Title
Fundamental Limits of Private Information Retrieval

Author
Sun, Hua

Publication Date
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Fundamental Limits of Private Information Retrieval

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical Engineering

by

Hua Sun

Dissertation Committee:
Professor Syed Jafar, Chair
Professor Hamid Jafarkhani
Professor Ender Ayanoglu

2017

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I have been truly lucky to be advised by Professor Syed Jafar, from whom I learned everything about information theory. I have benefited tremendously from numerous discussions during the past 6 years. I would always remember and learn from his insight, vision, commitment to excellence in my career.

I would like to thank Professor Hamid Jafarkhani, Ender Ayanoglu for serving on my dissertation committee, and Professor A. Lee Swindlehurst, Yaming Yu for serving on my qualifying examination committee.

It has been a great pleasure to have coauthored with Tiangao Gou, Chunhua Geng, Chenwei Wang, Xinping Yi, David Gesbert, Bofeng Yuan, Zhen Chen and Zhuqing Jia. My thanks extend to former and current colleagues including Hamed Maleki, Xiaoshi Song, Sundar Krishnamurthy, Yingyuan Gao, Arash Gholami and Yao-Chia Chan. I would also like to thank Croucher Summer Course in Information Theory 2015, where I first heard and started to learn about the topic of this dissertation.

My friends make my life easy and happy at Irvine. I would mention Jiang, Feng, Hanzi and Haoyu in particular. Finally, I would like to thank my family.

CURRICULUM VITAE

Hua Sun

EDUCATION

Doctor of Philosophy in Electrical Engineering **2017**
University of California, Irvine *Irvine, California*

Master of Science in Electrical and Computer Engineering **2013**
University of California, Irvine *Irvine, California*

Bachelor of Engineering in Communication Engineering **2011**
Beijing University of Posts and Telecommunications *Beijing, China*

ABSTRACT OF THE DISSERTATION

Fundamental Limits of Private Information Retrieval

By

Hua Sun

Doctor of Philosophy in Electrical Engineering

University of California, Irvine, 2017

Professor Syed Jafar, Chair

The modern information age is heralded by exciting paradigms ranging from big data, cloud computing to internet of things. As information becomes increasingly available, privacy concerns are starting to take center-stage, especially in the communication networks that are used for information storage, repair, retrieval or transfer. The focus of this dissertation is on the private information retrieval (PIR) problem. PIR originated in theoretical computer science and cryptography, and has only recently started receiving attention in information and coding theory. PIR seeks the most efficient way for a user to retrieve a desired message from a set of N distributed servers, each of which stores all K messages, without revealing any information about which message is being retrieved to any individual server. It is a canonical problem with deep connections to a number of other prominent problems such as oblivious transfer, multiparty computation, locally decodable codes, batch codes and blind interference alignment.

We will first identify the capacity of PIR, i.e., the maximum number of bits of desired information that can be privately retrieved per bit of downloaded information. This result is inspired by the discovery of an intriguing connection between PIR and blind interference alignment in wireless networks. Then we will discuss four extensions of PIR. The first extension is the TPIR problem, where we increase the privacy level, i.e., instead of requiring

privacy to each individual server, we require privacy to any colluding set of up to T servers. We will characterize the capacity of TPIR and generalize the result to include robustness constraints, where we have M databases, out of which any $M - N$ may fail to respond. The second extension is the MDS-TPIR problem, where we further generalize the storage constraint, i.e., instead of data replication, an MDS storage code is used to store the messages. In particular, we will disprove a recent conjecture on the capacity of MDS-TPIR. The third extension is the SPIR problem, a form of oblivious transfer where the privacy constraint is extended symmetrically to protect both the user and the servers. We will identify the capacity of SPIR. The final extension is MPIR, where the user and the servers communicate in multiple rounds. We will show that multiple rounds do not increase the capacity of PIR, but reduce the storage overhead. The results will shed light into the necessity for non-linear schemes and non-Shannon information inequalities.

Chapter 1

Introduction

Information and communication technology forms the backbone of the modern society. The rapid increase in the amount of information or data motivates revolutionary applications ranging from big data, large scale learning, internet of things to cloud computing. The development of new technologies and applications continues to benefit a wide array of fields including advertising, healthcare, manufacturing, retail, transportation and education. As the amount of available data grows, the risk of information leakage increases, which brings information privacy to the center of current research challenges.

The focus of this dissertation is on the private information retrieval (PIR) problem, a canonical problem in information privacy. PIR originated in theoretical computer science and cryptography, and has received much recent attention in information and coding theory. Let us start with a brief background on PIR.

1.1 Background

Introduced in 1995 by Chor, Kushilevitz, Goldreich and Sudan [23, 24], the private information retrieval (PIR) problem seeks the most efficient way for a user to retrieve a desired message from a set of distributed servers, each of which stores all the messages, without revealing any information about which message is being retrieved to any individual server. The user can hide his interests trivially by requesting all the information, but that could be very inefficient (expensive). The goal of the PIR problem is to find the most efficient solution.

Besides its direct applications, PIR is of broad interest because it shares intimate connections to many other prominent problems. PIR protocols are the essential ingredients of oblivious transfer [36], instance hiding [32, 1, 9], multiparty computation [10] and secret sharing schemes [60, 13], which are important information theoretic cryptography primitives (enabling premises and building blocks for much more sophisticated tasks). Beyond security and privacy, PIR is further related to locally decodable codes [72] and batch codes [40], which are used in distributed storage and repair. Through the connection between locally decodable and locally recoverable codes [37], PIR also connects to distributed data storage repair [26], index coding [18] and the entire umbrella of network coding [2] in general. As such PIR holds tremendous promise as a point of convergence of complementary perspectives.

The PIR problem is comprised of N servers, each stores K messages and each message is of size L bits. A user wants one of the messages, but requires each server to learn absolutely nothing (in the information theoretic sense)[1] about the retrieved message index. To do so, the user generates N queries, one for each server. After receiving the query, each server returns an answering string to the user. The user must be able to obtain the desired message

[1]There is another line of research, where privacy needs to be satisfied only for computationally bounded servers [34, 73, 54].

from all N answers. To be private, each query and each answer must be independent of the desired message index.

The PIR problem was initially studied among the theoretical computer science community in the setting where we have a large number of messages (typically $K \to \infty$), and each message is one bit long ($L = 1$) [23, 24]. The cost of a PIR scheme is measured by the total amount of communication (communication complexity) between the user and the servers, i.e., the sum of lengths of each query string (upload) and each answering string (download), in order functions of the number of messages, K. The pursuit of communication complexity of PIR has attracted extensive attention for the past two decades [4, 12, 14, 72, 28].

The focus of this dissertation is on the Shannon theoretic formulation in information theory, where we have a few messages (K is a constant) and message size is allowed to be arbitrarily large ($L \to \infty$). When the message size becomes large, the upload cost is negligible compared to the download cost [22][2], so that we only need to consider the download cost, measured relative to the message size. The reciprocal of download cost is the rate, i.e., the number of bits of desired information that is privately retrieved per downloaded information bit. The maximum rate possible for the PIR problem is its information theoretic capacity, C. The goal of this dissertation is to characterize the fundamental capacity limits of PIR and its variants. The main contributions of this dissertation are summarized in the next section.

1.2 Overview of the Dissertation

We start with Chapter 2, which considers the basic model of PIR. The main result of Chapter 2 is the complete characterization of the capacity of PIR for all choices of parameters. For K messages and N servers, we show that the PIR capacity is $C_{\text{PIR}} =$

[2]The justification argument (traces back to Proposition 4.1.1 of [24]) is that the upload cost does not scale with the message size. This is because we can reuse the original query functions for each part of the message.

$\left(1+1/N+1/N^2+\cdots+1/N^{K-1}\right)^{-1}$. The capacity achieving coding scheme is constructed based on a recursive algorithm, which iteratively retrieves sums of randomly permuted bits from each subset of all messages. A key feature of the scheme is that from each server, the interference (bits from undesired messages) remains the same, while desired signals are distinct. The information theoretic converse that establishes the optimality of the coding scheme also has a recursive nature. We boil down the PIR problem with K messages gradually to that with a smaller number of message, using the property of entropy functions and the privacy and decoding constraints. From that, we obtain a tight (thus precious) converse for PIR, a rare situation in network information theory.

We then proceed to consider a natural extension of PIR - PIR with colluding servers in Chapter 3. The motivation is to increase the privacy level, such that instead of requiring privacy to each individual server, we require privacy to any colluding set of up to T servers (the problem is therefore called TPIR). We characterize the capacity of TPIR to be $C_{\text{TPIR}} = \left(1+T/N+T^2/N^2+\cdots+T^{K-1}/N^{K-1}\right)^{-1}$. The novel aspect of the capacity achieving scheme for TPIR is that instead of using message bits over the binary field, we need to code the message symbols over a large field size. Although the query structure is similar to that of PIR (sums of symbols from each subset of all messages), a further layer of random mixing is required. In PIR, the randomness comes solely from permutations, while in TPIR, randomness comes in the form of random linear combinations (random matrices). Combining random matrices with generic mixing (MDS codes), we have the required elements to build the optimal scheme. The converse for TPIR is a fairly straightforward extension of that for PIR, where in order to deal with the constraint that any T servers might collude, we need an averaging argument that corresponds to Han's inequality on the dependency of average entropy on subset cardinality. We emphasize that when T divides N, the capacity of TPIR is the same as the capacity of PIR with N/T servers. The converse is immediate, while the achievability is highly non-trivial and surprising. The reason is that in PIR with N/T servers, in essence the N servers are divided into N/T disjoint colluding sets, with

T servers in each set while in TPIR, we allow arbitrary T servers to collude. However, no restriction on colluding servers does not hurt the rate. We also extend the TPIR problem to include unresponsive servers (called robust PIR). We show that not knowing in advance which N servers will respond out of $M \geq N$ servers does not hurt the rate. The coding scheme is similarly based on MDS codes.

Traditional PIR formulation assumes that each server stores all the messages, i.e., replication is used to store the messages. It is natural to explore the setting where the underlying storage is given by a distributed storage code. In Chapter 4, we consider the TPIR problem with MDS coded messages, where each message is separately coded using an MDS code. Recall that the TPIR problem requires MDS codes to construct the queries and here the messages are coded using MDS codes (hence the name MDS-PIR). When the two MDS codes are incompatible, the joint MDS-TPIR problem might be non-trivial. The focus of Chapter 4 is on a recent conjecture [33] on the capacity of MDS-TPIR, which generalizes the capacity of TPIR and the capacity of MDS-PIR [6] in a natural simple combination form. The main result of Chapter 4 is that we disprove the conjecture, showing that the capacity of MDS-TPIR is not a trivial combination of the capacity of two extreme cases that we have fully understood. The disproof is based on a novel PIR scheme, where we need richer structures in the queries. In particular, simple sums of message symbols (even combined with generic mixing) are not enough. We need to design the linear combination coefficients in a highly non-trivial way to simultaneously satisfy the privacy constraint and the correctness constraint in a download efficient manner. Although we have disproved the conjecture, the capacity of MDS-TPIR remains wide open in general. We expect novel techniques to be needed in the pursuit of the general answer.

We turn our attention to server privacy in Chapter 5. The topic of Chapter 5 is a form of oblivious transfer, where beyond user privacy, we further require server privacy, i.e., the server does not want to leak any information about the undesired messages. In other words,

we extend the privacy constraint symmetrically to include that of the servers' (hence the name symmetric PIR, or SPIR in short), so that the user only obtains his desired message and learns nothing about all other messages. We are able to find the exact capacity of SPIR, i.e., $C_{\text{SPIR}} = 1 - 1/N$, when the servers share a common random variable that is at least as long as $1/(N-1)$ of each message, and otherwise SPIR is not feasible and the capacity is zero. Interestingly, the capacity achieving scheme builds upon the optimal scheme for PIR when the number of messages approaches infinity (where the downloads are made up of linear combinations of all K messages and there is no exposed space for any single message). The converse of SPIR is another innovation, where due to the additional server privacy constraint, the derivations deviate tremendously from that of regular PIR.

Chapter 6 concentrates on the role of multiple rounds of communication. In Chapter 2 to Chapter 5, we assume the communication between the user and the server happens in one round. Naturally, the user could talk to the servers in multiple rounds and this could bring feedback and interaction, which are potentially useful. The problem is thus referred to as multiround PIR (MPIR in short). The benefits of MPIR over PIR are missing for a long time in the literature and no evidence exists that shows multiple rounds strictly help. Along the same line, we show that the capacity of MPIR is indeed the same as that of PIR, i.e., $C_{\text{MPIR}} = C_{\text{PIR}}$. To prove this result, we need a converse that holds under multiple rounds of communication. On a contrasting thought, we show that if we switch the metric from rate to storage overhead, then surprisingly and perhaps more interestingly, we show that multiple rounds strictly help (when combined with non-linear codes and ϵ-error PIR schemes). The question we ask is that in order to achieve the capacity of PIR, how much storage is required at the servers (so that each server no longer stores everything). The study of this question not only reveals the benefits of multiple rounds, but also sheds light on a number of topics that are center-stage in information theory, e.g., non-linear codes, non-Shannon inequalities, and zero-error versus ϵ-error schemes.

1.3 Notation

For $n_1, n_2 \in \mathbb{Z}, n_1 \leq n_2$, we use the notation $[n_1 : n_2] = \{n_1, n_1 + 1, \cdots, n_2\}$. The notation $X \sim Y$ is used to indicate that X and Y are identically distributed. For an index vector $\mathcal{I} = (i_1, i_2, \cdots, i_n)$, the notation $A_{\mathcal{I}}$ represents the vector $(A_{i_1}, A_{i_2}, \cdots, A_{i_n})$. Similarly, the notation $A(\mathcal{I})$ represents the vector $(A(i_1), A(i_2), \cdots, A(i_n))$. For a matrix S, the notation $S[\mathcal{I}, :]$ represents the submatrix of S formed by retaining only the rows corresponding to the elements of the vector $\mathcal{I}$. For an element j_θ in the set $\mathcal{J} = \{j_1, j_2, \cdots, j_n\}$, i.e., $j_\theta \in \mathcal{J}$, the notation $\overline{j_\theta}$ represents the complement of $\{j_\theta\}$, i.e., $\overline{j_\theta} \stackrel{\triangle}{=} \{j_1, \cdots, j_{\theta-1}, j_{\theta+1}, \cdots, j_n\}$. $(V_1; V_2; \cdots; V_n)$ refers to a matrix whose i-th row vector is $V_i, i \in [1 : n]$.

Chapter 2

Capacity of PIR

There is much recent interest in exploring the fundamental limits of PIR protocols. A PIR scheme is any mechanism by which a user may retrieve one desired message among K messages from N distributed servers (each stores all K messages) without revealing any information about which message is being retrieved to any individual server. The reason that the user could retrieve the desired information privately is that the user has multiple views, i.e., the user can download information from multiple servers, while each individual server only has a single view. In other words, we are using the distributed nature of the information retrieval system as the relative strength to protect the privacy of the user. Furthermore, we are interested in the strongest guarantee of privacy, called information theoretic privacy, i.e., even if the servers are computationally unbounded, they still obtain absolutely no information about the user's preference. The fundamental capacity limit of PIR (C_{PIR}) is the maximum number of bits of desired information that can be privately retrieved per bit of downloaded information (i.e., maximum rate).

In general, for arbitrary N and K, the best previously known achievable rate for PIR, reported in [59], is $1 - \frac{1}{N}$. Since 1 is a trivial upper bound on capacity, we know that

$1 \geq C_{\text{PIR}} \geq 1 - \frac{1}{N}$. The bounds present a reasonable approximation of capacity for large number of servers. However, in this chapter, we seek the *exact* information theoretic capacity C_{PIR} of the PIR problem, for *arbitrary* number of messages K and *arbitrary* number of servers N. Our interest in this topic started with the discovery of an intriguing connection [65] between PIR and Blind Interference Alignment [43], a problem previously studied in wireless communications. Inspired by this connection, we characterize the capacity of PIR, when we have $K = 2$ messages and the number of servers, N, is arbitrary. Building upon this preliminary success, we finally characterize the capacity of PIR, for all choices of parameters, to be

$$C_{\text{PIR}} = \left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1} \tag{2.1}$$

This chapter is devoted to prove (2.1). The organization of this chapter is as follows. Section 2.1 presents the problem statement. The exact capacity of PIR is characterized in Section 2.2. Section 2.3 presents a novel PIR scheme, and Section 2.4 provides the information theoretic converse (i.e., a tight upper bound) to establish its optimality. Section 2.5 contains a discussion of the results.

2.1 Problem Statement

Consider K independent messages $W_1, \cdots, W_K$ of size L bits each.

$$H(W_1, \cdots, W_K) = H(W_1) + \cdots + H(W_K), \tag{2.2}$$
$$H(W_1) = \cdots = H(W_K) = L. \tag{2.3}$$

There are N servers (databases) and each server stores all the messages $W_1, \cdots, W_K$. In PIR a user privately generates $\theta \in [1 : K]$ and wishes to retrieve W_θ while keeping θ a secret from each server. Depending on θ, there are K strategies that the user could employ to

privately retrieve his desired message. For example, if $\theta = k$, then in order to retrieve W_k, the user employs N queries $Q_1^{[k]}, \cdots, Q_N^{[k]}$. Since the queries are determined by the user with no knowledge of the realizations of the messages, the queries must be independent of the messages,

$$\forall k \in [1:K], \qquad I(W_1, \cdots, W_K; Q_1^{[k]}, \cdots, Q_N^{[k]}) = 0. \tag{2.4}$$

The user sends query $Q_n^{[k]}$ to the n-th server. Upon receiving $Q_n^{[k]}$, the n-th server generates an answering string $A_n^{[k]}$, which is a function of $Q_n^{[k]}$ and the data stored (i.e., all messages $W_1, \cdots, W_K$).

$$\forall k \in [1:K], \forall n \in [1:N], \quad H(A_n^{[k]} | Q_n^{[k]}, W_1, \cdots, W_K) = 0. \tag{2.5}$$

Each server returns to the user its answer $A_n^{[k]}$. From all the information that is now available to the user, he must be able to decode the desired message W_k, with probability of error P_e. The probability of error must approach zero as the size of each message L approaches infinity[1]. From Fano's inequality, we have

$$\text{[Correctness]} \quad \frac{1}{L} H(W_k | A_1^{[k]}, \cdots, A_N^{[k]}, Q_1^{[k]}, \cdots, Q_N^{[k]}) = o(L) \tag{2.6}$$

where $o(L)$ represents any term whose value approaches zero as L approaches infinity.

To protect the user's privacy, the K strategies must be indistinguishable (identically distributed) from the perspective of each server, i.e., the following privacy constraint must be satisfied[2] $\forall n \in [1:N], \forall k \in [1:K]$:

$$\text{[Privacy]} \quad (Q_n^{[1]}, A_n^{[1]}, W_1, \cdots, W_K) \sim (Q_n^{[k]}, A_n^{[k]}, W_1, \cdots, W_K) \tag{2.7}$$

[1] If P_e is required to be exactly zero, then the $o(L)$ terms can be replaced with 0.

[2] The privacy constraint is equivalently expressed as $I(\theta; Q_n^{[\theta]}, A_n^{[\theta]}, W_1, W_2, \cdots, W_K) = 0$.

The PIR *rate* characterizes how many bits of desired information are retrieved per downloaded bit, and is defined as follows.

$$R \triangleq L/D \tag{2.8}$$

where D is the expected value (over random queries) of the total number of bits downloaded by the user from all the servers. Note that because of the privacy constraint (2.7), the expected number of downloaded bits for each message must be the same.

A rate R is said to be ϵ-error achievable if there exists a sequence of PIR schemes, each of rate greater than or equal to R, for which $P_e \to 0$ as $L \to \infty$.[3] The supremum of ϵ-error achievable rates is called the ϵ-error capacity C_ϵ. A stronger (more constrained) notion of capacity is the zero-error capacity C_o, which is the supremum of zero-error achievable rates. A rate R is said to be zero-error achievable if there exists a PIR scheme of rate greater than or equal to R for which $P_e = 0$. From the definitions, it is evident that $C_o \leq C_\epsilon$. While in noise-less settings, the two are often the same, in general the inequality can be strict. Our goal is to characterize both the zero-error capacity, C_o, and the ϵ-error capacity, C_ϵ, of PIR.

2.2 Main Result: Capacity of PIR

Theorem 2.1 states the main result.

Theorem 2.1. *For the PIR problem with K messages and N servers, the capacity is*

$$C_o = C_\epsilon = \left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1}. \tag{2.9}$$

The following observations are in order.

1. For $N > 1$ servers, the capacity expression can be equivalently expressed as $(1 - \frac{1}{N})/(1 - \left(\frac{1}{N}\right)^K)$.

[3] Equivalently, for any $\epsilon > 0$, there exists a finite L_ϵ such that $P_e < \epsilon$ for all $L > L_\epsilon$.

2. The capacity is strictly higher than the previously best known achievable rate of $1 - 1/N$.

3. The capacity is a strictly decreasing function of the number of messages, K, and when the number of messages approaches infinity, the capacity approaches $1 - 1/N$.

4. The capacity is strictly increasing in the number of servers, N. As the number of servers approaches infinity, the capacity approaches 1.

5. Since the download cost is the reciprocal of the rate, Theorem 2.1 equivalently characterizes the optimal download cost per message bit as $\left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)$ bits.

6. The achievability proof for Theorem 2.1 to be presented in the next section, shows that message size approaching infinity is not necessary to approach capacity. In fact, it suffices to have messages of size equal to any positive integer multiple of N^K bits (or N^K symbols in any finite field) each to achieve a rate exactly equal to capacity, and with zero-error. A further reduction of the message size to N^{K-1} bits each for capacity achieving schemes is discussed in Section 2.5.

7. The upper bound proof will show that no PIR scheme can achieve a rate higher than capacity with $P_e \to 0$ as message size $L \to \infty$. Unbounded message size is essential to the information theoretic formulation of capacity. However, from a practical standpoint, it is natural to ask what this means if the message size is limited. The optimal rate for limited message size is found in Section 2.5. We note that regardless of message size, C_o (and therefore also C_ϵ) is always an upper bound on zero-error rate.

2.3 Theorem 2.1: Achievability

We present a zero-error PIR scheme for $L = N^K$ bits per message in this section, whose rate is equal to capacity. Note that a zero-error scheme with finite message length can always be

repeatedly applied to create a sequence of schemes with message-lengths approaching infinity for which the probability of error approaches (is) zero. Thus, the same scheme will suffice as the proof of achievability for both zero-error and ϵ-error capacity.

Let us illustrate the intuition behind the achievable scheme with a few simple examples. Then, based on the examples, we will present an algorithmic description of the achievable scheme for arbitrary number of messages, K and arbitrary number of servers, N. We will then revisit the examples in light of the algorithmic formulation. Finally, we will prove that the scheme is both correct and private, and that its rate is equal to the capacity.

2.3.1 Two Examples to Illustrate the Key Ideas

The capacity achieving PIR scheme has a myopic or greedy character, in that it starts with a narrow focus on the retrieval of the desired message bits from the first server, but grows into a full fledged scheme based on iterative application of three principles:

(1) *Enforcing Symmetry Across Servers*

(2) *Enforcing Message Symmetry within the Query to Each Server*

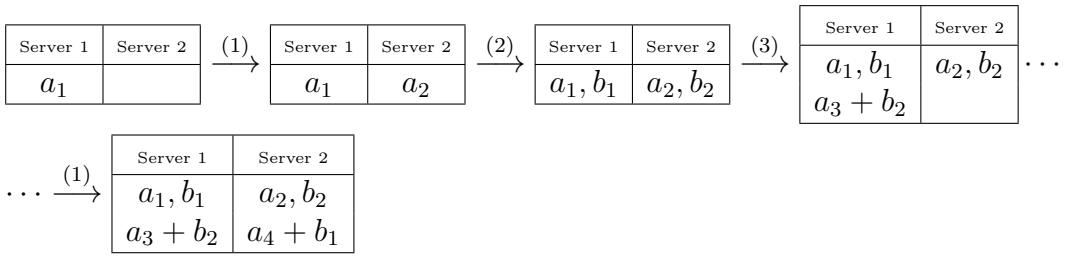
(3) *Exploiting Side Information of Undesired Messages to Retrieve New Desired Information*

2.3.1.1 Example 1: $N = 2, K = 2$

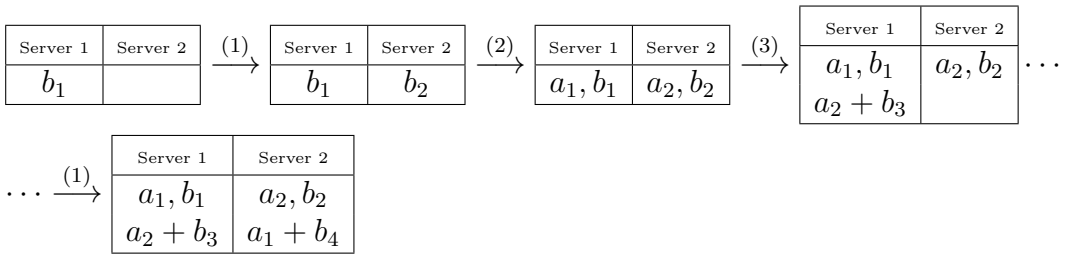
Consider the simplest PIR setting, with $N = 2$ servers, and $K = 2$ messages with $L = N^K = 4$ bits per message. Let $[a_1, a_2, a_3, a_4]$ represent a random permutation of $L = 4$ bits from W_1. Similarly, let $[b_1, b_2, b_3, b_4]$ represent an independent random permutation of $L = 4$ bits from W_2. These permutations are generated privately and uniformly by the user.

Suppose the desired message is W_1, i.e., $\theta = 1$. We start with a query that requests the first bit a_1 from the first server (Server 1). Applying server symmetry, we simultaneously request a_2 from the second server (Server 2). Next, we enforce message symmetry, by including

queries for b_1 and b_2 as the counterparts for a_1 and a_2. Now we have side information of b_2 from Server 2 to be exploited in an additional query to Server 1, which requests a new desired information bit a_3 mixed with b_2. Finally, applying server symmetry we have the corresponding query $a_4 + b_1$ for Server 2. At this point the queries satisfy symmetry across servers, message symmetry within the query to each server, and all undesired side information is exploited, so the construction is complete. The process is explained below, where the number above an arrow indicates which of the three principles highlighted above is used in each step.



Similarly, the queries for $\theta = 2$ are constructed as follows.



Privacy is ensured by noting that $[a_1, a_2, a_3, a_4]$ is a random permutation of W_1 and $[b_1, b_2, b_3, b_4]$ is an independent random permutation of W_2. These permutations are only known to the user and not to the servers. Therefore, regardless of the desired message, each server is asked for one randomly chosen bit of each message and a sum of a different pair of randomly chosen bits from each message. Since the permutations are uniform, all possible realizations are equally likely, and privacy is guaranteed.

To verify correctness, note that every desired bit is either downloaded directly or added with known side information which can be subtracted to retrieve the desired bit value. Thus, the desired message bits are successfully recoverable from the downloaded information.

Now, consider the rate of this scheme. The total number of downloaded bits is 6 and the number of desired bits is 4. Thus, the rate of this scheme is $4/6 = 2/3$ which matches the capacity for this case.

Finally, let us represent the structure of the queries (to any server) in the matrix shown below.

$$\begin{bmatrix} \underline{a} \\ \underline{b} \\ \underline{a}+\underline{b} \end{bmatrix}$$

$\underline{a}$ ($\underline{b}$) represents a place-holder for a distinct element of a_i (b_j). The key to the structure is that it is made up of sums (a single variable is also named a (trivial) sum) of message bits, no message bit appears more than once, and all possible assignments of message bits to these place-holders are equally likely. The structure matrix will be useful for the algorithmic description later.

2.3.1.2 Example 2: $N = 3, K = 3$

The second example is when $N = 3$, $K = 3$. In this case, all messages have $L = N^K = 27$ bits. The construction of the optimal PIR scheme for $N = 3, K = 3$ is illustrated below, where $[a_1, \cdots, a_{27}], [b_1, \cdots, b_{27}], [c_1, \cdots, c_{27}]$ are three i.i.d. uniform permutations of bits from W_1, W_2, W_3, respectively. The construction of the queries from each server when $\theta = 1$ may be visualized as follows.

Server 1	Server 2	Server 3
a_1		

$\xrightarrow{(1)}$

Server 1	Server 2	Server 3
a_1	a_2	a_3

$\xrightarrow{(2)}$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3

$\cdots$

$\cdots \xrightarrow{(3)}$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_4 + b_2$		
$a_5 + c_2$		
$a_6 + b_3$		
$a_7 + c_3$		

$\xrightarrow{(1)}$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_4 + b_2$	$a_8 + b_1$	$a_{12} + b_1$
$a_5 + c_2$	$a_9 + c_1$	$a_{13} + c_1$
$a_6 + b_3$	$a_{10} + b_3$	$a_{14} + b_2$
$a_7 + c_3$	$a_{11} + c_3$	$a_{15} + c_2$

$\cdots$

$\cdots \xrightarrow{(2)}$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_4 + b_2$	$a_8 + b_1$	$a_{12} + b_1$
$a_5 + c_2$	$a_9 + c_1$	$a_{13} + c_1$
$a_6 + b_3$	$a_{10} + b_3$	$a_{14} + b_2$
$a_7 + c_3$	$a_{11} + c_3$	$a_{15} + c_2$
$b_4 + c_4$	$b_6 + c_6$	$b_8 + c_8$
$b_5 + c_5$	$b_7 + c_7$	$b_9 + c_9$

$\xrightarrow{(3)}$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_4 + b_2$	$a_8 + b_1$	$a_{12} + b_1$
$a_5 + c_2$	$a_9 + c_1$	$a_{13} + c_1$
$a_6 + b_3$	$a_{10} + b_3$	$a_{14} + b_2$
$a_7 + c_3$	$a_{11} + c_3$	$a_{15} + c_2$
$b_4 + c_4$	$b_6 + c_6$	$b_8 + c_8$
$b_5 + c_5$	$b_7 + c_7$	$b_9 + c_9$
$a_{16} + b_6 + c_6$		
$a_{17} + b_7 + c_7$		
$a_{18} + b_8 + c_8$		
$a_{19} + b_9 + c_9$		

$\cdots$

$\cdots \xrightarrow{(1)}$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_4 + b_2$	$a_8 + b_1$	$a_{12} + b_1$
$a_5 + c_2$	$a_9 + c_1$	$a_{13} + c_1$
$a_6 + b_3$	$a_{10} + b_3$	$a_{14} + b_2$
$a_7 + c_3$	$a_{11} + c_3$	$a_{15} + c_2$
$b_4 + c_4$	$b_6 + c_6$	$b_8 + c_8$
$b_5 + c_5$	$b_7 + c_7$	$b_9 + c_9$
$a_{16} + b_6 + c_6$	$a_{20} + b_4 + c_4$	$a_{24} + b_4 + c_4$
$a_{17} + b_7 + c_7$	$a_{21} + b_5 + c_5$	$a_{25} + b_5 + c_5$
$a_{18} + b_8 + c_8$	$a_{22} + b_8 + c_8$	$a_{26} + b_6 + c_6$
$a_{19} + b_9 + c_9$	$a_{23} + b_9 + c_9$	$a_{27} + b_7 + c_7$

Similarly, the queries when $\theta = 2, 3$ are as follows.

$\theta = 2$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_2 + b_4$	$a_1 + b_8$	$a_1 + b_{12}$
$b_5 + c_2$	$b_9 + c_1$	$b_{13} + c_1$
$a_3 + b_6$	$a_3 + b_{10}$	$a_2 + b_{14}$
$b_7 + c_3$	$c_3 + b_{11}$	$b_{15} + c_2$
$a_4 + c_4$	$a_6 + c_6$	$a_8 + c_8$
$a_5 + c_5$	$a_7 + c_7$	$a_9 + c_9$
$a_6 + b_{16} + c_6$	$a_4 + b_{20} + c_4$	$a_4 + b_{24} + c_4$
$a_7 + b_{17} + c_7$	$a_5 + b_{21} + c_5$	$a_5 + b_{25} + c_5$
$a_8 + b_{18} + c_8$	$a_8 + b_{22} + c_8$	$a_6 + b_{26} + c_6$
$a_9 + b_{19} + c_9$	$a_9 + b_{23} + c_9$	$a_7 + b_{27} + c_7$

$\theta = 3$

Server 1	Server 2	Server 3
a_1, b_1, c_1	a_2, b_2, c_2	a_3, b_3, c_3
$a_2 + c_4$	$a_1 + c_8$	$a_1 + c_{12}$
$b_2 + c_5$	$b_1 + c_9$	$b_1 + c_{13}$
$a_3 + c_6$	$a_3 + c_{10}$	$a_2 + c_{14}$
$b_3 + c_7$	$b_3 + c_{11}$	$b_2 + c_{15}$
$a_4 + b_4$	$a_6 + b_6$	$a_8 + b_8$
$a_5 + b_5$	$a_7 + b_7$	$a_9 + b_9$
$a_6 + b_6 + c_{16}$	$a_4 + b_4 + c_{20}$	$a_4 + b_4 + c_{24}$
$a_7 + b_7 + c_{17}$	$a_5 + b_5 + c_{21}$	$a_5 + b_5 + c_{25}$
$a_8 + b_8 + c_{18}$	$a_8 + b_8 + c_{22}$	$a_6 + b_6 + c_{26}$
$a_9 + b_9 + c_{19}$	$a_9 + b_9 + c_{23}$	$a_7 + b_7 + c_{27}$

The structure of the queries is summarized in the structure matrix presented below. Note again that the structure matrix is made up of sums of place-holders of message bits, no message bit appears more than once, and the assignment of all messages bits to these place-holders is equally likely.

$\underline{a}$
$\underline{b}$
$\underline{c}$
$\underline{a} + \underline{b}$
$\underline{a} + \underline{b}$
$\underline{a} + \underline{c}$
$\underline{a} + \underline{c}$
$\underline{b} + \underline{c}$
$\underline{b} + \underline{c}$
$\underline{a} + \underline{b} + \underline{c}$
$\underline{a} + \underline{b} + \underline{c}$
$\underline{a} + \underline{b} + \underline{c}$
$\underline{a} + \underline{b} + \underline{c}$

The examples illustrated above generalize naturally to arbitrary N and K. As we proceed to proofs of privacy and correctness and to calculate the rate for arbitrary parameters, a more formal algorithmic description will be useful.

2.3.2 Formal Description of Achievable Scheme

For all $k \in [1:K]$, define[4] vectors $U_k = [u_k(1), u_k(2), \cdots, u_k(N^K)]$. We will use the terminology k-**sum** to denote an expression representing the sum of k distinct variables, each drawn from a *different* U_j vector, i.e., $u_{j_1}(i_1) + u_{j_2}(i_2) + \cdots + u_{j_k}(i_k)$, where $j_1, j_2, \cdots, j_k \in [1:K]$ are all *distinct* indices. Furthermore, we will define such a k-sum to be of **type** $\{j_1, j_2, \cdots, j_k\}$.

The achievable scheme is comprised of the following elements: 1) a fixed query set structure, 2) an algorithm to generate the query set as a deterministic function of θ, and 3) a random mapping from U_k variables to message bits, which will produce the actual queries to be sent to the servers. The random mapping will be privately generated by the user, unknown to the servers. These elements are described next.

2.3.2.1 A Fixed Query Set Structure

For all $\text{DB} \in [1:N], \theta \in [1:K]$, let us define 'query sets': $Q(\text{DB}, \theta)$, which must satisfy the following structural properties. Each $Q(\text{DB}, \theta)$ must be the union of K disjoint subsets called "blocks", that are indexed by $k \in [1:K]$. Block k must contain only k-sums. Note that there are only $\binom{K}{k}$ possible "*types*" of k-sums. Block k must contain all of them. We require that block k contains exactly $(N-1)^{k-1}$ distinct instances of *each type* of k-sum. This requirement is chosen following the intuition from the three principles, and as we will prove shortly, it ensures that the resulting scheme is capacity achieving. Thus, the total number of elements contained in block k must be $\binom{K}{k}(N-1)^{k-1}$, and the total number of elements in each query set must be $|Q(\text{DB}, \theta)| = \sum_{k=1}^{K} \binom{K}{k}(N-1)^{k-1}$. For example, for $N = 3, K = 3$, as illustrated previously, there are $\binom{3}{1} = 3$ types of 1-sums (a, b, c) and we have $(3-1)^{1-1} = 1$ instances of each; there are $\binom{3}{2} = 3$ types of 2-sums ($a+b$, $b+c$, $c+a$) and we have $(3-1)^{2-1} = 2$ instances of each; and there is $\binom{3}{3} = 1$ type of 3-sum ($a+b+c$)

[4]Since the number of messages, K, can be arbitrary, and we have only 26 letters in the English alphabet, instead of a_i, b_j, c_k, etc., we now use $u_1(i), u_2(j), u_3(k)$, etc., to represent random permutations of bits from different messages.

and we have $(3-1)^{3-1} = 4$ instances of it. The query to each server has this structure. Furthermore, no message symbol can appear more than once in a query set for any given server.

The structure of Block k of the query $Q(\text{DB}, \theta)$, enforced by the constraints described above, is illustrated in Figure 2.1 through an enumeration of all its elements. In the figure, each $\underline{U_j}$ represents a place-holder for a *distinct* element of U_j. Note that the structure as represented in Figure 2.1 is fixed regardless of θ and DB. All query sets must have the same fixed structure.

Type No.	Type of k-sum	Instance No.	Enumerated elements of Block k
1.	$\{1, 2, \cdots, k-2, k-1, k\}$	1. 2. $\vdots$ $(N-1)^{k-1}$.	$\underline{U_1} + \underline{U_2} + \cdots + \underline{U_{k-2}} + \underline{U_{k-1}} + \underline{U_k}$ $\underline{U_1} + \underline{U_2} + \cdots + \underline{U_{k-2}} + \underline{U_{k-1}} + \underline{U_k}$ $\vdots$ $\underline{U_1} + \underline{U_2} + \cdots + \underline{U_{k-2}} + \underline{U_{k-1}} + \underline{U_k}$
2.	$\{1, 2, \cdots, k-2, k-1, k+1\}$	1. 2. $\vdots$ $(N-1)^{k-1}$.	$\underline{U_1} + \underline{U_2} + \cdots + \underline{U_{k-2}} + \underline{U_{k-1}} + \underline{U_{k+1}}$ $\underline{U_1} + \underline{U_2} + \cdots + \underline{U_{k-2}} + \underline{U_{k-1}} + \underline{U_{k+1}}$ $\vdots$ $\underline{U_1} + \underline{U_2} + \cdots + \underline{U_{k-2}} + \underline{U_{k-1}} + \underline{U_{k+1}}$
	$\vdots$	$\vdots$	$\vdots$
i.	$\{i_1, i_2, \cdots, i_k\}$	1. 2. $\vdots$ $(N-1)^{k-1}$.	$\underline{U_{i_1}} + \underline{U_{i_2}} + \cdots + \underline{U_{i_k}}$ $\underline{U_{i_1}} + \underline{U_{i_2}} + \cdots + \underline{U_{i_k}}$ $\vdots$ $\underline{U_{i_1}} + \underline{U_{i_2}} + \cdots + \underline{U_{i_k}}$
	$\vdots$	$\vdots$	$\vdots$
$\binom{K}{k}$.	$\{K-k+1, K-k+2, \cdots, K\}$	1. 2. $\vdots$ $(N-1)^{k-1}$.	$\underline{U_{K-k+1}} + \underline{U_{K-k+2}} + \cdots + \underline{U_K}$ $\underline{U_{K-k+1}} + \underline{U_{K-k+2}} + \cdots + \underline{U_K}$ $\vdots$ $\underline{U_{K-k+1}} + \underline{U_{K-k+2}} + \cdots + \underline{U_K}$

Figure 2.1: Structure of Block k of $Q(\text{DB}, \theta)$.

2.3.2.2 A Deterministic Algorithm

Next we present the algorithm which will produce $Q(\mathrm{DB}, \theta)$ for all $\mathrm{DB} \in [1:N]$ as function of θ alone. In particular, this algorithm will determine which U_j variable is assigned to each place-holder value in the query structure described earlier. To present the algorithm we need these definitions.

For each $k \in [1:K]$, let $\texttt{new}(U_k)$ be a function that, starting with $u_k(1)$, returns the "next" variable in U_k each time it is called with U_k as its argument. So, for example, the following sequence of calls to this function: $\texttt{new}(U_2), \texttt{new}(U_1), \texttt{new}(U_1), \texttt{new}(U_1) + \texttt{new}(U_2)$ will produce $u_2(1), u_1(1), u_1(2), u_1(3) + u_2(2)$ as the output.

Let us partition each block k into two subsets — a subset $\mathcal{M}$ that contains the k-sums which include a variable from U_θ, and a subset $\mathcal{I}$ which contains all the remaining k-sums which contain no symbols from U_θ.[5]

Using these definitions the algorithm is presented next. Algorithm 1 appears in the next page.

Algorithm 1 realizes the 3 principles as follows. The for-loop in steps 5 to 14 ensures server symmetry (principle (1)). The for-loop in steps 10 to 13 ensures message symmetry within one server (principle (2)). Steps 7 to 8 retrieve new desired information using existing side information (principle (3)).

The proof that the $Q(\mathrm{DB}, \theta)$ produced by this algorithm indeed satisfy the query structure described before, is presented in Lemma 2.1.

2.3.2.3 Ordered Representation and Mapping to Message Bits to Produce $Q^{[\theta]}_{\mathbf{DB}}$

It is useful at this point to have an ordered vector representation of the query structure, as well as the query set $Q(\mathrm{DB}, \theta)$. For the query structure, let us first order the blocks in

[5]The nomenclature $\mathcal{M}$ and $\mathcal{I}$ corresponds to 'message' and 'interference', respectively.

[6]For any set Q, when accessing its elements in an algorithm (e.g., for all $q \in Q$, do ...), the output of the algorithm will in general depend on the order in which the elements are accessed. However, for our algorithmic descriptions the order is not important, i.e., any form of ordered access produces an optimal PIR scheme. By default, a natural lexicographic ordering may be assumed.

Algorithm 1 Input: θ. Output: Query sets $Q(\text{DB}, \theta)$, $\forall \text{DB} \in [1:N]$

Initialize: All query sets are initialized as null sets. Also initialize Block $\leftarrow 1$;
for DB $= 1:N$ **do**

$$Q(\text{DB}, \theta, \text{Block}, \mathcal{M}) \leftarrow \{\texttt{new}(U_\theta)\} \tag{2.10}$$

$$Q(\text{DB}, \theta, \text{Block}, \mathcal{I}) \leftarrow \bigcup_{k \in [1:K], k \neq \theta} \{\texttt{new}(U_k)\} \tag{2.11}$$

end for
for Block $= 2:K$ **do** {Generate each block...}
  for DB $= 1:N$ **do** {for each server...}
    for each DB$' = 1:N$ **and** DB$' \neq$ DB **do** {by looking at all 'other' servers, and...}
      for each[6] $q \in Q(\text{DB}', \theta, \text{Block} - 1, \mathcal{I})$ **do** {use the '$\mathcal{I}$' terms from their previous block...}

$$Q(\text{DB}, \theta, \text{Block}, \mathcal{M}) \leftarrow Q(\text{DB}, \theta, \text{Block}, \mathcal{M}) \cup \{\texttt{new}(U_\theta) + q\} \tag{2.12}$$

{...to create new $\mathcal{M}$ terms for this block by adding a new U_θ variable to each term.}
      end for (q)
    end for (DB$'$)
    for all distinct $\{i_1, i_2, \cdots, i_{\text{Block}}\} \subset [K]/\{\theta\}$ **do** {For all "types" that do not include θ...}
      for $i = 1:(N-1)^{\text{Block}-1}$ **do** {generate exactly $(N-1)^{\text{Block}-1}$ new instances of each.}

$$Q(\text{DB}, \theta, \text{Block}, \mathcal{I}) \leftarrow Q(\text{DB}, \theta, \text{Block}, \mathcal{I}) \cup \{\texttt{new}(U_{i_1}) + \texttt{new}(U_{i_2}) + \cdots + \texttt{new}(U_{i_{\text{Block}}})\}$$

      end for (i)
    end for ($\{i_1, i_2, \cdots, i_{\text{Block}}\}$)
  end for (DB)
end for (Block)
for DB $= 1:N$ **do**
  $Q(\text{DB}, \theta) \leftarrow \bigcup_{\text{Block} \in [K]} \left(Q(\text{DB}, \theta, \text{Block}, \mathcal{I}) \cup Q(\text{DB}, \theta, \text{Block}, \mathcal{M})\right)$
end for

increasing order of block index. Then within the k-th block, $k \in [1:K]$, arrange the "types" of k-sums by first sorting the indices into $(i_1, i_2, \cdots, i_k)$ such that $i_1 < i_2 < \cdots < i_k$, and then arranging the k-tuples $(i_1, i_2, \cdots, i_k)$ in increasing lexicographic order. For the query set, we have the same arrangement for blocks and types, but then for each given type, we further sort the multiple instances of that type by the i index of the $u_k(i)$ term with the smallest k value in that type. Let $\vec{Q}(\text{DB}, \theta)$ denote the ordered representation of $Q(\text{DB}, \theta)$. Next we will map the $u_k(i)$ variables to message bits to produce a query vector.

Suppose each message W_k, $k \in [1:K]$, is represented by the vector $W_k = [w_k(1), w_k(2), \cdots, w_k(N^K)]$, where $w_k(i)$ is the binary random variable representing the i-th bit of W_k. The user privately chooses permutations $\gamma_1, \gamma_2, \cdots, \gamma_K$, uniformly randomly from all possible $(N^K)!$ permutations over the index set $[1:N^K]$, so that the permutations are independent of each other and of θ. The U_k variables are mapped to the messages W_k through the random permutation γ_k, $\forall k \in [1:K]$. Let Γ denote an operator that replaces every instance of $u_k(i)$ with $w_k(\gamma_k(i))$, $\forall k \in [1:K], i \in [1:N^K]$. For example, $\Gamma(\{u_1(2), u_3(4) + u_5(6)\}) = \{w_1(\gamma_1(2)), w_3(\gamma_3(4)) + w_5(\gamma_5(6))\}$. This random mapping, applied to $\vec{Q}(\text{DB}, \theta)$ produces the actual query vector $Q_{\text{DB}}^{[\theta]}$ that is sent to server DB as

$$Q_{\text{DB}}^{[\theta]} \quad = \quad \text{“}\Gamma\big(\vec{Q}(\text{DB}, \theta)\big)\text{”} \tag{2.13}$$

We use the double-quotes notation around a random variable to represent the *query* about its realization. For example, while $w_1(1)$ is a random variable, which may take the value 0 or 1, in our notation "$w_1(1)$" is not random, because it only represents the *question*: "what is the value of $w_1(1)$?" This is an important distinction, in light of constraints such as (2.4) which require that queries must be independent of messages, i.e., message realizations. Note that our queries are indeed independent of message realizations because the queries are generated by the user with no knowledge of message realizations. Also note that the only randomness in $Q_{\text{DB}}^{[\theta]}$ is because of the θ and the random permutation Γ.

2.3.3 The Two Examples Revisited

To illustrate the algorithmic formulation, let us revisit the two examples that were presented previously from an intuitive standpoint.

2.3.3.1 Example 1. $N = 2, K = 2$

Consider the simplest PIR setting, with $N = 2$ servers, and $K = 2$ messages with $L = N^K = 4$ bits per message. Instead of our usual notation, i.e., $U_1 = [u_1(1), u_1(2), u_1(3), u_1(4)]$, for this example it will be less cumbersome to use the notation $U_1 = [a_1, a_2, a_3, a_4]$. Similarly, $U_2 = [b_1, b_2, b_3, b_4]$. The query structure and the outputs produced by the algorithm for $\theta = 1$ as well as for $\theta = 2$ are shown below. The blocks are separated by horizontal lines. Within each block the $\mathcal{I}$ terms are highlighted in red and the $\mathcal{M}$ terms are in black. Note that there are no terms in $\mathcal{I}$ for the last block (Block K), because there are no K-sums that do not include the U_θ variables.

Query Structure

	$\vec{Q}(\text{DB}, \theta)$
Block 1	$\underline{U_1}$
	$\underline{U_2}$
Block 2	$\underline{U_1} + \underline{U_2}$

Ordered Output of Algorithm 1 for $\theta = 1$

$\vec{Q}(\text{DB1}, \theta = 1)$	$\vec{Q}(\text{DB2}, \theta = 1)$
a_1	a_2
b_1	b_2
$a_3 + b_2$	$a_4 + b_1$

Ordered Output of Algorithm 1 for $\theta = 2$

$\vec{Q}(\text{DB1}, \theta = 2)$	$\vec{Q}(\text{DB2}, \theta = 2)$
a_1	a_2
b_1	b_2
$a_2 + b_3$	$a_1 + b_4$

To verify that the scheme is correct, note that whether $\theta = 1$ or $\theta = 2$, every desired bit is either downloaded directly (block 1) or appears with known side information that is available from the other server. To see why privacy holds, recall that the queries are ultimately presented to the server in terms of the message variables and the mapping from U_k to W_k is uniformly random and independent of θ. So, consider an arbitrary realization of the query with (distinct) message bits $w_1(i_1), w_2(i_2)$ from W_1 and $w_2(j_1), w_2(j_2)$ from W_2.

$\Gamma(\vec{Q}(\text{DB}, \theta))$
$w_1(i_1)$
$w_2(j_1)$
$w_1(i_2) + w_2(j_2)$

Given this query, the probability that it was generated for $\theta = 1$ is $((\frac{1}{4})(\frac{1}{3}))^2 = \frac{1}{144}$, which is the same as the probability that it was generated for $\theta = 2$. Thus, the query provides the server no information about θ, and the scheme is private. This argument is presented in detail and generalized to arbitrary K and N in Lemma 2.3. Finally, consider the rate of this scheme. The total number of downloaded bits is 6, and the number of desired bits downloaded is 4, so the rate of this scheme is $4/6 = 2/3$ which matches the capacity for this case.

2.3.3.2 Example 2. $N = 3, K = 3$

The second example is when $K = 3$, $N = 3$. In this case, both messages have $L = N^K = 27$ bits. $U_1 = [a_1, a_2, \cdots, a_{27}], U_2 = [b_1, b_2, \cdots, , b_{27}], U_3 = [c_1, c_2, \cdots, c_{27}]$. The query structure and the output of the algorithm for $\theta = 1$ are shown below.

Query Strucure

	$\vec{Q}(\text{DB}, \theta)$
Block 1	$\underline{U_1}$
	$\underline{U_2}$
	$\underline{U_3}$
Block 2	$\underline{U_1} + \underline{U_2}$
	$\underline{U_1} + \underline{U_2}$
	$\underline{U_1} + \underline{U_3}$
	$\underline{U_1} + \underline{U_3}$
	$\underline{U_2} + \underline{U_3}$
	$\underline{U_2} + \underline{U_3}$
Block 3	$\underline{U_1} + \underline{U_2} + \underline{U_3}$
	$\underline{U_1} + \underline{U_2} + \underline{U_3}$
	$\underline{U_1} + \underline{U_2} + \underline{U_3}$
	$\underline{U_1} + \underline{U_2} + \underline{U_3}$

Ordered Output of Algorithm 1 for $\theta = 1$

$\vec{Q}(\text{DB1}, \theta = 1)$	$\vec{Q}(\text{DB2}, \theta = 1)$	$\vec{Q}(\text{DB3}, \theta = 1)$
a_1	a_2	a_3
b_1	b_2	b_3
c_1	c_2	c_3
$a_4 + b_2$	$a_8 + b_1$	$a_{12} + b_1$
$a_6 + b_3$	$a_{10} + b_3$	$a_{14} + b_2$
$a_5 + c_2$	$a_9 + c_1$	$a_{13} + c_1$
$a_7 + c_3$	$a_{11} + c_3$	$a_{15} + c_2$
$b_4 + c_4$	$b_6 + c_6$	$b_8 + c_8$
$b_5 + c_5$	$b_7 + c_7$	$b_9 + c_9$
$a_{16} + b_6 + c_6$	$a_{20} + b_4 + c_4$	$a_{24} + b_4 + c_4$
$a_{17} + b_7 + c_7$	$a_{21} + b_5 + c_5$	$a_{25} + b_5 + c_5$
$a_{18} + b_8 + c_8$	$a_{22} + b_8 + c_8$	$a_{26} + b_6 + c_6$
$a_{19} + b_9 + c_9$	$a_{23} + b_9 + c_9$	$a_{27} + b_7 + c_7$

The output of Algorithm 1, for $\theta = 2$, is shown next.

$\vec{Q}(\text{DB1}, \theta = 2)$
a_1
b_1
c_1
$a_2 + b_4$
$a_3 + b_6$
$a_4 + c_4$
$a_5 + c_5$
$b_5 + c_2$
$b_7 + c_3$
$a_6 + b_{16} + c_6$
$a_7 + b_{17} + c_7$
$a_8 + b_{18} + c_8$
$a_9 + b_{19} + c_9$

$\vec{Q}(\text{DB2}, \theta = 2)$
a_2
b_2
c_2
$a_1 + b_8$
$a_3 + b_{10}$
$a_6 + c_6$
$a_7 + c_7$
$b_9 + c_1$
$b_{11} + c_3$
$a_4 + b_{20} + c_4$
$a_5 + b_{21} + c_5$
$a_8 + b_{22} + c_8$
$a_9 + b_{23} + c_9$

$\vec{Q}(\text{DB3}, \theta = 2)$
a_3
b_3
c_3
$a_1 + b_{12}$
$a_3 + b_{14}$
$a_8 + c_8$
$a_9 + c_9$
$b_{13} + c_1$
$b_{15} + c_3$
$a_4 + b_{24} + c_4$
$a_5 + b_{25} + c_5$
$a_6 + b_{26} + c_6$
$a_7 + b_{27} + c_7$

The output of Algorithm 1, for $\theta = 3$, is shown next.

$\vec{Q}(\text{DB1}, \theta = 3)$
a_1
b_1
c_1
$a_4 + b_4$
$a_5 + b_5$
$a_2 + c_4$
$a_3 + c_6$
$b_2 + c_5$
$b_3 + c_7$
$a_6 + b_6 + c_{16}$
$a_7 + b_7 + c_{17}$
$a_8 + b_8 + c_{18}$
$a_9 + b_9 + c_{19}$

$\vec{Q}(\text{DB2}, \theta = 3)$
a_2
b_2
c_2
$a_6 + b_6$
$a_7 + b_7$
$a_1 + c_8$
$a_3 + c_{10}$
$b_1 + c_9$
$b_3 + c_{11}$
$a_4 + b_4 + c_{20}$
$a_5 + b_5 + c_{21}$
$a_8 + b_8 + c_{22}$
$a_9 + b_9 + c_{23}$

$\vec{Q}(\text{DB3}, \theta = 3)$
a_3
b_3
c_3
$a_8 + b_8$
$a_9 + b_9$
$a_1 + c_{12}$
$a_2 + c_{14}$
$b_1 + c_{13}$
$b_2 + c_{15}$
$a_4 + b_4 + c_{24}$
$a_5 + b_5 + c_{25}$
$a_6 + b_6 + c_{26}$
$a_7 + b_7 + c_{27}$

Note that this construction retrieves 27 desired message bits out of a total of 39 downloaded bits, so its rate is $27/39 = 9/13$, which matches the capacity for this case.

2.3.4 Proof of Correctness, Privacy and Achieving Capacity

The following lemma confirms that the query set produced by the algorithm satisfies the required structural properties.

Lemma 2.1. (**Structure of** $Q(DB, \theta)$) *For any* $\theta \in [1 : K]$ *and for any* $DB \in [1 : N]$, *the* $Q(DB, \theta)$ *produced by Algorithm 1 satisfies the following properties.*

1. *For all* $k \in [1 : K]$, *block* k *contains exactly* $(N-1)^{k-1}$ *instances of* k*-sums of each possible type.*

2. *No* $u_k(i), i \in [1 : N^K]$ *variable appears more than once within* $Q(DB, \theta)$ *for any given DB.*

3. *Exactly* N^{K-1} *variables for each* U_k, $k \in [1 : K]$, *appear in the query set* $Q(DB, \theta)$.

4. *The size of* $Q(DB, \theta)$ *is* $N^{K-1} + \frac{1}{N-1}(N^{K-1} - 1)$.

Proof. 1. Fix any arbitrary N. The proof is based on induction on the claim $S(k)$, defined as follows.

$S(k)$: "*Block* k *contains exactly* $(N-1)^{k-1}$ *instances of* k*-sums of all possible types.*"

The basis step is when $k = 1$. This step is easily verified, because a 1-sum is simply one variable, of which there are K possible types, and from (2.10), (2.11) in Algorithm 1, we note that the first block always consists of one variable of each vector $U_k, k \in [1 : K]$.

We next proceed to the inductive step. Suppose $S(k)$ is true. Then we wish to prove that $S(k+1)$ must be true as well. Here we have $\text{Block} = k + 1$. First, consider $(k+1)$-sums of type $\{i_1, i_2, \cdots, i_{k+1}\} \subset [1 : K]/\{\theta\}$ where none of the indices is θ. These belong in $Q(\text{DB}, \theta, k+1, \mathcal{I})$, and from line 11 of the algorithm it is verified that exactly $(N-1)^{\text{Block}-1} = (N-1)^k$ instances are generated of this type. Next, consider the $(k+1)$-sums of type $\{i_1, i_2, \cdots, i_k, \theta\}$ where one of the indices is θ. These belong to $Q(\text{DB}, \theta, k+1, \mathcal{M})$ and are obtained by adding $\texttt{new}(U_\theta)$ to each of the k-sums of type $\{i_1, i_2, \cdots, i_k\}$ that belong to $Q(\text{DB}', \theta, k, \mathcal{I})$ for all $\text{DB}' \neq \text{DB}$. Therefore, the number of instances of $(k+1)$-sums of type $\{i_1, i_2, \cdots, i_k, \theta\}$ in $Q(\text{DB}, \theta, k+1, \mathcal{M})$ must be equal to the product of the number of 'other' servers DB', which is equal to $N-1$, and the number of instances of type $\{i_1, i_2, \cdots, i_k\}$ in each server DB', which is

equal to $(N-1)^{k-1}$ because $S(k)$ is assumed to be true as the induction hypothesis. $(N-1) \times (N-1)^{k-1} = (N-1)^k$, and thus, we have shown that $S(k+1)$ is true, completing the proof by induction.

2. From (2.10),(2.12), we see that for each block, the desired variables, i.e., the U_θ variables appear only through the $\texttt{new}(U_\theta)$ function so that each of them only appears once. For the non-desired variables $U_k, k \neq \theta$, we see that the only time that they do not appear through the $\texttt{new}(U_k)$ function is when they enter through q in (2.12). However, from (2.12) we see that these variables come from the $\mathcal{I}$ part of the previous block of other servers, where each of them was only introduced once through a $\texttt{new}(U_k)$ function. Moreover, each term from the $\mathcal{I}$ part of the previous block of other servers is used exactly once. Therefore, these U_k variables also appear no more than once in the query set of a given server.

3. Since we have shown that no variable appears more than once, we only need to count the number of times each vector $U_k, k \in [1:K]$ is invoked within $Q(\mathrm{DB}, \theta)$. Consider any particular vector, say U_j. The number of possible types of k-sums that include index j is $\binom{K-1}{k-1}$. As we have also shown, the k-th block contains $(N-1)^{k-1}$ instances of k-sums of each type. Therefore, the number of instances of vector U_j in block k is $(N-1)^{k-1}\binom{K-1}{k-1}$. Summing over all K blocks within $Q(\mathrm{DB}, \theta)$ we find

$$\begin{aligned} & \sum_{k=1}^{K} (N-1)^{k-1} \binom{K-1}{k-1} \\ = \; & (N-1+1)^{K-1} \quad \text{(Binomial Identity)} && (2.14) \\ = \; & N^{K-1} && (2.15) \end{aligned}$$

4. The k-th block of $Q(\mathrm{DB}, \theta)$ contains $(N-1)^{k-1}$ instances of k-sums of each possible type, and there are $\binom{K}{k}$ possible types of k-sums. Therefore, the cardinality of $Q(\mathrm{DB}, \theta)$ is

$$
\begin{aligned}
& |Q(\mathrm{DB},\theta)| \\
= \quad & \sum_{k=1}^{K}(N-1)^{k-1}\binom{K}{k} = \sum_{k=1}^{K}(N-1)^{k-1}\left[\binom{K-1}{k}+\binom{K-1}{k-1}\right] && (2.16)\\
& && (2.17)\\
\overset{(2.15)}{=} \quad & N^{K-1}+\sum_{k=1}^{K-1}(N-1)^{k-1}\binom{K-1}{k} && (2.18)\\
= \quad & N^{K-1}+\frac{1}{N-1}(N^{K-1}-1) && (2.19)
\end{aligned}
$$

■

We are now ready to prove that the achievable scheme is correct, private and achieves the capacity, in the following two lemmas.

Lemma 2.2. *The scheme described in Algorithm 1 is correct and the rate achieved is* $(1+1/N+\cdots+1/N^{K-1})^{-1}$*, which matches the capacity.*

Proof. The scheme is correct, i.e., all desired variables, U_θ, are decodable (with zero error probability), because either they appear with no interference (the first block) or they appear with interference q that is also downloaded separately from another server DB' so it can be subtracted. From Lemma 2.1 we know that there are N^{K-1} desired bit-variables in each $Q(\mathrm{DB},\theta)$. Note that desired variables always appear through $\texttt{new}(U_\theta)$, so they do not repeat across servers. Thus, the total number of desired bits that are retrieved is $N\times N^{K-1}=N^K$. We next compute the rate. The total number of desired bits retrieved is N^K, and the total number of downloaded bits from all servers is $N\times|Q(\mathrm{DB},\theta)|$ in every case. Therefore, the rate,

$$
\begin{aligned}
R \quad &= \quad \frac{N^K}{N\times|Q(\mathrm{DB},\theta)|} && (2.20)\\
&= \quad \frac{N^K}{N[N^{K-1}+\frac{1}{N-1}(N^{K-1}-1)]} = \left(\frac{N^{K-1}+\frac{1}{N-1}(N^{K-1}-1)}{N^{K-1}}\right)^{-1} && (2.21)
\end{aligned}
$$

$$= \left(1+\frac{\frac{1}{N-1}(N^{K-1}-1)}{N^{K-1}}\right)^{-1} = \left(1+\frac{\frac{1}{N}(1-\frac{1}{N^{K-1}})}{1-\frac{1}{N}}\right)^{-1} \tag{2.22}$$

$$= \left(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-1}}\right)^{-1} \tag{2.23}$$

■

Lemma 2.3. *The scheme described in Algorithm 1 is private.*

Proof. The intuition is quite straightforward. Regardless of θ, every realization of the query vector that fits the query structure is equally likely because of the uniformly random permutation Γ. To formalize this intuition, let us calculate the probability of an arbitrary query realization.

For any $\text{DB} \in [1:N], \theta \in [1:K]$, consider the ordered query vector representation $\vec{Q}(\text{DB},\theta)$. For each U_k, $k \in [1:K]$, denote the order in which these symbols appear in $\vec{Q}(\text{DB},\theta)$, as $\vec{u}_k(\text{DB},\theta) = [u_k(i_{k,\text{DB},\theta,1}), u_k(i_{k,\text{DB},\theta,2}), \cdots, u_k(i_{k,\text{DB},\theta,N^{K-1}})]$. Since the ordered query structure is already fixed regardless of θ and DB, and no variable occurs more than once, $\vec{Q}(\text{DB},\theta)$ is completely determined by $(\vec{u}_1(\text{DB},\theta), \vec{u}_2(\text{DB},\theta), \cdots, \vec{u}_K(\text{DB},\theta))$. Similarly, for each $k \in [1:K]$, denote an *arbitrary* N^{K-1}-tuple of bits from message W_k by $\vec{w}_k = [w_k(i'_{k_1}), w_k(i'_{k_2}), \cdots, w_k(i'_{k_{N^{K-1}}})]$. Recall that $u_k(i) = w_k(\gamma_k(i))$, $\forall k \in [1:K], i \in [1:N^K]$, and $\gamma_1, \gamma_2, \cdots, \gamma_K$ are uniform permutations chosen independently of each other and also independently of θ. Therefore, for all $(\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_K)$, we have

$$\begin{aligned}
&\text{Prob}\Big(\Gamma\left(\vec{u}_1(\text{DB},\theta), \vec{u}_2(\text{DB},\theta), \cdots, \vec{u}_K(\text{DB},\theta)\right)(\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_K)\Big) \\
&= \prod_{k=1}^{K} \text{Prob}\Big(\Gamma(\vec{u}_k(\text{DB},\theta)) = \vec{w}_k\Big) \qquad (2.24) \\
&= \left(\left(\frac{1}{N^K}\right)\left(\frac{1}{N^K-1}\right)\cdots\left(\frac{1}{N^K-N^{K-1}+1}\right)\right)^K \qquad (2.25)
\end{aligned}$$

which does not depend on θ. Thus, the distribution of $\vec{Q}(\text{DB},\theta)$ does not depend on θ. Since $Q_{\text{DB}}^{[\theta]}$ is a function of $\vec{Q}(\text{DB},\theta)$, $Q_{\text{DB}}^{[\theta]}$ must be independent of θ as well. Next, we show that privacy requirement (2.7) must be satisfied.

$$
\begin{aligned}
& I(\theta; Q_{\text{DB}}^{[\theta]}, A_{\text{DB}}^{[\theta]}, W_{1:K}) \\
= \; & I(\theta; Q_{\text{DB}}^{[\theta]}) + I(\theta; W_{1:K}|Q_{\text{DB}}^{[\theta]}) + I(\theta; A_{\text{DB}}^{[\theta]}|W_{1:K}, Q_{\text{DB}}^{[\theta]}) && (2.26) \\
= \; & 0 + 0 + 0 = 0 && (2.27)
\end{aligned}
$$

where $I(\theta; Q_{\text{DB}}^{[\theta]}) = 0$ because we have already proved that $Q_{\text{DB}}^{[\theta]}$ is independent of θ, $I(\theta; W_{1:K}|Q_{\text{DB}}^{[\theta]}) = 0$ because the desired message index and the query are generated privately by the user with no knowledge of the messages, and $I(\theta; A_{\text{DB}}^{[\theta]}|W_{1:K}, Q_{\text{DB}}^{[\theta]}) = 0$ because the answer is deterministic function of the query and messages. Therefore, all information available to server DB $(Q_{\text{DB}}^{[\theta]}, A_{\text{DB}}^{[\theta]}, W_1, \cdots, W_K)$ is independent of θ and the scheme is private. ■

Remark: From the proofs of privacy and correctness, note that the key is the query structure and the random mapping, Γ, of message bits to the query structure. In particular, no assumption is required on the statistics of the messages themselves. So the scheme works and a rate equal to C_o remains achievable even if the messages are not independent, although it may no longer be the capacity for this setting. For example, if $N = K = 2$ and the two messages are identical, $W_1 = W_2$, then clearly the capacity is 1, which is higher than $C_o = 2/3$. The independence of the messages is, however, needed for the converse.

We end this section with a lemma that highlights a curious property of our capacity achieving PIR scheme – that if the scheme is projected onto any subset of messages by eliminating the remaining messages, it also achieves the PIR capacity for that subset of messages.

Lemma 2.4. *Given a capacity achieving scheme generated by Algorithm 1 for K messages, if we set $\Delta, 1 \leq \Delta \leq K-1$ messages to be null, then the scheme achieves the capacity for the remaining $K - \Delta$ messages.*

Proof. We first prove that the scheme is correct after eliminating messages. This is easy to see as eliminating messages does not hurt (influence) the decoding procedure. Note that the eliminated messages can not include the desired one. We next prove that the scheme is also private. This is also easy to see as the permutations of the messages are independent, so that after eliminating messages, the bits of the remaining messages still distribute identically, no

matter which message is desired. We finally compute the rate and show that the scheme achieves the capacity for the remaining messages. Note that the total number of desired bits does not change, i.e., it is still N^K. The total number of downloaded equations decreases, as Δ messages are set to 0. In particular, the following number of equations becomes 0.

$$\begin{aligned} & N\sum_{k=1}^{\Delta}\binom{\Delta}{k}(N-1)^{k-1} \\ = \; & N\frac{1}{N-1}\left[\sum_{k=0}^{\Delta}\binom{\Delta}{k}(N-1)^{k}-1\right] && (2.28) \\ = \; & N\frac{1}{N-1}(N^{\Delta}-1) && (2.29) \end{aligned}$$

Subtracting above from $N|Q(\mathrm{DB},\theta)|$, we have the total number of downloaded equations. Therefore, the rate achieved is

$$\begin{aligned} R \; = \; & \frac{N^K}{N|Q(\mathrm{DB},\theta)|-N\frac{1}{N-1}(N^{\Delta}-1)} && (2.30) \\ = \; & \frac{N^K}{N[N^{K-1}+\frac{1}{N-1}(N^{K-1}-1)-\frac{1}{N-1}(N^{\Delta}-1)]} && (2.31) \\ = \; & \left(\frac{N^{K-1}+\frac{1}{N-1}(N^{K-1}-N^{\Delta})}{N^{K-1}}\right)^{-1}=\left(1+\frac{\frac{1}{N-1}(N^{K-1}-N^{\Delta})}{N^{K-1}}\right)^{-1} && (2.32) \\ = \; & \left(1+\frac{\frac{1}{N}(1-\frac{1}{N^{K-\Delta-1}})}{1-\frac{1}{N}}\right)^{-1}=\left(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-\Delta-1}}\right)^{-1} && (2.33) \end{aligned}$$

which matches the capacity.

■

2.4 Theorem 2.1: Converse

Note that the converse is proved for arbitrary L, i.e., we no longer assume that $L = N^K$. Let us start with two useful lemmas. Note that in the proofs, the relevant equations needed

to justify each step are specified by the equation numbers set on top of the (in)equality symbols.

Lemma 2.5. $I(W_{2:K}; Q_{1:N}^{[1]}, A_{1:N}^{[1]}|W_1) \le L(1/R - 1 + o(L))$.

Proof.

$$
\begin{aligned}
& I(W_{2:K}; Q_{1:N}^{[1]}, A_{1:N}^{[1]}|W_1) \\
\overset{(2.2)}{=} \quad & I(W_{2:K}; Q_{1:N}^{[1]}, A_{1:N}^{[1]}, W_1) && (2.34)\\
= \quad & I(W_{2:K}; Q_{1:N}^{[1]}, A_{1:N}^{[1]}) + I(W_{2:K}; W_1|Q_{1:N}^{[1]}, A_{1:N}^{[1]}) && (2.35)\\
\overset{(2.6)}{=} \quad & I(W_{2:K}; Q_{1:N}^{[1]}, A_{1:N}^{[1]}) + o(L)L && (2.36)\\
\overset{(2.4)}{=} \quad & I(W_{2:K}; A_{1:N}^{[1]}|Q_{1:N}^{[1]}) + o(L)L && (2.37)\\
= \quad & H(A_{1:N}^{[1]}|Q_{1:N}^{[1]}) - H(A_{1:N}^{[1]}|Q_{1:N}^{[1]}, W_{2:K}) + o(L)L && (2.38)\\
\le \quad & D - H(W_1, A_{1:N}^{[1]}|Q_{1:N}^{[1]}, W_{2:K}) + H(W_1|A_{1:N}^{[1]}, Q_{1:N}^{[1]}, W_{2:K}) + o(L)L && (2.39)\\
\overset{(2.8)(2.5)(2.6)}{=} \quad & L/R - H(W_1|Q_{1:N}^{[1]}, W_{2:K}) + o(L)L && (2.40)\\
\overset{(2.4)(2.2)(2.3)}{=} \quad & L/R - L + o(L)L = L(1/R - 1 + o(L)) && (2.41)
\end{aligned}
$$

■

Lemma 2.6. *For all* $k \in [2:K]$,

$$
I(W_{k:K}; Q_{1:N}^{[k-1]}, A_{1:N}^{[k-1]}|W_{1:k-1}) \ge \frac{1}{N} I(W_{k+1:K}; Q_{1:N}^{[k]}, A_{1:N}^{[k]}|W_{1:k}) + \frac{L(1-o(L))}{N}.
$$

Proof.

$$
\begin{aligned}
& N I(W_{k:K}; Q_{1:N}^{[k-1]}, A_{1:N}^{[k-1]}|W_{1:k-1}) \\
\ge \quad & \sum_{n=1}^{N} I(W_{k:K}; Q_n^{[k-1]}, A_n^{[k-1]}|W_{1:k-1}) && (2.42)\\
\overset{(2.7)}{=} \quad & \sum_{n=1}^{N} I(W_{k:K}; Q_n^{[k]}, A_n^{[k]}|W_{1:k-1}) && (2.43)\\
\ge \quad & \sum_{n=1}^{N} I(W_{k:K}; A_n^{[k]}|W_{1:k-1}, Q_n^{[k]}) && (2.44)\\
\overset{(2.5)}{=} \quad & \sum_{n=1}^{N} H(A_n^{[k]}|W_{1:k-1}, Q_n^{[k]}) && (2.45)
\end{aligned}
$$

$$
\begin{aligned}
&\geq \sum_{n=1}^{N} H(A_n^{[k]}|W_{1:k-1}, Q_{1:N}^{[k]}, A_{1:n-1}^{[k]}) && (2.46)\\
&\overset{(2.5)}{=} \sum_{n=1}^{N} I(W_{k:K}; A_n^{[k]}|W_{1:k-1}, Q_{1:N}^{[k]}, A_{1:n-1}^{[k]}) && (2.47)\\
&= I(W_{k:K}; A_{1:N}^{[k]}|W_{1:k-1}, Q_{1:N}^{[k]}) && (2.48)\\
&\overset{(2.4)(2.2)}{=} I(W_{k:K}; Q_{1:N}^{[k]}, A_{1:N}^{[k]}|W_{1:k-1}) && (2.49)\\
&\overset{(2.6)}{=} I(W_{k:K}; W_k, Q_{1:N}^{[k]}, A_{1:N}^{[k]}|W_{1:k-1}) - o(L)L && (2.50)\\
&= I(W_{k:K}; W_k|W_{1:k-1}) + I(W_{k:K}; Q_{1:N}^{[k]}, A_{1:N}^{[k]}|W_{1:k}) - o(L)L && (2.51)\\
&\overset{(2.2)(2.3)}{=} L + I(W_{k:K}; Q_{1:N}^{[k]}, A_{1:N}^{[k]}|W_{1:k}) - o(L)L && (2.52)\\
&= I(W_{k+1:K}; Q_{1:N}^{[k]}, A_{1:N}^{[k]}|W_{1:k}) + L(1 - o(L)) && (2.53)
\end{aligned}
$$

■

With these lemmas we are ready to prove the converse.

Proof of Converse of Theorem 2.1

Starting from $k = 2$ and applying Lemma 2.6 repeatedly for $k = 3$ to K,

$$
\begin{aligned}
& I(W_{2:K}; Q_{1:N}^{[1]}, A_{1:N}^{[1]}|W_1)\\
\geq\ & \frac{L}{N}(1 - o(L)) + \frac{1}{N} I(W_{3:K}; Q_{1:N}^{[2]}, A_{1:N}^{[2]}|W_1, W_2) && (2.54)\\
\geq\ & \frac{L}{N}(1 - o(L)) + \frac{1}{N}\Big[\frac{L}{N}(1 - o(L)) + \frac{1}{N} I(W_{4:K}; Q_{1:N}^{[3]}, A_{1:N}^{[3]}|W_{1:3})\Big] && (2.55)\\
=\ & L(1 - o(L))(\frac{1}{N} + \frac{1}{N^2}) + \frac{1}{N^2} I(W_{4:K}; Q_{1:N}^{[3]}, A_{1:N}^{[3]}|W_{1:3}) && (2.56)\\
\geq\ & \cdots && (2.57)\\
\geq\ & L(1 - o(L))(\frac{1}{N} + \cdots + \frac{1}{N^{K-2}}) + \frac{1}{N^{K-2}} I(W_K; Q_{1:N}^{[K-1]}, A_{1:N}^{[K-1]}|W_{1:K-1}) && (2.58)\\
\geq\ & L(1 - o(L))(\frac{1}{N} + \cdots + \frac{1}{N^{K-1}}) && (2.59)
\end{aligned}
$$

Combining Lemma 2.5 and (2.59), we have

$$
L(\frac{1}{R} - 1 + o(L)) \geq L(1 - o(L))(\frac{1}{N} + \cdots + \frac{1}{N^{K-1}}) \tag{2.60}
$$

Dividing both sides by L and letting L go to infinity gives us

	Prob. 1/2		Prob. 1/2	
	Want W_1	Want W_2	Want W_1	Want W_2
Server 1	$u_1, v_1, u_2 + v_2$	$u_1, v_1, u_2 + v_2$	$u_3, v_3, u_4 + v_4$	$u_3, v_3, u_4 + v_4$
Server 2	$u_4, v_2, u_3 + v_1$	$u_2, v_4, u_1 + v_3$	$u_2, v_4, u_1 + v_3$	$u_4, v_2, u_3 + v_1$

$$\frac{1}{R} - 1 \geq \left(\frac{1}{N} + \cdots + \frac{1}{N^{K-1}} \right) \Rightarrow R \leq \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}} \right)^{-1} \tag{2.61}$$

thus, completing the proof.

2.5 Discussion

In this section we share some interesting insights beyond the capacity characterization.

Upload Cost

To ensure privacy, we appealed to randomization arguments. To specify the randomly chosen query to the servers incurs an upload cost. For large messages the upload cost is negligible relative to the download cost, so it was ignored in this work. However, if the upload cost is a concern then it could be optimized as well. Random permutations of message bits are sufficient for privacy, but it is easy to see that the upload cost can be reduced by reducing the number of possibilities to be considered. For example, consider the $K = 2$ messages, $N = 2$ servers setting. We can group the bits, i.e., we can divide the 4 bits of each message into 2 groups, so that when we choose 2 bits, we only choose 2 bits from the same group. This reduces the choice to 1 out of 2 groups (rather than 2 out of 4 bits). Further, it may be possible to avoid random permutations among the chosen bits (group). For the same $K = 2$ messages and $N = 2$ servers example, we can fix the order within each group and the scheme becomes the following, shown at the top of this page. We denote the messages bits as $W_1 = \{u_1, u_2, u_3, u_4\}, W_2 = \{v_1, v_2, v_3, v_4\}$.

Note that regardless of which message is desired, the user is equally likely to request either $u_1, v_1, u_2 + v_2$ or $u_3, v_3, u_4 + v_4$ from Server 1, and either $u_2, v_4, u_1 + v_3$ or $u_4, v_2, u_3 + v_1$ from Server 2, so the scheme is private. However, each query is now limited to only 2 possibilities, thereby significantly reducing the upload cost. Also note that instead of storing all 8 bits that constitute the two messages, each server only needs to store 6 bits in this case, corresponding to the two possible queries that it may face.

Another interesting question in this context is to determine the *upload constrained capacity*. An information theoretic perspective is still useful. For example, since we are able to reduce the upload cost for $K = 2, N = 2$ to two possibilities, one might wonder if it is possible to reduce the upload cost of the $K = 3, N = 2$ setting to 3 possibilities without loss of capacity. Let us label the three possible downloads from Server 1 as f_1, f_2, f_3 and the three possible downloads from Server 2 as g_1, g_2, g_3. We wish to find out if the original PIR capacity of $4/7$ is still achievable under these upload constraints. As we show next, the capacity is strictly reduced. With uploads limited to choosing one out of only 3 possibilities, the upload constrained capacity of the $K = 3, N = 2$ setting is $1/2$ instead of $4/7$. Eliminating trivial degenerate cases, in this case there is no loss of generality in assuming that we can recover W_1 from any one of these three possibilities: $(f_1, g_1), (f_2, g_2), (f_3, g_3)$; we can recover W_2 from any one of these three possibilities: $(f_1, g_2), (f_2, g_3), (f_3, g_1)$; and we can recover W_3 from any one of these three possibilities: $(f_1, g_3), (f_2, g_1), (f_3, g_2)$. Then, for the optimal scheme we have

$$
\begin{aligned}
H(W_1) &= I(W_1; f_1, g_1) && (2.62)\\
&\leq 2H(A) - H(f_1, g_1|W_1) && (2.63)\\
\text{Similarly, } H(W_1) &\leq 2H(A) - H(f_2, g_2|W_1) && (2.64)\\
\text{Adding the two, } 2H(W_1) &\leq 4H(A) - H(f_1, g_1, f_2, g_2|W_1) && (2.65)\\
&\leq 4H(A) - H(W_1, W_2, W_3|W_1) && (2.66)\\
&\leq 4H(A) - H(W_2, W_3) && (2.67)\\
\Rightarrow C &= H(W_1)/2H(A) \leq 1/2 && (2.68)
\end{aligned}
$$

Here, $2H(A)$ is the total download. (2.66) follows because from f_1, g_1, f_2, g_2 we can recover all three messages. Thus, if the upload can only resolve one out of three possibilities for the query to each server, then the capacity of such a PIR scheme cannot be more than $1/2$, which is strictly smaller than the PIR capacity without upload constraints, $4/7$. In fact, the upload constrained capacity in this case is exactly $1/2$, as shown by the following achievable scheme which is interesting in its own right for how it fully exploits interference alignment. Suppose W_1, W_2, W_3 are symbols from a sufficiently large finite field (e.g., $\mathbb{F}_5$). Then the following construction works.

$$
\begin{aligned}
f_1 &= W_1 + 2W_2 + W_3 && (2.69)\\
f_2 &= W_1 + 4W_2 + 3W_3 && (2.70)\\
f_3 &= 3W_1 + 4W_2 + 6W_3 && (2.71)\\
g_1 &= W_1 + 4W_2 + 2W_3 && (2.72)\\
g_2 &= 3W_1 + 4W_2 + 3W_3 && (2.73)\\
g_3 &= 2W_1 + 4W_2 + 6W_3 && (2.74)
\end{aligned}
$$

It is easy to verify that W_1 can be recovered from any one of $(f_1, g_1), (f_2, g_2), (f_3, g_3)$; W_2 can be recovered from any one of $(f_1, g_2), (f_2, g_3), (f_3, g_1)$; and W_3 can be recovered from any one of $(f_1, g_3), (f_2, g_1), (f_3, g_2)$. The reason we can recover the desired message symbol from two equations, even though all three message symbols are involved in those two equations, is because of this special construction, which forces the undesired symbols to align into one dimension in every case. Thus, the upload constrained capacity for $K = 3, N = 2$ when the randomness is limited to choosing one out of 3 possibilities, is $1/2$. Answering this question for arbitrary K, N and arbitrary upload constraints is an interesting direction for future work.

Message Size

The information theoretic formulation of the PIR problem allows the sizes of messages to grow arbitrarily large. A natural question is this – how large do we need each message to be

for the optimal scheme. In our scheme, each message consists of N^K bits. However, even for our capacity achieving PIR scheme, the size of a message may be reduced. As an example, for the same $K = 2$ messages and $N = 2$ servers setting, the following PIR scheme works just as well (still achieves the same capacity) when each message is only made up of 2 bits: $W_1 = (u_1, u_2)$, $W_2 = (v_1, v_2)$.

	Prob. 1/2		Prob. 1/2	
	Want W_1	Want W_2	Want W_1	Want W_2
Server 1	u_1, v_2	u_1, v_2	u_2, v_1	u_2, v_1
Server 2	$u_2 + v_2$	$u_1 + v_1$	$u_1 + v_1$	$u_2 + v_2$

In general, the smallest message size needed to achieve the PIR capacity is characterized in [66] to be N^{K-1} bits per message. Further, building upon the capacity achieving scheme with smallest message size, the message size constrained PIR capacity is found in [66] to be $C_{\text{LPIR}} = \frac{L}{\lceil L/C_{\text{PIR}} \rceil}$.

Similarities between PIR and Blind Interference Alignment:

The idea of blind interference alignment was introduced in [43] to take advantage of the diversity of coherence intervals that may arise in a wireless network. For instance, different channels may experience different coherence times and coherence bandwidths. A diversity of coherence patterns can also be artificially induced by the switching of reconfigurable antennas in pre-determined patterns. As one of the simplest examples of BIA, consider a K user interference channel, where the desired channels have coherence time 1, i.e., they change after every channel use, while the cross channels (which carry interference) have coherence time 2, i.e., they remain unchanged over two channel uses. The transmitters are aware of the coherence times but otherwise have no knowledge of the channel coefficients. The BIA scheme operates over two consecutive channel uses. Over these two channel uses, each transmitter repeats its information symbol, and each receiver simply calculates the difference of its received signals. Since the transmitted symbols remain the same and the

cross channels do not change, the difference of received signals from the two channel uses eliminates all interference terms. However, because the desired channels change, the desired information symbols survive the difference at each receiver. Thus, one desired information symbol is successfully sent for each message over 2 channel uses, free from interference, achieving $\frac{1}{2}$ DoF per message. Remarkably, this is essentially identical to the first non-trivial scheme of PIR (see Section 3.1 of [24]).

The number of users in the BIA problem translates into the number of messages in the PIR problem. The received signals for user θ in BIA, translate into the answering strings when message W_θ is the desired message in the PIR problem. The channel vectors associated with user θ in the BIA problem translate into the query vectors for desired message W_θ in the PIR problem. The privacy requirement of the PIR scheme takes advantage of the observation that in BIA, over each channel use, the received signal at each receiver is statistically equivalent, because the transmitter does not know the channel values and the channel to each receiver has the same distribution. The most involved aspect of translating from BIA to PIR is that in BIA, the knowledge of the channel realizations across channel uses reveals the switching pattern, which in turn reveals the identity of the receiver. To remove this identifying feature of the BIA scheme, the channel uses are divided into subgroups such that the knowledge of the switching pattern within each group reveals nothing about the identity of the receiver. Each sub-group of channel uses is then associated with a different server. Since the servers are not allowed to communicate with each other, and each sub-group of queries (channel uses) reveals nothing about the message (user), the resulting scheme guarantees privacy. Finally, the symmetric degrees of freedom (DoF) value per user in BIA is the ratio between the number of desired message symbols and the number of channel uses (received signal equations), and the rate R in PIR is the ratio between the number of symbols of the desired message and the total number of equations in all answering strings. In this way, the DoF value achieved with BIA translates into the rate of the corresponding PIR protocol, i.e., $R = \text{DoF}$. We summarize these connections in the following table.

Table 2.1: PIR and BIA

PIR	BIA
Message	Receiver
Queries	Channel Coefficients
Answers	Received Signals
Rate	DoF

Recognizing this connection between PIR and BIA directly leads to capacity achieving PIR schemes for $K = 2$ messages, and arbitrary number of servers N, as in [65], by translating from known optimal BIA schemes. However, for $K > 2$, the PIR framework generalizes the BIA framework. This is because the coherence patterns that are assumed to exist in BIA are typically motivated by the distinct coherence times, coherence bandwidths, or antenna switching patterns that are feasible in wireless settings. However, since PIR is not bound by wireless phenomena, it allows for arbitrary coherence patterns, including many possibilities that would be considered infeasible in wireless settings. Even the simple scheme of BIA for the K user interference channel presented earlier, was originally noted in BIA [43] merely as a matter of curiosity rather than having any physical significance. As such, while our initial insights into PIR came by viewing it as a special case of existing BIA schemes, the new capacity achieving PIR schemes introduced in this work go well beyond existing results in BIA, by allowing arbitrary coherence patterns.

Chapter 3

Capacity of Robust PIR with Colluding Servers

In the previous chapter we considered the basic model of PIR, where a user wants to retrieve a desired message from a set of N distributed servers, each of which stores all K messages, without revealing anything (in the information theoretic sense) about which message is being retrieved to any individual server. There are several interesting extensions of PIR that explore its limitations under additional constraints. These include extensions where up to T of the N servers may collude [12, 8] (T-private PIR, or TPIR in short); where some of the servers may not respond [16] (Robust PIR); where both the privacy of the user and the servers must be protected [36] (Symmetric PIR); where only one server holds all the messages and all other servers hold independent information [35]; where retrieval operations are unsynchronized [30]; and where beyond communications, computation is also a concern [15]. There is also much recent work in the distributed storage setting [59, 22, 31, 69] (the servers form a distributed storage system) where the main focus is on how the coding of the storage system works jointly with PIR.

In this chapter, we mainly consider TPIR in the Shannon theoretic setting, where we have an arbitrary number of messages (K), arbitrary number of servers (N), each server stores

all the messages, the messages are allowed to be arbitrarily large, and the privacy of the desired message index must be guaranteed even if any T of the N servers collude. The main contribution of this chapter is to show that the information theoretic capacity of TPIR is

$$C_{\text{TPIR}} = \left(1 + T/N + T^2/N^2 + \cdots + T^{K-1}/N^{K-1}\right)^{-1} \tag{3.1}$$

We further consider the extension to *robust* TPIR, where we have $M \geq N$ servers, out of which any $M - N$ servers may not respond, so that with answers from any N servers, we need to ensure both privacy and correctness. In this context, the contribution of this chapter is to show that the information theoretic capacity of robust TPIR remains the same as that of TPIR, i.e., there is no capacity cost from not knowing in advance *which* N servers will respond.

3.1 Problem Statement

3.1.1 TPIR

Consider K independent messages $W_1, \cdots, W_K$ of size L bits each.

$$H(W_1, \cdots, W_K) = H(W_1) + \cdots + H(W_K), \tag{3.2}$$
$$H(W_1) = \cdots = H(W_K) = L. \tag{3.3}$$

There are N servers. Each server stores all the messages $W_1, \cdots, W_K$. A user wants to retrieve $W_\theta, \theta \in [1 : K]$ subject to T-privacy, i.e., without revealing anything about the message identity, θ, to any colluding susbset of up to T out of the N servers. θ is uniformly distributed over $[1 : K]$.

Suppose $\theta = k$. To retrieve W_k privately, the user generates N queries $Q_1^{[k]}, \cdots, Q_N^{[k]}$, where the superscript denotes the desired message index. Since the queries are generated with

no knowledge of the realizations of the messages, the queries must be independent of the messages,

$$I(W_1, \cdots, W_K; Q_1^{[k]}, \cdots, Q_N^{[k]}) = 0. \tag{3.4}$$

The user sends query $Q_n^{[k]}$ to the n-th server, $\forall n \in [1:N]$. Upon receiving $Q_n^{[k]}$, the n-th server generates an answering string $A_n^{[k]}$, which is a deterministic function of $Q_n^{[k]}$ and the data stored (i.e., all messages $W_1, \cdots, W_K$),

$$H(A_n^{[k]} | Q_n^{[k]}, W_1, \cdots, W_K) = 0. \tag{3.5}$$

Each server returns to the user its answer $A_n^{[k]}$. From all answers $A_1^{[k]}, \cdots, A_N^{[k]}$, the user can decode the desired message W_k,

$$\text{[Correctness]}\ H(W_k | A_1^{[k]}, \cdots, A_N^{[k]}, Q_1^{[k]}, \cdots, Q_N^{[k]}) = 0. \tag{3.6}$$

To satisfy the privacy constraint that any T colluding servers learn nothing about the desired message index θ information theoretically, information available to any T servers (queries, answers and the stored messages) must be independent of θ. Let $\mathcal{T}$ be a subset of $[1:N]$ and its cardinality be denoted by $|\mathcal{T}|$. $Q_{\mathcal{T}}^{[\theta]}$ represents the subset $\{Q_n^{[\theta]}, n \in \mathcal{T}\}$. $A_{\mathcal{T}}^{[\theta]}$ is defined similarly. To satisfy the T-privacy requirement we must have

$$\text{[Privacy]} \quad I(Q_{\mathcal{T}}^{[\theta]}, A_{\mathcal{T}}^{[\theta]}, W_1, \cdots, W_K; \theta) = 0, \forall \mathcal{T} \subset [1:N], |\mathcal{T}| = T. \tag{3.7}$$

To underscore that any set of T or fewer answering strings is independent of the desired message index, we may suppress the superscript and write $A_{\mathcal{T}}$ directly instead of $A_{\mathcal{T}}^{[k]}$, and express the elements of such a set as A_n instead of $A_n^{[k]}$.

The metric that we study in this chapter is the PIR rate[1], which characterizes how many bits of desired information are retrieved per downloaded bit. Note that the PIR rate is the reciprocal of download cost. The rate R of a PIR scheme is defined as follows.

$$R \triangleq L/D \tag{3.8}$$

where D is the expected value of the total number of bits downloaded by the user from all the servers. The capacity, C_{TPIR}, is the supremum of R over all PIR schemes.

3.1.2 Robust TPIR

The robust TPIR problem is defined similar to the TPIR problem. The only difference is that instead of N servers, we have $M \geq N$ servers, and the correctness condition needs to be satisfied when the user collects *any* N out of the M answering strings.

3.2 Main Result: Capacity of Robust TPIR

The following theorem states the main result.

Theorem 3.1. *For TPIR with K messages and N servers, the capacity is*

$$C_{TPIR} = \left(1 + T/N + T^2/N^2 + \cdots + T^{K-1}/N^{K-1}\right)^{-1}. \tag{3.9}$$

The capacity of PIR with T colluding servers generalizes the case without T-privacy constraints, where $T = 1$ (refer to Chapter 2). The capacity is a strictly decreasing function of T. When $T = N$, the capacity is $1/K$, meaning that the user has to download all K messages to be private, as in this case, the colluding servers are as strong as the user. Similar to the $T = 1$ case, the capacity is strictly deceasing in the number of messages, K, and strictly increasing

[1]In the Shannon theoretic formulation where the message size is allowed to grow, the upload cost (the length of the query strings) is negligible in the capacity formulation because when we double the message size, we can reuse the same query functions for both parts of the messages such that the upload cost does not scale with the message size. A more detailed treatment may be found in Proposition 4.1.1 of [24].

in the number of servers, N. When the number of messages approaches infinity, the capacity approaches $1 - T/N$, and when the number of servers approaches infinity (T remains constant), the capacity approaches 1. Finally, note that since the download cost is the reciprocal of the rate, the capacity characterization in Theorem 3.1 equivalently characterizes the optimal download cost per message bit for TPIR as $\left(1 + T/N + T^2/N^2 + \cdots + T^{K-1}/N^{K-1}\right)$ bits. Note that when $N \neq T$, the capacity expression can be equivalently expressed as $(1 - \frac{T}{N})/(1 - (\frac{T}{N})^K)$.

The capacity-achieving scheme that we construct for TPIR, generalizes easily to incorporate robustness constraints. As a consequence, we are also able to characterize the capacity of robust TPIR (RTPIR). This result is stated in the following theorem.

Theorem 3.2. *The capacity of RTPIR is*

$$C_{RTPIR} = \left(1 + T/N + T^2/N^2 + \cdots + T^{K-1}/N^{K-1}\right)^{-1}. \tag{3.10}$$

Since the capacity expressions are the same, we note that there is no capacity penalty from not knowing in advance which N servers will respond. Even though this uncertainty increases as M increases, capacity is not a function of M. However, we note that the communication complexity of our capacity achieving scheme does increase with M.

3.3 Proof of Theorem 3.1: Achievability

There are two key aspects of the achievable scheme – 1) the query structure, and 2) the specialization of the query structure to ensure T-privacy and correctness. While the query structure is different from the $T = 1$ setting in Chapter 2, it draws upon the iterative application of the same three principles that were identified in Chapter 2. These principles are listed below.

(1) *Enforcing Symmetry Across Servers*

(2) *Enforcing Message Symmetry within the Query to Each Server*

(3) *Exploiting Previously Acquired Side Information of Undesired Messages to Retrieve New Desired Information*

The specialization of the structure to ensure T-privacy and correctness is another novel element of the achievable scheme. To illustrate how these ideas work together in an iterative fashion, we will present a few simple examples corresponding to small values of K, N and T, and then generalize it to arbitrary K, N and T. Let us begin with a lemma.

Lemma 3.1. *Let $S_1, S_2, \cdots, S_K \in \mathcal{F}_q^{\alpha\times\alpha}$ be K random matrices, drawn independently and uniformly from all $\alpha \times \alpha$ full-rank matrices over $\mathcal{F}_q$. Let $G_1, G_2, \cdots, G_K \in \mathcal{F}_q^{\beta\times\beta}$ be K invertible square matrices of dimension $\beta\times\beta$ over $\mathcal{F}_q$. Let $\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_K \in \mathbb{N}^{\beta\times 1}$ be K index vectors, each containing β distinct indices from $[1:\alpha]$. Then*

$$(G_1S_1[\mathcal{I}_1,:], G_2S_2[\mathcal{I}_2,:], \cdots, G_KS_K[\mathcal{I}_K,:]) \sim (S_1[(1:\beta),:], S_2[(1:\beta),:], \cdots, S_K[(1:\beta),:]) \tag{3.11}$$

where $S_i[\mathcal{I}_i,:], i \in [1:K]$ are $\beta\times\alpha$ matrices comprised of the rows of S_i with indices in $\mathcal{I}_i$.

Proof: We wish to prove that the left hand side of (3.11) is identically distributed (recall that the notation $X \sim Y$ means that X and Y are identically distributed) to the right hand side of (3.11). Because the rank of a matrix does not depend on the ordering of the rows, we have

$$(S_1[\mathcal{I}_1,:], S_2[\mathcal{I}_2,:], \cdots, S_K[\mathcal{I}_K,:]) \sim (S_1[(1:\beta),:], S_2[(1:\beta),:], \cdots, S_K[(1:\beta),:])$$

Since S_i are picked uniformly from all full-rank matrices, conditioned on any feasible value of the remaining rows $S_i[(\beta+1:\alpha),:]$, the first β rows $S_i[(1:\beta),:]$ are uniformly distributed over all possibilities that preserve full-rank for S_i. Now note that the mapping from $S_i[(1:\beta),:]$ to $G_iS_i[(1:\beta),:]$ is bijective, and $S_i[(1:\beta),:]$ spans the same row space as $G_iS_i[(1:\beta),:]$,

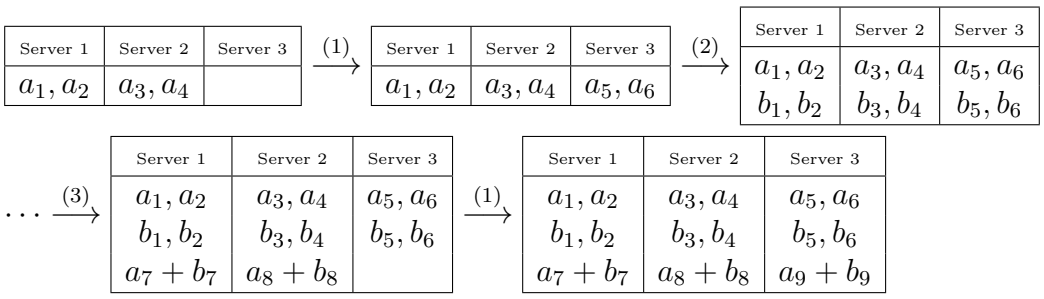
i.e., replacing $S_i[(1:\beta),:]$ with $G_i S_i[(1:\beta),:]$, preserves S_i as a full-rank matrix. Therefore, conditioned on any feasible $S_i[(\beta+1:\alpha),:]$, the set of feasible values of $S_i[(1:\beta),:]$ is the same as the set of feasible $G_i S_i[(1:\beta),:]$ values. Therefore, $G_i S_i[(1:\beta),:]$ is also uniformly distributed over the same set. Finally, since the S_i are chosen independently, the statement of Lemma 3.1 follows. ■

3.3.1 Example: $K = 2$, $N = 3$, $T = 2$

The capacity for this setting, is $C = \left(1 + \frac{2}{3}\right)^{-1} = \frac{3}{5}$.

3.3.1.1 Query Structure

We begin by constructing a query structure, which will then be specialized to achieve correctness and privacy. Without loss of generality, let $[a_k]$ denote the symbols of the desired message, and $[b_k]$ the symbols of the undesired message.



We start by requesting the first $T^{K-1} = 2$ symbols from each of the first $T = 2$ servers: a_1, a_2 from Server 1, and a_3, a_4 from Server 2. Applying server symmetry, we simultaneously request a_5, a_6 from Server 3. Next, we enforce message symmetry, by including queries for $b_1, \cdots, b_6$ as the counterparts for $a_1, \cdots, a_6$. Now consider the first $T = 2$ servers, i.e., Server 1 and Server 2, which can potentially collude with each other. Unknown to these servers the user has acquired two symbols of external side information, b_5, b_6, comprised of undesired message symbols received from Server 3. Splitting the two symbols of external side information among Server 1 and Server 2 allows the user one symbol of side information

for each of Server 1 and Server 2 that it can exploit to retrieve new desired information symbols. In our construction of the query structure, we will assign new labels (subscripts) to the external side-information exploited within each server, e.g., b_7 for Server 1 and b_8 for Server 2, with the understanding that eventually when the dependencies within the structure are specialized, b_7, b_8 will turn out to be functions of previously acquired side-information. Using its assigned side information, each Server acquires a new symbol of desired message, so that Server 1 requests $a_7 + b_7$ and Server 2 requests $a_8 + b_8$. Finally, enforcing symmetry across servers, Server 3 requests $a_9 + b_9$. At this point, the construction is symmetric across servers, the query to any server is symmetric in itself across messages, and the amount of side information exploited within any T colluding servers equals the amount of side information available external to those T servers. So the skeleton of the query structure is complete.

Note that if Server 1 and Server 2 collude, then the external side information is b_5, b_6, so we would like the side-information that is exploited by Server 1 and Server 2, i.e., b_7, b_8 to be functions of the external side information that is available, i.e., b_5, b_6. However, since *any* $T = 2$ servers can collude, it is also possible that Server 1 and Server 3 collude instead, in which case we would like b_7, b_9 to be functions of side information that is external to Server 1 and Server 3, i.e., b_3, b_4. Similarly, if Server 2 and Server 3 collude, then we would like b_8, b_9 to be functions of b_1, b_2. How to achieve such dependencies in a manner that preserves privacy and ensures correctness is the remaining challenge. Intuitively, the key is to make b_7, b_8, b_9 depend on *all* side-information $b_1, b_2, \cdots, b_6$ in a generic sense. In other words, we will achieve the desired functional dependencies by viewing $b_1, b_2, \cdots, b_9$ as the outputs of a $(9, 6)$ MDS code, so that any 3 of these b_k are functions of the remaining 6. The details of this specialization are described next.

3.3.1.2 Specialization to Ensure Correctness and Privacy

Let each message consist of $N^K = 9$ symbols from a sufficiently large[2] finite field $\mathbb{F}_q$. The messages W_1, $W_2 \in \mathbb{F}_q^{9\times 1}$ are then represented as 9×1 vectors over $\mathbb{F}_q$. Let $S_1, S_2 \in \mathbb{F}_q^{9\times 9}$ represent random matrices chosen privately by the user, independently and uniformly from all 9×9 full-rank matrices over $\mathbb{F}_q$. Without loss of generality, let us assume that W_1 is the desired message. Define the 9×1 vectors $a_{[1:9]} \in \mathbb{F}_q^{9\times 1}$ and $b_{[1:9]} \in \mathbb{F}_q^{9\times 1}$, as follows

$$a_{[1:9]} = S_1 W_1 \tag{3.12}$$
$$b_{[1:9]} = \text{MDS}_{9\times 6} S_2[(1:6), :] W_2 \tag{3.13}$$

where $S_2[(1:6), :]$ is a 6×9 matrix comprised of the first 6 rows of S_2. $\text{MDS}_{9\times 6}$ is the generator matrix of a $(9, 6)$ MDS code (e.g., a Reed Solomon code). The generator matrix does not need to be random, i.e., it may be globally known. Note that because of the MDS property, any 6 rows of $\text{MDS}_{9\times 6}$ form a 6×6 invertible matrix. Therefore, from any 6 elements of $b_{[1:9]}$, all 9 elements of $b_{[1:9]}$ can be recovered. For example, from $b_1, b_2, \cdots, b_6$, one can recover b_7, b_8, b_9. The queries from each server are constructed according to the structure described earlier.

Server 1	Server 2	Server 3
a_1, a_2	a_3, a_4	a_5, a_6
b_1, b_2	b_3, b_4	b_5, b_6
$a_7 + b_7$	$a_8 + b_8$	$a_9 + b_9$

(3.14)

Correctness is easy to see, because the user recovers $b_{[1:6]}$ explicitly, from which it can recover all $b_{[1:9]}$, thereby allowing it to recover all of $a_{[1:9]}$. Let us see why privacy holds. The queries for any $T = 2$ colluding servers are comprised of 6 variables from $a_{[1:9]}$ and 6 variables from $b_{[1:9]}$. Let the indices of these variables be denoted by the 6×1 vectors $\mathcal{I}_a, \mathcal{I}_b \in \mathbb{N}^{6\times 1}$, respectively, so that,

[2]The requirements on the size of the field have to do with the existence of MDS codes that are used in the construction. In this case $q \geq N^K$ is sufficient.

$$(a_{\mathcal{I}_a}, b_{\mathcal{I}_b}) = (S_1[\mathcal{I}_a, :]W_1, \text{MDS}_{9\times 6}[\mathcal{I}_b, :]S_2[(1:6), :]W_2) \quad (3.15)$$
$$\sim (S_1[(1:6), :]W_1, S_2[(1:6), :]W_2) \quad (3.16)$$

where (3.16) follows from Lemma 3.1 because $\text{MDS}_{9\times 6}[\mathcal{I}_b, :]$ is an invertible 6×6 matrix. Therefore, the random map from W_1 to $a_{\mathcal{I}_a}$ variables is i.i.d. as the random map from W_2 to $b_{\mathcal{I}_b}$, and privacy is guaranteed. Note that since 9 desired symbols are recovered from a total of 15 downloaded symbols, the rate achieved by this scheme is $9/15 = 3/5$, which matches the capacity for this setting. While this specialization suffices for our purpose (it achieves capacity), we note that further simplifications of the scheme are possible, which allow it to operate over smaller fields and with lower upload cost. Such an example is provided in the discussion section of this chapter.

3.3.2 Example: $K = 3$, $N = 3$, $T = 2$

The capacity for this setting, is $C = \left(1 + \frac{2}{3} + (\frac{2}{3})^2\right)^{-1} = \frac{9}{19}$.

3.3.2.1 Query Structure

The query structure is constructed as follows.

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	

$\xrightarrow{(1)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$

$\xrightarrow{(2)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$

$\xrightarrow{(3)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$
$a_{13} + b_{13}$	$a_{15} + b_{15}$	
$a_{14} + b_{14}$	$a_{16} + b_{16}$	
$a_{17} + c_{13}$	$a_{19} + c_{15}$	
$a_{18} + c_{14}$	$a_{20} + c_{16}$	

$\xrightarrow{(1)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$
$a_{13} + b_{13}$	$a_{15} + b_{15}$	$a_{21} + b_{17}$
$a_{14} + b_{14}$	$a_{16} + b_{16}$	$a_{22} + b_{18}$
$a_{17} + c_{13}$	$a_{19} + c_{15}$	$a_{23} + c_{17}$
$a_{18} + c_{14}$	$a_{20} + c_{16}$	$a_{24} + c_{18}$

$\xrightarrow{(2)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$
$a_{13} + b_{13}$	$a_{15} + b_{15}$	$a_{21} + b_{17}$
$a_{14} + b_{14}$	$a_{16} + b_{16}$	$a_{22} + b_{18}$
$a_{17} + c_{13}$	$a_{19} + c_{15}$	$a_{23} + c_{17}$
$a_{18} + c_{14}$	$a_{20} + c_{16}$	$a_{24} + c_{18}$
$b_{19} + c_{19}$	$b_{21} + c_{21}$	$b_{23} + c_{23}$
$b_{20} + c_{20}$	$b_{22} + c_{22}$	$b_{24} + c_{24}$

$\xrightarrow{(3)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$
$a_{13} + b_{13}$	$a_{15} + b_{15}$	$a_{21} + b_{17}$
$a_{14} + b_{14}$	$a_{16} + b_{16}$	$a_{22} + b_{18}$
$a_{17} + c_{13}$	$a_{19} + c_{15}$	$a_{23} + c_{17}$
$a_{18} + c_{14}$	$a_{20} + c_{16}$	$a_{24} + c_{18}$
$b_{19} + c_{19}$	$b_{21} + c_{21}$	$b_{23} + c_{23}$
$b_{20} + c_{20}$	$b_{22} + c_{22}$	$b_{24} + c_{24}$
$a_{25} + b_{25} + c_{25}$	$a_{26} + b_{26} + c_{26}$	

$\xrightarrow{(1)}$

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$
$a_{13} + b_{13}$	$a_{15} + b_{15}$	$a_{21} + b_{17}$
$a_{14} + b_{14}$	$a_{16} + b_{16}$	$a_{22} + b_{18}$
$a_{17} + c_{13}$	$a_{19} + c_{15}$	$a_{23} + c_{17}$
$a_{18} + c_{14}$	$a_{20} + c_{16}$	$a_{24} + c_{18}$
$b_{19} + c_{19}$	$b_{21} + c_{21}$	$b_{23} + c_{23}$
$b_{20} + c_{20}$	$b_{22} + c_{22}$	$b_{24} + c_{24}$
$a_{25} + b_{25} + c_{25}$	$a_{26} + b_{26} + c_{26}$	$a_{27} + b_{27} + c_{27}$

Starting with $T^{K-1} = 4$ symbols each requested from the first $T = 2$ servers, we proceed through iterative steps (1) and (2) to enforce symmetries across servers and messages. In step (3) we consider the first $T = 2$ servers together (Server 1 and Server 2) and account for the external side information, which in this case contains 4 symbols from $[b_k]$ and 4 symbols from $[c_k]$. Distributed evenly among Server 1 and Server 2, this allows a budget of 2 symbols of side information from $[b_k]$ and 2 symbols from $[c_k]$ per server to be exploited to recover new symbols of desired information. Proceeding again through steps (1) and (2) to enforce symmetries across servers and messages, we end up with new downloads that contain only undesired information symbols, which can now be used to download new desired information symbols. Once again, we consider Server 1 and Server 2 together, and account for the new external side information, $b_{23}+c_{23}, b_{24}+c_{24}$. Thus the external side information is comprised of two symbols, each of which is a sum of the form $b_k + c_k$. Dividing the side information evenly among servers Server 1 and Server 2, each is assigned one side-information symbol of the form $b_k + c_k$ with new labels. Thus, $a_{25} + b_{25} + c_{25}$ is added to the query from Server 1, and $a_{26} + b_{26} + c_{26}$ is added to the query from Server 2. Finally, applying symmetry across servers, we include $a_{27}+b_{27}+c_{27}$ to the query from Server 3. At this point, all symmetries are

satisfied, all external and exploited side-information amounts are balanced, and therefore, the query structure is complete.

3.3.2.2 Specialization

Let each message consist of $N^K = 27$ symbols from a sufficiently large finite field $\mathbb{F}_q$. The messages $W_1, W_2, W_3 \in \mathbb{F}_q^{27\times 1}$ are then represented as 27×1 vectors over $\mathbb{F}_q$. Let $S_1, S_2, S_3 \in \mathbb{F}_q^{27\times 27}$ represent random matrices chosen privately by the user, independently and uniformly from all 27×27 full-rank matrices over $\mathbb{F}_q$. Without loss of generality, let us assume that W_1 is the desired message. Define 27×1 vectors $a_{[1:27]}, b_{[1:27]}, c_{[1:27]} \in \mathbb{F}_q^{27\times 1}$, as follows

$$
\begin{aligned}
a_{[1:27]} &= S_1 W_1 && (3.17)\\
b_{[1:18]} &= \mathrm{MDS}_{18\times 12} S_2[(1:12),:] W_2 && (3.18)\\
c_{[1:18]} &= \mathrm{MDS}_{18\times 12} S_3[(1:12),:] W_3 && (3.19)\\
b_{[19:27]} &= \mathrm{MDS}_{9\times 6} S_2[(13:18),:] W_2 && (3.20)\\
c_{[19:27]} &= \mathrm{MDS}_{9\times 6} S_3[(13:18),:] W_3 && (3.21)
\end{aligned}
$$

where $S_2[(1:18),:]$ is a 18×27 matrix comprised of the first 18 rows of S_2. $\mathrm{MDS}_{18\times 12}$ is the generator matrix of a $(18, 12)$ MDS code, and $\mathrm{MDS}_{9\times 6}$ is the generator matrix of a $(9, 6)$ MDS code. In particular, note that the *same* generator matrix is used in (3.18) and (3.19). Similarly, the same generator matrix is used in (3.20) and (3.21). This is important because it allows us to write

$$
b_{[19:27]} + c_{[19:27]} = \mathrm{MDS}_{9\times 6} \left(S_2[(13:18),:] W_2 + S_3[(13:18),:] W_3 \right) \quad (3.22)
$$

so that all 9 elements of the vector $b_{[19:27]} + c_{[19:27]}$ can be recovered from any 6 of its elements, e.g., from $b_{[19:24]} + c_{[19:24]}$ one can also recover $b_{25} + c_{25}, b_{26} + c_{26}, b_{27} + c_{27}$. This observation is the key to understanding the role of interference alignment in this construction. The effective number of *resolvable* undesired symbols is minimized due to interference alignment. For example, b_{19} and c_{19} are always aligned together into one symbol $b_{19} + c_{19}$ in all the

downloaded equations. The two are unresolvable from each other and act as effectively one undesired symbol in the downloaded equations, thus reducing the effective number of undesired symbols, so that the same number of downloaded equations can be used to retrieve a greater number of desired symbols. Note also that desired symbols are always resolvable. These values are plugged into the query structure derived previously.

Server 1	Server 2	Server 3
a_1, a_2, a_3, a_4	a_5, a_6, a_7, a_8	$a_9, a_{10}, a_{11}, a_{12}$
b_1, b_2, b_3, b_4	b_5, b_6, b_7, b_8	$b_9, b_{10}, b_{11}, b_{12}$
c_1, c_2, c_3, c_4	c_5, c_6, c_7, c_8	$c_9, c_{10}, c_{11}, c_{12}$
$a_{13} + b_{13}$	$a_{15} + b_{15}$	$a_{21} + b_{17}$
$a_{14} + b_{14}$	$a_{16} + b_{16}$	$a_{22} + b_{18}$
$a_{17} + c_{13}$	$a_{19} + c_{15}$	$a_{23} + c_{17}$
$a_{18} + c_{14}$	$a_{20} + c_{16}$	$a_{24} + c_{18}$
$b_{19} + c_{19}$	$b_{21} + c_{21}$	$b_{23} + c_{23}$
$b_{20} + c_{20}$	$b_{22} + c_{22}$	$b_{24} + c_{24}$
$a_{25} + b_{25} + c_{25}$	$a_{26} + b_{26} + c_{26}$	$a_{27} + b_{27} + c_{27}$

Correctness is straightforward. Let us see why T-privacy holds. The queries for any $T = 2$ colluding servers are comprised of 18 variables from $a_{[1:27]}$, 12 variables from $b_{[1:18]}$, 6 variables from $b_{[19:27]}$, 12 variables from $c_{[1:18]}$ and 6 variables from $c_{[19:27]}$. Let the indices of these variables be denoted by the vectors $\mathcal{I}_a \in \mathbb{N}^{18\times 1}, \mathcal{I}_{b,12} \in \mathbb{N}^{12\times 1}, \mathcal{I}_{b,6} \in \mathbb{N}^{6\times 1}, \mathcal{I}_{c,12} \in \mathbb{N}^{12\times 1}$ and $\mathcal{I}_{c,6} \in \mathbb{N}^{6\times 1}$, respectively, so that,

$$
\begin{aligned}
a_{\mathcal{I}_a} &= S_1[\mathcal{I}_a, :]W_1 && (3.23)\\
b_{\mathcal{I}_{b,12}} &= \mathrm{MDS}_{18\times 12}[\mathcal{I}_{b,12}, :]S_2[(1:12), :]W_2 && (3.24)\\
b_{\mathcal{I}_{b,6}} &= \mathrm{MDS}_{9\times 6}[\mathcal{I}_{b,6}, :]S_2[(13:18), :]W_2 && (3.25)\\
c_{\mathcal{I}_{c,12}} &= \mathrm{MDS}_{18\times 12}[\mathcal{I}_{c,12}, :]S_3[(1:12), :]W_3 && (3.26)\\
c_{\mathcal{I}_{c,6}} &= \mathrm{MDS}_{9\times 6}[\mathcal{I}_{c,6}, :]S_3[(13:18), :]W_3 && (3.27)
\end{aligned}
$$

From Lemma 3.1, we have

$$
(a_{\mathcal{I}_a}, (b_{\mathcal{I}_{b,12}};\ \ b_{\mathcal{I}_{b,6}}), (c_{\mathcal{I}_{c,12}};\ \ c_{\mathcal{I}_{c,6}})) \ \sim\ (S_1[(1:18), :]W_1, S_2[(1:18), :]W_2, S_3[(1:18), :]W_3) \quad (3.28)
$$

Thus privacy is guaranteed. Finally, note that since 27 desired symbols are recovered from a total of 57 downloaded symbols, the rate achieved by this scheme is $27/57 = 9/19$, which matches the capacity for this setting.

3.3.3 Arbitrary K, Arbitrary N, Arbitrary T

3.3.3.1 Query Structure

For arbitrary K, N, T, we follow the same iterative procedure, briefly summarized below[3].

- Step 1: Initialization. Download T^{K-1} desired symbols each from the first T servers.
- Step 2: Invoke symmetry across servers to determine corresponding downloads from Server $T+1$ to Server N.
- Step 3: Invoke symmetry of messages to determine additional downloaded equations (comprised only of undesired symbols) from each server.
- Step 4: Consider the first T servers together. Divide the new external side information generated in the previous step evenly among the first T servers to determine the side-information budget per server. For each side information symbol allocated to a server create an additional query of the same form as the assigned side information (with new labels) combined with a new desired symbol.
- Step 5: Go back to Step 2 and run Step 2 to Step 4 a total of $(K-1)$ times.

3.3.3.2 Specialization

We now map the message symbols to the symbols in the query structure. Let each message consist of N^K symbols from a sufficiently large finite field $\mathbb{F}_q$. The messages $W_1, \cdots, W_K \in$

[3]To be more specific, server symmetry refers to the property that each server downloads a equal number of instances for each type of sums, and message symmetry refers to the property that within each server, the symbols from each message are equivalent up to permutations. A more detailed treatment can be found in [66]. We initialize by downloading T^{K-1} symbols such that in Step 4 when we divide side information symbols, each server always obtains an integer number of side information symbols.

$\mathbb{F}_q^{N^K \times 1}$ are represented as $N^K \times 1$ vectors over $\mathbb{F}_q$. Let $S_1, \cdots, S_K \in \mathbb{F}_q^{N^K \times N^K}$ represent random matrices chosen privately by the user, independently and uniformly from all $N^K \times N^K$ full-rank matrices over $\mathbb{F}_q$. Suppose W_l, $l \in [1:K]$, is the desired message.

Consider any undesired message index $k \in [1:K]/\{l\}$, and all distinct $\Delta = 2^{K-2}$ subsets of $[1:K]$ that contain k and do not contain l. Assign distinct labels to each subset, e.g., $\mathcal{K}_1, \mathcal{K}_2, \cdots \mathcal{K}_\Delta$. For each $k \in [1:K]/\{l\}$, define the vector

$$\begin{bmatrix} x^{[k]}_{\mathcal{K}_1} \\ \hline x^{[k]}_{\mathcal{K}_1 \cup \{l\}} \\ \hline x^{[k]}_{\mathcal{K}_2} \\ \hline x^{[k]}_{\mathcal{K}_2 \cup \{l\}} \\ \hline \vdots \\ \hline x^{[k]}_{\mathcal{K}_\Delta} \\ \hline x^{[k]}_{\mathcal{K}_\Delta \cup \{l\}} \end{bmatrix} = \left[\begin{array}{cccc} \mathrm{MDS}_{\frac{N}{T}\alpha_1 \times \alpha_1} & 0 & 0 & 0 \\ \hline 0 & \mathrm{MDS}_{\frac{N}{T}\alpha_2 \times \alpha_2} & 0 & 0 \\ \hline 0 & \cdots & \ddots & 0 \\ \hline 0 & 0 & 0 & \mathrm{MDS}_{\frac{N}{T}\alpha_\Delta \times \alpha_\Delta} \end{array}\right] S_k[(1 : TN^{K-1}), :] W_k$$

where $\alpha_i, i \in [1:\Delta]$ is defined as[4] $N(N-T)^{|\mathcal{K}_i|-1}T^{K-|\mathcal{K}_i|}$, each $x^{[k]}_{\mathcal{K}_i}$ is an $\alpha_i \times 1$ vector, and each $x^{[k]}_{\mathcal{K}_i \cup \{l\}}$ is an $(\frac{N-T}{T})\alpha_i \times 1$ vector over $\mathbb{F}_q$.

Now consider the desired message index l, and all distinct $\delta = 2^{K-1}$ subsets of $[1:K]$ that contain l. Assign distinct labels to each subset, e.g., $\mathcal{L}_1, \mathcal{L}_2, \cdots \mathcal{L}_\delta$. Define the vector

$$\begin{bmatrix} x^{[l]}_{\mathcal{L}_1} \\ x^{[l]}_{\mathcal{L}_2} \\ \vdots \\ x^{[l]}_{\mathcal{L}_\delta} \end{bmatrix} = S_l W_l$$

where the length of $x^{[l]}_{\mathcal{L}_i}, i \in [1:\delta]$ is $N(N-T)^{|\mathcal{L}_i|-1}T^{K-|\mathcal{L}_i|}$.

For each non-empty subset $\mathcal{K} \subset [1:K]$ generate the query vector

[4]The choice of α_i is to ensure both correctness and privacy. More specifically, it guarantees that over each layer, exposed undesired symbols suffice to decode the undesired symbols that interfere the desired symbols and the number of symbols seen by any colluding set matches the MDS code dimension such that they appear uniformly random. The proof appears later. For example, consider the setting in Section 3.3.2, where the desired message index $l = 1$. Consider the undesired message index $k = 2$. Here $\Delta = 2$, i.e., $\mathcal{K}_1 = \{2\}, \mathcal{K}_2 = \{2,3\}$, $\alpha_1 = 12$ and $\alpha_2 = 6$.

$$\sum_{k\in\mathcal{K}} x_{\mathcal{K}}^{[k]} \tag{3.29}$$

Distribute the elements of the query vector evenly among the N servers. This completes the specialized construction of the queries.

The construction has K layers. Over the j-th layer, from each server, we download $(N-T)^{j-1}T^{K-j}\binom{K}{j}$ equations[5] that are comprised of sums of j symbols, out of which $(N-T)^{j-1}T^{K-j}\binom{K-1}{j-1}$ involve desired data symbols. Our construction ensures that the interference $x_{\mathcal{K}_i\cup\{l\}}^{[k]}, k\neq l, i\in[1:\Delta]$ in the $(|\mathcal{K}_i|+1)$-th layer can be recovered from the corresponding symbols $x_{\mathcal{K}_i}^{[k]}$ in the $|\mathcal{K}_i|$-th layer. Therefore correctness is guaranteed.

Let us see why privacy holds. The queries for any T colluding servers are comprised of TN^{K-1} variables from each $x^{[k]}, k\in[1:K]$. In particular, $\forall k\neq l$, the variables from $x^{[k]}$ consist of α_i variables out of $\frac{N}{T}\alpha_i$ variables $x_{\mathcal{K}_i}^{[k]}, x_{\mathcal{K}_i\cup\{l\}}^{[k]}$, for each set $\mathcal{K}_i, i\in[1:\Delta]$. Note that these α_i variables are generated by the generator matrix of an $(\frac{N}{T}\alpha_i,\alpha_i)$ MDS code, so that they have full rank. Let the indices of the appeared variables be denoted by the vectors $\mathcal{I}_{x^{[k]}}\in\mathbb{N}^{TN^{K-1}\times 1}, \forall k\in[1:K]$. From Lemma 3.1, we have

$$x_{\mathcal{I}_{x^{[k]}}}^{[k]} \quad\sim\quad S_k[(1:TN^{K-1}),:]W_k \tag{3.30}$$

which in turn implies that $S_k[(1:TN^{K-1}),:]$ are independent and identically distributed. Thus privacy is guaranteed.

Finally, we compute the ratio of the number of desired symbols to the number of total downloaded symbols,

$$R \;=\; \frac{T^{K-1}\binom{K-1}{0}+(N-T)T^{K-2}\binom{K-1}{1}+(N-T)^2T^{K-3}\binom{K-1}{2}+\cdots+(N-T)^{K-1}\binom{K-1}{K-1}}{T^{K-1}\binom{K}{1}+(N-T)T^{K-2}\binom{K}{2}+(N-T)^2T^{K-3}\binom{K}{3}+\cdots+(N-T)^{K-1}\binom{K}{K}} \tag{3.31}$$

[5]Over the j-th layer, the downloads are in the form of sums of j symbols, each from one distinct message. The term $(N-T)^{j-1}T^{K-j}$ comes from the side information exploitation step (Step 4) and can be verified recursively. A more detailed analysis in similar flavor can be found in [66].

$$= \frac{N}{N} \frac{N^{K-1}}{\frac{1}{N-T}\left[(N-T)T^{K-1}\binom{K}{1} + (N-T)^2T^{K-2}\binom{K}{2} + \cdots + (N-T)^K\binom{K}{K}\right]} \tag{3.32}$$

$$= \frac{\frac{1}{N}N^K}{\frac{1}{N-T}\left(N^K - T^K\right)} = \frac{1-\frac{T}{N}}{1-\frac{T^K}{N^K}} \tag{3.33}$$

$$= \left(1 + \frac{T}{N} + \frac{T^2}{N^2} + \cdots + \frac{T^{K-1}}{N^{K-1}}\right)^{-1} \tag{3.34}$$

Thus, the PIR rate achieved by the scheme always matches the capacity.

Remark: When we set $T = 1$, Theorem 3.1 recovers the PIR capacity result in Chapter 2. The two schemes achieve the same rate (capacity achieving), but the two differ in that although the query structures are the same, the specialization here uses MDS codes over a large field while the specialization in Chapter 2 uses permutations over message bits.

3.4 Proof of Theorem 3.1: Converse

For compact notation, let us define

$$\mathcal{Q} \triangleq \{Q_n^{[k]} : k \in [1:K], n \in [1:N]\} \tag{3.35}$$

$$A_{\mathcal{I}}^{[k]} \triangleq \{A_n^{[k]} : n \in \mathcal{I}\} \tag{3.36}$$

$$\mathcal{H}_T \triangleq \frac{1}{\binom{N}{T}} \sum_{\mathcal{T}:|\mathcal{T}|=T} \frac{H(A_{\mathcal{T}}|\mathcal{Q})}{T}, \mathcal{T} \subset [1:N] \tag{3.37}$$

We first state Han's inequality (Theorem 17.6.1 in [25]), which will be used later and is described here for the sake of completeness.

Theorem 3.3. *(Han's inequality, Theorem 17.6.1 in [25])*

$$\mathcal{H}_T \geq \frac{H(A_1^{[k]}, A_2^{[k]}, \cdots, A_N^{[k]}|\mathcal{Q})}{N} \tag{3.38}$$

We next proceed to the converse proof. The proof of outer bound for Theorem 3.1 is based on an induction argument. To set up the induction, we will prove the outer bound for $K = 1$ (the trivial case) for arbitrary N, T, and then proceed to the case of arbitrary K.

$K = 1$ Message, N Servers

$$
\begin{aligned}
L = H(W_1) &= H(W_1|\mathcal{Q}) && (3.39)\\
&= I(A_1^{[1]}, A_2^{[1]}, \cdots, A_N^{[1]}; W_1|\mathcal{Q}) && (3.40)\\
&= H(A_1^{[1]}, A_2^{[1]}, \cdots, A_N^{[1]}|\mathcal{Q}) && (3.41)\\
&\leq N\mathcal{H}_T && (3.42)\\
&\leq \sum_{n=1}^{N} H(A_n|\mathcal{Q}) && (3.43)\\
\Rightarrow R &= \frac{L}{D} \leq \frac{L}{\sum_{n=1}^{N} H(A_n|\mathcal{Q})} \leq 1 && (3.44)
\end{aligned}
$$

where (3.42) follows from Han's inequality, and (3.43) is due to the property that dropping conditioning does not reduce entropy.

$K \geq 2$ Messages, N Servers

Consider $\mathcal{T} \subset [1:N]$ with cardinality $|\mathcal{T}| = T$. From $A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}, \mathcal{Q}$, we can decode all K messages $W_1, \cdots, W_K$.

$$
\begin{aligned}
& KL = H(W_1, \cdots, W_K|\mathcal{Q}) && (3.45)\\
= \; & I(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}; W_1, \cdots, W_K|\mathcal{Q}) && (3.46)\\
= \; & H(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}|\mathcal{Q}) && (3.47)\\
= \; & H(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}|\mathcal{Q}) + H(A_{\overline{\mathcal{T}}}^{[2]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}|A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}, \mathcal{Q}) && (3.48)\\
\leq \; & N\mathcal{H}_T + H(A_{\overline{\mathcal{T}}}^{[2]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}|A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}, W_1, \mathcal{Q}) && (3.49)\\
\leq \; & N\mathcal{H}_T + H(A_{\overline{\mathcal{T}}}^{[2]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}|A_{\mathcal{T}}, W_1, \mathcal{Q}) && (3.50)\\
= \; & N\mathcal{H}_T + H(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[2]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}|W_1, \mathcal{Q}) - H(A_{\mathcal{T}}|W_1, \mathcal{Q}) && (3.51)\\
= \; & N\mathcal{H}_T + H(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[2]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}, W_2, \cdots, W_K|W_1, \mathcal{Q}) - H(A_{\mathcal{T}}|W_1, \mathcal{Q}) && (3.52)\\
= \; & N\mathcal{H}_T + (K-1)L - H(A_{\mathcal{T}}|W_1, \mathcal{Q}) && (3.53)
\end{aligned}
$$

where (3.49) is due to the fact that W_1 is a function of $(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[1]}, \mathcal{Q})$. (3.52) follows from the fact that $W_2, \cdots, W_K$ is a function of $(A_{\mathcal{T}}, A_{\overline{\mathcal{T}}}^{[2]}, \cdots, A_{\overline{\mathcal{T}}}^{[K]}, \mathcal{Q})$. In (3.53), the second term is due to the fact that the answers are deterministic functions of the messages and queries, and the messages are independent.

Consider (3.53) for all subsets of $[1:N]$ that have exactly T elements and average over all such subsets. We have

$$N\mathcal{H}_T \geq L + \frac{1}{\binom{N}{T}} \sum_{\mathcal{T}:|\mathcal{T}|=T} H(A_{\mathcal{T}}|W_1, \mathcal{Q}) \tag{3.54}$$

To proceed, we note that for the last term of (3.54), conditioning on W_1, the setting reduces to a PIR problem with $K-1$ messages and N servers. Thus, (3.54) sets up an induction argument, which claims that for the K messages setting,

$$N\mathcal{H}_T \geq L\left(1 + \frac{T}{N} + \cdots + \frac{T^{K-1}}{N^{K-1}}\right) \tag{3.55}$$

We have proved the basis cases of $K = 1$ in (3.42). Suppose now the bound (3.55) holds for $K-1$. Then plugging in (3.54), we have that the bound (3.55) holds for K. Since both the basis and the inductive step have been performed, by mathematical induction, we have proved that (3.55) holds for all K. The desired outer bound follows as

$$R = \frac{L}{D} \leq \frac{L}{\sum_{n=1}^{N} H(A_n|\mathcal{Q})} \leq \frac{L}{N\mathcal{H}_T} \leq \left(1 + \frac{T}{N} + \cdots + \frac{T^{K-1}}{N^{K-1}}\right)^{-1} \tag{3.56}$$

Thus, the proof of the outer bound is complete.

Remark: For the converse proof, we could follow from similar lines as in Chapter 2. Here we adopt a slightly different approach to give more insights. From a high level view, the proof in Chapter 2 is an interference channel based approach, while the proof here is a multiple access channel based appraoch.

3.5 Proof of Theorem 3.2

Clearly the capacity of robust TPIR cannot be larger than the capacity of TPIR. Therefore, we only need to prove that the capacity of TPIR can be achieved in the robust PIR setting.

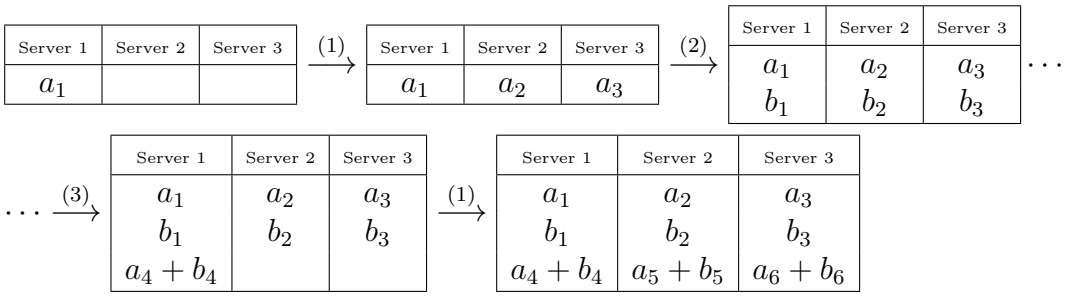
To this end, we build upon the scheme presented in Section 3.3. Before proceeding to the general proof, we first give an example to illustrate the key idea in a simpler setting.

3.5.1 Example: $K = 2$, $M = 3$, $N = 2$, $T = 1$

The capacity for this setting, is $C = \left(1 + \frac{1}{2}\right)^{-1} = \frac{2}{3}$.

3.5.1.1 Query Structure

We first construct the query structure, following the same 3 iterative principles for TPIR. Without loss of generality, let $[a_k]$ denote the symbols of the desired message, and $[b_k]$ the symbols of the undesired message.



We start by requesting the first $T^{K-1} = 1$ symbol from the first $T = 1$ server, i.e., a_1 from Server 1. Applying server symmetry, we simultaneously request a_2 from Server 2 and a_3 from Server 3. Next, we enforce message symmetry, by including queries for b_1, b_2, b_3 as the counterparts for a_1, a_2, a_3. Note that only $N = 2$ servers may respond. As a result, from the perspective of any individual server, we have only one symbol of external side information (from the other surviving server). We then exploit this side information symbol to retrieve a new desired symbol, i.e., we download $a_4 + b_4$ from Server 1, $a_5 + b_5$ from Server 2 and $a_6 + b_6$ from Server 3. The construction is complete.

We want to ensure that no matter which 2 servers respond, we can gather enough desired symbols to decode the desired message and privacy is preserved to each individual server. These are guaranteed by the following specialization.

3.5.1.2 Specialization to Ensure Correctness and Privacy

Let each message consist of $N^K = 4$ symbols from a sufficiently large field. The messages W_1, $W_2 \in \mathbb{F}_q^{4\times 1}$ are then represented as 4×1 vectors over $\mathbb{F}_q$. Let $S_1, S_2 \in \mathbb{F}_q^{4\times 4}$ represent random matrices chosen privately by the user, independently and uniformly from all 4×4 full-rank matrices over $\mathbb{F}_q$. Without loss of generality, let us assume that W_1 is the desired message. Define the 6×1 vectors $a_{[1:6]} \in \mathbb{F}_q^{6\times 1}$ and $b_{[1:6]} \in \mathbb{F}_q^{6\times 1}$, as follows

$$a_{[1:6]} = \text{MDS}_{6\times 4} S_1 W_1 \tag{3.57}$$
$$b_{[1:6]} = \text{MDS}_{6\times 2} S_2[(1:2),:] W_2 \tag{3.58}$$

where $S_2[(1:2),:]$ is a 2×4 matrix comprised of the first 2 rows of S_2. $\text{MDS}_{6\times 4}/\text{MDS}_{6\times 2}$ is the generator matrix of a $(6,4)/(6,2)$ MDS code.

Server 1	Server 3	Server 3
a_1	a_2	a_3
b_1	b_2	b_3
$a_4 + b_4$	$a_5 + b_5$	$a_6 + b_6$

(3.59)

Correctness is easy to see, because after receiving answers from any $N = 2$ servers, the user recovers all $b_{[1:6]}$ (refer to (3.58)). Then the user subtracts out $b_{[1:6]}$ and then recovers 4 symbols in $a_{[1:6]}$, from which all $a_{[1:6]}$ are recovered (refer to (3.57)). The query for any individual server is comprised of 2 variables from $a_{[1:6]}$ and 2 variables from $b_{[1:6]}$. Let the indices of these variables be denoted by the 2×1 vectors $\mathcal{I}_a, \mathcal{I}_b \in \mathbb{N}^{2\times 1}$, respectively, so that,

$$(a_{\mathcal{I}_a}, b_{\mathcal{I}_b}) = (\text{MDS}_{6\times 4}[\mathcal{I}_a,:] S_1 W_1, \text{MDS}_{6\times 2}[\mathcal{I}_b,:] S_2[(1:2),:] W_2) \tag{3.60}$$
$$\sim (S_1[(1:2),:] W_1, S_2[(1:2),:] W_2) \tag{3.61}$$

where (3.61) follows from Lemma 3.1. Therefore, the random map from W_1 to $a_{\mathcal{I}_a}$ variables is i.i.d. as the random map from W_2 to $b_{\mathcal{I}_b}$, and privacy is guaranteed. Note that since 4 desired symbols are recovered from a total of 6 downloaded symbols (from $N = 2$ responding

servers), the rate achieved by this scheme is $4/6 = 2/3$, which matches the capacity for this setting.

3.5.2 Arbitrary K, N, M, T

As before, let each message consist of N^K symbols from a sufficiently large finite field $\mathbb{F}_q$. The messages $W_1, \cdots, W_K \in \mathbb{F}_q^{N^K \times 1}$ are represented as $N^K \times 1$ vectors over $\mathbb{F}_q$. Let $S_1, \cdots, S_K \in \mathbb{F}_q^{N^K \times N^K}$ represent random matrices chosen privately by the user, independently and uniformly from all $N^K \times N^K$ full-rank matrices over $\mathbb{F}_q$. Suppose W_l, $l \in [1:K]$, is the desired message.

Consider any undesired message index $k \in [1:K]/\{l\}$, and all distinct $\Delta = 2^{K-2}$ subsets of $[1:K]$ that contain k and do not contain l. Assign distinct labels to each subset, e.g., $\mathcal{K}_1, \mathcal{K}_2, \cdots \mathcal{K}_\Delta$. For each $k \in [1:K]/\{l\}$, define the vector

$$\begin{bmatrix} x^{[k]}_{\mathcal{K}_1} \\ \hline x^{[k]}_{\mathcal{K}_1 \cup \{l\}} \\ \hline x^{[k]}_{\mathcal{K}_2} \\ \hline x^{[k]}_{\mathcal{K}_2 \cup \{l\}} \\ \hline \vdots \\ \hline x^{[k]}_{\mathcal{K}_\Delta} \\ \hline x^{[k]}_{\mathcal{K}_\Delta \cup \{l\}} \end{bmatrix} = \left[\begin{array}{cccc} \text{MDS}_{\frac{M}{T}\alpha_1 \times \alpha_1} & 0 & 0 & 0 \\ \hline 0 & \text{MDS}_{\frac{M}{T}\alpha_2 \times \alpha_2} & 0 & 0 \\ \hline 0 & \cdots & \ddots & 0 \\ \hline 0 & 0 & 0 & \text{MDS}_{\frac{M}{T}\alpha_\Delta \times \alpha_\Delta} \end{array}\right] S_k[(1 : TN^{K-1}), :] W_k$$

where $\alpha_i, i \in [1:\Delta]$ is defined as $N(N-T)^{|\mathcal{K}_i|-1}T^{K-|\mathcal{K}_i|}$, each $x^{[k]}_{\mathcal{K}_i}$ is an $\frac{M}{N}\alpha_i \times 1$ vector, and each $x^{[k]}_{\mathcal{K}_i \cup \{l\}}$ is an $\frac{M}{N}(\frac{N-T}{T})\alpha_i \times 1$ vector over $\mathbb{F}_q$.

Now consider the desired message index l, and all distinct $\delta = 2^{K-1}$ subsets of $[1:K]$ that contain l. Assign distinct labels to each subset, e.g., $\mathcal{L}_1, \mathcal{L}_2, \cdots \mathcal{L}_\delta$. Define the vector

$$\begin{bmatrix} x^{[l]}_{\mathcal{L}_1} \\ x^{[l]}_{\mathcal{L}_2} \\ \vdots \\ x^{[l]}_{\mathcal{L}_\delta} \end{bmatrix} = \text{MDS}_{\frac{M}{N}N^K \times N^K} S_l W_l$$

where the length of $x^{[l]}_{\mathcal{L}_i}, i \in [1:\delta]$ is $M(N-T)^{|\mathcal{L}_i|-1}T^{K-|\mathcal{L}_i|}$.

For each non-empty subset $\mathcal{K} \subset [1:K]$ generate the query vector

$$\sum_{k \in \mathcal{K}} x_{\mathcal{K}}^{[k]} \tag{3.62}$$

Distribute the elements of the query vector evenly among the M servers. This completes the construction of the queries.

Suppose the user collects answering strings from any N servers. For each set $\mathcal{K}_i$, from N servers, we download α_i symbols from $x_{\mathcal{K}_i}^{[k]}, k \neq l, i \in [1:\Delta]$, from which we can recover the interference $x_{\mathcal{K}_i \cup \{l\}}^{[k]}$, as they are generated by the generator matrix of an $(\frac{M}{T}\alpha_i, \alpha_i)$ MDS code. After subtracting out all the interference, we are left with N^K desired symbols, from which we can recover the desired message, as the symbols are generated by the generator matrix of an $(\frac{M}{N}N^K, N^K)$ MDS code. Therefore correctness is guaranteed.

Let us see why privacy holds. The queries for any T colluding servers are comprised of TN^{K-1} variables from each $x^{[k]}, k \in [1:K]$. When $k = l$, the TN^{K-1} desired symbols are generated by the generator matrix of an $(\frac{M}{N}N^K, N^K)$ MDS code such that these symbols have full rank. For each $k \neq l$, the TN^{K-1}variables from $x^{[k]}$ consist of α_i variables out of $\frac{M}{T}\alpha_i$ variables $x_{\mathcal{K}_i}^{[k]}, x_{\mathcal{K}_i \cup \{l\}}^{[k]}$, for each set $\mathcal{K}_i, i \in [1:\Delta]$. Note that these α_i variables are generated by the generator matrix of an $(\frac{M}{T}\alpha_i, \alpha_i)$ MDS code, so that they have full rank. Let the indices of the appeared variables be denoted by the vectors $\mathcal{I}_{x^{[k]}} \in \mathbb{N}^{TN^{K-1} \times 1}, \forall k \in [1:K]$. From Lemma 3.1, we have

$$x_{\mathcal{I}_{x^{[k]}}}^{[k]} \quad \sim \quad S_k[(1:TN^{K-1}), :]W_k \tag{3.63}$$

which in turn implies that $S_k[(1:TN^{K-1}), :]$ are independent and identically distributed. Thus privacy is guaranteed. Finally, the rate achieved is the same as that achieved in the setting without the robustness constraint. This completes the proof.

3.6 Discussion

We characterize the capacity of robust TPIR with arbitrary number of messages, arbitrary number of (responding) servers, and arbitrary privacy level. Let us summarize with a few observations. First, while in this chapter we adopt the zero error framework, we note that our converse extends in a straightforward manner to the ϵ-error framework as well, where the probability of error is only required to approach zero as the message size approaches infinity. Therefore, for robust TPIR, the ϵ-error capacity is the same as the zero error capacity. Second, recall that the capacity achieving scheme for PIR in Chapter 2 had a remarkable feature that if some of the messages were eliminated and the scheme projected onto a subset of messages, it remained capacity optimal for that subset of messages. The same phenomenon is observed for our achievable scheme for robust TPIR. On the other hand, an important point of distinction of the previous achievable scheme in Chapter 2 from the achievable scheme in this chapter is that the former directly uses each available side information symbol individually, whereas here we need MDS coded side information (uncoded side information symbols do not suffice). This is because of the T-privacy constraint which simultaneously creates multiple perspectives of external side-information depending upon which subset of servers decides to collude. Third, we note that in this chapter we require perfect privacy (refer to (3.7), $I(Q_{\mathcal{T}}^{[\theta]}, A_{\mathcal{T}}^{[\theta]}, W_1, \cdots, W_K; \theta) = 0$), and similar to the ϵ-error correctness constraint, we may require δ-privacy, where the leakage on the desired message index vanishes as the message size grows. That is, we replace the privacy constraint (3.7) by $I(Q_{\mathcal{T}}^{[\theta]}, A_{\mathcal{T}}^{[\theta]}, W_1, \cdots, W_K; \theta) = \delta$, where δ approaches zero as the message size approaches infinity. It turns out that the capacity under δ-privacy is the same as the capacity under perfect privacy. The converse proof extends by noting that the δ-privacy constraint implies $H(Q_{\mathcal{T}}^{[k_1]}, A_{\mathcal{T}}^{[k_1]}, W_1, \cdots, W_K) - H(Q_{\mathcal{T}}^{[k_2]}, A_{\mathcal{T}}^{[k_2]}, W_1, \cdots, W_K) = \delta'$ for any two message indices $k_1, k_2 \in [1 : K]$, where δ' vanishes with the message size and all other steps follow in the same manner.

Finally, we note that since we focus only on download cost, upload cost is not optimized in this work. However, even with T-privacy, significant optimizations of upload cost are possible through refinements of our achievable scheme. For example, the symbols may be grouped in a manner that randomizations are needed only within smaller groups, which may reduce the number of possible queries, and the size of the field of operations significantly. For example, consider the achievable scheme for $K = 2, N = 3, T = 2$ that was presented in Section 3.3.1, where each message is comprised of 9 symbols. We will operate over $\mathbb{F}_2$. Suppose we divide the 9 bits into 3 groups of 3 bits each, and label the groups so that A_1 represents the first three bits of W_1, A_2 the next three and A_3 represents the last three bits from W_1. Similarly, let B_1, B_2, B_3 represent three groups of three bits each from W_2. Now, for any group of 3 bits, say $X = (x_1, x_2, x_3)$, let $X(1), X(2), X(3)$ represent three randomly chosen linearly independent elements from the set $\{x_1, x_2, x_3, x_1 + x_2, x_1 + x_3, x_2 + x_3, x_1 + x_2 + x_3\}$, i.e., selected uniformly from among the choices that do not sum to zero in $\mathbb{F}_2$. This essentially means that $X(1), X(2)$ may be freely chosen as any two distinct elements of the set and then $X(3)$ is chosen uniformly from the 4 elements that are not $X(1), X(2)$ or $X(1) + X(2)$. The queries are constructed as follows.

Server 2	Server 2	Server 3
$A_1(1), A_2(1)$	$A_2(2), A_3(2)$	$A_3(3), A_1(3)$
$B_1(1), B_2(1)$	$B_2(2), B_3(2)$	$B_3(1+2), B_1(1+2)$
$A_3(1) + B_3(1)$	$A_1(2) + B_1(2)$	$A_2(3) + B_2(1+2)$

where we use the notation $X(1+2) = X(1) + X(2)$ for brevity. Note that for the undesired symbols B, we used the $(2, 3)$ MDS code $(B(1), B(2)) \longrightarrow (B(1), B(2), B(1+2))$ within each group. Due to the grouping of symbols the upload cost is significantly reduced. Moreover, because of the grouping we are able to operate over a smaller field. Whereas the original scheme presented in Section 3.3.1 uses $(6, 9)$ MDS codes which do not exist over $\mathbb{F}_2$, the refined example presented above uses only a $(2, 3)$ MDS code which does exist over $\mathbb{F}_2$. As

illustrated by this example, optimizations of upload costs as well as symbol size remain interesting avenues for future work.

Chapter 4

PIR from MDS Coded Data with Colluding Servers

The focus of this chapter is on a recent conjecture by Freij-Hollanti, Gnilke, Hollanti and Karpuk (FGHK conjecture, in short) in [33] which offers a capacity expression for a generalized form of PIR, called MDS-TPIR. Beyond the number of databases, N and the number of messages, K, MDS-TPIR involves two additional parameters: K_c and T, which generalize the storage and privacy constraints, respectively. Instead of replication, each message is encoded through an (N, K_c) MDS storage code, so that the information stored at any K_c servers is exactly enough to recover all K messages. Privacy must be preserved not just from each individual server, but from any colluding set of up to T servers. MDS-TPIR is a generalization of PIR, because setting both $T = 1$ and $K_c = 1$ reduces MDS-TPIR to the original PIR problem for which the capacity is already known (see Chapter 2).

The capacity of MDS-TPIR is known only at the degenerate extremes – when either T or K_c takes the value 1 or N. If either T or K_c is equal to N then by analogy to the single server setting it follows immediately that the user must download all messages, i.e., the capacity is $1/K$. If $K_c = 1$ or $T = 1$, then the problem specializes to TPIR, and MDS-PIR, respectively. The capacity of TPIR ($K_c = 1$) was shown in Chapter 3 to be

$$C_{\text{TPIR}} = \left(1+\frac{T}{N}+\frac{T^2}{N^2}+\cdots+\frac{T^{K-1}}{N^{K-1}}\right)^{-1} \tag{4.1}$$

The capacity of MDS-PIR ($T=1$) was characterized by Banawan and Ulukus in [6], as

$$C_{\text{MDS-PIR}} = \left(1+\frac{K_c}{N}+\frac{K_c^2}{N^2}+\cdots+\frac{K_c^{K-1}}{N^{K-1}}\right)^{-1} \tag{4.2}$$

It is notable that K_c and T play similar roles in the two capacity expressions.

The capacity achieving scheme of Banawan and Ulukus [6] improved upon a scheme proposed earlier by Tajeddine and Rouayheb in [69]. Tajeddine and Rouayheb also proposed an achievable scheme for MDS-TPIR for the $T=2$ setting. The scheme was generalized by Freij-Hollanti et al. [33] to the (K,N,T,K_c) setting, $T+K_c \leq N$, where it achieves the rate $1-\frac{T+K_c-1}{N}$. Remarkably, the rate achieved by this scheme does not depend on the number of messages, K. In support of the plausible asymptotic ($K\to\infty$) optimality of their scheme, and based on the intuition from existing capacity expressions for PIR, MDS-PIR and TPIR, Freij-Hollanti et al. conjecture that if $T+K_c \leq N$, then the capacity of MDS-TPIR is given by the following expression.

FGHK Conjecture [33]:

$$C^{\text{conj}}_{\text{MDS-TPIR}} = \left(1+\frac{T+K_c-1}{N}+\cdots+\frac{(T+K_c-1)^{K-1}}{N^{K-1}}\right)^{-1} \tag{4.3}$$

The conjecture is appealing for its generality and elegance as it captures all four parameters, K,N,T,K_c in a compact form. T and K_c appear as interchangeable terms, and the capacity expression appears to be a natural extension of the capacity expressions for TPIR and MDS-PIR. Indeed, the conjectured capacity recovers the known capacity of TPIR if we set $K_c=1$ and that of MDS-PIR if we set $T=1$. However, in all non-degenerate cases where $T,K_c \notin \{1,N\}$, the capacity of MDS-TPIR, and therefore the validity of the conjecture is unknown. In fact, in all these cases the problem is open on *both* sides, i.e., the conjectured

capacity expression is neither known to be achievable, nor known to be an outer bound. The lack of any non-trivial outer bounds for MDS-TPIR is also recently highlighted in [49]. This intriguing combination of plausibility, uncertainty and generality of the FGHK conjecture motivates our work.

As the main outcome of this chapter, we disprove the FGHK conjecture. For our counterexample, we consider the setting $(K, N, T, K_c) = (2, 4, 2, 2)$ where the data is stored using the $(2, 4)$ MDS code $(x, y) \to (x, y, x + y, x + 2y)$. The conjectured capacity for this setting is $4/7$. We show that the rate $3/5 > 4/7$ is achievable, thus disproving the conjecture. As a converse argument, we show that no (scalar or vector) linear PIR scheme can achieve a rate higher than $3/5$ for this MDS storage code subject to $T = 2$ privacy.

4.1 Problem Statement

Consider[1] K independent messages $W_1, \cdots, W_K \in \mathbb{F}_p^{L\times 1}$, each represented as an $L \times 1$ vector comprised of L i.i.d. uniform symbols from a finite field $\mathbb{F}_p$ for a prime p. In p-ary units,

$$\begin{aligned} H(W_1) &= \cdots = H(W_K) = L && (4.4)\\ H(W_1, \cdots, W_K) &= H(W_1) + \cdots + H(W_K) && (4.5) \end{aligned}$$

There are N servers. The n-th server stores $(W_{1n}, W_{2n}, \cdots, W_{Kn})$, where $W_{kn} \in \mathbb{F}^{\frac{L}{K_c}\times 1}$ represents L/K_c symbols from $W_k, k \in [1 : K]$.

$$H(W_{kn}|W_k) = 0,\ H(W_{kn}) = L/K_c \tag{4.6}$$

We require the storage system to satisfy the MDS property, i.e., from the information stored in any K_c servers, we can recover each message, i.e.,

$$\text{[MDS]}\ H(W_k|W_{k\mathcal{K}_c}) = 0, \forall \mathcal{K}_c \subset [1 : N], |\mathcal{K}_c| = K_c \tag{4.7}$$

[1]While the problem statement is presented in its general form, we will primarily consider cases with $K = 2$ messages in this chapter.

In this chapter, we generalize the system model by incorporating further randomness in the strategies taken by the servers. Previous results also hold under this generalized model. Let us use $\mathcal{F}$ to denote a random variable privately generated by the user, whose realization is not available to the servers. $\mathcal{F}$ represents the randomness in the strategies followed by the user. Similarly, $\mathcal{G}$ is a random variable that determines the random strategies followed by the servers, and whose realizations are assumed to be known to all the servers and to the user. The user privately generates θ uniformly from $[1:K]$ and wishes to retrieve W_θ while keeping θ a secret from each server. $\mathcal{F}$ and $\mathcal{G}$ are generated independently and before the realizations of the messages or the desired message index are known, so that

$$H(\theta, \mathcal{F}, \mathcal{G}, W_1, \cdots, W_K) = H(\theta) + H(\mathcal{F}) + H(\mathcal{G}) + H(W_1) + \cdots + H(W_K) \tag{4.8}$$

Suppose $\theta = k$. In order to retrieve $W_k, k \in [1:K]$ privately, the user privately generates N random queries, $Q_1^{[k]}, \cdots, Q_N^{[k]}$.

$$H(Q_1^{[k]}, \cdots, Q_N^{[k]} | \mathcal{F}) = 0, \forall k \in [1:K] \tag{4.9}$$

The user sends query $Q_n^{[k]}$ to the n-th server, $n \in [1:N]$. Upon receiving $Q_n^{[k]}$, the n-th server generates an answering string $A_n^{[k]}$, which is a function of the received query $Q_n^{[k]}$, the stored information $W_{1n}, \cdots, W_{Kn}$ and $\mathcal{G}$,

$$H(A_n^{[k]} | Q_n^{[k]}, W_{1n}, \cdots, W_{Kn}, \mathcal{G}) = 0 \tag{4.10}$$

Each server returns to the user its answer $A_n^{[k]}$.[2]

From all the information that is now available at the user $(A_{1:N}^{[k]}, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G})$, the user decodes the desired message W_k according to a decoding rule that is specified by the PIR scheme. Let P_e denote the probability of error achieved with the specified decoding rule.

[2]If the $A_n^{[k]}$ are obtained as inner products of query vectors and stored message vectors, then such a PIR scheme is called a linear PIR scheme.

To protect the user's privacy, the K strategies must be indistinguishable (identically distributed) from the perspective of any subset $\mathcal{T} \subset [1:N]$ of at most T colluding servers, i.e., the following privacy constraint must be satisfied.

$$[T\text{-Privacy}]\ (Q_{\mathcal{T}}^{[k]}, A_{\mathcal{T}}^{[k]}, \mathcal{G}, W_{1\mathcal{T}}, \cdots, W_{K\mathcal{T}}) \sim (Q_{\mathcal{T}}^{[k']}, A_{\mathcal{T}}^{[k']}, \mathcal{G}, W_{1\mathcal{T}}, \cdots, W_{K\mathcal{T}}), \quad \forall k, k' \in [1:K], \forall \mathcal{T} \subset [1:N], |\mathcal{T}| = T \tag{4.11}$$

The PIR rate characterizes how many bits of desired information are retrieved per downloaded bit and is defined as follows.

$$R = L/D \tag{4.12}$$

where D is the expected value of the total number of bits downloaded by the user from all the servers.

A rate R is said to be ϵ-error achievable if there exists a sequence of PIR schemes, indexed by L, each of rate greater than or equal to R, for which $P_e \to 0$ as $L \to \infty$. Note that for such a sequence of PIR schemes, from Fano's inequality, we must have

$$[\text{Correctness}]\ o(L) = \frac{1}{L} H(W_k | A_{1:N}^{[k]}, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) \tag{4.13}$$

$$\overset{(4.9)}{=} \frac{1}{L} H(W_k | A_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) \tag{4.14}$$

where $o(L)$ represents a term whose value approaches zero as L approaches infinity. The supremum of ϵ-error achievable rates is called the capacity C.[3]

4.2 Settling the Conjecture

Our main result, which settles the FGHK conjecture, is stated in the following theorem.

[3]Alternatively, the capacity may be defined with respect to zero error criterion, i.e., the supreme of zero error achievable rates where a rate R is said to be zero error achievable if there exists (for some L) a PIR scheme of rate greater than or equal to R for which $P_e = 0$.

Theorem 4.1. *For the MDS-TPIR problem with $K = 2$ messages, $N = 4$ servers, $T = 2$ privacy and the $(N, K_c) = (4, 2)$ MDS storage code $(x, y) \to (x, y, x + y, x + 2y)$, a rate of $3/5$ is achievable. Since the achievable rate exceeds the conjectured capacity of $4/7$ for this setting, the FGHK conjecture is false.*

Proof: We present a scheme that achieves rate 3/5. We assume that each message is comprised of $L = 12$ symbols from $\mathbb{F}_p$ for a sufficiently[4] large prime p. Define $\mathbf{a} \in \mathbb{F}_p^{6\times 1}$ as the 6×1 vector $(a_1; a_2; \cdots; a_6)$ comprised of i.i.d. uniform symbols $a_i \in \mathbb{F}_p$. Vectors $\mathbf{b}, \mathbf{c}, \mathbf{d}$ are defined similarly. Messages W_1, W_2 are defined in terms of these vectors as follows.

$$W_1 = (\mathbf{a}; \mathbf{b}) \qquad W_2 = (\mathbf{c}; \mathbf{d}) \tag{4.15}$$

4.2.1 Storage Code

The storage is specified as

$$(W_{11}, W_{12}, W_{13}, W_{14}) = (\mathbf{a}, \mathbf{b}, \mathbf{a} + \mathbf{b}, \mathbf{a} + \mathbf{2b}) \tag{4.16}$$
$$(W_{21}, W_{22}, W_{23}, W_{24}) = (\mathbf{c}, \mathbf{d}, \mathbf{c} + \mathbf{d}, \mathbf{c} + \mathbf{2d}) \tag{4.17}$$

Recall that W_{kn} is the information about message W_k that is stored at Server n. Thus, Server 1 stores $(\mathbf{a}, \mathbf{c})$, Server 2 stores $(\mathbf{b}, \mathbf{d})$, Server 3 stores $(\mathbf{a} + \mathbf{b}, \mathbf{c} + \mathbf{d})$, and Server 4 stores $(\mathbf{a} + \mathbf{2b}, \mathbf{c} + \mathbf{2d})$. In particular, each server stores 6 symbols for each message, for a total of 12 symbols per server. Any two servers store just enough information to recover both messages, thus the MDS storage criterion is satisfied.

[4]It suffices to choose $p = 349$ for Theorem 4.1. In general, the appeal to large field size, analogous to the random coding argument in information theory, is made to prove the existence of a scheme, but may not be essential to the construction of the PIR scheme. [67] includes some examples of MDS-TPIR capacity achieving schemes over small fields.

4.2.2 Construction of Queries

The query to each server $Q_n^{[k]}$ is comprised of two parts, denoted as $Q_n^{[k]}(W_1), Q_n^{[k]}(W_2)$. Each part contains 3 row vectors, also called query vectors, along which the server should project its corresponding stored message symbols.

$$Q_n^{[k]} = (Q_n^{[k]}(W_1), Q_n^{[k]}(W_2)) \tag{4.18}$$

In preparation for the construction of the queries, let us denote the set of all full rank 6×6 matrices over $\mathbb{F}_p$ as $\mathcal{S}$. The user privately chooses two matrices, S and S', independently and uniformly from $\mathcal{S}$. Label the rows of S as $V_1, V_2, V_3, V_4, V_5, V_6$, and the rows of S' as $U_0, U_1, U_2, U_3, U_4, U_5$. Define

$$\mathcal{V}_1 = \{V_1, V_2, V_3\}, \quad \mathcal{U}_1 = \{U_0, U_6, U_8\} \tag{4.19}$$
$$\mathcal{V}_2 = \{V_1, V_4, V_5\}, \quad \mathcal{U}_2 = \{U_0, U_7, U_9\} \tag{4.20}$$
$$\mathcal{V}_3 = \{V_2, V_4, V_6\}, \quad \mathcal{U}_3 = \{U_0, U_1, U_3\} \tag{4.21}$$
$$\mathcal{V}_4 = \{V_3, V_5, V_6\}, \quad \mathcal{U}_4 = \{U_0, U_2, U_4\} \tag{4.22}$$

U_6, U_7, U_8, U_9 are obtained as follows.

$$U_6 = U_1 + U_2, \quad U_7 = U_1 + 2U_2 \tag{4.23}$$
$$U_8 = U_3 + U_4, \quad U_9 = U_3 + 2U_4 \tag{4.24}$$

As a preview of what we are trying to accomplish, we note that for Server $n \in [1:4]$, $\mathcal{V}_n$ will be used as the query vectors for desired message symbols, while $\mathcal{U}_n$ will be used as query vectors for undesired message symbols. Since $K_c = 2$, the same query vector V_i sent to two different servers will recover 2 independent desired symbols. Each $V_i, i \in [1:6]$, is used exactly twice, so all queries for desired symbols will return independent information for a total of 12 independent desired symbols. On the other hand, for undesired symbols note that U_0 is used as the query vector to all 4 servers, but because $K_c = 2$, it can only produce 2 independent symbols, i.e., 2 of the 4 symbols are redundant. The dependencies introduced via

(4.23),(4.24) are carefully chosen to ensure that the queries along U_1, U_2, U_6, U_7 will produce only 3 independent symbols. Similarly, the queries along U_3, U_4, U_8, U_9 will produce only 3 independent symbols. Thus, all the queries for the undesired message will produce a total of only 8 independent symbols. The 12 independent desired symbols and 8 independent undesired symbols will be resolved from a total of $12 + 8 = 20$ downloaded symbols, to achieve the rate $12/20 = 3/5$. To ensure $T = 2$ privacy, the $\mathcal{U}_i$ and $\mathcal{V}_i$ queries will be made indistinguishable from the perspective of any 2 colluding servers. The key to the $T = 2$ privacy is that any $\mathcal{V}_n, \mathcal{V}_{n'}$, $n \neq n'$ have one element in common. Similarly, any $\mathcal{U}_n, \mathcal{U}_{n'}$, $n \neq n'$ also have one element in common. This is a critical aspect of the construction.

Next we provide a detailed description of the queries and downloads for message $W_k, k \in [1:2]$, both when W_k is desired and when it is not desired. To simplify the notation, we will denote $W_k = (\mathbf{x}; \mathbf{y})$. Note that when $k = 1$, $(\mathbf{x}; \mathbf{y}) = (\mathbf{a}; \mathbf{b})$ and when $k = 2$, $(\mathbf{x}; \mathbf{y}) = (\mathbf{c}; \mathbf{d})$.

4.2.2.1 *Case 1. W_k is Desired*

The query sent to Server n is a 3×6 matrix whose rows are the 3 vectors in $\mathcal{V}_n$. The ordering of the rows is uniformly random, i.e.,

$$\text{Server } n: \ Q_n^{[k]}(W_k) = \pi_n(\mathcal{V}_n), \qquad n \in [1:4] \tag{4.25}$$

For a set $\mathcal{V} = \{V_{i_1}, V_{i_2}, V_{i_3}\}$, $\pi_n(\mathcal{V})$ is equally likely to return any one of the 6 possibilities: $(V_{i_1}; V_{i_2}; V_{i_3})$, $(V_{i_1}; V_{i_3}; V_{i_2})$, $(V_{i_2}; V_{i_1}; V_{i_3})$, $(V_{i_2}; V_{i_3}; V_{i_1})$, $(V_{i_3}; V_{i_1}; V_{i_2})$ and $(V_{i_3}; V_{i_2}; V_{i_1})$. The π_n are independently chosen for each $n \in [1:4]$.

After receiving the 3 query vectors $Q_n^{[k]}(W_k)$, Server n projects its stored W_{kn} symbols along these vectors. This creates three linear combinations of W_{kn} symbols (denoted as $A_n^{[k]}(W_k)$).

$$A_n^{[k]}(W_k) = Q_n^{[k]}(W_k) W_{kn} \tag{4.26}$$

Define $k^c = 3-k$ as the complement of k, i.e., $k^c = 1$ if $k = 2$ and vice versa. The answers $A_n^{[k]}$ to be sent to the user will be constructed eventually by combining $A_n^{[k]}(W_k)$ and $A_n^{[k]}(W_{k^c})$, since separately sending these answers will be too inefficient. The details of this combining process will be specified later. Next we note an important property of the construction.

Desired Symbols Are Independent: We show that if the user can recover $A_{1:4}^{[k]}(W_k)$ from the downloads, then he can recover all 12 symbols of W_k. From $A_{1:4}^{[k]}(W_k)$ the user recovers the 12 symbols $V_1\mathbf{x}$, $V_2\mathbf{x}$, $V_3\mathbf{x}$, $V_1\mathbf{y}$, $V_4\mathbf{y}$, $V_5\mathbf{y}$, $V_2(\mathbf{x}+\mathbf{y})$, $V_4(\mathbf{x}+\mathbf{y})$, $V_6(\mathbf{x}+\mathbf{y})$, $V_3(\mathbf{x}+\mathbf{2y})$, $V_5(\mathbf{x}+\mathbf{2y})$, $V_6(\mathbf{x}+\mathbf{2y})$. From these 12 symbols, he recovers $V_i\mathbf{x}$ and $V_i\mathbf{y}$ for all $i \in [1:6]$. Since $S = (V_1; V_2; V_3; V_4; V_5; V_6)$ has full rank (invertible) and the user knows $V_{1:6}$, he recovers all symbols in $\mathbf{x}$ and $\mathbf{y}$ (thus W_k).

4.2.2.2 *Case 2. W_k is Undesired*

Similarly, the query sent to Server n is a 3×6 matrix whose rows are the 3 vectors in $\mathcal{U}_n$. The ordering of the rows is uniformly random for each n, and independent across all $n \in [1:4]$.

$$\text{Server } n: \ Q_n^{[k^c]}(W_k) = \pi'_n(\mathcal{U}_n), \qquad n \in [1:4] \tag{4.27}$$

Each server projects its stored W_{kn} symbols along the 3 query vectors to obtain,

$$A_n^{[k^c]}(W_k) = Q_n^{[k^c]}(W_k)W_{kn} \tag{4.28}$$

Interfering Symbols Have Dimension 8*:* $A_{1:4}^{[k^c]}(W_k)$ is comprised of $U_0\mathbf{x}$, $U_6\mathbf{x}$, $U_8\mathbf{x}$, $U_0\mathbf{y}$, $U_7\mathbf{y}$, $U_9\mathbf{y}$, $U_0(\mathbf{x}+\mathbf{y})$, $U_1(\mathbf{x}+\mathbf{y})$, $U_3(\mathbf{x}+\mathbf{y})$, $U_0(\mathbf{x}+\mathbf{2y})$, $U_2(\mathbf{x}+\mathbf{2y})$, $U_4(\mathbf{x}+\mathbf{2y})$. We now show that these 12 symbols are dependent and have dimension only 8.[5] Because of (4.23) and (4.24), we have

$$\begin{aligned} U_0\mathbf{x} + U_0\mathbf{y} &= U_0(\mathbf{x}+\mathbf{y}) \\ U_0\mathbf{x} + 2U_0\mathbf{y} &= U_0(\mathbf{x}+2\mathbf{y}) \end{aligned}$$

[5]Equivalently, the joint entropy of these 12 variables, conditioned on $U_{0:9}$ is only 8 p-ary units.

$$\begin{aligned} U_6\mathbf{x} + U_7\mathbf{y} - U_1(\mathbf{x}+\mathbf{y}) &= U_2(\mathbf{x}+2\mathbf{y}) \\ U_8\mathbf{x} + U_9\mathbf{y} - U_3(\mathbf{x}+\mathbf{y}) &= U_4(\mathbf{x}+2\mathbf{y}) \end{aligned} \tag{4.29}$$

Thus, of the 12 symbols recovered from $A_{1:4}^{[k^c]}(W_k)$, at least 4 are linear combinations of the remaining 8. It follows that $A_{1:4}^{[k^c]}(W_k)$ contains no more than 8 dimensions. The number of dimensions is also not less than 8 because, the following 8 undesired symbols (two symbols from each server) are independent,

$$\begin{aligned} &\text{Server 1 :} && U_0\mathbf{x}, U_6\mathbf{x} = (U_1+U_2)\mathbf{x} \\ &\text{Server 2 :} && U_0\mathbf{y}, U_9\mathbf{y} = (U_3+2U_4)\mathbf{y} \\ &\text{Server 3 :} && U_1(\mathbf{x}+\mathbf{y}), U_3(\mathbf{x}+\mathbf{y}) \\ &\text{Server 4 :} && U_2(\mathbf{x}+2\mathbf{y}), U_4(\mathbf{x}+2\mathbf{y}) \end{aligned} \tag{4.30}$$

To see that the 8 symbols are independent, we add 4 new symbols ($U_1\mathbf{x}$, $U_3\mathbf{y}$, $U_5\mathbf{x}$, $U_5\mathbf{y}$) such that from the 12 symbols, we can recover all 12 undesired symbols ($S'\mathbf{x}$, $S'\mathbf{y}$). Since the 4 new symbols cannot contribute more than 4 dimensions, the original 8 symbols must occupy at least 8 dimensions.

4.2.3 Combining Answers for Efficient Download

Based on the queries, each server has 3 linear combinations of symbols of W_1 in $A_n^{[k]}(W_1)$ and 3 linear combinations of symbols of W_2 in $A_n^{[k]}(W_2)$ for a total of 12 linear combinations of desired symbols and 12 linear combinations of undesired symbols across all servers. However, recall that there are only 8 independent linear combinations of undesired symbols. This is a fact that can be exploited to improve the efficiency of download. Specifically, we will combine the 6 queried symbols (i.e., the 6 linear combinations) from each server into 5 symbols to be downloaded by the user. Intuitively, 5 symbols from each server will give the user a total of 20 symbols, from which he can resolve the 12 desired and 8 undesired symbols.

The following function maps 6 queried symbols to 5 downloaded symbols.

$$\mathcal{L}(X_1, X_2, X_3, Y_1, Y_2, Y_3) \ = (X_1, X_2, Y_1, Y_2, X_3 + Y_3) \tag{4.31}$$

Note that the first four symbols are directly downloaded and only the last symbol is mixed. The desired and undesired symbols are combined to produce the answers as follows.

$$A_n^{[k]} = \mathcal{L}(C_n A_n^{[k]}(W_1), C_n A_n^{[k]}(W_2)) \tag{4.32}$$

where C_n are deterministic 3×3 matrices, that are required to satisfy the following two properties. Denote the first 2 rows of C_n as $\overline{C}_n$.

P1. All C_n must have full rank.

P2. For all $(3!)^4$ distinct realizations of $\pi'_n, n \in [1:4]$, the 8 linear combinations of the undesired message symbols that are directly downloaded (2 from each server), $\overline{C}_1 A_1^{[k]}(W_{k^c})$, $\overline{C}_2 A_2^{[k]}(W_{k^c})$, $\overline{C}_3 A_3^{[k]}(W_{k^c})$, $\overline{C}_4 A_4^{[k]}(W_{k^c})$ are independent.

It is not difficult to find matrices that satisfy these properties. In fact, these properties are 'generic', i.e., uniformly random choices of C_n matrices will satisfy these properties with probability approaching 1 as the field size approaches infinity. The appeal to generic property will be particularly useful as we consider larger classes of MDS-TPIR settings. Those (weaker) proofs apply here as well. However, for the particular setting of Theorem 4.1, based on a brute force search we are able to strengthen the proof by presenting the following explicit choice of $C_n, n \in [1:4]$ which satisfies both properties over $\mathbb{F}_{349}$.

$$C_1 = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 0 & 0 & 1 \end{pmatrix}, \ C_2 = \begin{pmatrix} 1 & 7 & 3 \\ 11 & 9 & 8 \\ 0 & 0 & 1 \end{pmatrix}, \ C_3 = \begin{pmatrix} 1 & 10 & 8 \\ 7 & 5 & 4 \\ 0 & 0 & 1 \end{pmatrix}, \ C_4 = \begin{pmatrix} 1 & 3 & 5 \\ 12 & 9 & 3 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.33}$$

Property $P1$ is trivially verified. Property $P2$ is verified by considering one by one, all of the 6^4 distinct realizations of $\pi'_n, n \in [1:4]$. To show how this is done, let us consider one case here. Suppose the realization of the permutations is such that

$$\pi_1'(\mathcal{U}_1) = (U_0, U_6, U_8) \tag{4.34}$$
$$\pi_2'(\mathcal{U}_2) = (U_0, U_9, U_7) \tag{4.35}$$
$$\pi_3'(\mathcal{U}_3) = (U_1, U_3, U_0) \tag{4.36}$$
$$\pi_4'(\mathcal{U}_4) = (U_2, U_4, U_0) \tag{4.37}$$

then we have

$$(\overline{C}_1 A_1^{[k]}(W_{k^c}); \cdots; \overline{C}_4 A_4^{[k]}(W_{k^c})) = \underbrace{\begin{pmatrix} 1 & 2 & 0 & -3 & 0 & 3 & 0 & 3 \\ 6 & 5 & 0 & -4 & 0 & 4 & 0 & 4 \\ 0 & -3 & 1 & 7 & 3 & 0 & 3 & 0 \\ 0 & -8 & 11 & 9 & 8 & 0 & 8 & 0 \\ 8 & 0 & 8 & 0 & 1 & 10 & 0 & 0 \\ 4 & 0 & 4 & 0 & 7 & 5 & 0 & 0 \\ 5 & 0 & 10 & 0 & 0 & 0 & 1 & 3 \\ 3 & 0 & 6 & 0 & 0 & 0 & 12 & 9 \end{pmatrix}}_{\triangleq \mathcal{C}} \begin{pmatrix} U_0\mathbf{x} \\ U_6\mathbf{x} \\ U_0\mathbf{y} \\ U_9\mathbf{y} \\ U_1(\mathbf{x}+\mathbf{y}) \\ U_3(\mathbf{x}+\mathbf{y}) \\ U_2(\mathbf{x}+2\mathbf{y}) \\ U_4(\mathbf{x}+2\mathbf{y}) \end{pmatrix} \tag{4.38}$$

The determinant of $\mathcal{C}$ over $\mathbb{F}_{349}$ is 321. Since the determinant is non-zero, all of its 8 rows are linearly independent. Note that the test for property $P2$ does not depend on the realizations of U_i vectors. To see why this is true, note that the 8 linear combinations of $(\mathbf{x}, \mathbf{y})$ in the rightmost column vector of (4.38) are linearly independent. Therefore, if $\mathcal{C}$ is an invertible matrix then the 8 directly downloaded linear combinations on the LHS of (4.38) are also independent (have joint entropy 8 p-ary units, conditioned on $U_{0:9}$).

At this point the construction of the scheme is complete. All that remains now is to prove that the scheme is correct, i.e., it retrieves the desired message, and that it is $T = 2$ private.

4.2.4 The Scheme is Correct (Retrieves Desired Message)

As noted previously, the first 4 variables in the output of the $\mathcal{L}$ function are obtained directly, i.e., $\overline{C}_1 A_1^{[k]}(W_1)$, $\overline{C}_2 A_2^{[k]}(W_1)$, $\overline{C}_3 A_3^{[k]}(W_1)$, $\overline{C}_4 A_4^{[k]}(W_1)$ and $\overline{C}_1 A_1^{[k]}(W_2)$, $\overline{C}_2 A_2^{[k]}(W_2)$, $\overline{C}_3 A_3^{[k]}(W_2)$, $\overline{C}_4 A_4^{[k]}(W_2)$ are all directly recovered. By property $P2$ of C_n, $\overline{C}_1 A_1^{[k]}(W_{k^c})$,

$\overline{C}_2 A_2^{[k]}(W_{k^c})$, $\overline{C}_3 A_3^{[k]}(W_{k^c})$, $\overline{C}_4 A_4^{[k]}(W_{k^c})$ are linearly independent. Since the user has recovered 8 independent dimensions of interference, and interference only spans 8 dimensions, all interference is recovered and eliminated. Once the interference is eliminated, since C_n matrices have full rank, the user is left with 12 independent linear combinations of desired symbols, from which he is able to recover the 12 desired message symbols. Therefore the scheme is correct.

4.2.5 The Scheme is Private (to Any $T = 2$ Colluding Servers)

To prove that the scheme is $T = 2$ private (refer to (4.11)), it suffices to show that the queries for any 2 servers are identically distributed, regardless of which message is desired. Since each query is made up of two independently generated parts, one for each message, it suffices to prove that the query vectors for a message (say W_k) are identically distributed, regardless of whether the message is desired or undesired,

$$\left(Q_{n_1}^{[k]}(W_k), Q_{n_2}^{[k]}(W_k)\right) \sim \left(Q_{n_1}^{[k^c]}(W_k), Q_{n_2}^{[k^c]}(W_k)\right), \ \forall n_1, n_2 \in [1:4], n_1 < n_2 \tag{4.39}$$

Note that

$$\left(Q_{n_1}^{[k]}(W_k), Q_{n_2}^{[k]}(W_k)\right) = (\pi_{n_1}(\mathcal{V}_{n_1}), \pi_{n_2}(\mathcal{V}_{n_2})) \tag{4.40}$$
$$\left(Q_{n_1}^{[k^c]}(W_k), Q_{n_2}^{[k^c]}(W_k)\right) = \left(\pi'_{n_1}(\mathcal{U}_{n_1}), \pi'_{n_2}(\mathcal{U}_{n_2})\right) \tag{4.41}$$

Therefore, to prove (4.39) it suffices to show the following.

$$\left(V_{i_1}, V_{i_2}, V_{i_3}, V_{i_4}, V_{i_5}\right) \sim \left(U_0, U_{j_1}, U_{j_2}, U_{j_3}, U_{j_4}\right) \tag{4.42}$$

where $\mathcal{V}_{n_1} = \{V_{i_1}, V_{i_2}, V_{i_3}\}$, $\mathcal{V}_{n_2} = \{V_{i_1}, V_{i_4}, V_{i_5}\}$, $\mathcal{U}_{n_1} = \{U_0, U_{j_1}, U_{j_2}\}$, $\mathcal{U}_{n_2} = \{U_0, U_{j_3}, U_{j_4}\}$. Because S is uniformly chosen from the set of all full rank matrices, we have

$$(V_{i_1}, V_{i_2}, V_{i_3}, V_{i_4}, V_{i_5}) \sim (V_1, V_2, V_3, V_4, V_5) \tag{4.43}$$

Next we note that there is a bijection between

$$(U_0, U_{j_1}, U_{j_2}, U_{j_3}, U_{j_4}) \quad \leftrightarrow \quad (U_0, U_1, U_2, U_3, U_4) \tag{4.44}$$

This is because $(U_0, U_{j_1}, U_{j_2}, U_{j_3}, U_{j_4})$ always includes U_0, two terms out of U_1, U_2, U_6, U_7 and two terms out of U_3, U_4, U_8, U_9. But from any two terms of U_1, U_2, U_6, U_7 there is a bijection to U_1, U_2, and from any two terms of U_3, U_4, U_8, U_9 there is a bijection to U_3, U_4. Now since $S' = (U_0; U_1; U_2; U_3; U_4; U_5)$ is picked uniformly from $\mathcal{S}$, conditioned on any feasible value of U_5, $(U_0, U_1, U_2, U_3, U_4)$ is uniformly distributed over all possible values that preserve full rank for S'. Since $(U_0, U_{j_1}, U_{j_2}, U_{j_3}, U_{j_4})$ spans the same space as $(U_0, U_1, U_2, U_3, U_4)$, they have the same set of feasible values. The bijection between them then means that $(U_0, U_{j_1}, U_{j_2}, U_{j_3}, U_{j_4})$ is also uniformly distributed over all possibilities that preserve full rank for S', conditioned on any feasible U_5. That means

$$(U_0, U_{j_1}, U_{j_2}, U_{j_3}, U_{j_4}) \sim (U_0, U_1, U_2, U_3, U_4) \tag{4.45}$$

Finally, we note that S and S' are identically distributed, so we have

$$(V_1, V_2, V_3, V_4, V_5) \sim (U_0, U_1, U_2, U_3, U_4) \tag{4.46}$$

Combining (4.43), (4.45) and (4.46), we arrive at (4.42) and (4.39).

4.2.6 Rate Achieved is $3/5$

The rate achieved is $12/20 = 3/5$, because we download 20 symbols in total (5 from each server) and the desired message size is 12 symbols.

4.3 Optimality of Rate $3/5$

We presented a scheme that achieves the rate $3/5$ for the setting $(K, N, T, K_c) = (2, 4, 2, 2)$ with the MDS storage code $(x, y) \to (x, y, x + y, x + 2y)$. But is the scheme optimal? i.e., is the rate $3/5$ the highest rate possible for this setting? To settle this question we need an upper bound. So far the best information theoretic upper bound that we are able to prove is $8/13$[6] (see [67]), which leaves the information theoretic capacity open for this setting. However, let us define the notion of "linear capacity" as the highest rate that can be achieved by any (scalar or vector) linear PIR scheme. It turns out that we are able to settle the linear capacity.

Theorem 4.2. *For the MDS-TPIR problem with* $(K, N, T, K_c) = (2, 4, 2, 2)$ *and the MDS storage code* $(x, y) \to (x, y, x + y, x + 2y)$*, the linear capacity is* $3/5$*.*

Proof: Since the achievability of $3/5$ has already been shown, we are left to prove the converse, i.e., the upper bound.

Let $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{F}_p^{L/2\times 1}$ be i.i.d. uniform $L/2 \times 1$ vectors over $\mathbb{F}_p$. Without loss of generality, the MDS storage code for message W_k is represented as follows.

$$W_1 = (\mathbf{a}; \mathbf{b}) \qquad W_2 = (\mathbf{c}; \mathbf{d}) \tag{4.47}$$

and the storage is specified as

$$\begin{aligned} (W_{11}, W_{12}, W_{13}, W_{14}) &= (\mathbf{a}, \mathbf{b}, \mathbf{a} + \mathbf{b}, \mathbf{a} + \mathbf{2b}) \\ (W_{21}, W_{22}, W_{23}, W_{24}) &= (\mathbf{c}, \mathbf{d}, \mathbf{c} + \mathbf{d}, \mathbf{c} + \mathbf{2d}) \end{aligned} \tag{4.48}$$

The scheme is linear so that the download from each server consists of linear combinations of the stored symbols of both messages. Furthermore, without loss of generality, we assume

[6] Remarkably, $8/13$ can be shown to be the capacity if the colluding sets of servers are restricted to servers $\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}$ (see [67]).

that the scheme is symmetric[7] and the download from each server is comprised of $d \leq L/2$ independent symbols from each message. Therefore, the downloads can be expressed as

$$\begin{aligned} A_n^{[k]} &= V_{1n}^{[k]} W_{1n} + V_{2n}^{[k]} W_{2n}, \forall n \in [1:4], k \in [1:2] && (4.49) \\ \text{rank}(V_{1n}^{[k]}) &= \text{rank}(V_{2n}^{[k]}) = d && (4.50) \end{aligned}$$

where $V_{in}^{[k]}$ are $D/4 \times L/2$ matrices that may be chosen randomly by the user (functions of $\mathcal{F}$). Clearly we must have $4d \geq L$ otherwise the L symbols of the desired message cannot be recovered. Define $\epsilon \geq 0$ such that

$$4d = L(1+\epsilon) \quad (4.51)$$

Without loss of generality, let us assume henceforth that W_2 is the desired message. For the next set of arguments, we focus only on the downloads corresponding to W_2, i.e., set all W_1 symbols to 0. Further, let us use the notation V to represent the row span of the matrix V. The symbols downloaded from Server n along $\mathsf{V} \subset \mathsf{V}_{2n}^{[2]}$, are called redundant if they can be expressed as linear combinations of symbols downloaded from other servers, i.e., they contribute no new information.

$$H(VW_{2n} | V_{2n_1}^{[2]} W_{2n_1}, V_{2n_2}^{[2]} W_{2n_2}, V_{2n_3}^{[2]} W_{2n_3}, \mathcal{F}, V) = 0 \quad (4.52)$$

where n, n_1, n_2, n_3 are distinct indices in $[1:4]$. Note that we download no more than a total of $L(1+\epsilon)$ (possibly dependent) symbols of W_2 from all 4 servers, from which we must be able to decode all L independent symbols of W_2. Therefore, we cannot have more than ϵL redundant symbols. Therefore, for any V that satisfies (4.52) we must have

$$\dim(\mathsf{V}) \leq \epsilon L \quad (4.53)$$

[7]Any scheme can be made symmetric, e.g., by repeating the original scheme for each of the $N!$ permutations of the servers to retrieve a correspondingly expanded message of length $L' = N!L$.

Next, let us consider the pairwise overlap between $\mathsf{V}_{2i}^{[2]}$ and $\mathsf{V}_{2j}^{[2]}$, $i < j, i, j \in [1:4]$. By the symmetry of the scheme, there exist V_{ij}, $\forall i, j \in [1:4], i \neq j$, and $\alpha \geq 0$ such that

$$\mathsf{V}_{ij} = \mathsf{V}_{2i}^{[2]} \cap \mathsf{V}_{2j}^{[2]}, \quad \dim(\mathsf{V}_{ij}) = \alpha d \tag{4.54}$$

The following lemma formalizes the intuition that the overlaps α must be small enough to ensure that we have enough independent symbols to recover W_2.

Lemma 4.1.

$$3\alpha d \leq d + 2\epsilon L \tag{4.55}$$

$$\textit{Equivalently, } \alpha \leq \frac{1}{3} + \frac{8}{3}\left(\frac{\epsilon}{1+\epsilon}\right) \tag{4.56}$$

Proof: First, we show that

$$\dim(\mathsf{V}_{12} \cap \mathsf{V}_{13}) \leq \epsilon L \tag{4.57}$$

For any vector $v \in \mathsf{V}_{12} \cap \mathsf{V}_{13}$ (note that v belongs simultaneously to $\mathsf{V}_{21}^{[2]}, \mathsf{V}_{22}^{[2]}, \mathsf{V}_{23}^{[2]}$), the symbol vW_{23} (downloaded from Server 3) is redundant because it is a linear combination of downloads from servers 1 and 2,

$$v(\mathbf{c}+\mathbf{d}) = v\mathbf{c} + v\mathbf{d} \tag{4.58}$$

$$\therefore vW_{23} = vW_{21} + vW_{22} \tag{4.59}$$

$$\Rightarrow H(vW_{23} | V_{21}^{[2]}W_{21}, V_{22}^{[2]}W_{22}, \mathcal{F}, v) = 0 \tag{4.60}$$

From (4.60) and (4.53), we have (4.57).

Second, we show that

$$\dim\left((\mathsf{V}_{12} \cup \mathsf{V}_{13}) \cap \mathsf{V}_{14}\right) \leq \epsilon L \tag{4.61}$$

Consider any vector $v \in \mathsf{V}_{12}$. Because v belongs to both $\mathsf{V}_{21}^{[2]}$ and $\mathsf{V}_{22}^{[2]}$, we have downloaded $vW_{21} = v\mathbf{c}$ and $vW_{22} = v\mathbf{d}$ from servers 1 and 2. Similarly, for any vector $v' \in \mathsf{V}_{13}$, we

have downloaded $v'W_{21} = v'\mathbf{c}$ and $v'W_{23} = v'(\mathbf{c}+\mathbf{d}) = v'W_{21} + v'W_{22}$ (from servers 1 and 3), from which we can recover $v'W_{21} = v'\mathbf{c}$ and $v'W_{22} = v'\mathbf{d}$. Consider now any vector $v^* \in (\mathsf{V}_{12} \cup \mathsf{V}_{13}) \cap \mathsf{V}_{14}$. Suppose $v^* = h_1 v + h_2 v', v \in \mathsf{V}_{12}, v' \in \mathsf{V}_{13}$ for constants h_1, h_2. The symbol $v^* W_{24} = v^*(\mathbf{c}+\mathbf{2d})$ (downloaded from Server 4) is redundant because it is a linear combination of downloads from servers 1, 2 and 3,

$$\begin{aligned} v^*W_{24} &= (h_1 v + h_2 v')(\mathbf{c}+\mathbf{2d}) && (4.62)\\ &= h_1 v\mathbf{c} + 2h_1 v\mathbf{d} + h_2 v'\mathbf{c} + 2h_2 v'\mathbf{d} && (4.63)\\ &= h_1 vW_{21} + 2h_1 vW_{22} + h_2 v'W_{21} + 2h_2 v'W_{22} && (4.64)\\ &\Rightarrow \quad H(v^*W_{24} | V_{21}^{[2]}W_{21}, V_{22}^{[2]}W_{22}, V_{23}^{[2]}W_{23}, \mathcal{F}, v^*) = 0 && (4.65) \end{aligned}$$

From (4.65) and (4.53), we have (4.61). Next, consider $\dim(\mathsf{V}_{12} \cup \mathsf{V}_{13})$.

$$\begin{aligned} & \dim(\mathsf{V}_{12} \cup \mathsf{V}_{13}) && (4.66)\\ = \; & \dim(\mathsf{V}_{12}) + \dim(\mathsf{V}_{13}) - \dim(\mathsf{V}_{12} \cap \mathsf{V}_{13}) && (4.67)\\ \geq \; & 2\alpha d - \epsilon L \quad (\text{from (4.54)(4.57)}) && (4.68) \end{aligned}$$

Finally, consider $\dim(\mathsf{V}_{12} \cup \mathsf{V}_{13} \cup \mathsf{V}_{14})$.

$$\begin{aligned} d &= \dim(\mathsf{V}_{21}^{[2]}) \geq \dim(\mathsf{V}_{12} \cup \mathsf{V}_{13} \cup \mathsf{V}_{14}) && (4.69)\\ &= \dim(\mathsf{V}_{12} \cup \mathsf{V}_{13}) + \dim(\mathsf{V}_{14}) - \dim\left((\mathsf{V}_{12} \cup \mathsf{V}_{13}) \cap \mathsf{V}_{14}\right) && (4.70)\\ &\geq \quad 2\alpha d - \epsilon L + \alpha d - \epsilon L \quad (\text{from (4.68)(4.54)(4.61)}) && (4.71)\\ &\Rightarrow \; 3\alpha d \leq d + 2\epsilon L && (4.72) \end{aligned}$$

■

We now proceed to complete the converse.

$$\begin{aligned} & D + o(L)L \geq H(A_{1:4}^{[1]} | \mathcal{F}, \mathcal{G}) + o(L)L && (4.73)\\ \overset{(4.14)}{=} \; & H(A_{1:4}^{[1]}, W_1 | \mathcal{F}, \mathcal{G}) && (4.74)\\ \overset{(4.8)}{=} \; & H(W_1) + H(A_1^{[1]} | W_1, \mathcal{F}, \mathcal{G}) + H(A_{2:4}^{[1]} | W_1, A_1^{[1]}\mathcal{F}, \mathcal{G}) && (4.75)\\ \geq \; & H(W_1) + H(A_1^{[1]} | W_1, \mathcal{F}, \mathcal{G}) + H(A_{3:4}^{[1]} | W_1, W_{21}, A_1^{[1]}, \mathcal{F}, \mathcal{G}) && (4.76)\\ \overset{(4.6)(4.9)(4.10)}{=} \; & H(W_1) + H(A_1^{[1]} | W_1, \mathcal{F}, \mathcal{G}) + H(A_{3:4}^{[1]} | W_1, W_{21}, \mathcal{F}, \mathcal{G}) && (4.77) \end{aligned}$$

$$\overset{(4.6)(??)}{=} H(W_1) + H(A_1^{[2]}|W_1, \mathcal{F}, \mathcal{G}) + H(A_{3:4}^{[2]}|W_1, W_{21}, \mathcal{F}, \mathcal{G}) \tag{4.78}$$

$$\overset{(4.47)(4.48)}{=} H(\mathbf{a}, \mathbf{b}) + H(V_{21}^{[2]}\mathbf{c}|\mathcal{F}) + H(V_{23}^{[2]}(\mathbf{c}+\mathbf{d}), V_{24}^{[2]}(\mathbf{c}+\mathbf{2d})|\mathbf{c}, \mathcal{F}) \tag{4.79}$$

$$= H(\mathbf{a}, \mathbf{b}) + H(V_{21}^{[2]}\mathbf{c}|\mathcal{F}) + H(V_{23}^{[2]}\mathbf{d}, 2V_{24}^{[2]}\mathbf{d}|\mathcal{F}) \tag{4.80}$$

$$\overset{(4.4)}{=} L + \dim(\mathsf{V}_{21}^{[2]}) + \dim(\mathsf{V}_{23}^{[2]} \cup \mathsf{V}_{24}^{[2]}) \tag{4.81}$$

$$\overset{(4.50)(4.54)}{=} L + d + 2d - \alpha d \tag{4.82}$$

$$\overset{(4.56)}{\geq} L + \left(3 - \frac{1}{3} - \frac{8}{3}\left(\frac{\epsilon}{1+\epsilon}\right)\right)\frac{(1+\epsilon)L}{4} \tag{4.83}$$

$$= 5L/3 \tag{4.84}$$

Letting $L \to \infty$, we have $R = L/D \leq 3/5$.

■

4.4 Discussion

We settle a conjecture on the capacity of MDS-TPIR by Freij-Hollanti et al. [33] by constructing a scheme that beats the conjectured capacity for one particular instance of MDS-TPIR. The rate achieved by the new scheme is shown to be the best possible rate that can be achieved by any linear scheme for the same MDS storage code.

[67] further contains the following generalizations of the results presented in this chapter. The insights from the counterexample lead us to characterize the exact capacity of various instances of MDS-TPIR. This includes all cases with $(K, N, T, K_c) = (2, N, T, N-1)$, where N and T can be arbitrary. The capacity for these cases turns out to be

$$C = \frac{N^2 - N}{2N^2 - 3N + T} \tag{4.85}$$

Note that this is the information theoretic capacity, i.e., for $K = 2$ messages, no $(N-1, N)$ MDS storage code and no PIR scheme (linear or non-linear) can beat this rate, which is achievable with the simple MDS storage code $(x_1, x_2, \cdots, x_{N-1}) \to (x_1, x_2, \cdots, x_{N-1}, \sum_{i=1}^{N-1} x_i)$ and a linear PIR scheme.

The general capacity expression for MDS-TPIR remains unknown. However, we are able to show that it cannot be symmetric in K_c and T, i.e., the two parameters are not interchangeable in general. Also, between K_c and T the capacity expression does not consistently favor one over the other.

Finally, taking an asymptotic view of capacity of MDS-TPIR, we show that if $T + K_c > N$, then the capacity collapses to 0 as the number of messages $K \to \infty$. This is consistent with the restriction of $T + K_c \leq N$ that is required by the achievable scheme of Freij-Hollanti et al. whose rate does not depend on K.

Chapter 5

Capacity of Symmetric PIR

The original formulation of PIR only considers the privacy of the user. The privacy of the undesired messages is ignored. However, it is often desirable to restrict the user to retrieve nothing beyond his chosen message. This new constraint is called server privacy, and with this constraint, the problem is called symmetric[1] PIR (SPIR) [36]. SPIR is especially challenging because the servers must individually learn nothing about the identity of the desired message, but must still collectively allow the user to retrieve his desired message in such a way that the user learns nothing about any other message besides his desired message. For example, the trivial solution of downloading everything, which works for PIR, is no longer acceptable. The main result of this chapter is the characterization of the capacity of SPIR, i.e., the maximum number of bits of desired message that can be privately retrieved by a user per bit of downloaded information, without leaking any information about undesired messages to the user. For K messages and N servers, we show that the capacity is $1-1/N$, if the servers have access to common randomness (not available to the user) that is independent of the messages, in the amount that is at least $1/(N-1)$ bits per desired message bit, and zero otherwise.

[1]Symmetry means that the privacy of both the user and the server is considered.

$$C_{\text{SPIR}} = \begin{cases} 1 - 1/N & \text{if } \rho \geq \frac{1}{N-1} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Besides its direct applications, PIR is especially significant as a fundamental problem that lies at the intersection of several open problems in cryptography [34, 73], coding theory [46, 72, 40] and complexity theory [39]. SPIR inherits many of these connections from PIR. For example, SPIR is essentially a (distributed) form of oblivious transfer [55, 29], where the typical objective is that the transmitter(s) should not know which message is received by the receiver and the receiver should obtain nothing more than the desired message. Oblivious transfer is an important building block (primitive) in cryptography, whose feasibility leads to many other cryptographic protocols [47, 41]. Fundamental limits on the communication efficiency of various forms of oblivious transfer therefore represent an important class of open problems [3, 53]. The capacity characterization of SPIR is a promising step in this direction. Let us start with the problem statement.

5.1 Problem Statement

Consider K independent messages $W_1, \cdots, W_K$, where W_k is represented as an $L \times 1$ vector comprised of L i.i.d. uniform bits from the finite field $\mathbb{F}_2$.

$$H(W_1, \cdots, W_K) = H(W_1) + \cdots + H(W_K), \tag{5.2}$$
$$H(W_1) = \cdots = H(W_K) = L. \tag{5.3}$$

There are N servers. Each server stores all the messages $W_1, \cdots, W_K$.

Let us use $\mathcal{F}$ to denote a random variable privately generated by the user, whose realization is not available to the servers. $\mathcal{F}$ represents the randomness in the strategies followed by the user. Similarly, $\mathcal{G}$ is a random variable that determines the random strategies followed by the servers, and whose realizations are assumed to be known to all the servers and to the user. The user privately generates θ uniformly from $[1 : K]$ and wishes to retrieve W_θ

privately. The servers do not want to give out any information beyond the one message of the user's choosing (W_θ). In order to achieve server-privacy, we assume that the servers share a common random variable S that is not known to the user. This common randomness model is canonical for SPIR in the sense that it is introduced in the first SPIR paper [36] and has been adopted ever since, e.g., in [35]. The model is also minimal because it has been shown that without such common randomness, SPIR is not feasible [36]. $\mathcal{F}$ and $\mathcal{G}$ are generated independently and before the realizations of the messages, the common randomness or the desired message index are known, so that

$$H(\theta, \mathcal{F}, \mathcal{G}, W_1, \cdots, W_K, S) = H(\theta) + H(\mathcal{F}) + H(\mathcal{G}) + H(W_1) + \cdots + H(W_K) + H(S) \quad (5.4)$$

Suppose $\theta = k$. In order to retrieve message $W_k, k \in [1 : K]$ privately, the user privately generates N queries $Q_1^{[k]}, \cdots, Q_N^{[k]}$.

$$H(Q_1^{[k]}, \cdots, Q_N^{[k]} | \mathcal{F}) = 0, \forall k \in [1 : K]. \quad (5.5)$$

The user sends query $Q_n^{[k]}$ to the n-th server, $n \in [1 : N]$. Upon receiving $Q_n^{[k]}$, the n-th server generates an answering string $A_n^{[k]}$, which is a function of $Q_n^{[k]}$, all messages $W_1, \cdots, W_K$, the common randomness S and $\mathcal{G}$,

$$H(A_n^{[k]} | Q_n^{[k]}, W_1, \cdots, W_K, S, \mathcal{G}) = 0. \quad (5.6)$$

Each server returns to the user its answer $A_n^{[k]}$. From all the information that is now available to the user $(Q_{1:N}^{[k]}, A_{1:N}^{[k]}, \mathcal{F}, \mathcal{G})$, the user decodes the desired message W_k according to a decoding rule that is specified by the SPIR scheme. Let P_e denote the probability of error achieved with the specified decoding rule.

To protect the user's privacy, the K strategies must be indistinguishable (identically distributed) from the perspective of any individual server, i.e., the following user-privacy constraint must be satisfied[2] $\forall k, k' \in [1:K], \forall n \in [1:N]$,

$$\text{[User-Privacy]} \quad (Q_n^{[k]}, A_n^{[k]}, W_{1:K}, S, \mathcal{G}) \sim (Q_n^{[k']}, A_n^{[k']}, W_{1:K}, S, \mathcal{G}) \tag{5.7}$$

Symmetric PIR also requires protecting the privacy of the server, i.e., it must be ensured that the user learns nothing more than the desired message W_k. So the vector $W_{\overline{k}} = (W_1, \cdots, W_{k-1}, W_{k+1}, \cdots, W_K)$, must be independent of all the information available to the user. Thus, the following server-privacy constraint must be satisfied:

$$\text{[Server-Privacy]} \quad I(W_{\overline{k}}\,; Q_{1:N}^{[k]}, A_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) = 0, \forall k \in [1:K] \tag{5.8}$$

The SPIR rate characterizes the amount of desired information retrieved per downloaded bit, and is defined as follows.

$$R \triangleq L/D \tag{5.9}$$

where D is the expected value of the total number of bit downloaded by the user from all the servers. The rate R is said to be ϵ-error achievable if there exists a sequence of PIR schemes, indexed by L, where the PIR rate is greater than or equal to R and $P_e \to 0$ as $L \to \infty$. Note that for such a sequence of SPIR schemes, from Fano's inequality, we must have

$$\begin{aligned} \text{[Correctness]}\ o(L) &= \frac{1}{L} H(W_k | Q_{1:N}^{[k]}, A_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.10) \\ &\overset{(5.5)}{=} \frac{1}{L} H(W_k | A_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.11) \end{aligned}$$

where $o(L)$ represents a term whose value approaches zero as L approaches infinity. The supremum of ϵ-error achievable rates is called the capacity $\mathcal{C}_{\text{SPIR}}$.

[2]The User-Privacy constraint is equivalently expressed as $I(\theta; Q_n^{[\theta]}, A_n^{[\theta]}, W_{1:K}, S, \mathcal{G}) = 0$.

5.2 Main Result: Capacity of Symmetric PIR

When there is only $K = 1$ message, note that the server-privacy constraint is satisfied trivially, so that SPIR reduces to the PIR setting and the capacity is 1. For $K \geq 2$, it is known that some common randomness S is necessary for the feasibility of SPIR. Let us define ρ as the amount of common randomness relative to the message size

$$\rho \quad = \quad \frac{H(S)}{H(W)} = \frac{H(S)}{L} \tag{5.12}$$

The capacity should depend on ρ, and because availability of common randomness at the servers is a non-trivial requirement, this dependence is of some interest.

When there is only $N = 1$ server, it is easy to see that the server-privacy constraint, the user-privacy constraint and correctness constraint conflict with each other such that SPIR is not feasible and the capacity is zero. The reason is as follows. First, because of the user-privacy constraint (5.7), the answer from the only server $A_1^{[k]}$ is identically distributed for all $k \in [1 : K]$. Second, from the correctness constraint (5.11), from $A_1^{[k]}, \mathbb{F}, \mathbb{G}$, one can decode W_k. Combining these two facts, we have that from $A_1^{[k]}$, one can decode all messages $W_1, \cdots, W_K$. This contradicts the server-privacy constraint (5.8). Therefore, when $N = 1$ and $K \geq 2$, SPIR is not feasible.

The following theorem states the capacity of SPIR.

Theorem 5.1. *For SPIR with $K \geq 2$ messages and $N \geq 2$ servers, the capacity is*

$$C_{SPIR} = \begin{cases} 1 - 1/N & \text{if } \rho \geq \frac{1}{N-1} \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

The achievability proof appears in Section 5.3. The converse proof appears in Section 5.4. The following observations place Theorem 5.1 in perspective.

1. We notice a surprising threshold phenomenon in the dependence of SPIR capacity, C_{SPIR}, on the amount of common randomness ρ. When $\rho < \frac{1}{N-1}$, SPIR is not feasible

and $C_{\text{SPIR}} = 0$. However, when $\rho \geq \frac{1}{N-1}$, SPIR is not only possible, but the rate can immediately be increased to the maximum possible, i.e., the capacity. Therefore, the minimum common randomness required to achieve any positive rate is already sufficient to achieve the capacity of SPIR. A pictorial illustration of the SPIR capacity and its dependency on the amount of common randomness appears in Figure 5.1.

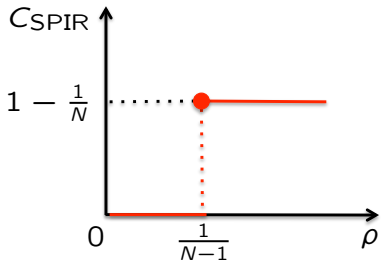


Figure 5.1: SPIR Capacity.

2. The capacity of SPIR is independent of the number of messages, K.

3. When the capacity is non-zero, the capacity is strictly increasing in the number of servers, N, and when N approaches infinity, the capacity approaches 1.

4. It is interesting to compare the capacity of SPIR and the capacity of PIR,

$$C_{\text{PIR}} = \left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1}. \tag{5.14}$$

We see that the capacity of SPIR is strictly smaller than the capacity of PIR (the additional requirement of preserving server-privacy strictly hurts) and the capacity of PIR approaches the capacity of SPIR when the number of messages, K, approaches infinity (in the large number of messages regime, the penalty vanishes), i.e., $C_{\text{PIR}} > C_{\text{SPIR}}$ for any finite K and $C_{\text{PIR}} \to C_{\text{SPIR}}$ when $K \to \infty$.

5. In the achievability proof for Theorem 5.1, the message size is $N - 1$ bits per message. Therefore, to achieve capacity, message size is not required to approach infinity. By employing the scheme multiple times, we know that when message size is equal to an

integer multiple of $N-1$ bits, the capacity is achieved as well. When the message size is not equal to an integer multiple of $N-1$ bits, it turns out that capacity can not be achieved and there is a penalty in the form of a ceiling operation. We note that the converse (upper bound) holds for arbitrary message size L when we require exactly zero error and the $o(L)$ terms can be replaced with 0.

6. The achievable scheme presented in Section 5.3 has exactly zero error. Therefore, SPIR capacity remains the same under both zero error and ϵ error criteria.

5.3 Theorem 5.1: Achievability

In this section, we present the scheme that achieves rate $1-1/N$, when $\rho = 1/(N-1)$. To this end, we assume each message consists of $N-1$ bits and each answering string is 1 bit. Specifically, we assume $W_k = (x_{k,1}, \cdots, x_{k,N-1}), \forall k \in [1:K]$ where each $x_{k,i}, i \in [1:N-1]$ is one bit. We further assume the entropy of the common random variable S is 1 bit, i.e., S is uniformly distributed over $\{0,1\}$. Note that S is independent of the messages.

Next we specify the queries. To retrieve W_k privately, the user first generates a random vector of length $(N-1)K$, $[h_{1,1}, \cdots, h_{1,N-1}, \cdots, h_{k,1}, \cdots, h_{K,N-1}]$, where each element is uniformly distributed over $\{0,1\}$. Then the queries are set as follows.

$$\begin{aligned} Q_1^{[k]} &= [h_{1,1}, \cdots, h_{k,1}, \cdots, h_{k,N-1}, \cdots, h_{K,N-1}] \\ Q_2^{[k]} &= [h_{1,1}, \cdots, h_{k,1}+1, \cdots, h_{k,N-1}, \cdots, h_{K,N-1}] \\ &\cdots \\ Q_N^{[k]} &= [h_{1,1}, \cdots, h_{k,1}, \cdots, h_{k,N-1}+1, \cdots, h_{K,N-1}] \end{aligned} \tag{5.15}$$

The answering strings are generated by using the query vector as the combining coefficients and producing the corresponding linear combination of message bits. We further add the common random variable to each answer.

$$
\begin{aligned}
A_1^{[k]} &= \sum_{j=1}^{K}\sum_{i=1}^{N-1} h_{j,i}x_{j,i} + S \\
A_2^{[k]} &= \sum_{j=1}^{K}\sum_{i=1}^{N-1} h_{j,i}x_{j,i} + x_{k,1} + S \\
&\cdots \\
A_N^{[k]} &= \sum_{j=1}^{K}\sum_{i=1}^{N-1} h_{j,i}x_{j,i} + x_{k,N-1} + S
\end{aligned} \tag{5.16}
$$

The user obtains $x_{k,i}, i \in [1:N-1]$ by subtracting $A_1^{[k]}$ from $A_{i+1}^{[k]}$. Therefore, the correctness condition is satisfied.

Privacy of the user is guaranteed because each query is independent of the desired message index k. This is because regardless of the desired message index k, each of the query vectors $Q_n^{[k]}, \forall n$ is individually comprised of elements that are i.i.d. uniform over $\{0,1\}$. Thus, each server learns nothing about which message is requested.

We now show that server-privacy is preserved as well.

$$
\begin{aligned}
& I(W_{\overline{k}}\,; A_1^{[k]}, A_2^{[k]}, \cdots, A_N^{[k]}, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.17)\\
= \;& I(W_{\overline{k}}\,; A_1^{[k]}, A_1^{[k]} + x_{k,1}, \cdots, A_1^{[k]} + x_{k,N-1}, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.18)\\
= \;& I(W_{\overline{k}}\,; A_1^{[k]}, x_{k,1}, \cdots, x_{k,N-1}, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.19)\\
= \;& I(W_{\overline{k}}\,; A_1^{[k]}, W_k, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.20)\\
= \;& I(W_{\overline{k}}\,; A_1^{[k]} | W_k, Q_{1:N}^{[k]}, \mathcal{F}, \mathcal{G}) && (5.21)\\
= \;& 0 && (5.22)
\end{aligned}
$$

where in each step, the transformation on the variables is invertible such that mutual information remains the same. The last step follows from the independence of the messages and the common randomness (refer to (5.4)).

Note that because each answering string is 1 bit and the message is $L = N-1$ bits, the rate achieved is $(N-1)/N = 1 - 1/N$ which matches the capacity. Also note that only the minimum threshold amount of common randomness is utilized, i.e., $\rho = 1/(N-1)$. ■

5.4 Theorem 5.1: Converse

For the converse we allow any feasible SPIR scheme, and prove that its rate cannot be larger than C_{SPIR}. Let us start with two lemmas that will be used later in the proof.

Lemma 5.1.

$$\begin{aligned} H(A_n^{[k]}|W_k, Q_n^{[k]}, \mathcal{G}) &= H(A_n^{[k']}|W_k, Q_n^{[k']}, \mathcal{G}) && (5.23)\\ H(A_n^{[k]}|Q_n^{[k]}, \mathcal{G}) &= H(A_n^{[k']}|Q_n^{[k']}, \mathcal{G}), \quad \forall n \in [1:N] && (5.24)\end{aligned}$$

Proof. Since the proofs of (5.23) and (5.24) follow from the same arguments, here we will present only the proof of (5.23). From the User-Privacy constraint (5.7) we know that $\forall k \in [1:K], \forall n \in [1:N]$, $I(\theta; A_n^{[\theta]}, W_k, Q_n^{[\theta]}, \mathcal{G}) = 0$. Therefore, we must have $\forall k' \in [1:K]$,

$$\begin{aligned} H(A_n^{[k]}, W_k, Q_n^{[k]}, \mathcal{G}) &= H(A_n^{[k']}, W_k, Q_n^{[k']}, \mathcal{G}) && (5.25)\\ H(W_k, Q_n^{[k]}, \mathcal{G}) &= H(W_k, Q_n^{[k']}, \mathcal{G}) && (5.26)\end{aligned}$$

Combining (5.25) and (5.26), we obtain $H(A_n^{[k]}|W_k, Q_n^{[k]}, \mathcal{G}) = H(A_n^{[k']}|W_k, Q_n^{[k']}, \mathcal{G})$. ■

Lemma 5.2.

$$H(A_n^{[k]}|W_k, \mathcal{F}, Q_n^{[k]}, \mathcal{G}) = H(A_n^{[k]}|W_k, Q_n^{[k]}, \mathcal{G}), \quad \forall n \in [1:N] \tag{5.27}$$

Proof. Since

$$H(A_n^{[k]}|W_k, Q_n^{[k]}, \mathcal{G}) - H(A_n^{[k]}|W_k, \mathcal{F}, Q_n^{[k]}, \mathcal{G}) = I(A_n^{[k]}; \mathcal{F}|W_k, Q_n^{[k]}, \mathcal{G}) \geq 0, \tag{5.28}$$

we only need to prove $I(A_n^{[k]}; \mathcal{F}|W_k, Q_n^{[k]}, \mathcal{G}) \leq 0$.

$$\begin{aligned} & I(A_n^{[k]}; \mathcal{F}|W_k, Q_n^{[k]}, \mathcal{G}) && (5.29)\\ \leq\ & I(A_n^{[k]}, W_1, \cdots, W_K, S; \mathcal{F}|W_k, Q_n^{[k]}, \mathcal{G}) && (5.30)\\ =\ & I(W_1, \cdots, W_K, S; \mathcal{F}|W_k, Q_n^{[k]}, \mathcal{G}) + \underbrace{I(A_n^{[k]}; \mathcal{F}|W_1, \cdots, W_K, S, W_k, Q_n^{[k]}, \mathcal{G})}_{=0} && (5.31)\\ \leq\ & I(W_1, \cdots, W_K, S; \mathcal{F}, \mathcal{G}, Q_n^{[k]}) && (5.32)\end{aligned}$$

$$= \quad 0 \tag{5.33}$$

where the second term in (5.31) is zero because of (5.6) and (5.33) follows from (5.4), (5.5).

■

The proof for $R \leq C_{\mathbf{SPIR}}$

For every feasible SPIR scheme, we must satisfy the server-privacy constraint (5.8),

$$0 = I(W_{\overline{k'}}\,; A_1^{[k']}, \cdots, A_N^{[k']}, \mathcal{F}, \mathcal{G}) \tag{5.34}$$

such that $\forall n \in [1:N], \forall k \in [1:K], k \neq k'$,

$$\begin{aligned}
0 &= I(W_k; A_n^{[k']}, Q_n^{[k']}, \mathcal{G}) = I(W_k; A_n^{[k']} | Q_n^{[k']}, \mathcal{G}) && (5.35)\\
&= H(A_n^{[k']} | Q_n^{[k']}, \mathcal{G}) - H(A_n^{[k']} | W_k, Q_n^{[k']}, \mathcal{G}) && (5.36)\\
&\overset{(5.23)}{=} H(A_n^{[k']} | Q_n^{[k']}, \mathcal{G}) - H(A_n^{[k]} | W_k, Q_n^{[k]}, \mathcal{G}) && (5.37)
\end{aligned}$$

Now, consider the answering strings $A_1^{[k]}, \cdots, A_N^{[k]}$, from which we can decode W_k.

$$\begin{aligned}
L = H(W_k) &\overset{(5.4)}{=} H(W_k | \mathcal{F}, \mathcal{G}) \leq I(W_k; A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) + o(L)L && (5.38)\\
&= H(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - H(A_1^{[k]}, \cdots, A_N^{[k]} | W_k, \mathcal{F}, \mathcal{G}) + o(L)L && (5.39)\\
&\overset{(5.5)}{\leq} H(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - H(A_n^{[k]} | W_k, \mathcal{F}, Q_n^{[k]}, \mathcal{G}) + o(L)L && (5.40)\\
&\overset{(5.27)}{=} H(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - H(A_n^{[k]} | W_k, Q_n^{[k]}, \mathcal{G}) + o(L)L && (5.41)\\
&\overset{(5.37)}{=} H(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - H(A_n^{[k']} | Q_n^{[k']}, \mathcal{G}) + o(L)L && (5.42)\\
&\overset{(5.24)}{=} H(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - H(A_n^{[k]} | Q_n^{[k]}, \mathcal{G}) + o(L)L && (5.43)\\
&\overset{(5.5)}{\leq} H(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - H(A_n^{[k]} | \mathcal{F}, \mathcal{G}) + o(L)L && (5.44)
\end{aligned}$$

Adding (5.44) for all $n \in [1:N]$, we have

$$NL \quad \leq \quad NH(A_1^{[k]}, \cdots, A_N^{[k]} | \mathcal{F}, \mathcal{G}) - \sum_{n \in [1:N]} H(A_n^{[k]} | \mathcal{F}, \mathcal{G}) + o(L)L \tag{5.45}$$

$$
\begin{aligned}
&\leq N\left(1-\frac{1}{N}\right) H(A_1^{[k]},\cdots,A_N^{[k]}|\mathcal{F},\mathcal{G})+o(L)L && (5.46)\\
&\leq N\left(1-\frac{1}{N}\right)\sum_{n=1}^{N} H(A_n^{[k]})+o(L)L && (5.47)\\
&\leq N\left(1-\frac{1}{N}\right) D+o(L)L && (5.48)\\
R_k &= \frac{L}{D}\leq 1-\frac{1}{N} \quad \text{(Letting } L\to\infty\text{)} && (5.49)
\end{aligned}
$$

Thus, the rate of any feasible SPIR scheme cannot be more than C_{SPIR}.

The proof for $\rho \geq 1/(N-1)$

Suppose a feasible SPIR scheme exists that achieves a non-zero SPIR rate. Then we will show in this section that it must have $\rho \geq 1/(N-1)$. Consider the answering strings $A_1^{[k]},\cdots,A_N^{[k]}$, from which we can decode W_k. From the server-privacy constraint, we have

$$
\begin{aligned}
0 &= I(W_{\overline{k}}\,; A_1^{[k]},\cdots,A_N^{[k]},\mathcal{F},\mathcal{G}) \overset{(5.4)}{=} I(W_{\overline{k}}\,; A_1^{[k]},\cdots,A_N^{[k]}|\mathcal{F},\mathcal{G}) && (5.50)\\
&\overset{(5.11)}{=} I(W_{\overline{k}}\,; A_1^{[k]},\cdots,A_N^{[k]},W_k|\mathcal{F},\mathcal{G})+o(L)L && (5.51)\\
&\overset{(5.4)}{=} I(W_{\overline{k}}\,; A_1^{[k]},\cdots,A_N^{[k]}|W_k,\mathcal{F},\mathcal{G})+o(L)L && (5.52)\\
&\geq I(W_{\overline{k}}\,; A_n^{[k]}|W_k,\mathcal{F},\mathcal{G})+o(L)L && (5.53)\\
&= H(A_n^{[k]}|W_k,\mathcal{F},\mathcal{G})-H(A_n^{[k]}|W_1,\cdots,W_K,\mathcal{F},\mathcal{G})+o(L)L && (5.54)\\
&\overset{(5.6)}{=} H(A_n^{[k]}|W_k,\mathcal{F},\mathcal{G})-H(A_n^{[k]}|W_1,\cdots,W_K,\mathcal{F},\mathcal{G})\\
&\quad + H(A_n^{[k]}|W_1,\cdots,W_K,\mathcal{F},\mathcal{G},S)+o(L)L && (5.55)\\
&= H(A_n^{[k]}|W_k,\mathcal{F},\mathcal{G})-I(S;A_n^{[k]}|W_1,\cdots,W_K,\mathcal{F},\mathcal{G})+o(L)L && (5.56)\\
&\geq H(A_n^{[k]}|W_k,\mathcal{F},\mathcal{G},Q_n^{[k]})-H(S)+o(L)L && (5.57)\\
&\overset{(5.27)}{=} H(A_n^{[k]}|W_k,Q_n^{[k]},\mathcal{G})-H(S)+o(L)L && (5.58)\\
&\overset{(5.37)}{=} H(A_n^{[k']}|Q_n^{[k']},\mathcal{G})-H(S)+o(L)L && (5.59)\\
&\overset{(5.24)}{=} H(A_n^{[k]}|Q_n^{[k]},\mathcal{G})-H(S)+o(L)L && (5.60)
\end{aligned}
$$

Adding (5.60) for $n\in[1:N]$, we have

$$
0 \geq \sum_{n\in[1:N]} H(A_n^{[k]}|Q_n^{[k]},\mathcal{G})-NH(S)+o(L)L \qquad (5.61)
$$

$$
\begin{aligned}
&\geq N\frac{1}{N}H(A_1^{[k]},\cdots,A_N^{[k]}|\mathcal{F},\mathcal{G}) - NH(S) + o(L)L && (5.62)\\
&\overset{(5.46)}{\geq} N\frac{1}{N-1}L - NH(S) + o(L)L && (5.63)\\
\Rightarrow H(S) &\geq \frac{1}{N-1}L + o(L)L && (5.64)\\
\Rightarrow \rho &= \frac{H(S)}{L} \geq \frac{1}{N-1} \quad \text{(Letting } L \to \infty\text{)} && (5.65)
\end{aligned}
$$

Thus, the amount of common randomness relative to the message size of any feasible SPIR scheme cannot be less than $1/(N-1)$.

Chapter 6

Multiround PIR: Capacity and Storage Overhead

The capacity has recently been characterized for PIR as well as several of its variants such as LPIR [66] – where message *lengths* can be arbitrary, TPIR (see Chapter 3) – where any set of up to T servers may collude, RPIR (see Chapter 3) – where *robustness* is required against unresponsive servers, SPIR (see Chapter 5) – which extends the privacy constraint *symmetrically* to protect both the user and the servers, MDS-PIR [6] and MDS-SPIR [71] – variants of PIR and SPIR, respectively, where each message is separately MDS coded.[1]

A common theme in these results is that there is no capacity advantage of non-linear schemes over linear schemes, or of ϵ-error schemes over zero-error schemes. This is a matter of some curiosity because the necessity of non-linear coding schemes has often been a key obstacle in network coding capacity problems [27, 21, 58, 20], and the capacity benefit of ϵ-error schemes over zero-error schemes for network coding problems in general [50] remains one of the key unresolved mysteries — with direct connections to the edge-removal question [45] and the

[1] As a caveat, we note that separate MDS coding of each message is a restrictive assumption. Consider the setting with $K = 2$ messages, $N = 3$ servers and the storage size of each server is equal to the size of one message. If separate MDS codes are employed for each message, then the maximum rate (capacity) is equal to $3/5$ [6]. However, Example 2 in [22] shows that rate $2/3$ ($> 3/5$) is achievable with a storage code that jointly encodes both messages.

existence of strong converses [48] in network information theory. Motivated by this curiosity, in this chapter we explore another important variant of PIR – *multiround* PIR (MPIR). Our contributions are summarized next.

The classical PIR setting assumes that all the queries are simultaneously generated by the user. This assumption is also made in all previous chapters. However, such a constraint is not essential to PIR. What if this constraint is relaxed, i.e., multiple rounds of queries and answers are allowed, such that the queries in each round of communication are generated by the user with the knowledge of the answers from all previous rounds? The resulting variant of the PIR problem is the *multiround* PIR (MPIR) problem (also known as interactive PIR [11, 12]). Multiround PIR has been noted as an intriguing possibility in several prior works [24, 11, 12]. However, it is not known whether there is any benefit of MPIR over single-round PIR. Answering this question from a capacity perspective is the first contribution of this chapter. Specifically, we show that the capacity of MPIR is the same as the capacity of PIR, i.e.,

$$C_{\text{MPIR}} = C_{\text{PIR}} = \left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1}. \tag{6.1}$$

Combined with previous results, this shows that there is no capacity advantage from multiround over single-round schemes, non-linear over linear schemes or from ϵ-error over zero-error schemes. Furthermore, we show that this is true even with T-privacy constraints.

To complement the capacity analysis, we consider another metric of interest – storage overhead. Classical PIR assumes replicated servers, i.e., each server stores all the messages. For larger datasets, replication schemes incur substantial storage costs. Coding has been shown to be an effective way to reduce the storage costs in distributed data storage systems. Applications of coding to reduce the storage overhead for PIR have attracted attention recently [59, 22, 31, 69, 56, 19, 61, 6, 75, 71]. In this context, our main contribution is an example ($N = 2$ servers, $K = 2$ messages) of a multiround, non-linear, ϵ-error PIR scheme that achieves a strictly smaller storage overhead than the best possible with a single-round,

linear, zero-error scheme. The simplicity of the scheme and the $N = K = 2$ setting makes it an attractive point of reference for future work toward understanding the role of linear versus non-linear schemes, zero-error versus ϵ-error capacity, and single-round versus multiround communications. Interestingly, the scheme reveals that coded storage is useful not only for reducing the storage overhead, but also it has a surprising benefit of enhancing the privacy of PIR.

6.1 Problem Statement

Let us start with a general problem statement that can then be specialized to various settings of interest. Consider K independent messages $W_1, \cdots, W_K$, each comprised of L i.i.d. uniform bits.

$$H(W_1, \cdots, W_K) = H(W_1) + \cdots + H(W_K), \tag{6.2}$$
$$H(W_1) = \cdots = H(W_K) = L. \tag{6.3}$$

There are N servers. Let S_n denote the information that is stored at the n-th server.

$$H(S_n|W_1, W_2, \cdots, W_K) \;=\; 0, \quad \forall n \in [1:N]. \tag{6.4}$$

Define the storage overhead α as the ratio of the total amount of storage used by all servers to the total amount of data.

$$\alpha \;\triangleq\; \frac{\sum_{n=1}^{N} H(S_n)}{KL}. \tag{6.5}$$

For replication based schemes, each server stores all K messages, so $S_n = (W_1, W_2, \cdots, W_K)$, $H(S_n) = KL$, and the storage overhead, $\alpha = N$.

A user privately generates θ uniformly from $[1:K]$ and wishes to retrieve W_θ while keeping θ a secret from each server.

Prior works on capacity of PIR and its variants make certain (implicitly justified) assumptions of deterministic behavior, e.g., that the answers provided by the servers are deterministic functions of queries and messages. Here we will follow, instead, an explicit formulation. We allow randomness in the strategies followed by the user and the servers. This is accomplished by representing the actions of the user and the servers as functions of random variables. Let us use $\mathbb{F}$ to denote a random variable privately generated by the user, whose realization is not available to the servers. Similarly, $\mathbb{G}$ is a random variable that determines the random strategies followed by the servers, and whose realizations are assumed to be known to all the servers and the user without loss of generality. $\mathbb{F}$ and $\mathbb{G}$ take values over the set of all deterministic strategies that the user or the servers can follow, respectively, associating each strategy with a certain probability. $\mathbb{F}$ and $\mathbb{G}$ are generated offline, i.e., before the realizations of the messages or the desired message index are known. Since these random variables are generated a-priori we must have

$$\begin{aligned} & H(\theta, \mathbb{F}, \mathbb{G}, W_1, \cdots, W_K) \\ = \; & H(\theta) + H(\mathbb{F}) + H(\mathbb{G}) + H(W_1) + \cdots + H(W_K) \end{aligned} \tag{6.6}$$

The multiround PIR scheme proceeds as follows. Suppose $\theta = k$. In order to retrieve $W_k, k \in [1:K]$ privately, the user communicates with the servers over Γ rounds. In the first round, the user privately generates N random queries, $Q_1^{[k]}(1), Q_2^{[k]}(1), \cdots, Q_N^{[k]}(1)$.

$$H(Q_1^{[k]}(1), Q_2^{[k]}(1), \cdots, Q_N^{[k]}(1)|\mathbb{F}) = 0, \quad \forall k \in [1:K] \tag{6.7}$$

The user sends query $Q_n^{[k]}(1)$ to the n-th server, $\forall n \in [1:N]$. Upon receiving $Q_n^{[k]}(1)$, the n-th server generates an answering string $A_n^{[k]}(1)$. Without loss of generality, we assume that the answering string is a function of $Q_n^{[k]}(1)$, the stored information S_n, and the random variable $\mathbb{G}$.

$$H(A_n^{[k]}(1)|Q_n^{[k]}(1), S_n, \mathbb{G}) = 0. \tag{6.8}$$

Each server returns to the user its answer $A_n^{[k]}(1)$.

Proceeding similarly[2] , over the γ-th round, $\gamma \in [2 : \Gamma]$, the user generates N queries $Q_1^{[k]}(\gamma), \cdots, Q_N^{[k]}(\gamma)$, which are functions of previous queries and answers and $\mathbb{F}$,

$$H(Q_{1:N}^{[k]}(\gamma)|Q_{1:N}^{[k]}(1:\gamma-1), A_{1:N}^{[k]}(1:\gamma-1), \mathbb{F}) = 0 \tag{6.9}$$

The user sends query $Q_n^{[k]}(\gamma)$ to the n-th server, which generates an answer $A_n^{[k]}(\gamma)$ and returns $A_n^{[k]}(\gamma)$ to the user. The answer is a function of all queries received so far, the stored information S_n, and $\mathbb{G}$,

$$H(A_n^{[k]}(\gamma)|Q_n^{[k]}(1:\gamma), S_n, \mathbb{G}) = 0. \tag{6.10}$$

At the end of Γ rounds, from all the information that is now available to the user ($A_{1:N}^{[k]}(1:\Gamma), Q_{1:N}^{[k]}(1:\Gamma), \mathbb{F}$), the user decodes the desired message W_k according to a decoding rule that is specified by the PIR scheme. Let P_e denote the probability of error achieved with the specified decoding rule.

To protect the user's privacy, the K possible values of the desired message index should be indistinguishable from the perspective of any subset $\mathcal{T} \subset [1 : N]$ of at most T colluding servers, i.e., the following privacy constraint must be satisfied.

$$[T\text{-Privacy}] \quad (Q_{\mathcal{T}}^{[k]}(1:\Gamma), A_{\mathcal{T}}^{[k]}(1:\Gamma), \mathbb{G}, S_{\mathcal{T}}) \sim (Q_{\mathcal{T}}^{[k']}(1:\Gamma), A_{\mathcal{T}}^{[k']}(1:\Gamma), \mathbb{G}, S_{\mathcal{T}}) \quad \forall k, k' \in [1:K], \forall \mathcal{T} \subset [1:N], |\mathcal{T}| = T \tag{6.11}$$

The PIR rate characterizes how many bits of desired information are retrieved per downloaded bit and is defined as follows.

[2]One might wonder if the setting can be further generalized by allowing sequential queries, i.e., allowing the query to each server to depend not only on the answers received from previous rounds, but also on the answers received from other servers queried previously within the same round. We note that sequential queries are already contained in our multiround framework, e.g., by querying only one server in each round (sending null queries to the remaining servers).

$$R = L/D \tag{6.12}$$

where D is the expected value[3] of the total number of bits downloaded by the user from all the servers over all Γ rounds.

A rate R is said to be ϵ-error achievable if there exists a sequence of PIR schemes, indexed by L, each of rate greater than or equal to R, for which $P_e \to 0$ as $L \to \infty$. Note that for such a sequence of PIR schemes, from Fano's inequality we must have

$$\begin{aligned} \text{[Correctness]} \quad o(L) &= \frac{1}{L} H(W_k | A_{1:N}^{[k]}(1:\Gamma), Q_{1:N}^{[k]}(1:\Gamma), \mathbb{F}) \\ &\overset{(6.7)(6.9)}{=} \frac{1}{L} H(W_k | A_{1:N}^{[k]}(1:\Gamma), \mathbb{F}), \qquad \forall k \in [1:K] \end{aligned} \tag{6.13}$$

where $o(L)$ represents any term whose value approaches zero as L approaches infinity. The supremum of ϵ-error achievable rates is called the ϵ-error capacity C_ϵ.

A rate R is said to be zero-error achievable if there exists (for some L) a PIR scheme of rate greater than or equal to R for which $P_e = 0$. The supremum of zero-error achievable rates is called the zero-error capacity C_o. From the definitions, it is evident that

$$C_o \leq C_\epsilon \tag{6.14}$$

6.2 Results

There are two main contributions in this chapter, summarized in the following sections.

[3]Alternatively, D may be defined as the maximum download needed by the PIR scheme which (similar to choosing zero-error instead of ϵ-error) weakens the converse and strengthens the achievability arguments in general. The capacity characterizations in this chapter, as well as results in previous chapters hold under either definition. This is because in every case, the upper bounds allow average download D, while the achievability only requires maximum download D.

6.2.1 Capacity Perspective

We first consider the capacity benefits of multiple rounds of communication in the classical setting where each server stores all messages, i.e., storage is unconstrained. We present our result in the general context of multiround PIR with T-privacy constraints (MTPIR). The MTPIR setting is obtained from the general problem statement by relaxing the storage overhead constraints, i.e.,

$$\begin{aligned} S_n &= (W_1, W_2, \cdots, W_K), \forall n \in [1:N] \\ \alpha &= N \end{aligned}$$

i.e., each server stores all the messages (replication). The following theorem presents the main result.

Theorem 6.1. *The capacity of MTPIR*

$$C_o = C_\epsilon = \left(1 + T/N + T^2/N^2 + \cdots + T^{K-1}/N^{K-1}\right)^{-1}.$$

The converse proof of Theorem 6.1 is presented in Section 6.3. Achievability follows directly from Chapter 3. The following observations place the result in perspective.

1. The capacity of MTPIR matches the capacity of TPIR found in Chapter 3, i.e., multiple rounds do not increase capacity.

2. Setting $T = 1$ gives us the capacity of multiround PIR (MPIR) without T-privacy constraints. The capacity of MPIR matches the capacity of PIR found in Chapter 2, i.e., multiple rounds do not increase capacity.

3. Since the achievability proofs in Chapter 2 and Chapter 3 only require linear and zero-error schemes, there is no capacity benefit of multiple rounds over single-round schemes, non-linear over linear schemes, or ϵ-error over zero-error schemes.

4. For all N, K, T, Γ the converse proof of Theorem 6.1 generalizes the converse proofs in Chapter 2 and Chapter 3. Remarkably, it requires only Shannon information inequalities, i.e., sub-modularity of entropy.

6.2.2 Storage Overhead Perspective

As summarized above, our first result shows that in a broad sense – with or without colluding servers – there is no capacity benefit of multiple rounds over single-round communication, ϵ-error over zero-error schemes or non-linear over linear schemes for PIR. This pessimistic finding may lead one to believe that there is little reason to further explore interactive communication, non-linear schemes or ϵ-error schemes for PIR. As our main contribution in this section, we offer an optimistic counterpoint by looking at the PIR problem from the perspective of storage overhead instead of capacity. The counterpoint is made through a counterexample. The counterexample is quite remarkable in itself as it shows from a storage overhead perspective not only the advantage of a multiround PIR scheme over all single-round PIR schemes, but also of a non-linear PIR scheme over all linear PIR schemes, and an ϵ-error scheme over all zero-error schemes.

For a counterexample the simplest setting is typically the most interesting. Therefore, in this section we will only consider the simplest non-trivial setting, with $K = 2$ messages, $N = 2$ servers, and $T = 1$, i.e., no collusion among servers. Recall that for this setting the capacity is $C = 2/3$. For our counterexample we explore the minimum storage overhead that is needed to achieve the rate $2/3$.

Theorem 6.2. *For $K = 2, N = 2, T = 1$, and for rate $2/3$,*

1. *there exists a multiround, non-linear and ϵ-error PIR scheme with storage overhead*

$$\alpha = 3/4 + 3/8 \log_2 3$$

which is less than $3/2$.

2. the storage overhead of any single-round, linear and zero-error PIR scheme is

$$\alpha \geq 3/2$$

The achievability arguments, including the multiround, non-linear and ϵ-error PIR scheme that proves the first part of Theorem 6.2 are presented in this section. The proof of the second claim notably utilizes Ingleton's inequality, which goes beyond submodularity, and is presented in Section 6.4.

6.2.2.1 A multiround, non-linear and ϵ-error PIR scheme for $K = 2, N = 2, T = 1$

Define w_1, w_2 as two independent uniform binary random variables. Further, define

$$\begin{aligned} x_1 &= w_1 \wedge w_2 && (6.15)\\ x_2 &= (\sim w_1) \wedge (\sim w_2) && (6.16)\\ y_1 &= w_1 \wedge (\sim w_2) && (6.17)\\ y_2 &= (\sim w_1) \wedge w_2 && (6.18) \end{aligned}$$

where $\wedge$ and $\sim$ are the logical AND and NOT operators. Note the following,

$$\begin{aligned} x_1 = 1 &\Rightarrow (w_1, w_2) = (1, 1) && (6.19)\\ x_2 = 1 &\Rightarrow (w_1, w_2) = (0, 0) && (6.20)\\ x_1 = 0 &\Rightarrow (w_1, w_2) = (y_1, y_2) && (6.21)\\ x_2 = 0 &\Rightarrow (w_1, w_2) = (\sim y_2, \sim y_1) && (6.22) \end{aligned}$$

For ease of exposition, consider first the case where each message is only one bit long. In this case, the messages W_1, W_2, directly correspond to w_1, w_2, respectively. Denote the first server as Server 1 and the second server as Server 2. Regardless of whether the user desires W_1 or W_2, he flips a private fair coin, and requests the value of either x_1 or x_2 from Server 1. If the answer is 1, then according to (6.19) and (6.20) the user knows the values of both

w_1, w_2 and no further information is requested from Server 2. If the answer is 0, then the user proceeds as follows.

- If $x_1 = 0$ and W_1 is desired, ask Server 2 for the value of y_1. Retrieve $w_1 = y_1$.
- If $x_1 = 0$ and W_2 is desired, ask Server 2 for the value of y_2. Retrieve $w_2 = y_2$.
- If $x_2 = 0$ and W_1 is desired, ask Server 2 for the value of y_2. Retrieve $w_1 =\sim y_2$.
- If $x_2 = 0$ and W_2 is desired, ask Server 2 for the value of y_1. Retrieve $w_2 =\sim y_1$.

Note that in order to answer the user's queries, Server 1 only needs to store (x_1, x_2), and Server 2 only needs to store (y_1, y_2). This observation is the key to not only the reduced storage overhead, but also the enhanced privacy of this scheme.

Further, in preparation for the proofs that follow, let us define another binary random variable u, which takes the value $u = 0$ if no response is needed from Server 2, and the value $u = 1$ otherwise. Note that $u = 0$ implies that $(y_1, y_2) = (0, 0)$. On the other hand, if $u = 1$, then (y_1, y_2) takes the values $(0, 0), (1, 0), (0, 1)$, each with probability $1/3$. Therefore,

$$\begin{aligned} H(y_1, y_2|u) &= 1/4 \times H(y_1, y_2|u = 0) + 3/4 \times H(y_1, y_2|u = 1) && (6.23) \\ &= 1/4 \times 0 + 3/4 \times H(1/3, 1/3, 1/3) = 3/4 \log_2 3 && (6.24) \end{aligned}$$

The correctness of the scheme is obvious from (6.19)-(6.22). Let us verify that the scheme is private. Start with Server 1. The query to Server 1 is equally likely to be x_1 or x_2, regardless of the desired message index and the message realizations. Therefore, Server 1 learns nothing about which message is retrieved. Next consider Server 2. Let us prove that $(Q_2^{[1]}, y_1, y_2) \sim (Q_2^{[2]}, y_1, y_2)$.

$(\theta = 1)$ $(\theta = 2)$

$(Q_2^{[1]}, y_1, y_2)$	Prob.
$(\emptyset, 0, 0)$	$1/4$
$(\text{“}y_1\text{”}, 0, 0)$	$1/8$
$(\text{“}y_2\text{”}, 0, 0)$	$1/8$
$(\text{“}y_1\text{”}, 0, 1)$	$1/8$
$(\text{“}y_2\text{”}, 0, 1)$	$1/8$
$(\text{“}y_1\text{”}, 1, 0)$	$1/8$
$(\text{“}y_2\text{”}, 1, 0)$	$1/8$

$\sim$

$(Q_2^{[2]}, y_1, y_2)$	Prob.
$(\emptyset, 0, 0)$	$1/4$
$(\text{“}y_1\text{”}, 0, 0)$	$1/8$
$(\text{“}y_2\text{”}, 0, 0)$	$1/8$
$(\text{“}y_1\text{”}, 0, 1)$	$1/8$
$(\text{“}y_2\text{”}, 0, 1)$	$1/8$
$(\text{“}y_1\text{”}, 1, 0)$	$1/8$
$(\text{“}y_2\text{”}, 1, 0)$	$1/8$

where the double quote notation around a random variable represents the query about its realization. The computation of the joint distribution values is straightforward. We present the derivation here for one case. All other cases follow similarly. From the law of total probability, we have

$$\begin{aligned}
&\Pr\left((Q_2^{[1]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1)\right) \\
&= \Pr\left((Q_2^{[1]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1) | (Q_1^{[1]}, w_1, w_2) = (\text{“}x_1\text{”}, 0, 1)\right) \times \Pr\left((Q_1^{[1]}, w_1, w_2) = (\text{“}x_1\text{”}, 0, 1)\right) \\
&+ \Pr\left((Q_2^{[1]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1) | (Q_1^{[1]}, w_1, w_2) = (\text{“}x_2\text{”}, 0, 1)\right) \times \Pr\left((Q_1^{[1]}, w_1, w_2) = (\text{“}x_2\text{”}, 0, 1)\right) \qquad (6.25) \\
&= 1 \times 1/8 + 0 \times 1/8 = 1/8 \qquad (6.26)
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\Pr\left((Q_2^{[2]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1)\right) \\
&= \Pr\left((Q_2^{[2]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1) | (Q_1^{[2]}, w_1, w_2) = (\text{“}x_1\text{”}, 0, 1)\right) \times \Pr\left((Q_1^{[2]}, w_1, w_2) = (\text{“}x_1\text{”}, 0, 1)\right) \\
&+ \Pr\left((Q_2^{[2]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1) | (Q_1^{[2]}, w_1, w_2) = (\text{“}x_2\text{”}, 0, 1)\right) \times \Pr\left((Q_1^{[2]}, w_1, w_2) = (\text{“}x_2\text{”}, 0, 1)\right) \qquad (6.27) \\
&= 0 \times 1/8 + 1 \times 1/8 = 1/8 \qquad (6.28)
\end{aligned}$$

Thus, $\Pr\left((Q_2^{[1]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1)\right) = \Pr\left((Q_2^{[1]}, y_1, y_2) = (\text{“}y_1\text{”}, 0, 1)\right)$. All other cases are verified similarly. Then, since the distribution of $(Q_2^{[\theta]}, y_1, y_2)$ does not depend on θ, and the answers are only deterministic functions of the query and the stored information, it follows that the scheme is private.

Next consider the L length extension of this PIR scheme, where each desired bit is retrieved independently as described above. Under the L length extension, $W_1, W_2, X_1, X_2, Y_1, Y_2, U$ are sequences of length L, such that the sequence of tuples $[(W_1(l), W_2(l),\ X_1(l), X_2(l), Y_1(l), Y_2(l), U(l))]_{l=1}^{L}$ is i.i.d. $\sim (w_1, w_2, x_1, x_2, y_1, y_2, u)$. Since the extension is obtained by repeated independent applications of the PIR scheme described above for retrieving each message bit, it follows trivially that the extended PIR scheme is also correct and private. The purpose for the L length extension, with $L \to \infty$, is to invoke fundamental limits of data compression which optimize both the data rates and the storage overhead as explained next.

Let us show that the rate $2/3$ is achieved asymptotically as $L \to \infty$. We take advantage of the fact that the answers from the servers are not uniformly distributed, and therefore the sequence of answers from each server is compressible. With optimal compression, the user downloads $H(1/4, 3/4)$ bits per desired message bit from Server 1. This is because, for each retrieved bit, the answer from Server 1 takes the value 1 with probability $1/4$ and 0 with probability $3/4$. From Server 2, we download $1/4 \times 0 + 3/4 \times H(1/3, 2/3) = 3/4H(1/3, 2/3)$ bits per desired message bit, because with probability $1/4$ (when the answer from Server 1 is 1), no response is requested from Server 2 and otherwise within the remaining space of probability measure $3/4$ (when the answer from Server 1 is 0), the answer from Server 2 is 1 with conditional probability $1/3$ and 0 with conditional probability $2/3$. Therefore the total download is $H(1/4, 3/4) + 3/4H(1/3, 2/3) = 3/2$ bits per desired message bit and the rate achieved is $2/3$.

Next let us determine the storage requirements of this scheme. Server 1 needs (X_1, X_2) to answer the user's queries, so with optimal compression, it needs to store $H(x_1, x_2) = H(1/4, 1/4, 1/2) = 3/2$ bits per desired message bit. One might naively imagine that the same storage requirement also applies to Server 2, because Server 2 similarly needs the values (Y_1, Y_2) to answer the user's queries. However, this is not true, because the query sent to

Server 2 already contains some information about the message realizations,[4] and this *side-information* allows Server 2 to reduce its storage requirement by taking advantage of Slepian Wolf coding [63, 25] (distributed compression with decoder side information).

The key is to realize that Server 2 does not need to know (Y_1, Y_2) until after it receives the query from the user. The query from the user includes U as side information. Therefore, using Slepian Wolf coding, Server 2 is able to optimally compress the i.i.d. sequence (Y_1, Y_2) to the conditional entropy $H(y_1, y_2|u)$ bits per desired message bit and still decode the (Y_1, Y_2) sequence when it is needed, i.e., after the query is provided by the user. Thus, the total storage required by this PIR scheme is $3/2 + 3/4 \log_2 3$ bits per bit of desired message. Since there are two messages, the storage overhead is $3/4 + 3/8 \log_2 3$.

The following observations are useful to place the new PIR scheme in perspective.

1. The optimal compression guarantees are only available in the ϵ-error sense. Therefore, this PIR scheme is essentially an ϵ-error scheme.

2. The multiround scheme is in fact a sequential PIR scheme that utilizes only one round of queries for each server (two rounds total since there are two servers).

3. The scheme is essentially non-linear because, e.g., the logical AND operator is non-linear.

4. Since the multiround, non-linear and ϵ-error aspects are all essential for *this* scheme to create an advantage in terms of storage overhead, it is an intriguing question whether all three aspects are necessary in *general* or if it is possible to achieve storage overhead less than 3/2 through another scheme while sacrificing at least one of the three aspects.

5. A key insight from this PIR scheme is the surprising privacy benefit of storage overhead optimization. By not storing all the information at each server, and by optimally compressing the stored information, not only do we reduce the storage overhead, but

[4]Note that the query sent to Server 2 is independent of the desired message index but not the message realizations.

also we enable stronger privacy guarantees than would hold otherwise. Note that if each server stores all the information (both W_1 and W_2), then the scheme is not private. To see this, suppose $(w_1, w_2) = (1, 1)$. This would be known to Server 2 because it stores both messages. Under this circumstance, Server 2 knows that if the user asks for y_2, then his desired message must be W_1 and if the user asks for y_1 then his desired message must be W_2. Thus, storing all the information at each server would result in loss of privacy. As another example, we note that if the data is not in its optimally compressed form, i.e., w_1 and/or w_2 are not uniformly distributed then again the PIR scheme would lose privacy. To see this, suppose $\Pr(w_1 = 1)$=$\Pr(w_2 = 1) > 1/2$. Then Server 2 is more likely to be asked for y_1 if the desired message is W_2 than if the desired message is W_1. On the other hand, note that optimal data compression is a pre-requisite in any case for the optimization of rate and storage overhead.[5]

6. Let us consider momentarily the restricted message size setting, where each message is only $L = 1$ bit long. Then it is easy to see that any single-round scheme (all queries generated simultaneously) must download at least 2 bits on average, but our multiround scheme requires an expected download of only $1 + 3/4 = 7/4$ bits. Thus, even though the download advantage of multiround PIR disappears under unconstrained message lengths, for constrained message lengths there are benefits of multiround PIR.

6.2.2.2 A single-round, linear and zero-error scheme for $K = 2, N = 2, T = 1$

For comparison, the corresponding scheme from Chapter 2 which also achieves rate 2/3 is reproduced below. This will be shown to be the optimal single-round, linear, zero-error scheme for storage overhead in Section 6.4. Denote the messages, each comprised of 4 bits, as $W_1 = (a_1, a_2, a_3, a_4), W_2 = (b_1, b_2, b_3, b_4)$. The downloaded information from each server is shown below.

[5]Since optimal compression limits are typically achieved asymptotically, if the data is not assumed to be uniform a-priori, then as noted by [11, 12] the privacy guarantees would also be subject to ϵ-leakage that approaches zero as message length approaches infinity.

	Prob. 1/2		Prob. 1/2	
	Want W_1	Want W_2	Want W_1	Want W_2
Server 1	$a_1, b_1, a_2 + b_2$	$a_1, b_1, a_2 + b_2$	$a_3, b_3, a_4 + b_4$	$a_3, b_3, a_4 + b_4$
Server 2	$a_4, b_2, a_3 + b_1$	$a_2, b_4, a_1 + b_3$	$a_2, b_4, a_1 + b_3$	$a_4, b_2, a_3 + b_1$

The scheme achieves rate 2/3 and is linear, single-round, and zero-error. A total of 6 bits are stored at each server

$$S_1 = (a_1, a_3, b_1, b_3, a_2 + b_2, a_4 + b_4) \tag{6.29}$$
$$S_2 = (a_2, a_4, b_2, b_4, a_3 + b_1, a_1 + b_3) \tag{6.30}$$

Thus, the storage overhead is 3/2.

6.3 Proof of Theorem 6.1

We first present two useful lemmas. Note that in the proofs, the relevant equations needed to justify each step are specified by the equation numbers set on top of the (in)equality symbols.

Lemma 6.1. *For all $k \in [2 : K]$,*

$$\begin{aligned} &I(W_{k:K}; Q_{1:N}^{[k-1]}(1:\Gamma), A_{1:N}^{[k-1]}(1:\Gamma), \mathbb{F}|W_{1:k-1}, \mathbb{G}) \\ &\quad \geq \frac{T}{N} I(W_{k+1:K}; Q_{1:N}^{[k]}(1:\Gamma), A_{1:N}^{[k]}(1:\Gamma), \mathbb{F}|W_{1:k}, \mathbb{G}) + \frac{LT}{N}(1 - o(L)). \end{aligned} \tag{6.31}$$

Proof:

$$\begin{aligned} &NI(W_{k:K}; Q_{1:N}^{[k-1]}(1:\Gamma), A_{1:N}^{[k-1]}(1:\Gamma), \mathbb{F}|W_{1:k-1}, \mathbb{G}) \\ &\quad \geq \frac{N}{\binom{N}{T}} \sum_{\mathcal{T} \subset [1:N], |\mathcal{T}|=T} I(W_{k:K}; Q_{\mathcal{T}}^{[k-1]}(1:\Gamma), A_{\mathcal{T}}^{[k-1]}(1:\Gamma)|W_{1:k-1}, \mathbb{G}) \qquad (6.32) \\ &\quad \overset{(6.11)}{=} \frac{N}{\binom{N}{T}} \sum_{\mathcal{T} \subset [1:N], |\mathcal{T}|=T} I(W_{k:K}; Q_{\mathcal{T}}^{[k]}(1:\Gamma), A_{\mathcal{T}}^{[k]}(1:\Gamma)|W_{1:k-1}, \mathbb{G}) \qquad (6.33) \end{aligned}$$

$$
\begin{aligned}
&= \frac{N}{\binom{N}{T}} \sum_{\mathcal{T}\subset[1:N],|\mathcal{T}|=T} \sum_{\gamma=1}^{\Gamma} I(W_{k:K}; Q_{\mathcal{T}}^{[k]}(\gamma), A_{\mathcal{T}}^{[k]}(\gamma) | Q_{\mathcal{T}}^{[k]}(1:\gamma-1), A_{\mathcal{T}}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{G}) \\
&\geq \frac{N}{\binom{N}{T}} \sum_{\mathcal{T}\subset[1:N],|\mathcal{T}|=T} \sum_{\gamma=1}^{\Gamma} I(W_{k:K}; A_{\mathcal{T}}^{[k]}(\gamma) | Q_{\mathcal{T}}^{[k]}(1:\gamma), A_{\mathcal{T}}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{G}) \qquad (6.34) \\
&\overset{(6.8)(6.10)}{=} \frac{N}{\binom{N}{T}} \sum_{\mathcal{T}\subset[1:N],|\mathcal{T}|=T} \sum_{\gamma=1}^{\Gamma} H(A_{\mathcal{T}}^{[k]}(\gamma) | Q_{\mathcal{T}}^{[k]}(1:\gamma), A_{\mathcal{T}}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{G}) \qquad (6.35) \\
&\geq \frac{N}{\binom{N}{T}} \sum_{\mathcal{T}\subset[1:N],|\mathcal{T}|=T} \sum_{\gamma=1}^{\Gamma} H(A_{\mathcal{T}}^{[k]}(\gamma) | Q_{1:N}^{[k]}(1:\gamma), A_{1:N}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{F}, \mathbb{G}) \qquad (6.36) \\
&\geq T \sum_{\gamma=1}^{\Gamma} H(A_{1:N}^{[k]}(\gamma) | Q_{1:N}^{[k]}(1:\gamma), A_{1:N}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{F}, \mathbb{G}) \quad \text{(Han's inequality)} \qquad (6.37) \\
&\overset{(6.8)(6.10)}{=} T \sum_{\gamma=1}^{\Gamma} I(W_{k:K}; A_{1:N}^{[k]}(\gamma) | Q_{1:N}^{[k]}(1:\gamma), A_{1:N}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{F}, \mathbb{G}) \qquad (6.38) \\
&\overset{(6.7)(6.9)}{=} T \sum_{\gamma=1}^{\Gamma} I(W_{k:K}; Q_{1:N}^{[k]}(\gamma), A_{1:N}^{[k]}(\gamma) | Q_{1:N}^{[k]}(1:\gamma-1), A_{1:N}^{[k]}(1:\gamma-1), W_{1:k-1}, \mathbb{F}, \mathbb{G}) \qquad (6.39) \\
&= TI(W_{k:K}; Q_{1:N}^{[k]}(1:\Gamma), A_{1:N}^{[k]}(1:\Gamma) | W_{1:k-1}, \mathbb{F}, \mathbb{G}) \qquad (6.40) \\
&\overset{(6.13)}{=} TI(W_{k:K}; W_k, Q_{1:N}^{[k]}(1:\Gamma), A_{1:N}^{[k]}(1:\Gamma) | W_{1:k-1}, \mathbb{F}, \mathbb{G}) - o(L)LT \qquad (6.41) \\
&= TI(W_{k:K}; W_k | W_{1:k-1}, \mathbb{F}, \mathbb{G}) - o(L)LT \\
&\quad + TI(W_{k+1:K}; Q_{1:N}^{[k]}(1:\Gamma), A_{1:N}^{[k]}(1:\Gamma) | W_{1:k}, \mathbb{F}, \mathbb{G}) \qquad (6.42) \\
&\overset{(6.6)}{=} LT(1-o(L)) + TI(W_{k+1:K}; Q_{1:N}^{[k]}(1:\Gamma), A_{1:N}^{[k]}(1:\Gamma) | W_{1:k}, \mathbb{F}, \mathbb{G}) \qquad (6.43) \\
&\overset{(6.6)}{=} LT(1-o(L)) + TI(W_{k+1:K}; Q_{1:N}^{[k]}(1:\Gamma), A_{1:N}^{[k]}(1:\Gamma), \mathbb{F} | W_{1:k}, \mathbb{G}) \qquad (6.44)
\end{aligned}
$$

■

Lemma 6.2.

$$
I(W_{2:K}; Q_{1:N}^{[1]}(1:\Gamma), A_{1:N}^{[1]}(1:\Gamma), \mathbb{F} | W_1, \mathbb{G}) \leq L(1/R - 1 + o(L)). \qquad (6.45)
$$

Proof:

$$
\begin{aligned}
&I(W_{2:K}; Q_{1:N}^{[1]}(1:\Gamma), A_{1:N}^{[1]}(1:\Gamma), \mathbb{F} | W_1, \mathbb{G}) \\
&\overset{(6.6)}{=} I(W_{2:K}; Q_{1:N}^{[1]}(1:\Gamma), A_{1:N}^{[1]}(1:\Gamma), W_1, \mathbb{F}, \mathbb{G}) \qquad (6.46) \\
&\overset{(6.7)(6.9)}{=} I(W_{2:K}; A_{1:N}^{[1]}(1:\Gamma), W_1, \mathbb{F}, \mathbb{G}) \qquad (6.47) \\
&= I(W_{2:K}; A_{1:N}^{[1]}(1:\Gamma), \mathbb{F}, \mathbb{G}) + I(W_{2:K}; W_1 | A_{1:N}^{[1]}(1:\Gamma), \mathbb{F}, \mathbb{G}) \qquad (6.48)
\end{aligned}
$$

$$\overset{(6.6)(6.13)}{=} I(W_{2:K}; A^{[1]}_{1:N}(1:\Gamma)|\mathbb{F},\mathbb{G}) + o(L)L \tag{6.49}$$
$$= H(A^{[1]}_{1:N}(1:\Gamma)|\mathbb{F},\mathbb{G}) - H(A^{[1]}_{1:N}(1:\Gamma)|\mathbb{F},\mathbb{G},W_{2:K}) + o(L)L \tag{6.50}$$
$$\overset{(6.12)}{\leq} L/R - H(A^{[1]}_{1:N}(1:\Gamma)|\mathbb{F},\mathbb{G},W_{2:K}) + o(L)L \tag{6.51}$$
$$\overset{(6.13)}{=} L/R - H(W_1, A^{[1]}_{1:N}(1:\Gamma)|\mathbb{F},\mathbb{G},W_{2:K}) + o(L)L \tag{6.52}$$
$$\leq L/R - H(W_1|\mathbb{F},\mathbb{G},W_{2:K}) + o(L)L \tag{6.53}$$
$$\overset{(6.6)}{=} L/R - L + o(L)L = L(1/R - 1 + o(L)) \tag{6.54}$$

■

With Lemma 6.1 and Lemma 6.2, we are ready to prove the converse.

Rate Outerbound

Starting from $k = 2$ and applying (6.31) repeatedly for $k \in [3:K]$,

$$\begin{aligned}
&I(W_{2:K}; Q^{[1]}_{1:N}(1:\Gamma), A^{[1]}_{1:N}(1:\Gamma), \mathbb{F}|W_1, \mathbb{G}) \\
&\overset{(6.31)}{\geq} \frac{T}{N} I(W_{3:K}; Q^{[2]}_{1:N}(1:\Gamma), A^{[2]}_{1:N}(1:\Gamma), \mathbb{F}|W_1, W_2, \mathbb{G}) + \frac{LT(1-o(L))}{N} \\
&\overset{(6.31)}{\geq} \cdots \\
&\overset{(6.31)}{\geq} \frac{T^{K-2}}{N^{K-2}} I(W_K; Q^{[K-1]}_{1:N}(1:\Gamma), A^{[K-1]}_{1:N}(1:\Gamma), \mathbb{F}|W_{1:K-1}, \mathbb{G}) \\
&\qquad + \frac{LT(1-o(L))}{N} + \cdots + \frac{LT^{K-2}(1-o(L))}{N^{K-2}} \\
&\overset{(6.31)}{\geq} \frac{T^{K-2}}{N^{K-2}} \frac{LT(1-o(L))}{N} + \frac{LT(1-o(L))}{N} + \cdots + \frac{LT^{K-2}(1-o(L))}{N^{K-2}} \qquad (6.55)(6.56) \\
&= L(1-o(L))(T/N + \cdots + T^{K-1}/N^{K-1}) \qquad (6.57)
\end{aligned}$$

Combining (6.57) and (6.45), we have

$$L(1/R - 1 + o(L)) \geq L(1-o(L))(T/N + \cdots + T^{K-1}/N^{K-1}) \tag{6.58}$$

Normalizing by L and letting L go to infinity gives us

$$1/R - 1 \geq T/N + \cdots + T^{K-1}/N^{K-1} \tag{6.59}$$
$$\Rightarrow R \leq (1 + T/N + \cdots + T^{K-1}/N^{K-1})^{-1} \tag{6.60}$$

thus, the proof is complete.

6.4 Proof of Theorem 6.2 – Statement 2.

We show that when $K = 2, N = 2, T = 1, \Gamma = 1$ and the rate equals $2/3$, the storage overhead of all zero-error, linear, and single-round PIR schemes is no less than $3/2$. Since we only consider single-round schemes in this section, we will simplify the notation, e.g., instead of $Q_2^{[1]}(1)$ we write simply $Q_2^{[1]}$. In addition, without loss of generality, let us make the following simplifying assumptions.

1. We assume that the PIR scheme is symmetric, in that

$$H(A_1^{[1]}|\mathbb{F},\mathbb{G}) = H(A_2^{[1]}|\mathbb{F},\mathbb{G}) = H(A_2^{[2]}|\mathbb{F},\mathbb{G}) \tag{6.61}$$
$$H(S_1) = H(S_2) \tag{6.62}$$

   Given any (asymmetric) PIR scheme that retrieves messages of size L, a symmetric PIR scheme with the same rate and storage overhead that retrieves messages of size NL is obtained by repeating the original scheme N times, and in the n-th repetition shifting the server indices cyclically by n. This symmetrization process is described in Theorem 6.3 (see Section 6.4.1).

2. We assume that $Q_1^{[1]} = Q_1^{[2]}$, i.e., the query for the first server is chosen without the knowledge of the desired message index. There is no loss of generality in this assumption because of the privacy constraint, which requires that $Q_1^{[\theta]}$ is independent of θ.[6] Note that this also means that $A_1^{[1]} = A_1^{[2]}$.

[6]Note that instead of $Q_1^{[1]} = Q_1^{[2]}$, we could equivalently assume that $Q_2^{[1]} = Q_2^{[2]}$ without of loss of generality (because privacy also requires that $Q_2^{[\theta]}$ is independent of θ). However, if we simultaneously assume both $Q_1^{[1]} = Q_1^{[2]}$ *and* $Q_2^{[1]} = Q_2^{[2]}$, then there is a loss of generality because together $(Q_1^{[\theta]}, Q_2^{[\theta]})$ is *not* required to be independent of θ by the privacy constraint.

Our goal is to prove a lower bound on the storage overhead. Since the PIR scheme is symmetric by assumption, the storage overhead is $(H(S_1) + H(S_2))/2L = H(S_2)/L$. Furthermore, $H(S_2) \geq H(A_2^{[1]}, A_2^{[2]}|\mathbb{F}, \mathbb{G})$, so we will prove a lower bound on $H(A_2^{[1]}, A_2^{[2]}|\mathbb{F}, \mathbb{G})$ instead. Let us start with a useful lemma that holds for all linear and non-linear schemes.

Lemma 6.3.

$$H(A_1^{[1]}|W_1, \mathbb{F}, \mathbb{G}) = H(A_2^{[2]}|W_1, \mathbb{F}, \mathbb{G}) = H(A_2^{[2]}|W_2, \mathbb{F}, \mathbb{G}) = L/2 \tag{6.63}$$

$$H(A_2^{[2]}|W_1, A_2^{[1]}, \mathbb{F}, \mathbb{G}) = H(A_2^{[2]}|W_2, A_2^{[1]}, \mathbb{F}, \mathbb{G}) = L/2 \tag{6.64}$$

Proof: We prove (6.63) first. On the one hand, after substituting[7] $R = 2/3$ in Lemma 6.2, from (6.47) we have

$$L/2 \geq I(W_2; A_1^{[1]}, A_2^{[1]}, W_1, \mathbb{F}, \mathbb{G}) \tag{6.65}$$

$$\overset{(6.6)}{=} I(W_2; A_1^{[1]}, A_2^{[1]}|W_1, \mathbb{F}, \mathbb{G}) \tag{6.66}$$

$$\overset{(6.7)(6.8)(6.4)}{=} H(A_1^{[1]}, A_2^{[1]}|W_1, \mathbb{F}, \mathbb{G}) \tag{6.67}$$

$$\Rightarrow L/2 \geq H(A_1^{[1]}|W_1, \mathbb{F}, \mathbb{G}) \tag{6.68}$$

$$\text{and } L/2 \geq H(A_2^{[1]}|W_1, \mathbb{F}, \mathbb{G}) \tag{6.69}$$

On the other hand, from (6.32) in Lemma 6.1, we have

$$L \leq I(W_2; Q_1^{[1]}, A_1^{[1]}|W_1, \mathbb{G}) + I(W_2; Q_2^{[1]}, A_2^{[1]}|W_1, \mathbb{G}) \tag{6.70}$$

$$\leq I(W_2; Q_1^{[1]}, A_1^{[1]}, \mathbb{F}|W_1, \mathbb{G}) + I(W_2; Q_2^{[1]}, A_2^{[1]}, \mathbb{F}|W_1, \mathbb{G}) \tag{6.71}$$

$$\overset{(6.6)}{=} I(W_2; Q_1^{[1]}, A_1^{[1]}|W_1, \mathbb{F}, \mathbb{G}) + I(W_2; Q_2^{[1]}, A_2^{[1]}|W_1, \mathbb{F}, \mathbb{G}) \tag{6.72}$$

$$\overset{(6.7)(6.8)(6.4)}{=} H(A_1^{[1]}|W_1, \mathbb{F}, \mathbb{G}) + H(A_2^{[1]}|W_1, \mathbb{F}, \mathbb{G}) \tag{6.73}$$

Combining (6.68), (6.69) and (6.73), we have shown that

$$H(A_1^{[1]}|W_1, \mathbb{F}, \mathbb{G}) = H(A_2^{[1]}|W_1, \mathbb{F}, \mathbb{G}) = L/2 \tag{6.74}$$

[7]Since we are considering only zero-error schemes, the $o(L)$ term in Lemma 6.2 is exactly 0.

Symmetrically, it follows that $H(A_2^{[2]}|W_2,\mathbb{F},\mathbb{G}) = L/2$. We are left to prove $H(A_2^{[2]}|W_1,\mathbb{F},\mathbb{G}) = L/2$. On the one hand, from (6.68) and (6.69), we have

$$\begin{aligned}
L/2 &\geq H(A_1^{[1]}|W_1,\mathbb{F},\mathbb{G}) = H(A_1^{[2]}|W_1,\mathbb{F},\mathbb{G}) \quad (\text{Using } A_1^{[1]} = A_1^{[2]}) && (6.75)\\
L/2 &\geq H(A_2^{[1]}|W_1,\mathbb{F},\mathbb{G}) && (6.76)\\
&\overset{(6.7)}{=} H(A_2^{[1]}|W_1,Q_2^{[1]},\mathbb{F},\mathbb{G}) && (6.77)\\
&= H(A_2^{[1]}|W_1,Q_2^{[1]},\mathbb{G}) && (6.78)\\
&= H(A_2^{[2]}|W_1,Q_2^{[2]},\mathbb{G}) && (6.79)\\
&= H(A_2^{[2]}|W_1,Q_2^{[2]},\mathbb{F},\mathbb{G}) && (6.80)\\
&\overset{(6.7)}{=} H(A_2^{[2]}|W_1,\mathbb{F},\mathbb{G}) && (6.81)
\end{aligned}$$

where (6.79) follows from the fact that for single-round PIR, the desired message index is independent of the messages, queries and answers, i.e., from (6.6), we have

$$\begin{aligned}
& I(\theta;W_1,W_2,\mathbb{F},\mathbb{G}) = 0 && (6.82)\\
\overset{(6.7)}{\Longrightarrow}\ & I(\theta;W_1,W_2,\mathbb{F},\mathbb{G},Q_2^{[\theta]}) = 0 && (6.83)\\
\overset{(6.8)(6.4)}{\Longrightarrow}\ & I(\theta;W_1,W_2,\mathbb{F},\mathbb{G},Q_2^{[\theta]},A_2^{[\theta]}) = 0 && (6.84)\\
\Longrightarrow\ & A_2^{[1]},W_1,Q_2^{[1]},\mathbb{G} \sim A_2^{[2]},W_1,Q_2^{[2]},\mathbb{G} && (6.85)
\end{aligned}$$

(6.78) and (6.80) are due to the Markov chain $\mathbb{F} - (W_1,Q_2^{[k]},\mathbb{G}) - A_2^{[k]}, k = 1,2$, which is proved as follows.

$$\begin{aligned}
I(A_2^{[k]};\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G}) &\leq I(A_2^{[k]},S_2;\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G}) && (6.86)\\
&= I(S_2;\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G}) + I(A_2^{[k]};\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G},S_2) && (6.87)\\
&\overset{(6.8)}{=} I(S_2;\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G}) && (6.88)\\
&\leq I(S_2,W_2;\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G}) && (6.89)\\
&= I(W_2;\mathbb{F}|W_1,Q_2^{[k]},\mathbb{G}) + I(S_2;\mathbb{F}|Q_2^{[k]},\mathbb{G},W_1,W_2) && (6.90)\\
&\overset{(6.4)}{\leq} I(W_2;\mathbb{F},W_1,Q_2^{[k]},\mathbb{G}) && (6.91)\\
&\overset{(6.7)(6.6)}{=} 0 && (6.92)
\end{aligned}$$

On the other hand, from (6.70), we have

$$
\begin{aligned}
L \quad & \leq \quad I(W_2; Q_1^{[1]}, A_1^{[1]} | W_1, \mathbb{G}) + I(W_2; Q_2^{[1]}, A_2^{[1]} | W_1, \mathbb{G}) && (6.93)\\
& \overset{(6.11)}{=} \quad I(W_2; Q_1^{[2]}, A_1^{[2]} | W_1, \mathbb{G}) + I(W_2; Q_2^{[2]}, A_2^{[2]} | W_1, \mathbb{G}) && (6.94)\\
& \leq \quad I(W_2; Q_1^{[2]}, A_1^{[2]}, \mathbb{F} | W_1, \mathbb{G}) + I(W_2; Q_2^{[2]}, A_2^{[2]}, \mathbb{F} | W_1, \mathbb{G}) && (6.95)\\
& \overset{(6.6)}{=} \quad I(W_2; Q_1^{[2]}, A_1^{[2]} | W_1, \mathbb{F}, \mathbb{G}) + I(W_2; Q_2^{[2]}, A_2^{[2]} | W_1, \mathbb{F}, \mathbb{G}) && (6.96)\\
& \overset{(6.7)(6.8)(6.4)}{=} \quad H(A_1^{[2]} | W_1, \mathbb{F}, \mathbb{G}) + H(A_2^{[2]} | W_1, \mathbb{F}, \mathbb{G}) && (6.97)
\end{aligned}
$$

Combining (6.75), (6.81) and (6.97), we have shown that $H(A_2^{[2]}|W_1, \mathbb{F}, \mathbb{G}) = L/2$. The proof of (6.63) is complete.

Next we prove (6.64). On the one hand,

$$
H(A_2^{[2]}|W_1, A_2^{[1]}, \mathbb{F}, \mathbb{G}) \leq H(A_2^{[2]}|W_1, \mathbb{F}, \mathbb{G}) \overset{(6.63)}{=} L/2 \tag{6.98}
$$

On the other hand, from sub-modularity of entropy functions we have

$$
\begin{aligned}
& H(A_2^{[2]}, A_2^{[1]} | W_1, \mathbb{F}, \mathbb{G}) \\
& \geq \quad -H(A_2^{[1]}, A_1^{[1]} | W_1, \mathbb{F}, \mathbb{G}) + H(A_1^{[1]}, A_2^{[2]}, A_2^{[1]} | W_1, \mathbb{F}, \mathbb{G}) + H(A_2^{[1]} | W_1, \mathbb{F}, \mathbb{G}) && (6.99)\\
& \geq \quad -L/2 + H(A_1^{[1]}, A_2^{[2]}, A_2^{[1]}, W_2 | W_1, \mathbb{F}, \mathbb{G}) + L/2 \qquad \text{from (6.67)(6.13)(6.74)} && (6.100)\\
& \geq \quad H(W_2 | W_1, \mathbb{F}, \mathbb{G}) \overset{(6.6)}{=} L && (6.101)\\
& \Rightarrow \quad H(A_2^{[2]} | W_1, A_2^{[1]}, \mathbb{F}, \mathbb{G}) = H(A_2^{[2]}, A_2^{[1]} | W_1, \mathbb{F}, \mathbb{G}) - H(A_2^{[1]} | W_1, \mathbb{F}, \mathbb{G}) \overset{(6.74)}{\geq} L/2 && (6.102)
\end{aligned}
$$

Note that the second term of (6.100) follows from the assumption that $A_1^{[1]} = A_1^{[2]}$ so that from $A_1^{[1]}, A_2^{[2]}$, we can decode W_2 just as from $A_1^{[2]}, A_2^{[2]}$, we can decode W_2. Combining (6.98), (6.102), we have proved $H(A_2^{[2]}|W_1, A_2^{[1]}, \mathbb{F}, \mathbb{G}) = L/2$. Symmetrically, it follows that $H(A_2^{[2]}|W_2, A_2^{[1]}, \mathbb{F}, \mathbb{G}) = L/2$. Therefore, the desired inequality (6.64) is obtained. ■

From Lemma 6.3, we know that $I(A_2^{[1]}; A_2^{[2]}|W_1, \mathbb{F}, \mathbb{G}) = I(A_2^{[1]}; A_2^{[2]}|W_2, \mathbb{F}, \mathbb{G}) = 0$. Plugging in Ingleton's inequality [38] that holds for linear schemes but not for non-linear schemes, we have

$$
\begin{aligned}
I(A_2^{[1]}; A_2^{[2]}|\mathbb{F},\mathbb{G}) &\leq I(A_2^{[1]}; A_2^{[2]}|W_1,\mathbb{F},\mathbb{G}) + I(A_2^{[1]}; A_2^{[2]}|W_2,\mathbb{F},\mathbb{G}) + \underbrace{I(W_1;W_2|\mathbb{F},\mathbb{G})}_{=0,\text{ from (6.6)}} \\
&= 0 && (6.103)\\
\Rightarrow H(A_2^{[1]}, A_2^{[2]}|\mathbb{F},\mathbb{G}) &= H(A_2^{[1]}|\mathbb{F},\mathbb{G}) + H(A_2^{[2]}|\mathbb{F},\mathbb{G}) && (6.104)\\
&\overset{(6.61)}{=} H(A_2^{[1]}|\mathbb{F},\mathbb{G}) + H(A_1^{[1]}|\mathbb{F},\mathbb{G}) && (6.105)\\
&\geq H(A_1^{[1]}, A_2^{[1]}|\mathbb{F},\mathbb{G}) && (6.106)\\
&\overset{(6.13)}{=} H(W_1, A_1^{[1]}, A_2^{[1]}|\mathbb{F},\mathbb{G}) && (6.107)\\
&= H(W_1|\mathbb{F},\mathbb{G}) + H(A_1^{[1]}, A_2^{[1]}|W_1,\mathbb{F},\mathbb{G}) && (6.108)\\
&\overset{(6.6)}{\geq} L + H(A_1^{[1]}|W_1,\mathbb{F},\mathbb{G}) \overset{(6.63)}{=} 3L/2 && (6.109)\\
\Rightarrow \alpha = H(S_2)/L &\geq H(A_2^{[1]}, A_2^{[2]}|\mathbb{F},\mathbb{G})/L \geq 3/2 && (6.110)
\end{aligned}
$$

Remark: The above converse uses Ingleton's inequality. It turns out that the best storage overhead bound from Shannon inequalities is $5/4$*, which can be tightened to* $4/3$ *with Zhang-Yeung non-Shannon type inequality.*

6.4.1 Symmetrization

Theorem 6.3. *Consider the single-round PIR problem with* $K = 2$ *messages and* $N = 2$ *servers. Suppose we have a scheme described by* $\bar{L}, \bar{W}_1, \bar{W}_2, \bar{S}_1, \bar{S}_2, \bar{Q}_{1:2}^{[1]}, \bar{Q}_{1:2}^{[2]}, \bar{A}_{1:2}^{[1]}, \bar{A}_{1:2}^{[2]}, \bar{\mathbb{F}}, \bar{\mathbb{G}}$*. Then we can construct a symmetric PIR scheme, also for* $K = N = 2$*, described by* $L, W_1, W_2, S_1, S_2, Q_{1:2}^{[1]}, Q_{1:2}^{[2]}, A_{1:2}^{[1]}, A_{1:2}^{[2]}, \mathbb{F}, \mathbb{G}$ *such that*

$$
\begin{aligned}
H(A_1^{[1]}|\mathbb{F},\mathbb{G}) &= H(A_2^{[1]}|\mathbb{F},\mathbb{G}) = H(A_2^{[2]}|\mathbb{F},\mathbb{G}) && (6.111)\\
H(S_1) &= H(S_2) && (6.112)\\
L &= 2\bar{L} && (6.113)
\end{aligned}
$$

such that the symmetric PIR scheme has the same rate and storage overhead as the original PIR scheme.

Proof: Consider two independent implementations of the asymmetric PIR scheme. Let us use the 'bar' notation for the first implementation and the 'tilde' notation for the second

implementation. In the first implementation, there are two messages $\bar{W}_1, \bar{W}_2$, each of length $\bar{L}$, two servers Server $\bar{1}$ and Server $\bar{2}$ which store $\bar{S}_1, \bar{S}_2$, respectively. In the second implementation, there are two messages $\tilde{W}_1, \tilde{W}_2$, each of length $\tilde{L} = \bar{L}$, two servers Server $\tilde{2}$ and Server $\tilde{1}$ which store $\tilde{S}_1, \tilde{S}_2$, respectively. Note the critical detail that the server indices are switched in the second implementation. The asymmetric PIR scheme specifies the queries for each implementation such that the user can privately retrieve an arbitrarily chosen message from each implementation.

The symmetric PIR scheme is created by combining the two implementations. In the combined scheme, there are two messages $W_1 = (\bar{W}_1, \tilde{W}_1)$ and $W_2 = (\bar{W}_2, \tilde{W}_2)$, each of length $L = 2\bar{L}$, two servers Server 1 and Server 2 which store $(\bar{S}_1, \tilde{S}_2)$ and $(\bar{S}_2, \tilde{S}_1)$, respectively. Retrieval works exactly as before. For example, if the user wishes to privately retrieve $W_1 = (\bar{W}_1, \tilde{W}_1)$, then it retrieves $\bar{W}_1$ exactly as in the first implementation, and $\tilde{W}_1$ exactly as in the second implementation.

Since the symmetric scheme is comprised of two independent implementations of the original PIR scheme, the message size, total download size, total storage size, are all doubled relative to the original PIR scheme. As a result the rate and storage overhead, which are normalized quantities, remain unchanged in the new scheme. Symmetry is achieved because each server from the original PIR scheme is equally represented within each server in the new PIR scheme.

Mathematically,

$$W_1 = (\bar{W}_1, \tilde{W}_1), W_2 = (\bar{W}_2, \tilde{W}_2) \tag{6.114}$$
$$S_1 = (\bar{S}_1, \tilde{S}_2), S_2 = (\bar{S}_2, \tilde{S}_1) \tag{6.115}$$
$$\mathbb{F} = (\bar{\mathbb{F}}, \tilde{\mathbb{F}}), \mathbb{G} = (\bar{\mathbb{G}}, \tilde{\mathbb{G}}) \tag{6.116}$$
$$Q_n^{[k]} = (\bar{Q}_n^{[k]}, \tilde{Q}_{3-n}^{[k]}), n = 1, 2, k = 1, 2 \tag{6.117}$$
$$A_n^{[k]} = (\bar{A}_n^{[k]}, \tilde{A}_{3-n}^{[k]}) \tag{6.118}$$

where each random variable with a bar symbol is independent of and identically distributed with the same random variable with a tilde symbol. We are now ready to prove the first equality in (6.111).

$$\begin{aligned}
H(A_1^{[1]}|\mathbb{F},\mathbb{G}) &= H(\bar{A}_1^{[1]},\tilde{A}_2^{[1]}|\mathbb{F},\mathbb{G}) && (6.119)\\
&= H(\bar{A}_1^{[1]}|\bar{\mathbb{F}},\bar{\mathbb{G}}) + H(\tilde{A}_2^{[1]}|\tilde{\mathbb{F}},\tilde{\mathbb{G}}) && (6.120)\\
&= H(\tilde{A}_1^{[1]}|\tilde{\mathbb{F}},\tilde{\mathbb{G}}) + H(\bar{A}_2^{[1]}|\bar{\mathbb{F}},\bar{\mathbb{G}}) && (6.121)\\
&= H(\bar{A}_2^{[1]},\tilde{A}_1^{[1]}|\mathbb{F},\mathbb{G}) && (6.122)\\
&= H(A_2^{[1]}|\mathbb{F},\mathbb{G}) && (6.123)
\end{aligned}$$

where (6.120) and (6.122) follow from the fact that the two copies of the given scheme are independent and (6.121) is due to the property that the two copies are identically distributed. Consider the second equality in (6.111).

$$\begin{aligned}
H(A_2^{[1]}|\mathbb{F},\mathbb{G}) &\overset{(6.7)}{=} H(A_2^{[1]}|Q_2^{[1]},\mathbb{F},\mathbb{G}) && (6.124)\\
&= H(A_2^{[1]}|Q_2^{[1]},\mathbb{G}) && (6.125)\\
&\overset{(6.11)}{=} H(A_2^{[2]}|Q_2^{[2]},\mathbb{G}) && (6.126)\\
&= H(A_2^{[2]}|Q_2^{[2]},\mathbb{F},\mathbb{G}) && (6.127)\\
&\overset{(6.7)}{=} H(A_2^{[2]}|\mathbb{F},\mathbb{G}) && (6.128)
\end{aligned}$$

where (6.125) and (6.127) are due to the Markov chain $\mathbb{F} - (Q_2^{[k]},\mathbb{G}) - A_2^{[k]}, k = 1, 2$, which is proved as follows.

$$\begin{aligned}
I(A_2^{[k]};\mathbb{F}|Q_2^{[k]},\mathbb{G}) &\leq I(A_2^{[k]},S_2;\mathbb{F}|Q_2^{[k]},\mathbb{G}) && (6.129)\\
&= I(S_2;\mathbb{F}|Q_2^{[k]},\mathbb{G}) + I(A_2^{[k]};\mathbb{F}|Q_2^{[k]},\mathbb{G},S_2) && (6.130)\\
&\overset{(6.8)}{=} I(S_2;\mathbb{F}|Q_2^{[k]},\mathbb{G}) && (6.131)\\
&\leq I(S_2,W_1,W_2;\mathbb{F}|Q_2^{[k]},\mathbb{G}) && (6.132)\\
&= I(W_1,W_2;\mathbb{F}|Q_2^{[k]},\mathbb{G}) + I(S_2;\mathbb{F}|Q_2^{[k]},\mathbb{G},W_1,W_2) && (6.133)\\
&\overset{(6.4)}{\leq} I(W_1,W_2;\mathbb{F},Q_2^{[k]},\mathbb{G}) && (6.134)\\
&\overset{(6.7)(6.6)}{=} 0 && (6.135)
\end{aligned}$$

Finally, we prove (6.112).

$$
\begin{aligned}
H(S_1) &= H(\bar{S}_1, \tilde{S}_2) && (6.136)\\
&= H(\bar{S}_1) + H(\tilde{S}_2) && (6.137)\\
&= H(\tilde{S}_1) + H(\bar{S}_2) && (6.138)\\
&= H(\bar{S}_2, \tilde{S}_1) && (6.139)\\
&= H(S_2) && (6.140)
\end{aligned}
$$

where (6.137) and (6.139) follow from the fact that the two copies of the given scheme are independent and (6.138) is due to the property that the two copies are identically distributed.

■

6.5 Discussion

We showed that the capacity of MPIR is equal to the capacity of PIR, both with and without T-privacy constraints. Our result implies that there is no advantage in terms of capacity from multiround over single-round schemes, non-linear over linear schemes, or ϵ-error over zero-error schemes. We also offered a counterpoint to this pessimistic result by exploring optimal storage overhead instead of capacity. Specifically, we constructed a simple multiround, non-linear, ϵ-error PIR scheme that achieves a strictly smaller storage overhead than the best possible with any single-round, linear, zero-error PIR scheme. The simplicity of the scheme makes it an attractive point of reference for future work toward understanding the role of linear versus non-linear schemes, zero-error versus ϵ-error capacity, and single-round versus multiple round communications. Another interesting insight revealed by the scheme is the privacy benefit of reduced storage overhead. By not storing all the information at each server, and by optimally compressing the stored information, not only do we reduce the storage overhead, but also we enable privacy where it wouldn't hold otherwise.

Chapter 7

Conclusion

In this dissertation, we have explored the fundamental capacity limits of PIR and some of its variants. Starting from the basic model of PIR, we have characterized the exact capacity of PIR (Chapter 2), TPIR and RTPIR (Chapter 3), SPIR (Chapter 5) and MPIR (Chapter 6), for all choices of parameters. These results produce a new class of random codes that is marginally uniformly random if we view one part (i.e., from a local view), and is almost deterministic conditioning on one part (i.e., from a global view). A full understanding of this class of 'private' codes will benefit much beyond the field of information privacy, e.g., immediate applications are found in distributed storage and computing, through the connection between PIR and locally decodable codes. One remarkable feature of this class of codes is that the code structure admits a recursive algorithmic description, leaving much room for further optimization of computational complexity. Further, the random aspect of the code is decomposable with the code structure; we could first construct the code and then plug in the randomization. It remains interesting future work to discover new code structures, as well as new randomization techniques. The information theoretic converse involves a unusual type of constraint, i.e., same marginal distribution, which is not directly applicable in entropy terms. The tight converse characterization reveals immense potential of information theoretic reasoning. In the pursuit of optimal answers, new techniques from

both converse and achievability sides are of great interest. On the other hand, in spit of the above exact characterizations, when the PIR problem becomes more complicated, e.g., when more elements are involved, PIR becomes highly non-trivial as well. In Chapter 4, when we combine the T-privacy constraint and MDS-storage codes, the MDS-TPIR problem goes much beyond the two special cases. Although we have disproved a natural conjecture on the capacity of MDS-TPIR, the capacity remains a mystery in general. The depth of the problem requires a clean understanding of the interplay of multiple subspaces, which has been a key challenge in the study of signal dimensions in wireless networks [42]. There is some potential to develop the MDS-TPIR problem as a canonical setting for the study of the interactions of linear spaces, similar to the degrees of freedom problem of Gaussian multiple antenna interference channels [42]. Another intriguing element that we explore in Chapter 6 is the metric storage overhead. The problem of how much storage is needed to achieve a certain rate for the simplest setting turns out to be as profound as one would imagine. We encounter the necessity of non-linear codes (AND and OR functions and Slepian Wolf), non-Shannon information inequalities, ϵ-error schemes for a noiseless problem, and feedback (multiple rounds of communication). Simple as the setting is, one could easily connect to other fields with similar flavor, e.g., one could construct an equivalent network coding instance that requires non-Shannon inequalities and the network instance appears to the simplest available in the literature. We hope the PIR problem would serve as the playground to advance our understanding of non-linear codes, non-Shannon inequalities, ϵ-error schemes, and feedback and interaction. This prompts further generalizations and analysis of the unusual storage code proposed in Chapter 6, both in the field of PIR and to other topics, e.g., distributed storage and computing.

Chapter 2 characterizes the optimal communication rate of retrieving one message out of K messages from N distributed replicated servers privately. Banawan and Ulukus consider the setting [7] where the user wants to retrieve multiple ($P > 1$) messages simultaneously. A first question is that is separation optimal, i.e., could we do better than retrieving one message

at a time? The answer turns out to be positive and [7] proposes interesting joint coding schemes that combine answers from multiple messages. The exact capacity is characterized when we want no less than half of the messages ($P \geq K/2$) and when P divides K. Other settings are open and an intriguing extension on the query structure (beyond sums of subsets of message symbols) might be necessary.

The extension of PIR to include colluding servers and unresponsive servers is considered in Chapter 3. It turns out that a further layer of MDS codes suffices to achieve the capacity for both cases. One restriction on the problem statement is the underlying symmetry assumption, where we assume any T databases might collude and any N databases might respond. A recent work by Tajeddine et al. [68] initiates the topic of restricted colluding patterns. A topological view of the PIR problem will shed light on applications in heterogeneous network access structure.

The focus of Chapter 4 is on the MDS-TPIR problem. A disproof of the recent conjecture on the capacity of MDS-TPIR [33] asks for further investigation on this problem, e.g., Zhang and Ge [74] recently propose a new code for MDS-TPIR that works for all choices of parameters and outperforms the code in [33] (where the main target is the regime with infinite number of messages) for some range of parameters. It is intriguing that the combination of MDS storage codes and colluding servers turns out to be non-trivial. This motivates a more detailed classification of synergistic elements for PIR, i.e., when would we encounter problems that require significant generalizations of known techniques and when do current codes suffice to guarantee optimality? A list of candidates includes finite message length, unequal message length, colluding servers, unresponsive servers, storage codes, database privacy, topology considerations, multiple rounds of communication, etc.

The capacity of a form of oblivious transfer - SPIR is characterized in Chapter 5. The problem of MDS-SPIR is considered and the capacity is characterized by Wang and Skoglund in [71]. Oblivious transfer is a canonical problem in cryptography and its feasibility and communication efficiency over a point to point channel has been a topic of central interest.

Obvious transfer generally needs further randomness in the channel or network in the form of noise, common random variables, etc. SPIR represents an interesting class of extensions of oblivious transfer to the network setting. Combing SPIR with network characteristic remains an interesting future research topic.

The topic of Chapter 6 is multiple rounds of communication, i.e., MPIR. MPIR is particularly interesting in the metric - storage overhead. Although we have found the capacity of MPIR, the storage overhead of PIR stands out as an intriguing open problem. We mention that there is a recent and independent research line [31, 56, 19, 75, 62, 5, 70, 52] that targets the storage overhead problem. The focus there is on a class of codes with locality, where the defining property is that for each message symbol, there exists a number of disjoint recovering set. This property ensures that any linear PIR scheme can be simulated over such a locally coded storage system. Therefore, this line is mainly a decomposition based approach and focuses on the design of the local storage code, which can be combined with known PIR protocols.

To conclude, this dissertation contains an information theoretic treatment of the PIR problem, which originated in theoretical computer science. Information theorists commonly study the optimal coding rates of communication problems dealing with a few messages, each carrying an asymptotically large number of bits, while computer scientists often study the computational complexity of problems dealing with an asymptotically large number of messages, each carrying only a few bits (e.g., 1 bit per message). The occasional crossover of problems between the two fields opens up exciting opportunities for new insights. A prominent example is the index coding problem [17, 18], originally posed by computer scientists and recently studied from an information theoretic perspective. The information theoretic capacity characterization for the index coding problem is now recognized as perhaps one of the most important open problems in network information theory, because of its fundamental connections to a broad range of questions that includes topological interference management [44], network coding [58], distributed storage [51], hat guessing [57], and non-Shannon in-

formation inequalities [64]. Like index coding, the PIR problem also involves non-trivial interference alignment principles and is related to problems like blind interference alignment [43]. Further, PIR belongs to another rich class of problems studied in computer science, with deep connections to oblivious transfer [36], instance hiding [32, 1, 9], multiparty computation [10], secret sharing schemes [60, 13] and distributed computation with untrusted servers [10]. Bringing this class of problems into the domain of information theoretic studies holds much promise for new insights and fundamental progress. The results presented in this dissertation represent a step in this direction.

Bibliography

[1] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 195–203. ACM, 1987.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. Inform. Theory*, 46(4):1204–1216, Jul. 2000.

[3] R. Ahlswede and I. Csiszár. On oblivious transfer capacity. In *Information Theory, Combinatorics, and Search Theory*, pages 145–166. Springer, 2013.

[4] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Automata, Languages and Programming*, pages 401–407. Springer, 1997.

[5] H. Asi and E. Yaakobi. Nearly Optimal Constructions of PIR and Batch Codes. *arXiv preprint arXiv:1701.07206*, 2017.

[6] K. Banawan and S. Ulukus. The Capacity of Private Information Retrieval from Coded Databases. *arXiv preprint arXiv:1609.08138*, 2016.

[7] K. Banawan and S. Ulukus. Multi-Message Private Information Retrieval: Capacity Results and Near-Optimal Schemes. *arXiv preprint arXiv:1702.01739*, 2017.

[8] O. Barkol, Y. Ishai, and E. Weinreb. On locally decodable codes, self-correctable codes, and t-private PIR. *Algorithmica*, 58(4):831–859, 2010.

[9] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *STACS 90*, pages 37–48. Springer, 1990.

[10] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Locally random reductions: Improvements and applications. *Journal of Cryptology*, 10(1):17–36, 1997.

[11] A. Beimel and Y. Ishai. Information-theoretic private information retrieval: A unified construction. In *Automata, Languages and Programming*, pages 912–926. Springer, 2001.

[12] A. Beimel, Y. Ishai, and E. Kushilevitz. General constructions for information-theoretic private information retrieval. *Journal of Computer and System Sciences*, 71(2):213–247, 2005.

[13] A. Beimel, Y. Ishai, E. Kushilevitz, and I. Orlov. Share Conversion and Private Information Retrieval. In *Proceedings of the 27th Annual Conference on Computational Complexity*, pages 258–268. IEEE, 2012.

[14] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond. Breaking the $\mathcal{O}(n^{1/(2k-1)})$ barrier for information-theoretic Private Information Retrieval. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science.*, pages 261–270. IEEE, 2002.

[15] A. Beimel, Y. Ishai, and T. Malkin. Reducing the servers computation in private information retrieval: PIR with preprocessing. In *Advances in CryptologyCRYPTO 2000*, pages 55–73. Springer, 2000.

[16] A. Beimel and Y. Stahl. Robust information-theoretic private information retrieval. *Journal of Cryptology*, 20(3):295–321, 2007.

[17] Y. Birk and T. Kol. Informed-source coding-on-demand (ISCOD) over broadcast channels. In *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM'98*, volume 3, pages 1257–1264, 1998.

[18] Y. Birk and T. Kol. Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients. *IEEE Trans. on Information Theory*, 52(6):2825–2830, June 2006.

[19] S. Blackburn and T. Etzion. PIR Array Codes with Optimal PIR Rate. *arXiv preprint arXiv:1607.00235*, 2016.

[20] A. Blasiak, R. Kleinberg, and E. Lubetzky. Lexicographic products and the power of non-linear network coding. *ArXiv:1108.2489*, Aug. 2011.

[21] T. H. Chan and A. Grant. Dualities between entropy functions and network codes. *IEEE Trans. Inf. Theory*, 54(10):4470 – 4487, Oct. 2008.

[22] T. H. Chan, S.-W. Ho, and H. Yamamoto. Private Information Retrieval for Coded Storage. *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2842–2846, 2015.

[23] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 41–50, 1995.

[24] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private Information Retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2006.

[26] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh. A survey on network codes for distributed storage. *Proceedings of the IEEE*, 99:476–489, 2011.

[27] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Transactions on Information Theory*, 51(8):2745 – 2759, Aug. 2005.

[28] Z. Dvir and S. Gopi. 2-Server PIR with Sub-polynomial Communication. *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC'15*, pages 577–584, 2015.

[29] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[30] G. Fanti and K. Ramchandran. Efficient private information retrieval over unsynchronized databases. *Selected Topics in Signal Processing, IEEE Journal of*, 9(7):1229–1239, 2015.

[31] A. Fazeli, A. Vardy, and E. Yaakobi. Codes for distributed PIR with low storage overhead. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2852–2856, 2015.

[32] J. Feigenbaum. Encrypting problem instances. In *Advances in Cryptology – CRYPTO85 Proceedings*, pages 477–488. Springer, 1985.

[33] R. Freij-Hollanti, O. Gnilke, C. Hollanti, and D. Karpuk. Private Information Retrieval from Coded Databases with Colluding Servers. *arXiv preprint arXiv:1611.02062*, 2016.

[34] W. Gasarch. A Survey on Private Information Retrieval. In *Bulletin of the EATCS*, 2004.

[35] Y. Gertner, S. Goldwasser, and T. Malkin. A random server model for private information retrieval. In *Randomization and Approximation Techniques in Computer Science*, pages 200–217. Springer, 1998.

[36] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 151–160. ACM, 1998.

[37] P. Gopalan, C.Huang, H. Simitci, and S. Yekhanin. On the Locality of Codeword Symbols. *IEEE Transactions on Information Theory*, 58(11):6925–6934, Nov. 2012.

[38] A. W. Ingleton. Representation of matroids in combinatorial mathematics and its applications. *Combinatorial Mathematics and Its Applications*, 44:149 – 167, Jul. 1971.

[39] Y. Ishai and E. Kushilevitz. On the hardness of information-theoretic multiparty computation. In *Advances in Cryptology-EUROCRYPT 2004*, pages 439–455. Springer, 2004.

[40] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Batch codes and their applications. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 262–271. ACM, 2004.

[41] Y. Ishai, M. Prabhakaran, and A. Sahai. Founding cryptography on oblivious transfer–efficiently. In *Annual International Cryptology Conference*, pages 572–591. Springer, 2008.

[42] S. Jafar. Interference Alignment: A New Look at Signal Dimensions in a Communication Network. In *Foundations and Trends in Communication and Information Theory*, pages 1–136, 2011.

[43] S. A. Jafar. Blind Interference Alignment. *IEEE Journal of Selected Topics in Signal Processing*, 6(3):216–227, June 2012.

[44] S. A. Jafar. Topological Interference Management through Index Coding. *IEEE Trans. on Inf. Theory*, 60(1):"529–568", Jan. 2014.

[45] S. Jalali, M. Effros, and T. Ho. On the impact of a single edge on the network coding capacity. In *Information Theory and Applications Workshop (ITA), 2011*, pages 1–5. IEEE, 2011.

[46] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86. ACM, 2000.

[47] J. Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 20–31. ACM, 1988.

[48] O. Kosut and J. Kliewer. On the relationship between edge removal and strong converses. In *Proceedings of International Symposium on Information Theory (ISIT)*, 2016.

[49] S. Kumar, E. Rosnes, and A. G. i Amat. Private Information Retrieval in Distributed Storage Systems Using an Arbitrary Linear Code. *arXiv preprint arXiv:1612.07084*, 2016.

[50] M. Langberg and M. Effros. Network Coding: Is zero error always possible? In *49th Allerton Conference on Communication, Control and Computing.*, pages 1478–1485, 2011.

[51] A. Mazumdar. Storage Capacity of Repairable Networks. *IEEE Trans. on Inf. Theory*, 61(11), Nov. 2015.

[52] P. V. K. Myna Vajha, Vinayak Ramkumar. Binary, Shortened Projective Reed Muller Codes for Coded Private Information Retrieval. *arXiv preprint arXiv:1702.05074*, 2017.

[53] A. C. Nascimento and A. Winter. On the oblivious-transfer capacity of noisy resources. *IEEE Transactions on Information Theory*, 54(6):2572–2581, 2008.

[54] R. Ostrovsky and W. E. Skeith III. A Survey of Single-database Private Information Retrieval: Techniques and Applications. In *Public Key Cryptography–PKC 2007*, pages 393–411. Springer, 2007.

[55] M. O. Rabin. How to exchange secrets with oblivious transfer. 1981.

[56] S. Rao and A. Vardy. Lower Bound on the Redundancy of PIR Codes. *arXiv preprint arXiv:1605.01869*, 2016.

[57] S. Riis. Information Flows, Graphs and their Guessing Numbers. *The Electronic Journal of Combinatorics*, 14(1):R44, 2007.

[58] S. Rouayheb, A. Sprintson, and C. Georghiades. On the Index Coding Problem and Its Relation to Network Coding and Matroid Theory. *IEEE Trans. on Inf. Theory*, 56(7):3187–3195, July 2010.

[59] N. Shah, K. Rashmi, and K. Ramchandran. One Extra Bit of Download Ensures Perfectly Private Information Retrieval. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 856–860, 2014.

[60] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[61] T. E. Simon R. Blackburn and M. B. Paterson. PIR schemes with small download complexity and low storage requirements. *arXiv preprint arXiv:1609.07027*, 2016.

[62] V. Skachek. Batch and PIR Codes and Their Connections to Locally Repairable Codes. *arXiv preprint arXiv:1611.09914*, 2016.

[63] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4):471–480, 1973.

[64] H. Sun and S. A. Jafar. Index Coding Capacity: How far can one go with only Shannon Inequalities? *IEEE Trans. on Inf. Theory*, 61(6):3041–3055, 2015.

[65] H. Sun and S. A. Jafar. Blind Interference Alignment for Private Information Retrieval. *arXiv preprint arXiv:1601.07885*, 2016.

[66] H. Sun and S. A. Jafar. Optimal Download Cost of Private Information Retrieval for Arbitrary Message Length. *arXiv preprint arXiv:1610.03048*, 2016.

[67] H. Sun and S. A. Jafar. Private Information Retrieval from MDS Coded Data with Colluding Servers: Settling a Conjecture by Freij-Hollanti et al. *arXiv preprint arXiv:1701.07807*, 2017.

[68] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. E. Rouayheb. Private Information Retrieval Schemes for Coded Data with Arbitrary Collusion Patterns. *arXiv preprint arXiv:1701.07636*, 2017.

[69] R. Tajeddine and S. E. Rouayheb. Private Information Retrieval from MDS Coded Data in Distributed Storage Systems. *arXiv preprint arXiv:1602.01458*, 2016.

[70] E. K. Thomas and V. Skachek. Explicit Constructions and Bounds for Batch Codes with Restricted Size of Reconstruction Sets. *arXiv preprint arXiv:1701.07579*, 2017.

[71] Q. Wang and M. Skoglund. Symmetric Private Information Retrieval For MDS Coded Distributed Storage. *arXiv preprint arXiv:1610.04530*, 2016.

[72] S. Yekhanin. *Locally Decodable Codes and Private Information Retrieval Schemes*. PhD thesis, Massachusetts Institute of Technology, 2007.

[73] S. Yekhanin. Private Information Retrieval. *Communications of the ACM*, 53(4):68–73, 2010.

[74] Y. Zhang and G. Ge. A general private information retrieval scheme for MDS coded databases with colluding servers. *arXiv preprint arXiv:1704.06785*, 2017.

[75] Y. Zhang, X. Wang, H. Wei, and G. Ge. On private information retrieval array codes. *arXiv preprint arXiv:1609.09167*, 2016.