = 2$, the efficiency also improves.

### 4.6.1 Generalized ORE

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. Let $P_d(x_1, x_2) = x_1 \in \{x_2 + 1, \ldots, x_2 + 2^d - 1\}$ let $\Gamma = (\mathcal{K}_h, \mathcal{H}, \mathcal{T})$ be a generalized $PPH$ scheme with respect to predicate $P_d$ (a construction from SXDH is given in the follow section).

We define our ORE scheme $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ as follows:

- $\mathcal{K}(1^\lambda)$: on input the security parameter $\lambda$, the algorithm picks a uniform key $\mathsf{k} \in \mathcal{K}$ for the PRF $F$ and runs the Setup algorithm of the generalized $PPH$ $\Gamma.\mathcal{K}_h$ to obtain the hash and test keys $(\mathsf{hk}, \mathsf{tk})$. It sets the comparison key $\mathsf{ck} = \mathsf{tk}$ and secret key $\mathsf{sk} = (k, \mathsf{hk})$.

- $\mathsf{E}(\mathsf{sk}, m)$: on input a secret key $\mathsf{sk}$ and a message $m \in \{0,1\}^n$, encryption parses $m$ as $m = b_1 || \ldots || b_{n/d}$ (later we denote $\ell = n/d$), where $b_i \in \{0,1\}^d$ Then it computes

$$u_i = F(k, (i, b_1 b_2 \cdots b_{(i-1)d} || 0^{n-(i-1)d})) + b_i, \quad t_i = \Gamma.\mathcal{H}(\mathsf{hk}, u_i).$$

  (Here we abuse the notation $b_i$ as an integer value according to its binary representation.) Then it chooses a random permutation $\pi : [\ell] \to [\ell]$, and sets $v_i = t_{\pi(i)}$. The algorithm outputs $\mathsf{CT} = (v_1, \ldots, v_\ell)$.

- $\mathcal{C}(\mathsf{ck}, \mathsf{CT}_1, \mathsf{CT}_2)$: on input the public parameter, two ciphertexts $\mathsf{CT}_1, \mathsf{CT}_2$ where

$$\mathsf{CT}_1 = (v_1, \ldots, v_\ell); \mathsf{CT}_2 = (v'_1, \ldots, v'_\ell),$$

  the algorithm runs the test algorithm $\Gamma.\mathcal{T}(\mathsf{tk}, v_i, v'_j)$ and $\Gamma.\mathcal{T}(\mathsf{tk}, v'_i, v_j)$ for every $i, j \in [\ell]$. If there exists a pair $(i^*, j^*)$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v_{i^*}, v'_{j^*}) = 1$, then the algorithm outputs 1, meaning $m_1 > m_2$; else if there exists a pair $(i^*, j^*)$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v'_{i^*}, v_{j^*}) = 1$, then the algorithm outputs 0, meaning $m_1 < m_2$; otherwise it outputs it outputs $\perp$, meaning $m_1 = m_2$.

**Correctness of the generalized ORE.** For two messages $m_1, m_2$, let $(b_1, \ldots b_\ell)$ and $(b'_1, \ldots, b'_\ell)$ be their $d$-bit block representations. We know that if $m_1 > m_2$, then there must exists a unique index $i^* \in [\ell]$ such that the prefixes of their $d$-bit block representations up to $i^*$, say $u = (b_1, \ldots, b_{i^*})$, $u' = (b'_1, \ldots, b'_{i^*})$, satisfy the following relation: $u = u' + i, i = 1, \ldots, 2^d$. By the correctness of the generalized PPH, we know that, with overwhelming probability:

$$\Gamma.\mathcal{T}(\Gamma.\mathcal{H}(\mathsf{hk}, u), \Gamma.\mathcal{H}(\mathsf{hk}, u')) = 1.$$

We can use the same argument for the case $m_1 < m_2$.

For the case $m_1 = m_2$, we know that all prefixes of the two messages are identical. For this case, the $\mathsf{Test}$ of $\Gamma$ outputs $\perp$ (for all possible pairs) with overwhelming probability. This proves the correctness of our ORE scheme.

**Leakage profile.** Next, we present the leakage profile. For two messages $m_1, m_2$, let $(b_1, \ldots b_\ell)$ and $(b'_1, \ldots, b'_\ell)$ as their $d$-bit block representations, and denote by $\mathsf{msddb}(m_1, m_2)$ their most significant different $d$-bit-block. More precisely,

$$\mathsf{msddb}(m_1, m_2) = \min\{i : b_i \neq b'_i\} \cup \{n/d + 1\}.$$

The leakage profile takes in input a vector of messages $\vec{m} = (m_1, \ldots, m_n)$ and produces

the following:

$$\mathcal{L}_f^*(m_1,\ldots,m_t) := \left\{ \begin{array}{l} \mathbf{1}(m_i < m_j), \\ \mathbf{1}(\mathsf{msddb}(m_i,m_j) = \mathsf{msddb}(m_i,m_k)) \\ \text{for } 1 \le i,j,k \le q \end{array} \right\}.$$

**Theorem 4.6.1** *The generalized ORE scheme* $\Pi$ *is* $\mathcal{L}_{\mathsf{f}}^*$*-non-adaptively-simulation secure, assuming* $F$ *is a secure PRF and* $\Gamma$ *is augmented-restricted-chosen-input secure.*

The proof this theorem is of little change to that of Theorem 4.4.3, and we skip it here.

### 4.6.2 Generalized PPH

In this section, we present a PPH for a family of predicts $P_d$, $d \ge 1$ that generalizes the predicate $P$ above as follows. We let $P_d(x,y) = 1$ if $x \in \{y+1,\ldots,y+2^d-1\}$ and 0 otherwise.

**Construction.** As before, we use a PRF $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$, and we will sometimes view the output of $F$ as the binary representation of a $\lambda$-bit integer. We now describe our PPH $\Gamma^d = (\mathcal{K}_h^d, \mathcal{H}^d, \mathcal{T}^d)$ for the generalized predicate $P_d$, where $d \ge 1$ is a parameter to adjusted.

- $\mathcal{K}_h^d(1^\lambda)$. This algorithm is identical to $\mathcal{K}_h$ given in $\Gamma$.
- $\mathcal{H}^d(\mathsf{hk}, x)$ This algorithm takes as input the hash key $\mathsf{hk}$, an input $x$. For $i = 0,\ldots,2^d-1$ it picks random $r_i \leftarrow \mathbb{Z}_p$, then it samples a random permutation $\pi$ on $[2^d]$ , and then it computes

  $$(A,B) = (g^{r_0}, g^{r_0 \cdot F(K,x)}) \in \mathbb{G} \times \mathbb{G}.$$

  Then, for $i = 1,\ldots,T$ it computes

  $$(X_i, Y_i) = (\hat{g}^{r_i}, \hat{g}^{r_i \cdot F(K, x+\pi(i))}) \in \hat{\mathbb{G}} \times \hat{\mathbb{G}}.$$

  It outputs $(A_0, B_0, X_1, Y_1, \ldots, X_{2^d-1}, Y_{2^d-1})$.

Game $\text{IND}^{\mathsf{pph-aug}}_{\Gamma,P}(\mathcal{A})$:

$(\mathsf{hk}, \mathsf{tk}) \xleftarrow{\$} \mathcal{K}_h(1^\lambda)$; $x^*, y^* \xleftarrow{\$} D$

$(x_1, y_1), \ldots, (x_s, y_s) \xleftarrow{\$} \{(x, y) : \forall i, j \in [s], P(x^*, x_i) = P(y^*, y_i) = 1;\ P(x_i, x_j) = P(y_i, y_j)\}$

$\vec{h_0} = (\mathcal{H}(\mathsf{hk}, x^*), \mathcal{H}(\mathsf{hk}, x_1), \ldots, \mathcal{H}(\mathsf{hk}, x_s))$; $\vec{h_1} = (\mathcal{H}(\mathsf{hk}, y^*), \mathcal{H}(\mathsf{hk}, y_1), \ldots, \mathcal{H}(\mathsf{hk}, y_s))$

$b' \xleftarrow{\$} \mathcal{A}^{\text{HASH}}(\mathsf{tk}, x^*, y^*, x_1, \ldots, x_s, y_1, \ldots, y_s, \vec{h_b})$

Return $(b \stackrel{?}{=} b')$

HASH$(x)$:

If $\exists z \in \{x, y, x_1, \ldots, x_s, y_1, \ldots, y_s\}$, $P(z, x) = 1$ or $P(x, z) = 1$

Then $h \leftarrow \perp$,Else $h \xleftarrow{\$} \mathcal{H}(\mathsf{hk}, x)$

Return $h$

Table 4.4: Game $\text{IND}^{\mathsf{pph-aug}}_{\Gamma,P}(\mathcal{A})$.

- $\mathcal{T}^d(\mathsf{tk}, h_1, h_2)$. The test algorithm parses each $h_j$ as

$$(A^j, B^j, X^j_1, Y^j_1, \ldots, X^j_{2^d-1}, Y^j_{2^d-1})$$

  Then it tests if there exists an $i \in \{1, \ldots, 2^d - 1\}$ such that

$$e(A^1, Y^2_i) = e(B^1, X^2_i).$$

  If it finds such an $i$ it outputs 1, and otherwise it outputs 0.

And the domain $D$ is $\{0,1\}^\lambda$ and the range $R$ is $\mathbb{G}^2 \times \hat{\mathbb{G}}^{2^{d+1}-2}$.

**Correctness.** It is easy to show that $\Gamma_d$ is computationally correct for the predicate $P_d$, via the same methods as with $\Gamma$, assuming that $F$ is a PRF.

**Security.** Our ORE construction will require a slightly stronger version of PPH security illustrated in Figure 4.4. We call this version of PPH security *augmented-restricted-chosen-input* security, and define the advantage of an adversary $\mathcal{A}$ against PPH scheme $\Gamma$ via

$$\mathsf{Adv}^{\mathsf{pph-aug}}_{\Gamma,P,\mathcal{A}}(\lambda) = 2\Pr[\text{IND}^{\mathsf{pph-aug}}_{\Gamma,P} = 1] - 1.$$

We say that $\Gamma$ is *augmented-restricted-chosen-input secure* if for all efficient adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{pph-aug}}_{\Gamma,P,\mathcal{A}}(\lambda)$ is negligible.

**Theorem 4.6.2** *For each $d \geq 1$, our PPH $\Gamma^d$ is augmented-restricted-chosen-input secure, assuming $F$ is a PRF and the SXDH assumption hold with respect to the appropriate groups and pairing.*

The proof of this theorem is very similar to that of Theorem 4.5.1, despite the augmented security definition. It follows via standard game transitions using the SXDH assumptions.

### 4.6.3 Efficiency for small $d$

For generalized ORE, when $d$ is small, $d = 2, 3$ for instance, the efficiency (ciphertext size and pairing operations in each single comparison) is better than basic ORE. We measure the ciphertext size by number of group elements, and calculate the average number of pairing operations needed in a comparison. When $d = 2$, the construction is strictly better than basic construction, and $d = 3$ has some trade-off in ciphertext size and pairing operation with basic ORE.

## 4.7 More efficient comparisons

Our construction from section 4.4 need to evaluate the PPH test algorithm $O(n^2)$ times to compare to numbers. In this part, we present a variant ORE achieving better efficiency but with a weaker leakage profile, which only requires $O(n)$ pairings in each individual comparison. And what's more interesting is that this weaker leakage profile is also smoothed CLWW, which means we can construct a parameter hiding ORE with much better efficiency. High level speaking, instead of sampling a fresh random permutation in each encryption procedure, we choose a *fixed* permutation $\pi$ for all ciphertexts (and stores $\pi$ in the key). With this modification, we only need $O(n)$ PPH test evaluations for each comparison.

**Construction.** Let $F$ be a secure PRF with the same syntax as above, let $P(x, y) = \mathbf{1}(x = y + 1)$ be the relation predicate that outputs 1 if and only if $x = y + 1$, and let $\Gamma = (\mathcal{K}_h, \mathcal{H}, \mathcal{T})$ be a PPH scheme with respect to $P$, as before. We define our ORE scheme $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ as follows:

- $\mathcal{K}(1^\lambda)$: On input the security parameter, the algorithm chooses a key $k$ uniformly at random for $F$, runs $\Gamma.\mathcal{K}_h$ to obtain the hash and test keys $(\mathsf{hk}, \mathsf{tk})$, and samples a random permutation $\pi : [n] \to [n]$. It sets $\mathsf{ck} \leftarrow \mathsf{tk}$, $\mathsf{sk} \leftarrow (k, \mathsf{hk}, \pi)$ and outputs $(\mathsf{ck}, \mathsf{sk})$.

- $\mathsf{E}(\mathsf{sk}, m)$: On input the secret key $\mathsf{SK}$ and a message $m$, the algorithm computes the binary representation of $m = (b_1, \ldots, b_n)$, and then calculates:

$$u_i = F(k, (i, b_1 b_2 \cdots b_{i-1} || 0^{n-i+1})) + b_i, \quad t_i = \Gamma.\mathcal{H}(\mathsf{hk}, u_i).$$

  Then it sets $v_i = t_{\pi(i)}$ and outputs $\mathsf{CT} = (v_1, \ldots, v_n)$.

- $\mathcal{C}(\mathsf{ck}, \mathsf{CT_1}, \mathsf{CT_2})$: on input the public parameter, two ciphertexts $\mathsf{CT_1}, \mathsf{CT_2}$ where $\mathsf{CT_1} = (v_1, \ldots, v_n), \mathsf{CT_2} = (v'_1, \ldots, v'_n)$, the algorithm runs $\Gamma.\mathcal{T}(\mathsf{tk}, v_i, v'_i)$ for every $i \in [n]$. If there exists $i^*$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v_{i^*}, v'_{i^*}) = 1$, then the algorithm outputs 1, meaning $m_1 > m_2$; else if there exists a pair $i^*$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v'_{i^*}, v_{i^*}) = 1$, then the algorithm outputs 0, meaning $m_1 < m_2$; otherwise it outputs it outputs $\perp$, meaning $m_1 = m_2$.

Now, we give the description of the leakage profile, which takes $\vec{m} = \{m_1, \ldots, m_q\}$ as input and produces:

$$\mathcal{L}'_f(m_1, \ldots, m_q) := (\forall 1 \le i, j, k, l \le q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathsf{msdb}(m_i, m_j) = \mathsf{msdb}(m_k, m_l)))$$

Compared to $\mathcal{L}_f$, $\mathcal{L}'_f$ gives extra information that $\mathbf{1}(\mathsf{msdb}(m_i, m_j) = \mathsf{msdb}(m_k, m_l))$ even when $i \neq k$. However, $\mathcal{L}'_f$ is still strictly stronger than CLWW, and for any $\vec{m}$, it's obvious that $\mathcal{L}'_f(\vec{m}) = \mathcal{L}'_f(2\vec{m})$, which gives evidence that $\mathcal{L}'_f$ is also smoothed CLWW. And for its simulation based security, applying exactly the same argument as the proof of Theorem 4.4.3, we can establish the following theorem.

**Theorem 4.7.1** *The ORE scheme* $\Pi$ *is* $\mathcal{L}'_f$*-non-adaptive-simulation secure, assuming* $F$ *is a secure PRF and* $\Gamma$ *is restricted-chosen-input secure.*

## 4.8 Left/Right ORE Construction

In Lewi and Wu [2016], Lewi and Wu introduced a Left/Right framework for ORE. In the usual sense, an ORE encryption algorithm takes a message and outputs a ciphertext, and the comparison algorithm then takes two ciphertexts and outputsthe comparison relation on the two underlying messages. In the left/right framework, the encryption algorithm consists of two functions: a "left" encryption function and a "right" encryption function. Each of these encryption functions takes a message and the secret key, and outputs either a "left" or "right" ciphertext, respectively. Next, instead of taking two ciphertexts, the comparison algorithm takes a left ciphertext and a right ciphertext, and outputs the comparison relation between the two underlying messages. It's obvious that any ORE scheme in the left/right framework can be converted to an ORE scheme in the usual sense, and in the following, we show that our ORE can be converted to the left/right framework. Firstly, we convert our PPH to the left/right framework and based on it we construct the left/right ORE.

### 4.8.1 Left-Right PPH

In this section, we present a variant PPH scheme for predicate $P$ ( $P(x, y) = 1$ if and only if $x = y + 1$ ), called Left/Right PPH (LRPPH). Our construction uses an asymmetric bilinear map and a PRF $F$, as before. Below, we let $F$ be a PRF and $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ be a bilinear pairing over groups $(\mathbb{G}, \hat{\mathbb{G}})$ of prime order $p$, with generators $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$ respectively. Let $F : \{0,1\}^\lambda \times (\{0,1\}^\lambda) \to \{0,1\}^\lambda$ and in the following we will sometimes view the output of $F$ as the binary representation of a $\lambda$-bit integer.

**Construction.** We now define our LRPPH $\Gamma = (\mathcal{K}_h, \mathcal{H}_L, \mathcal{H}_R, \mathcal{T})$.

- $\mathcal{K}_h(1^\lambda)$ This algorithm takes the security parameter as input, generates a bilinear groups map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ with the generators $g, \hat{g}$, and chooses $k \xleftarrow{\$} \{0,1\}^\lambda$. Then it sets the hash key $\mathsf{hk} \leftarrow (k, g, \hat{g})$, the test key $\mathsf{tk} \leftarrow (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e)$, a description of the bilinear map and groups, and outputs $(\mathsf{hk}, \mathsf{tk})$.

- $\mathcal{H}_L(\mathsf{hk}, x)$ This algorithm takes as input the hash key $\mathsf{hk}$, an input $x$, picks a random non-zero $r \in \mathbb{Z}_p$ and outputs

$$\mathcal{H}_L(\mathsf{hk}, x) = (g^r, g^{r \cdot F(k,x)}).$$

- $\mathcal{H}_R(\mathsf{hk}, x)$ This algorithm takes as input the hash key $\mathsf{hk}$, an input $x$, picks a random non-zero $r \in \mathbb{Z}_p$ and outputs

$$\mathcal{H}_R(\mathsf{hk}, x) = (\hat{g}^r, \hat{g}^{r \cdot F(k,x+1)}).$$

- $\mathcal{T}(\mathsf{tk}, h_1, h_2)$ To test two hash values $h_1 = (A_1, A_2)$ and $h_2 = (B_1, B_2)$ where $h_1$ is a left hash value while $h_2$ is a right hash value, $\mathcal{T}$ outputs 1 if

$$e(A_1, B_2) = e(A_2, B_1),$$

  and otherwise it outputs 0.

Hence the domain $D$ is $\{0,1\}^\lambda$ and the range $R$ is $(\mathbb{G}^2, \hat{\mathbb{G}}^2)$.

**Correctness.** The condition tested is equivalent to $F(k, y+1) = F(k, x)$ If $x = y + 1$ then this is obviously true. If not, then it is easily shown that finding $x, y$ with this property with non-negligible probability leads to an adversary that contradicts the assumption that $F$ is a PRF.

**Security of LRPPH.** We give two variant versions of security notion.

**Definition 4.8.1** *Let $P$ be some predicate and $\Gamma = (\mathcal{K}_h, \mathcal{H}_L, \mathcal{H}_R, \mathcal{T})$ be a LRPPH scheme with respect to $P$. For an adversary $\mathcal{A}$ we define the game $\mathrm{IND}^{\mathsf{LRPPH}}_{\Gamma,P}(\mathcal{A})$ in Figure 4.5. The* left-right restricted-chosen-input advantage of $\mathcal{A}$ *is defined to be*

$$\mathsf{Adv}^{\mathsf{LRPPH}}_{\Gamma,P,\mathcal{A}}(\lambda) = (2\Pr[\mathrm{IND}^{\mathsf{L\text{-}PPH}}_{\Gamma,P}(\mathcal{A}) = 1] - 1) + (2\Pr[\mathrm{IND}^{\mathsf{R\text{-}PPH}}_{\Gamma,P}(\mathcal{A}) = 1] - 1).$$

*We say that $\Gamma$ is* left-right restricted-chosen-input secure *if for all efficient adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{LRPPH}}_{\Gamma,P,\mathcal{A}}(\lambda)$ is negligible.*

| Game $\text{IND}^{\mathsf{L\text{-}PPH}}_{\Gamma,P}(\mathcal{A})$: | Game $\text{IND}^{\mathsf{R\text{-}PPH}}_{\Gamma,P}(\mathcal{A})$: |
|---|---|
| $(\mathsf{hk},\mathsf{tk}) \xleftarrow{\$} \mathcal{K}_h(1^\lambda);\ x^* \xleftarrow{\$} D$<br>$h_0 \xleftarrow{\$} \mathcal{H}_L(\mathsf{hk}, x^*);\ h_1 \xleftarrow{\$} \mathbb{G}^2$<br>$b \xleftarrow{\$} \{0,1\}$<br>$b' \xleftarrow{\$} \mathcal{A}^{\text{HASH}_L,\text{HASH}_R}(\mathsf{tk}, x^*, h_b)$<br>Return $(b \stackrel{?}{=} b')$ | $(\mathsf{hk},\mathsf{tk}) \xleftarrow{\$} \mathcal{K}_h(1^\lambda);\ x^* \xleftarrow{\$} D$<br>$h_0 \xleftarrow{\$} \mathcal{H}_R(\mathsf{hk}, x^*);\ h_1 \xleftarrow{\$} \hat{\mathbb{G}}^2$<br>$b \xleftarrow{\$} \{0,1\}$<br>$b' \xleftarrow{\$} \mathcal{A}^{\text{HASH}_L,\text{HASH}_R}(\mathsf{tk}, x^*, h_b)$<br>Return $(b \stackrel{?}{=} b')$ |
| $\text{HASH}_L(x)$:<br>Return $\mathcal{H}_L(x)$ | $\text{HASH}_R(x)$:<br>Return $\mathcal{H}_R(x)$ |
| $\text{HASH}_R(x)$:<br>If $P(x^*, x) = 1$, then $h \leftarrow \perp$<br>Else $h \xleftarrow{\$} \mathcal{H}_R(\mathsf{hk}, x)$<br>Return $h$ | $\text{HASH}_L(x)$:<br>If $P(x, x^*) = 1$, then $h \leftarrow \perp$<br>Else $h \xleftarrow{\$} \mathcal{H}_L(\mathsf{hk}, x)$<br>Return $h$ |

Table 4.5: Game $\text{IND}^{\mathsf{LRPPH}}_{\Gamma,P}(\mathcal{A})$.

**Theorem 4.8.2** *Our PPH $\Gamma$ is left-right restricted-chosen-input secure, assuming $F$ is a PRF and the SXDH assumption hold with respect to the appropriate groups and pairing.*

The proof of this theorem is very similar to that of Theorem 4.5.1, despite the left-right security definition. It follows via standard game transitions using the SXDH assumptions.

### 4.8.2 Left-Right ORE

In this section, we show how to convert our ORE to the left/right framework. Formally:
**Construction.** Let $F : K \times ([n] \times \{0,1\}^n) \to \{0,1\}^\lambda$ be a secure PRF. Let $P(x, y) = \mathbf{1}(x = y + 1)$ be the relation predicate that outputs 1 if and only if $x = y + 1$, and let $\Gamma = (\mathcal{K}_h, \mathcal{H}_L, \mathcal{H}_R, \mathcal{T})$ be a LRPPH scheme with respect to $P$. In our construction, we interpret the output of $F$ as a $\lambda$-bit integer, which is also the input domain of the LRPPH $\Gamma$. We define our LR-ORE scheme $\Pi = (\mathcal{K}, \mathsf{E}_L, \mathsf{E}_R, \mathcal{C})$ as follows:

- $\mathcal{K}(1^\lambda)$: On input the security parameter, the algorithm chooses a key $k$ uniformly at random for $F$, and runs the Setup algorithm of the property preserving hash

function $\Gamma.\mathcal{K}_h$ to obtain the hash and test keys $(\mathsf{hk}, \mathsf{tk})$. It sets $\mathsf{ck} \leftarrow \mathsf{tk}$, $\mathsf{sk} \leftarrow (k, \mathsf{hk})$ and outputs $(\mathsf{ck}, \mathsf{sk})$.

- $\mathsf{E}_L(\mathsf{sk}, m)$: On input the secret key $\mathsf{sk}$ and a message $m$, the algorithm computes the binary representation of $m = (b_1, \ldots, b_n)$, and then calculates:

$$u_i = F(k, (i, b_1 b_2 \cdots b_{i-1} || 0^{n-i+1})) + b_i, t_i = \Gamma.\mathcal{H}_L(\mathsf{hk}, u_i).$$

  Then it chooses a random permutation $\pi : [n] \to [n]$, and sets $v_i = t_{\pi(i)}$. The algorithm outputs $\mathsf{CT}_L = (v_1, \ldots, v_n)$.

- $\mathsf{E}_R(\mathsf{sk}, m)$: On input the secret key $\mathsf{sk}$ and a message $m$, the algorithm computes the binary representation of $m = (b_1, \ldots, b_n)$, and then calculates:

$$u_i = F(k, (i, b_1 b_2 \cdots b_{i-1} || 0^{n-i+1})) + b_i, t_i = \Gamma.\mathcal{H}_R(\mathsf{hk}, u_i).$$

  Then it chooses a random permutation $\pi : [n] \to [n]$, and sets $v_i = t_{\pi(i)}$. The algorithm outputs $\mathsf{CT}_R = (v_1, \ldots, v_n)$.

- $\mathcal{C}(\mathsf{ck}, \mathsf{CT}_1, \mathsf{CT}_2)$: on input the public parameter, one left ciphertexts $\mathsf{CT_1}$ and one right ciphertext $\mathsf{CT_2}$ where

$$\mathsf{CT_1} = (v_1, \ldots, v_n); \mathsf{CT_2} = (v'_1, \ldots, v'_n),$$

  the algorithm runs $\Gamma.\mathcal{T}(\mathsf{tk}, v_i, v_j)$ for every $i, j \in [n]$. If there exists a pair $(i^*, j^*) in [n]^2$ such that $\Gamma.\mathcal{T}(\mathsf{tk}, v_{i^*}, v'_{j^*}) = 1$, then the algorithm outputs 1, meaning $m_1 > m_2$; else output 0, meaning $m_1 \leq m_2$.

Correctness of our scheme is implied by our basic ORE.

Next, we present the corresponding leakage profile, which takes in put two vectors

of message $\vec{m} = \{m_1, \ldots, m_{q_1}\}; \vec{m^*} = \{m_1^*, \ldots m_{q_2}^*\}$ and produce the following:

$$\bar{\mathcal{L}}_f(\vec{m}, \vec{m^*}) := \left\{ \begin{array}{l} \mathbf{1}(m_i > m_j^*), \\ \mathbf{1}(\mathsf{msdb}(m_i, m_k^*) = \mathsf{msdb}(m_i, m_l^*)) \\ \mathbf{1}(\mathsf{msdb}(m_i, m_k^*) = \mathsf{msdb}(m_j, m_k^*)) \\ \text{for } i, j \in [q_1], k, l \in [q_2] \end{array} \right\}.$$

If $\vec{m} = \vec{m^*}$, it's obvious that $\bar{\mathcal{L}_f} = \mathcal{L}_f$. Using a similar argument as the proof of Theorem 4.4.3, we are able to establish the following theorem:

**Theorem 4.8.3** *The ORE scheme* $\Pi$ *is* $\bar{\mathcal{L}}_f$*-non-adaptive-simulation secure, assuming* $F$ *is a secure PRF and* $\Gamma$ *is left-right restricted-chosen-input secure.*

## 4.9 Impossibility of Parameter-Hiding OPE

Here, we show that any order *preserving* encryption (OPE) scheme cannot achieve parameter hiding. The intuition is that in OPE, one can take the difference of ciphertexts, and use this as a proxy for the difference between plaintexts. Because we need the indistinguishability of both very small differences and very large differences, we cannot hope to fully hide the scale.

**Warm-up.** First, we show that OPE cannot have ideal security for even two messages, improving on Boldyreva et al. [2009] which required three messages, and re-proving Chenette et al. [2016]. Our proof is also much simpler, which will allow us to later extend it to the parameter-hiding setting.

**Theorem 4.9.1** *For any OPE scheme with message space* $[0, M-1]$ *and ciphertext space* $[0, C-1]$*, there is an two-message attack with advantage at least* $\Omega(\frac{\log M}{\log C})$ *that runs in time* $\text{poly}(\log M, \log C)$*.*

Since $\log C$ is the bit-length of ciphertexts, this means that the bit-length must be super-polynomial in order to get a non-negligible advantage.

**Proof:** Our attack is as follows. First, choose two random adjacent messages $(i, i+1)$ for $i \in [0, M-2]$. Then submit the following two pairs $(i, i+1), (0, M-1)$. In response, we receive $c_0, c_1$, which are either then encryptions of $i, i+1$, or encryptions of $0, M-1$. For now, we will consider a non-uniform attacker, which is given some advice $L \in [0, C-1]$. The attacker simply computes $c_1 - c_0$, and outputs 1 if and only if the result is greater than $L$.

We now analyze the scheme. Let $W$ be the random variable for $c_1 - c_0$ in the case where $(i, i+1)$ are encrypted for a random $i$. Let $R_i$ be the random variable that represents $c_1 - c_0$ in the case where $i, i+1$ are encrypted for a given $i$. Let $V$ be the random variable representing $c_1 - c_0$ when $1, M$ are encrypted.

We immediately observe that $V = \sum_{i=1}^{M-1} R_i$. We also observe that

$$\begin{aligned}\mathbf{E}[\log W] &= \frac{1}{M-1}\sum_{i=1}^{M-1}\mathbf{E}[\log R_i] = \mathbf{E}[\frac{1}{M-1}\sum_i \log R_i] \\ &\leq \mathbf{E}[\log\left(\frac{1}{M-1}\sum_i R_i\right)] = \mathbf{E}[\log V] - \log(M-1).\end{aligned}$$

We now use the following lemma:

**Lemma 4.9.2** *Let $X, Y$ be two random variables in $[0,1]$ such that $\mathbf{E}[Y] - \mathbf{E}[X] \geq \delta$. Then there is a threshold $\alpha$ such that $\Pr[Y \geq \alpha] - \Pr[X \geq \alpha] \geq \delta$*

Let $P_X(t), P_Y(t)$ be the PDFs for $X, Y$, and let $C_X(t), C_Y(t)$ be the CDFs. We know that

$$\delta \leq \mathbf{E}[Y] - \mathbf{E}[X] = \int_0^1 t(P_Y(t) - P_X(t))dt.$$

Using integration by parts, we see that

$$\int t(P_Y(t) - P_X(t))dt = t(C_Y(t) - C_X(t)) - \int (C_Y(t) - C_X(t))dt.$$

Since $C_X(1) = C_Y(1) = 1$, we have that

$$\delta \leq \int_0^1 (C_X(t) - C_Y(t))dt.$$

By the mean value theorem, there is some value $\alpha$ such that $C_X(\alpha) - C_Y(\alpha) \geq \delta$. As $C_X(t) = 1 - \Pr[X \geq t], C_X(t) = 1 - \Pr[X \geq t]$, the lemma follows.

We now apply this lemma to the variables $\log W / \log C$ and $\log V / \log C$, which satisfy the conditions for the lemma with $\delta = \log(M-1)/\log C$. We therefore set $L = 2^\alpha$, and the attack succeeds with the desired probability.

We can easily turn this into a uniform attacker by estimating $L$. We simply estimate $C_X, C_Y$ offline by choosing several random keys and encrypting either a random $i, i+1$ or $1, M$, and measuring the difference of the two ciphertexts. We can obtain estimates to within $\ll \delta$, and then we can choose the $\alpha$ that maximizes the difference $C_X(\alpha) - C_Y(\alpha)$, which will be a reasonably good threshold. This completes the proof. ∎

**The Full Impossibility.** We now show how to extend the impossibility above to work for parameter-hiding ORE.

**Theorem 4.9.3** *Fix an OPE scheme with message space $[0, M-1]$ and ciphertext space $[0, C-1]$. Consider a distribution $D$ over $[0,1]$, and let $\gamma$ be the minimum scaling allowed for $D$. Let $D_1 = \lfloor D_{\mathsf{scale}}^{M,0} \rceil$. Suppose $D_1$ has the property that, with overwhelming probability over independent samples $x, y$ from $D_1$, we have that $|x - y| \geq 3\gamma$.*

*Then there is a two-message attack for the $(\gamma, D)$-parameter hiding OPE with advantage at least $\Omega(\frac{\log 2}{\log C})$ that runs in time $\mathrm{poly}(\log M, \log C)$, where $\gamma$ is the minimum scaling allowed by the scheme.*

We note that the conditions on $D$ are met for smooth $D$, provided $M$ is exponentially larger than $\gamma$.

**Proof:** The theorem is a simple extension of the attack above. Let $D_0 = \lfloor D_{\mathsf{scale}}^{\gamma,0} \rceil$ and $D_1 = \lfloor D_{\mathsf{scale}}^{M,0} \rceil$.

Consider the following adversary: choose random $x, y$ according to $D_1$. If $x \geq y$, flip $x$ and $y$. Then choose a random $\ell \in [x, y - \gamma]$ (which will be non-empty whp by our condition on $D_1$). Let the left challenge parameters be $(\gamma, \ell)$, and the right be $(M, 0)$. Obtain the two ciphertexts, take the difference, and then compare to $L$.

Our goal, as before, is to show that an $L$ exists that distinguishes these two distributions. Let $W$ be the random variable for $c_1 - c_0$ in the case where the distribution is $(\gamma, \ell)$ for a random $\ell$ as sampled above. Let $W_{x,y}$ be the corresponding difference conditioned on sampling $x, y$. Let $W_\ell$ be the difference conditioned on a given $\ell$. Let $V$ be the random variable representing $c_1 - c_0$ when the parameters are $(M, 0)$. Let $R_i$ be the random variable that represents $c_1 - c_0$ when encrypting $i, i + 1$.

In either case, two values $x, y$ are chosen according for $D_1$, and then permuted to ensure $x \leq y$. For the analysis, we will actually slightly change the distributions on $x, y$ to be conditioned on $y - x \geq 3\gamma$. By our conditions on $D_1$, this negligibly affects the distribution. Let $P^{(1)}_{x,y}$ be the PDF for this distribution on pairs.

In the right case, $x$ and $y$ are encrypted; in the left case, a random $\ell$ is chosen in $[x, y-\delta]$, then random $x', y'$ are chosen according to $D_0$ (swapped if necessary so that $x' \leq y'$), and $x' + \ell, y' + \ell$ are encrypted. Given that parameter-hiding requires the min-entropy to be high, we note that $x' \neq y'$ with overwhelming probability. Therefore, we will let $P^{(0)}_{x',y'}$ be the PDF for the distribution over $(x', y')$ conditioned on $x' < y'$, which is negligibly close to the correct distribution. Under our slightly perturbed distributions, we can therefore write:

$$\mathbf{E}[\log V] = \sum_{x,y} P^{(1)}_{x,y} \mathbf{E}[\log \sum_{i=x}^{y} R_i],$$

$$\mathbf{E}[\log W] = \sum_{x,y} P^{(1)}_{x,y} \sum_{\ell=x}^{y-\gamma} \sum_{x',y'} P^{(0)}_{x',y'} \mathbf{E}[\log \sum_{i=x'+\ell}^{y'+\ell} R_i].$$

We use the concavity of log to bring the average over $\ell$ inside the log and get

$$\mathbf{E}[\log W] = \sum_{x,y} P^{(1)}_{x,y} \sum_{x',y'} P^{(0)}_{x',y'} \mathbf{E}[\frac{1}{y - x - \gamma} \sum_{\ell=x}^{y-\gamma} \log \sum_{i=x'+\ell}^{y'+\ell} R_i].$$

For formula $\frac{1}{y-x-\gamma}\sum_{\ell=x}^{y-\gamma}\sum_{i=x'+\ell}^{y'+\ell} R_i$, we observe that $R_i$ only has weight for $i \in [x, y]$, and for any fixed "$i, x', y'$", there are at most $y' - x'$ copies appear in the formula,which means

$$\frac{1}{y-x-\gamma}\sum_{\ell=x}^{y-\gamma}\sum_{i=x'+\ell}^{y'+\ell} R_i \leq \frac{y'-x'}{y-x-\gamma}\sum_{i=x}^{y} R_i.$$

Therefore, we can bound

$$\begin{aligned}\mathbf{E}[\log W] &\leq \sum_{x,y} P^{(1)}_{x,y} \sum_{x',y'} P^{(0)}_{x',y'} \mathbf{E}[\log\left(\frac{y'-x'}{y-x-\gamma}\sum_{i=x}^{y} R_i\right)] \\ &= \mathbf{E}[\log V] - (\mathbf{E}[\log(y-x-\gamma)] - \mathbf{E}[\log(y'-x')]).\end{aligned}$$

Moreover, the condition that $y-x$ is almost always greater than $3\gamma$ means that $\mathbf{E}[\log(y-x-\gamma)] \geq \log 2\gamma$. Similarly, we always have that $y' - x' \leq \gamma$, so $\mathbf{E}[\log(y'-x')] \leq \gamma$. Thus,

$$\mathbf{E}[\log W] \leq E[\log V] - \log 2.$$

The rest of the proof is essentially identical to the proof of Theorem 4.9.1: under our perturbed distribution, $\log W/\log C$ and $\log V/\log C$ are two variables on $[0, 1]$ whose expectations differ by at least $\log 2/\log C$. Therefore, we can choose a threshold $L$ to distinguish these two. Since the true distributions of $W, V$ are statistically close to the perturbed versions, $L$ distinguishes these as well. ∎

# Chapter 5

# A Ciphertext-Size Lower Bound for Order-Preserving Encryption with Limited Leakage

In this chapter, we give our lower-bound of the ciphertext size for MSDB-secure OPE.

## 5.1 Technique overview

We first recall the constructions of MSDB-secure ORE and OPE, then we give the high-level intuition of our technique.

### 5.1.1 MSDE-secure ORE/OPE

We recall a version that is slightly different from theirs in that it is *perfectly* correct. The scheme $\Pi_{\text{clww-ore}} = (\mathcal{K}^{\text{ore}}, \mathcal{E}^{\text{ore}}, \mathcal{C}^{\text{ore}})$ uses a PRF

$$F : \{0,1\}^\lambda \times ([m] \times \{0,1\}^m) \to (\{0,1\}^\lambda \setminus \{1^\lambda\}).$$

Thus the input domain of $F$ is $[m] \times \{0,1\}^m$, and it outputs a $\lambda$-bit string that is assumed to never be $1^\lambda$ (of course we can modify any PRF so that this is true without affecting asymptotic security).

- Key generation $\mathcal{K}^{\text{ore}}(1^\lambda)$ outputs a random PRF key $K \xleftarrow{\$} \{0,1\}^\lambda$.
- Encryption $\mathcal{E}^{\text{ore}}_K(x)$, on input a message $x \in \{0,1\}^m$, the algorithm computes for each $i = 1, \ldots, m$ the value

$$u_i = F(K, i \,\|\, x[1, \ldots, i-1] \,\|\, 0^{m-i+1}) + x[i], \tag{5.1}$$

where the addition is done by interpreting the bitstrings as members of $\{0, \ldots, 2^\lambda - 1\}$. Encryption outputs $(u_1, \ldots, u_m)$.

- The comparison algorithm $\mathcal{C}^{\text{ore}}((u_1, \ldots, u_m), (u'_1, \ldots, u'_m))$ takes as input two ciphertexts. It finds the smallest $i$ such that $u_i \neq u'_i$, and it outputs 1 if $\mathbf{1}(u_i < u'_i)$.

Correctness follows by observing that the $u_i$ will be equal until the $u_i, u'_i$ corresponding to the first differing bit in the plaintexts. At that position, $u_i$ and $u'_i$ will differ by 1 (additively) and the smaller plaintext has the smaller value. CLWW proved that $\Pi_{\text{clww-ore}}$ (and the variants below) are $\mathcal{L}_{\text{clww}}$-secure, assuming that $F$ is a PRF. It is straightforward to derive from their proof that $\Pi_{\text{clww-ore}}$ is also statistically-secure with the same leakage profile in the random-oracle model.

**Conversion to OPE.** CLWW showed how to convert this construction to an OPE scheme $\Pi_{\text{clww-ope}}$ by simply concatenating the members of a ciphertext to form a bitstring in $\{0,1\}^{\lambda m}$ that is interpreted as a number for comparison. This scheme is perfectly correct because of our assumption that $F$ never outputs the all-ones string, and thus the addition in (5.1) will never wrap modulo $2^\lambda$.

**Compressing ORE ciphertexts.** CLWW showed that one can modify $\Pi_{\text{clww-ore}}$ to a new ORE scheme which has shorter ciphertext. More precisely, the new scheme use a PRF $F'$ with range only $\{0,1,2\}$ instead of $F$, where

$$F' : \{0,1\}^\lambda \times ([m] \times \{0,1\}^m) \to \{0,1,2\}.$$

Now encryption uses $F'$, and for $i = 1, \ldots, m$ computes

$$u_i = F'(K, i \,\|\, x[1, \ldots, i-1] \,\|\, 0^{m-i+1}) + x[i] \mod 3. \tag{5.2}$$

It outputs the vector $(u_1, \ldots, u_n) \in \{0,1,2\}^m$.

Comparison now takes as input $(u_1, \ldots, u_m)$ $(u'_1, \ldots, u'_m)$. As before, it finds the first $i$ such that $u_i \neq u'_i$. But now it outputs 1 if $u'_i = u_i + 1 \mod 3$, and otherwise it outputs 0.

A ciphertext for an $m$-bit input is now a vector in $\{0,1,2\}^m$, which can be represented using $log_2(3)m + O(1) \approx 1.58m$ bits.

### 5.1.2 High-level intuition

Now we consider what the $\epsilon$-$\mathcal{L}_{\text{clww}}$-statistical security implies about our random variables $X_0, \ldots, X_{2^m-1}$. For every possible pair of vectors of messages $\mathbf{m}_0, \mathbf{m}_1$ that does not automatically lose the game because of the leakage requirement, we get a condition about the statistical distance of the distributions of two tuples of random variables. For instance, if the adversary requests singleton vectors $\mathbf{m}_0 = i$ or $\mathbf{m}_1 = j \in \{0,1\}^m$ then the leakage $\mathcal{L}_{\text{clww}}(i) = \mathcal{L}_{\text{clww}}(j) = \emptyset$, so we must have that

$$\Delta(X_i, X_j) \leq \varepsilon$$

for every $i, j$. More generally, for any two vectors $\mathbf{i} = (i_1, \ldots, i_q)$ and $\mathbf{j} = (j_1, \ldots, j_q)$ in $(\{0,1\}^m)^q$ with $\mathcal{L}_{\text{clww}}(\mathbf{i}) = \mathcal{L}_{\text{clww}}(\mathbf{j})$, we must have

$$\Delta((X_{i_1}, \ldots, X_{i_q}), (X_{j_1}, \ldots, X_{j_q})) \leq \epsilon.$$

Thus we need to understand which $\mathbf{i}, \mathbf{j}$ satisfy $\mathcal{L}_{\text{clww}}(\mathbf{i}) = \mathcal{L}_{\text{clww}}(\mathbf{j})$. Fortunately, our proof will only require inputs of a particular structure. We observe that the following qualify for $t = 0, \ldots, m-1$:

$$\mathbf{i} = (0, 2^{t+1} - 1) \quad \text{and} \quad \mathbf{j} = (2^t - 1, 2^t).$$

In binary, $\mathbf{i}$ is $(0^m, 0^{m-t-1}1^{t+1})$ and $\mathbf{j}$ is $(0^{m-t}1^t, 0^{m-t-1}10^t)$. In both cases, the most significant differing bit is in the $t+1$-st least significant position (and the messages are in the same order), so the leakage in the same.

But why should this choice be useful? It represents the most extreme cases of two "distant" plaintexts and two "close" plaintexts that must appear indistinguishable. At a very high level, the scheme must "waste" a lot of its ciphertext space in order to make pairs like this appear indistinguishable. This is because the $\mathbf{i}$ side must have ciphertexts

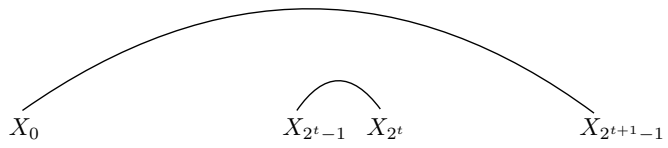


Figure 5.1: Two indistinguishable pairs of r.v.s by the security definition.

that are far apart (by roughly $2^{t+1}$) simply because correctness forces many ciphertexts to be between $X_0$ and $X_{2^{t+1}-1}$, namely $X_1, X_2, \ldots, X_{2^{t+1}-2}$. In order to appear indistinguishable, $X_{2^t-1}$ and $X_{2^t}$ must also be far apart, with no other ciphertexts between them (again by correctness). Moreover, as $t$ grows we get a *nested* sequence of pairs, where the space wasted by the previous pair force the next to waste even more.

Our proof will argue that this wasted space grows to the quoted bound. We consider the nested sequence of these tuples above, and then proceed by induction to show that a large ciphertext-space is needed for security. The key step in our induction is that, since the tuples $(X_0, X_{2^{t+1}-1})$ and $(X_{2^t-1}, X_{2^t})$ must have statistical distance at most $\epsilon$, then their *gaps*

$$G_1 = X_{2^{t+1}-1} - X_0 \quad \text{and} \quad G_2 = X_{2^t} - X_{2^t-1}$$

must also satisfy $\Delta(G_1, G_2) \leq \varepsilon$ by the data processing inequality. But the gap measured by $G_2$ is a subset of the gap measured by $G_1$, so $G_2 < G_1$. In fact, as we show via induction on $t$, $G_2$ must often be much less than $G_1$ (since $G_1$ contains the gap from $X_{2^t-1}$ and $X_0$, which is the previous step of the induction). Using this fact, we apply the following lemma that will be proven in next section.

**Lemma 5.1.1** *For any two variables $X \geq Y \in [N-1]'$, and distinct positive integers $d_1, \ldots, d_k$ such that $Pr[X = Y + d_i] = p_i$, we have*

$$\Delta(X, Y) \geq \frac{\sum_{i=1}^{k} p_i \cdot d_i}{N-1}.$$

Intuitively, this lemma says that if one of the random variables is often much bigger than the other, then they must have large statistical distance.

**Contrast with big jump.** The *big jump* attack of Boldyreva et al. [2011] gave a ciphertext-size lower bound for any ideal OPE. With ideal ORE, *every* pair of two random variables $X_{i_1} < X_{i_2}$ and $X_{j_1} < X_{j_2}$ must be indistinguishable, which gives the attack more flexibility and results in an exponential bound (without resorting to recursion). Instead our bound works with a particular nested set of $m$ pairs, with each step using a pair to increase the bound by roughly $\lambda$ bits.

## 5.2 Ciphertext lower-bound for OPE with CLWW leakage

We can now state our theorem formally.

**Theorem 5.2.1 (Lower Bound for Ciphertext size)** *Suppose* $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ *is an order-preserving encryption scheme with associated message space* $\{0,1\}^m$ *and ciphertext space* $\{0,1\}^n$*, and that* $\Pi$ *is* $2^{-\lambda}$*-*$\mathcal{L}_{\text{clww}}$*-statistically-secure. Then we have*

$$n \geq \lambda m - m \log m + m \log e$$

In any practical OPE scenario we are aware of, we have $\log m - \log e < \lambda$ and thus our bound is nontrivial. For example, considering the message space is 40 bytes, $\log m - \log e = \log 320/e < 7$, while in real world encryption, the secure parameter is always set to be 80 or larger.

Before our proof for the theorem, we here introduce an additional technique lemma.

**Lemma 5.2.2** *Let* $X > Y \in [N-1]'$ *be random variables such that* $\Delta(X,Y) \leq \delta$*. Let* $i \geq 1$ *and assume that for all* $q \in [0,1]$*,* $\Pr[X > Y + \frac{(1-q)^i}{\delta^i . i!}] \geq q$*. Then for all* $q \in [0,1]$ *we have*

$$\Pr[X > \frac{(1-q)^{i+1}}{\delta^{i+1}(i+1)!}] \geq q.$$

### 5.2.1 Proof for Theorem 5.2.1

In this part, we present the proof for our lower bound theorem, by making use of Lemma 5.1.1 and Lemma 5.2.2.

**Proof:** Let $\Pi = (\mathcal{K}, \mathsf{E})$ be an OPE scheme with associated message space $\{0,1\}^m$ and ciphertext space $\{0,1\}^n$, and assume $\Pi$ is $2^{-\lambda}$-$\mathcal{L}_{\text{clww}}$-statistically-secure.

Below, for $i \in [2^m - 1]'$, we let $X_i = \mathsf{E}_K(i)$ where $K \xleftarrow{\$} \mathcal{K}(1^\lambda)$ as in the proof sketch. That is, the $X_i$ are dependent random variables that represent the encryption of message $i$ under a random key. Note that $X_0 < X_1 < \cdots < X_{2^m-1}$.

We will prove the theorem using following claim. Here, we let $\varepsilon = 2^{-\lambda}$.

**Lemma 5.2.3** *For $i \in [2^m - 1]'$, let $X_i$ be defined as above. Then for $1 \leq j \leq m$ and $q \in [0,1]$,*

$$\Pr[X_{2^j-1} - X_0 \geq \frac{(1-q)^{j-1}}{\varepsilon^{j-1} \cdot (j-1)!}] \geq q$$

The proof is by induction on $j$.

**Case** $j = 1$. This case reduces to $\Pr[X_1 - X_0 \geq 1] = 1$, which is true by the correctness of the scheme.

**Case** $j \implies j+1$. We need to show that for any $q \in [0,1]$

$$\Pr[X_{2^{j+1}-1} - X_0 \geq \frac{(1-q)^{j}}{\varepsilon^{j} \cdot (j)!}] \geq q.$$

By the correctness of the scheme, we have that

$$X_{2^{j+1}-1} - X_0 \geq (X_{2^j} - X_{2^j-1}) + (X_{2^j-1} - X_0) + 1. \tag{5.3}$$

Now define "gap" random variables $G_1 = X_{2^{j+1}-1} - X_0$ and $G_2 = (X_{2^j} - X_{2^j-1})$. By induction we know that for any $q \in [0,1]$

$$\Pr[X_{2^j-1} - X_0 \geq \frac{(1-q)^{j-1}}{\varepsilon^{j-1} \cdot (j-1)!}] \geq q.$$

Plugging this, and the definitions of $G_1, G_2$ into (5.3), we have

$$\Pr[G_1 > G_2 + \frac{(1-q)^{j-1}}{\varepsilon^{j-1} \cdot (j-1)!}] \geq q.$$

Moreover, we know by the $\varepsilon$-$\mathcal{L}_{\text{clww}}$-statistical security of $\Pi$ and Lemma 2.1.1 that $\Delta(G_1, G_2) \leq \varepsilon$.

We now want to apply Lemma 5.2.2 to $G_1$ and $G_2$, to show that $G_1$ must be large and then conclude the induction. In the lemma, we set $G_1 = X, G_2 = Y, i = j$, and $\delta = \varepsilon$. The lemma gives

$$\Pr[G_1 > \frac{(1-q)^j}{\varepsilon^j \cdot (j)!}] \geq q,$$

obtaining the induction step.

We can now complete the proof of Theorem 5.2.1. The above lemma with $j = m$ tells us that for any $q \in [0, 1]$

$$\Pr[X_{2^m-1} > X_0 + \frac{(1-q)^{m-1}}{\varepsilon^{m-1} \cdot (m-1)!}] \geq q,$$

and thus for any $j \leq D = 1/\varepsilon^{m-1}(m-1)!$,

$$\Pr[X_{2^m-1} > X_0 + j] \geq 1 - ((m-1)! \cdot j)^{1/m-1}\varepsilon$$

and

$$\sum_{\ell=1}^{j} \Pr[X_{2^m-1} = X_0 + \ell] \leq ((m-1)! \cdot j)^{1/m-1}\varepsilon.$$

Besides, we claim $D \leq N - 1$, if not, then there exists $q > 0$ such that

$$N - 1 = \frac{(1-q)^{m-1}}{\varepsilon^{m-1} \cdot (m-1)!}$$

referring to

$$\Pr[X_{2^m-1} > X_0 + N - 1] \geq q > 0,$$

which contradicts $X_i \in [N-1]'$.

Now we denote $p_\ell = \Pr[X_{2^m-1} = X_0 + \ell]$, and according to Lemma 5.1.1, we get that

$$\varepsilon \geq \Delta(X_{2^m-1}, X_0) \geq \frac{\sum_{\ell=1}^{N-1} p_\ell \cdot \ell}{N-1} \tag{5.4}$$

and

$$\begin{aligned}
\sum_{\ell=1}^{N-1} p_\ell \cdot \ell &= (p_1 + \cdots + p_{N-1}) + (p_2 + \cdots + p_{N-1}) + \cdots + p_{N-1} \\
&\geq 1 + (1 - p_1) + (1 - p_1 - p_2) + \cdots + (1 - p_1 - \cdots - p_{D-1}) \\
&\geq 1 + \sum_{\ell=1}^{D-1} (1 - ((m-1)!\ell)^{\frac{1}{m-1}} \cdot \varepsilon) \\
&= D - (m-1)!^{\frac{1}{m-1}} \cdot \varepsilon \sum_{\ell=1}^{D-1} \ell^{\frac{1}{m-1}} \\
&\geq D - (m-1)!^{\frac{1}{m-1}} \cdot \varepsilon \cdot \int_0^D x^{\frac{1}{m-1}} dx \\
&= \frac{1}{\varepsilon^{m-1}(m-1)!} \cdot \frac{1}{m} = \frac{1}{\varepsilon^{m-1} m!}.
\end{aligned}$$

Returning to (5.4), we have $N - 1 \geq 1/\varepsilon^m m!$. By setting $\varepsilon = 2^{-\lambda}$, we get

$$n \geq \lambda m - \log(m!) \geq \lambda m - \log((m/e)^m) = \lambda m - m \log m + m \log e,$$

which we complete the entire proof assuming Lemma 5.1.1 and 5.2.2. ▌

### 5.2.2 Proof for Lemma 5.1.1

In this part, we give the proof for Lemma 5.1.1.

**Proof:** We will show that one of the distinguishers $\mathcal{D}_i$, $i \in [N-1]$, has the needed advantage, where $\mathcal{D}_i$ is defined as follows: Given input $T \in [N-1]'$, $\mathcal{D}_i$ outputs 1 if and only if $T \geq i$.

The advantage of $\mathcal{D}_i$ is $\delta_i = \Pr[X \geq i] - \Pr[Y \geq i]$. We have that

$$\begin{aligned}
\sum_{i=1}^{N-1} \delta_i &= \sum_{i=1}^{N-1} \Pr[X \geq i] - \sum_{i=1}^{N-1} \Pr[Y \geq i] \\
&= \sum_{i=0}^{N-1} \Pr[X \geq i] - \sum_{i=0}^{N-1} \Pr[Y \geq i] = E(X - Y) \geq \sum_{i=1}^{k} p_i d_i.
\end{aligned}$$

Thus some $\delta_i$ must be at least this sum divided by $N - 1$. ▌

### 5.2.3 Proof for Lemma 5.2.2

In this part, we give the proof for Lemma 5.2.2.

**Proof:** Suppose for contradiction that there exists $q^* \in [0,1]$ such that

$$\hat{q} := \Pr[X > t] < q^*,$$

where $t = (1-q^*)^{i+1}/\delta^{i+1}(i+1)!$.

We will show that $\Delta(X,Y) > \delta$, violating the assumption in the lemma. We will prove this by showing the following "truncated" r.v.s $W, Z$ satisfy $\Delta(X,Y) \geq \Delta(W,Z) > \delta$, where $W, Z$ are defined via the joint distribution

$$\Pr[W=a, Z=b] = \begin{cases} \Pr[X=a, Y=b] & \text{if } (a,b) \in [t]^2 \setminus (0,0), \\ \hat{q} & \text{if } (a,b) = (0,0) \\ 0 & \text{otherwise} \end{cases}.$$

According to the definition of $(W,Z)$, we show $\Delta(X,Y) \geq \Delta(W,Z)$. For simplifying, we denote

$$p_{i,j} = \Pr[X=i, Y=j];\ p_j = \sum_{k=0}^{t} p_{k,j};\ p_j^* = \sum_{k=t+1}^{N-1} p_{k,j};\ \forall i,j \in [t]$$

and it's obvious to note that for $j \in [t]$: 1) $\Pr[X=j] = \Pr[W=j]$; 2) $\Pr[Z=j] = p_j$;

3) $\Pr[Y = j] = p_j + p_j^*$; 4)$\sum_{k=0}^{t} p_j^* = \sum_{k=t+1}^{N-1}(\Pr[X = k] - \Pr[Y = k])$. Hence:

$$\begin{aligned}
2\Delta(X,Y) &= \sum_{j=0}^{N-1} |\Pr[X = j] - \Pr[Y = j]| \\
&= \sum_{j=0}^{t} |\Pr[X = j] - \Pr[Y = j]| + \sum_{j=t+1}^{N-1} |\Pr[X = j] - \Pr[Y = j]| \\
&\geq \sum_{j=0}^{t} |\Pr[X = j] - \Pr[Y = j]| + \sum_{j=t+1}^{N-1} (\Pr[X = j] - \Pr[Y = j]) \\
&= \sum_{j=0}^{t} |\Pr[X = j] - \Pr[Y = j]| + \sum_{j=0}^{t} p_j^* \\
&= \sum_{j=0}^{t} |\Pr[W = j] - \Pr[Z = j] - p_j^*| + \sum_{j=0}^{t} p_j^* \\
&\geq \sum_{j=0}^{t} |\Pr[W = j] - \Pr[Z = j]| = 2\Delta(W,Z).
\end{aligned}$$

In the following, it suffices to show that $\Delta(W, Z) > \delta$. We denote $d_j = \Pr[W = Z + j]$. Applying Lemma 5.1.1,

$$\Delta(W,Z) \geq \frac{\sum_{\ell=1}^{t} d_\ell \cdot \ell}{t}.$$

We now show that $\sum_{\ell=1}^{t} d_\ell \cdot \ell > \delta t$, completing the proof. Below we use the following technical claim, which we establish below:

**Claim 5.2.4** *In the notation of the proof, we have the following:*

1. $\sum_{\ell=1}^{t} d_\ell = 1 - \hat{q}$,
2. *For each* $j$, $\sum_{\ell=1}^{j} d_\ell \leq (i! \cdot j)^{1/i}\delta$,
3. $t \geq \hat{t}$, *where* $\hat{t} = (1 - \hat{q})^i/\delta^i i!$.

Using the claim, we have

$$\begin{aligned}
\sum_{\ell=1}^{t} d_\ell \cdot \ell &\geq \sum_{\ell=1}^{\hat{t}} d_\ell \cdot \ell = (d_1 + \ldots + d_{\hat{t}}) + (d_2 + \ldots + d_{\hat{t}}) + \ldots + (d_{\hat{t}}) \\
&\geq (1-\hat{q}) + ((1-\hat{q}) - d_1) + ((1-\hat{q}) - d_1 - d_2) + \ldots + ((1-\hat{q}) - d_1 - \ldots - d_{\hat{t}-1}) \\
&\geq (1-\hat{q})\hat{t} - \sum_{\ell=1}^{\hat{t}-1} (\ell i!)^{1/i}\delta \\
&\geq (1-\hat{q})\hat{t} - (i!)^{1/i}\delta \int_0^{\hat{t}} x^{1/i} dx \\
&= (1-\hat{q})^{i+1}\hat{t} - (i!)^{1/i}\delta \cdot \frac{i}{i+1}\hat{t}^{\frac{i+1}{i}} = \frac{(1-\hat{q})^{i+1}}{\delta^i (i+1)!} > \delta t.
\end{aligned}$$

We now prove the claim. The first part follows easily from the definition of $W, Z$. For the second part, we have

$$\sum_{\ell=1}^{j} d_\ell \leq \sum_{\ell=1}^{j} \Pr[X = Y + \ell] = 1 - \Pr[X > Y + j] \leq (i!j)^{1/i}\delta,$$

where the last inequality follows since $\Pr[X > Y + (1-q)^i/\delta^i i!] \geq q$ holds for all $q \in [0,1]$, and particular $q = 1 - (i!j)^{1/i}\delta$.

For the third part of the claim, suppose for contradiction that $t < \hat{t}$. Then

$$\Pr[X > t] \geq \Pr[X > Y + t] \geq 1 - (i!t)^{1/i}\delta > 1 - (i!\hat{t})^{1/i}\delta = \hat{q}.$$

(The second inequality is another application of the condition in the lemma, similar to the proof of the second part.) But this contradicts the definition $\hat{q} = \Pr[X > t]$ and proves the third part of the claim. ∎

## 5.3 Extensions

Our lower bound applies to the specific definition, say CLWW leakage, and it is possible to circumvent the bound by targeting a different, but hopefully satisfactory, notion of

security. In this section we identify an abstract property, which we term *inner-distance-indistinguishablity*, for which a similar lower bound applies. Thus, to avoid the bound for OPE with another definition, one must avoid this property, and the authors are not aware of an approach for doing so.

We also show how to apply our proof technique to give an essentially-tight lower bound on the ciphertext length of the "base-$d$" OPE variants suggested by Chenette et al., which achieve a weakened version of security with shorter ciphertexts.

**Inner-distance-indistinguishablity.** The following property seems mostly useful as a tool for understanding and generalizing the lower bound, and not as a stand-alone target for OPE security in practice.

**Definition 5.3.1** *Let $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ be an OPE scheme with associated message space $M$, $d \geq 1$ be an integer, and $\varepsilon > 0$. We say that $\Pi$ is (statistically) $\varepsilon$-*inner-distance-indistinguishable for width $d$ (denoted $\varepsilon$-IDI$_d$) *if for all $i < j \in M$ such that $j - i > d$, there exist $k, \ell \in M$ such that*

1. *$i \leq k < \ell \leq j$*
2. *$\ell - k \leq d$*
3. *$\Delta(D_1, D_2) \leq \varepsilon$, where $D_1 = \mathsf{E}_K(j) - \mathsf{E}_K(i)$ and $D_2 = \mathsf{E}_K(k) - \mathsf{E}_K(\ell)$ and $K$ is random key.*

Intuitively, $\varepsilon$-IDI$_d$ says that the distance between every encrypted pair of messages must be indistinguishable from the gap between two encrypted messages which both lie between them, and moreover the latter gap is required to be small, namely $d$ or less.

The CLWW notion implies $\varepsilon$-IDI$_1$ security. That is, for every pair $i < j$, $\mathsf{E}_K(j) - \mathsf{E}_K(i)$ is distinguishable from $\mathsf{E}_K(k+1) - \mathsf{E}_K(k)$ for some $k$ between $i$ and $j$ (when $d = 1$, we must have $\ell = k + 1$ in the definition).

To see this, fix some $i, j$, with $j > i + 1$, and consider their binary expansions. We may write $i$ in the form $p \,\|\, 0 \,\|\, x$ and $j$ in the form $p \,\|\, 1 \,\|\, y$, where $p$ is the longest

common prefix and $i$ and $j$, and $x, y \in \{0,1\}^L$ for some $L \geq 1$. Then consider

$$k = p \,\|\, 0 \,\|\, 1^L \quad \text{and} \quad \ell = p \,\|\, 1 \,\|\, 0^L.$$

We have that $\ell = k + 1$ (treating $\ell, k$ as numbers), and that either $k \neq i$ or $\ell \neq j$. Moreover the CLWW security notion ensures that the condition of $\text{IDI}_1$ security holds for this choice of $k, \ell$.

The following theorem generalizes Theorem 5.2.1.

**Theorem 5.3.2** *Suppose $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ is an order-preserving encryption scheme with security parameter $\lambda$ and associated message space $\{0,1\}^m$ and ciphertext space $\{0,1\}^n$, and $\Pi$ is $2^{-\lambda}$-$\text{IDI}_d$ secure for some $d \geq 1$. Let $m' = m - \lceil \log d \rceil$. Then we have*

$$n \geq \lambda m' - m' \log m' + m' \log e.$$

**Proof:** Let $\Pi = (\mathcal{K}, \mathsf{E}, \mathcal{C})$ be an OPE scheme with the syntax and conditions in the theorem. Below, for $i \in \{0,1\}^m$, we write $X_i = \mathsf{E}_K(i)$, and let $m'$ be as defined in the theorem.

We will show how to carry out the same strategy used in the proof of Theorem 5.2.1. We will prove a version of Lemma 5.2.3 for a different nested sequence of pairs of messages $(i_j^L, i_j^R)_{j=1}^{m'}$ that we define inductively from $m'$ down to 1 now.

- Base: $i_{m'}^L = 0, i_{m'}^R = 2^m - 1$.
- Step: Given $(i_j^L, i_j^R)$, let $k < \ell$ be the pair between $i_j^L$ and $i_j^R$ guaranteed by $\text{IDI}_d$ security. We distinguish two cases:
  1. If $k - i_j^L > i_j^R - \ell$ then set $(i_{j-1}^L, i_{j-1}^R)$ to be $(i_j^L, k)$.
  2. Otherwise, set $(i_{j-1}^L, i_{j-1}^R)$ to $(\ell, i_j^R)$.

Intuitively, we use the $\text{IDI}_d$ security property to find a nested sequence by moving to the "larger" gap at each step, and this continues for at least $m'$ steps. Using this sequence,

the rest of the proof of Lemma 5.2.3 can be carried out. Finally, the rest of the proof of Theorem 5.2.1 can be applied exactly as before. ∎

**Other variants.** We can also extend our proof of Theorem 5.2.1 to the "$d$-ary" variants of CLWW. That construction saved a modest amount of space over the main CLWW construction via additional leakage, which is described via the following leakage profile $\mathcal{L}^d_{\text{clww}}$:

$$\mathcal{L}^d_{\text{clww}}(x_1, \ldots, x_q) := \{(i, j, \mathsf{ind}^{(d)}_{\mathsf{diff}}(x_i, x_j), \mathbf{1}(x_i < x_j)) \ : \ 1 \le i < j \le q\},$$

where $\mathsf{ind}^{(d)}_{\mathsf{diff}}(a, b)$ writes its inputs in base $d$ as $a = (a[1], \ldots, a[m])$ and $b = (b[1], \ldots, b[m])$, and outputs $(k, |b[k] - a[k]|)$, where $k$ is the smallest index such that $b[k] \neq a[k]$. If there is not such index (i.e. $a = b$) then it outputs $(m + 1, 0)$.

Intuitively, this leakage outputs the index of the first base-$d$ digit where each pair of messages differ, and additionally outputs the absolute difference in that digit. (When $d = 2$ the additional output is trivial, since it is always 1.)

We will show how to carry out the same strategy used in the proof of Theorem 5.2.1. Here we denote $m^* = m/\log d - 1$, and we will prove a version of Lemma 5.2.3 for a different nested sequence of pairs of messages $(i^L_j, i^R_j)_{j=1}^{m^*}$ that we define as follows:

$$i^L_j = 0, \quad i^R_j = 0^{m^*-j}||1||(d-1)^j.$$

And we define the pair $\hat{i}^L_j, \hat{i}^R_j$ as:

$$\hat{i}^L_j = 0^{m^*-j}||0||(d-1)^j, \quad \hat{i}^R_j = 0^{m^*-j}||1||0^j.$$

According to the leakage profile, we have $(\mathsf{E}_K(i^L_j), \mathsf{E}_K(i^R_j)\ )$ and $(\mathsf{E}_K(\hat{i}^L_j), \mathsf{E}_K(\hat{i}^R_j))$ are statistical indistinguishable. Using the sequence $(i^L_j, i^R_j)_{j=1}^{m^*}$, the rest of the proof of Lemma 5.2.3 can be carried out. Finally, the rest of the proof of Theorem 5.2.1 can be applied exactly as before. Hence we have the lower bound:

$$n \ge \lambda(m/\log d) - (m/\log d)\log(m/\log d).$$

# Chapter 6

# Conclusion

Order revealing encryption has shown to be an important primitive in leaky cryptography and encrypted database protocols, despite many known attacks indicate that, under the condition that the adversary has a good estimate of the message distribution, ORE provides little privacy. This dissertation explores several directions on ORE and presents three of our current results, which includes new security notions, new constructions and black box separations.

First, we study ORE with ideal leakage, where *only* the order of the underlying plaintexts are leaked and nothing else. We show a black box separation between ideal ORE and weaker models, such as random oracle model and generic group model, which means we can not build an ideal ORE from any symmetric key primitives, as well as any public key primitives that are implied by generic group model. Thus, despite of providing the strongest security, ideal ORE has to suffer from somewhat inefficiency, and in fact, those inefficiency is unacceptable in the current state.

Next, we attempt to develop more efficient schemes, by relaxing the security requirement, say allowing somewhat additional leakage. Based on this idea, we propose a new security notion called parameter-hiding, where in such a notion we consider about the privacy of the distribution they came from, and we show how to protect the statistical information, such as *mean* and *variance*, about the underlying data distribution. Then we build the corresponding scheme *only* based on bi-linear map.

After that, we turn our attention to OPE. We know that OPE can not achieve ideal-secure, EP-MSDB-secure and parameter-hiding, and for MSDB-secure OPE, the current construction has ciphertext length—$\lambda m$. We then show a lower bound on ciphertext size which indicates that the current construction is almost optimal.

Reviewing our results, we know that, ideal ORE is unpractical in the real world and OPE with CLWW leakage has to suffer a long ciphertext. On the side, our new privacy notion, indeed gives some hope on ORE, but our notion only works under the condition that the adversary does not have a good estimate of the underlying message distribution. Thus, the big-open question is *what's the future of ORE?* In our perspective, a potential solution against those attacks is to develop a protocol that use ORE as a primitive, rather than using it directly. We actually illustrate our intuitive idea in Sec 1.1.4, while lacking of security analysis, and we seriously treat it as a future research direction for ORE.

# Bibliography

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 2013a. URL `https://www.microsoft.com/en-us/research/publication/orthogonal-security-with-cipherbase/`.

A. Arasu, K. Eguro, R. Kaushik, and R. Ramamurthy. Querying encrypted data (tutorial). In *ICDE*, 2013b.

S. Bajaj and R. Sion. Trusteddb: A trusted hardware-based database with privacy and data confidentiality. *TKDE*, 26(3):752–765, 2014.

M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Annual International Cryptology Conference*, pages 535–552. Springer, 2007.

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.

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.

D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

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. ISBN 978-3-540-34547-3.

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 *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 563–594. Springer, 2015.

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. ISBN 978-3-642-54242-8. doi: 10.1007/978-3-642-54242-8_1. URL `https://doi.org/10.1007/978-3-642-54242-8_1`.

Z. Brakerski and G. Segev. Function-private functional encryption in the private-key setting. In *Theory of Cryptography Conference*, pages 306–324. Springer, 2015.

R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Annual International Cryptology Conference*, pages 455–469. Springer, 1997.

R. Canetti, Y. T. Kalai, and O. Paneth. On obfuscation with random oracles. In *Theory of Cryptography Conference*, pages 456–467. Springer, 2015.

D. Cash and C. Zhang. A ciphertext-size lower bound for order-preserving encryption with limited leakage. In *Theory of Cryptography Conference*, pages 159–176. Springer, 2018.

D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner. Highly-scalable searchable symmetric encryption with support for boolean queries. In *Advances in Cryptology–CRYPTO 2013*, pages 353–373. Springer, 2013.

D. Cash, P. Grubbs, J. Perry, and T. Ristenpart. Leakage-abuse attacks against searchable encryption. In *CCS*, 2015.

D. Cash, F.-H. Liu, A. O'Neill, and C. Zhang. Reducing the leakage in practical order-revealing encryption. Cryptology ePrint Archive, Report 2016/661, 2016. `https://eprint.iacr.org/2016/661`.

D. Cash, F.-H. Liu, A. O'Neill, M. Zhandry, and C. Zhang. Parameter-hiding order revealing encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 181–210. Springer, 2018.

Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *ACNS*, 2005.

M. Chase and S. Kamara. Structured encryption and controlled disclosure. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 577–594. Springer, 2010.

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.

J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 3–12, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. ISBN 978-3-662-46800-5.

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.

J.-S. Coron, M. S. Lee, T. Lepoint, and M. Tibouchi. Cryptanalysis of ggh15 multilinear maps. In M. Robshaw and J. Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 607–628, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg. ISBN 978-3-662-53008-5.

T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN 0471241954.

R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *CCS*, 2006.

J. L. Dautrich Jr and C. V. Ravishankar. Compromising privacy in precise query protocols. In *EDBT*, 2013.

F. B. Durak, T. M. DuBuisson, and D. Cash. What else is revealed by order-revealing encryption? In *ACM CCS*, 2016. To appear.

P. Erdős and V. Sós. On a problem of graph theory.

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.

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. ISBN 978-3-662-44371-2.

E.-J. Goh et al. Secure indexes. *IACR Cryptology ePrint Archive*, 2003:216, 2003.

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.

H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 216–227, New York, NY, USA, 2002. ACM.

B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu. Secure multidimensional range queries over outsourced data. *VLDBJ*, 21(3):333–358, 2012.

M. S. Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *NDSS*, 2012.

M. S. Islam, M. Kuzu, and M. Kantarcioglu. Inference attack against encrypted range queries on outsourced databases. In *CODASPY*, 2014.

M. Joye and A. Passelègue. Function-revealing encryption. 2016.

S. Kamara and T. Moataz. Sql on structurally-encrypted databases. Cryptology ePrint Archive, Report 2016/453, 2016. `http://eprint.iacr.org/`.

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. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978376. URL `http://doi.acm.org/10.1145/2976749.2978376`.

C. Liu, L. Zhu, M. Wang, and Y.-a. Tan. Search pattern leakage in searchable encryption: Attacks and new construction. *Information Sciences*, 265:176–188, 2014.

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.

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.

A. Mandal and A. Roy. Relational hash: Probabilistic hash for verifying relations, secure against forgery and more. In *Annual Cryptology Conference*, pages 518–537. Springer, 2015.

E. Miles, A. Sahai, and M. Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. In *Proceedings, Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology — CRYPTO 2016 - Volume 9815*, pages 629–658, Berlin, Heidelberg, 2016. Springer-Verlag. ISBN 978-3-662-53007-8. doi: 10.1007/978-3-662-53008-5_22. URL https://doi.org/10.1007/978-3-662-53008-5_22.

P. Mohassel and M. Franklin. Efficiency tradeoffs for malicious two-party computation. In *International Workshop on Public Key Cryptography*, pages 458–473. Springer, 2006.

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.

O. Pandey and Y. Rouselakis. Property preserving symmetric encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–391. Springer, 2012.

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.

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.

G. Segev and I. Shahaf. Ciphertext expansion in limited-leakage order-preserving encryption: A tight computational lower bound. To appear in Preceeding of the 16th Theory of Cryptography Conference, 2018. https://eprint.iacr.org/2018/521.

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.

D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *SP*, 2000.

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.

M. Zhandry and C. Zhang. Impossibility of order-revealing encryption in idealized models. In *Theory of Cryptography Conference*, pages 129–158. Springer, 2018.