PROVIDING PRIVATE AND FAST DATA ACCESS FOR CLOUD SYSTEMS

A Dissertation

by

FATEMEH KAZEMIKORDASIABI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Alex Sprintson
Committee Members,	P.R. Kumar
	Chao Tian
	Natarajan Gautam
Head of Department,	Miroslav M. Begovic

May 2022

Major Subject: Electrical Engineering

ABSTRACT

Cloud storage and computing systems have become the backbone of many applications such as streaming (Netflix, YouTube), storage (Dropbox, Google Drive), and computing (Amazon Elastic Computing, Microsoft Azure). To address the ever growing demand for storage and computing requirements of these applications, cloud services are typically implemented over a large-scale distributed data storage system. Cloud systems are expected to provide the following two pivotal services for the users: 1) private content access and 2) fast content access. The goal of this thesis is to understand and address some of the challenges that need to be overcome to provide these two services.

The first part of this thesis focuses on private data access in distributed systems. In particular, we contribute to the areas of Private Information Retrieval (PIR) and Private Computation (PC). In the PIR problem, there is a user who wishes to privately retrieve a subset of files belonging to a database stored on a single or multiple remote server(s). In the PC problem, the user wants to privately compute functions of a subset of files in the database. The PIR and PC problems seek the most efficient solutions with the minimum download cost that enable the user to retrieve or compute what it wants privately.

We establish fundamental bounds on the minimum download cost required for guaranteeing the privacy requirement in some practical and realistic settings of the PIR and PC problems and develop novel and efficient privacy-preserving algorithms for these settings. In particular, we study the single-server and multi-server settings of PIR in which the user initially has a random linear combination of a subset of files in the database as side information, referred to as PIR with coded side information. We also study the multi-server setting of the PC in which the user wants to privately compute multiple linear combinations of a subset of files in the database, referred to as Private Linear Transformation.

The second part of this thesis focuses on fast content access in distributed systems. In particular, we study the use of erasure coding to handle data access requests in distributed storage and computing systems. Service rate region is an important performance metric for coded distributed systems, which expresses the set of all data access request rates that can be simultaneously served by the system. In this context, two classes of problems arise: 1) characterizing the service rate region of a given storage scheme and finding the optimal request allocation, and 2) designing the underlying erasure code to handle a given desired service rate region.

As contributions along the first class of problems, we characterize the service rate region of systems with some common coding schemes such as Simplex codes and Reed-Muller codes by introducing two novel techniques: 1) fractional matching and vertex cover on graph representation of codes, and 2) geometric representations of codes. Moreover, along the second class of code design, we establish some lower bounds on the minimum storage required to handle a desired service rate region for a coded distributed system and in some regimes, we design efficient storage schemes that provide the desired service rate region while minimizing the storage requirements.

DEDICATION

To my mother, to whom I owe everything.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisor, Prof. Alex Sprintson, for his constant support and encouragement at every stage of my Ph.D. journey. He taught me the art of tackling a complex problem, the correct way of conducting research, and the principles of academic writing and scientific presentation. He has given me the freedom to pursue new research ideas, while has always been available for guiding me whenever I needed help, whether for brainstorming new ideas or solving issues before deadlines. He always encouraged and helped me establish new collaborations, which was instrumental in my growth as a researcher. This dissertation would have not been successfully accomplished without all his supports for which I will always be indebted.

Moreover, I would like to express my very special appreciation to my academic mother, Prof. Emina Soljanin, who has been an invaluable part of my graduate life. It is beyond words to describe how supportive she has been and how much I have learned from her. I am extremely grateful to her for giving me the opportunity to spend a wonderful academic year in her lab at Rutgers University as a visiting student. I cannot thank her enough for being an excellent host and an extraordinary mentor. Her ingenuity to convert a complex research problem into an interesting fun puzzle and to present complex ideas in a simple way using nice pictures has always amazed me. I am also very grateful to her for helping me build connections in the academic community and establish new collaborations. I wholeheartedly thank her for being so caring, compassionate, and supportive.

I would also like to express my sincere appreciation to my committee members, Prof. P.R. Kumar, Prof. Chao Tian, and Prof. Natarajan Gautam, for kindly accepting to be on my dissertation committee and for their valuable comments and feedback that helped me to improve the quality and the presentation of my thesis.

I was fortunate to participate in multiple collaborative projects and to work with many wonderful collaborators over the past few years. My special thanks go to Anoosheh Heidarzadeh, from whom I learned a lot. Anoosheh is not only a brilliant researcher but also an excellent mentor. I am thankful to him for teaching me the principles of conducting rigorous research and writing a technical paper. I also had the pleasure to work with Prof. Sascha Kurz, who helped me advance my research. I am thankful to him for his insight, his valuable comments, and for all the helpful discussions we had over lengthy emails.

I would like to thank Prof. Zixiang Xiong, who supported me to receive the Graduate Teaching Fellowship award from the College of Engineering at Texas A&M University, to serve as an instructor for the Random Signal and Systems course during Spring 2021. I would also like to thank all of my teachers and the faculty members whom I interacted with over the years, in particular, Prof. Krishna Narayanan. I am also thankful to all of my wonderful friends and the amazing staff at our department for making my time at Texas A&M University a wonderful experience in a friendly environment.

I am deeply indebted to my loving parents, Mitra and Jamshid, for their unconditional love, endless support, and constant encouragement. My achievements would undoubtedly never have been possible without their countless sacrifices. I am also profoundly thankful to my dear brother, Nima, who always stands by my side no matter how far we are. Words truly cannot express the depth of my gratitude and appreciation to them.

Finally, and most importantly, I owe my deepest gratitude to my best friend, my forever companion, my colleague, and my husband, Navid, for his unconditional love and constant support, for always believing in me, and for being the source of my confidence and strength whenever I was overwhelmed and stressed. Being away from my family for these many years was the hardest part of this journey, and he was the only person who made it bearable for me. This thesis would be impossible without his support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Alex Sprintson (advisor) and Professors P. R. Kumar and Chao Tian of the Department of Electrical and Computer Engineering and Professor Natarajan Gautam of the Department of Industrial Engineering at the Texas A&M University. The work presented in Chapters 2 and 3 is carried out in collaboration with Dr. Anoosheh Heidarzadeh of the Department of Electrical and Computer Engineering at the Texas A&M University. The work presented in Chapters 5, 6 and 7, is carried out in collaboration with Professor Emina Soljanin of the Department of Electrical and Computer Engineering at the Rutgers University. The work presented in Chapters 6 and 7 is carried out in collaboration with Professor Sascha Kurz of the Department of Mathematics at the University of Bayreuth. All other work conducted for the dissertation was completed by the student independently.

Funding Sources

This work was supported in part by the National Science Foundation (NSF) under Grants No. CIF-1642983 and CIF-1718658, and the Graduate Teaching Fellowship from the College of Engineering at Texas A&M University.

NOMENCLATURE

DSS	Distributed Storage System(s)
PIR	Private Information Retrieval
PLC	Private Linear Computation
PLT	Private Linear Transformation
PC	Private Computation
MDS	Maximum Distance Separable
RM	Reed-Muller
LP	Linear Program
ML	Machine Learning
$\mathbb{P}(\cdot)$	Probability
$\mathbb{P}(\cdot\|\cdot)$	Conditional probability
$H(\cdot)$	(Shannon) Entropy
$H(\cdot\|\cdot)$	Conditional entropy
$I(\cdot;\cdot)$	Mutual information
$I(\cdot;\cdot\|\cdot)$	Conditional mutual information
$\mathbb{F}_q$	Finite field for a prime power q
$\mathbb{F}_q^\times$	Multiplicative group of $\mathbb{F}_q$
$\mathbb{F}_{q^l}$	Extension field of $\mathbb{F}_q$
$\mathbb{F}_q^S$	S-dimensional vector space over $\mathbb{F}_q$
$\mathbb{F}_q^{k\times n}$	$k \times n$-dimensional matrix space over $\mathbb{F}_q$

$\mathbb{Z}_{\geq 0}$	Set of non-negative integers
$\mathbb{N}$	Set of positive integers
$[i]$	Set of integers $\{1, \ldots, i\}$
$\mathcal{K}$	Set of integers $\{1, \ldots, k\}$
$\mathbf{0}_k$	All-zero vector of length k
$\mathbf{1}_k$	All-one vector of length k
$[n, k]_q$	Linear code of length n and dimension k over $\mathbb{F}_q$
$\mathcal{C}$	Linear code
$\#\mathcal{S}$	Cardinality of the set (or multiset) $\mathcal{S}$
$\mathrm{PG}(k-1, q)$	$k-1$-dimensional projective space over $\mathbb{F}_q$
$\mathbf{e}_i$	Vector of size k, having 1 at position i and 0 elsewhere
$\langle \mathcal{S} \rangle$	Span of the set of vectors $\mathcal{S}$
$\mathrm{conv}(\mathcal{S})$	Convex hull of the set of vectors $\mathcal{S}$
λ_i	Request arrival rate for file f_i
$\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$	Demand vector
$\mathcal{Y}_i$	Set of t_i recovery sets for file f_i
$Y_{i,j}$	jth recovery set for file f_i
μ	Service rate of each server
$\lambda_{i,j}$	Portion of λ_i assigned to the recovery set $Y_{i,j}$
$(\mathbf{G}, \boldsymbol{\mu})$ system	System with storage scheme $\mathbf{G}$, service rate $\boldsymbol{\mu}$ of servers
$\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$	Service rate region of $(\mathbf{G}, \boldsymbol{\mu})$ system

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Background and Motivation

Cloud storage and computing systems have become the backbone of today's widely used applications such as streaming (Netflix, YouTube), storage (Dropbox, Google Drive), computing (Amazon Elastic Computing, Microsoft Azure), and data analytics. To address the ever growing demand for storage and computing requirements of these applications, cloud services are implemented over a large-scale distributed data storage system that provides the desired content to the users. As a result, the performance of a cloud system and the quality of user experience rely on the efficiency of underlying distributed storage system. In addition to providing low-cost and reliable content access, cloud systems are expected to provide the following two pivotal services for the users: 1) private content access and 2) fast content access.

Providing private content access is very important for cloud services because online service providers such as (Google, YouTube, Netflix, etc.) collect significant amounts of users data (such as email addresses, phone numbers, etc.) for various purposes, most notably for building their advertising platforms through which advertisers can target platform users. This data that uniquely identify users may pose a severe threat to users' privacy, for example, if accessed by untrusted parties. Thus, privacy is a major concern for online users who may unknowingly reveal critical personal information (such as political proclivity, medical conditions, etc.) through daily online activities. As a result, protecting the privacy of user's information is of paramount importance for the online service providers. This well-acknowledged concern has led to many interesting theoretical problems such as anonymity, differential privacy, private information retrieval (PIR), and private computation [2–48].

Providing fast content access is vital for cloud services because delayed response turns away users which results in revenue loss. For example, Google recently reported a 20% loss of search traffic when the delay in loading search results increased only 0.5 sec. Content files stored on the cloud storage systems may be simultaneously requested by multiple users. Maximizing the number of users that can be served simultaneously by the system reduces the users' experienced latency, particularly in a high traffic regime, which is very important for delay-sensitive applications such as video streaming, as well as collaborative applications such as Dropbox and Google Docs, where many users wish to access the same content at the same time. Thus, cloud services must be able to serve a large number of users simultaneously.

The goal of this thesis is to understand and address some of the challenges that arise in providing private and fast content access in cloud systems. In particular, the objective of this thesis is to provide mathematical underpinnings and practical solutions that enable private and fast content access in distributed storage and computing systems by developing insights based on fundamental ideas from different areas of information theory, coding theory, probability, graph theory, optimization, and combinatorics. Next section outlines the key contributions of the two parts of this thesis, and describes the organization of the chapters within these two parts.

1.2 Private Data Access in Distributed Systems

The first part of this thesis focuses on private content access and private computation in distributed storage systems. In particular, my work contributes to the areas of Private Information Retrieval (PIR) and Private Linear Transformation (PLT). In addition to its direct applications in privacy, PIR is intimately related to many fundamental problems in cryptography, coding theory, and network coding. Therefore, PIR represents an important focal point to tackle significant challenges across these fields.

In the PIR problem, there is a user who wishes to privately download a single or multiple files belonging to a database stored on a single or multiple (non-colluding or colluding) servers. There are two different types of PIR in the literature: *computational* and *information-theoretic*. In the computational PIR (e.g., [2, 3]), the identity of the requested message(s) must be protected from the server(s), assuming that the server(s) is computationally bounded. Aside from the computational PIR is the information-theoretic PIR, introduced by Chor *et al.* in [4], where no such assumption is made on the computational power of the server(s), and the identity of the requested message(s) need to be protected in an information-theoretic sense.

An information-theoretic PIR scheme seeks the most efficient solution with minimum download cost that enables the user to privately retrieve their demanded file(s) without revealing any information about the identity of the desired file(s) to any individual server. The drawback of this strong requirement is that in the single-server case or multi-server setting when all servers can fully collude, the user must download the entire database from the server [4]. This has led to an extensive body of work on the multi-server PIR when the servers do not fully collude (see, e.g., [5–18]) and the PIR settings in which the user has some side information (unknown to the server(s)) about the messages in the database [21–33].

In the PLT problem, there are N servers, each of which stores an identical copy of a database consisting of K independent messages. Also, there is a user who wants to privately compute L linear combinations of a subset of D messages in the database without revealing any information to any server about the identity of the messages required for the computation, while downloading the minimum possible amount of information from the servers. In the PLT problem, the following two types of privacy requirements can be considered: (i) the individual privacy, where the identity of each individual message in the support set of the demanded linear combinations needs to be kept private [49]; and (ii) the

joint privacy, in which the identity of the entire set of messages in the support set of the demanded linear combinations must be kept private [50]. The joint privacy is a stronger notion of privacy, for which the query must protect the correlation between the indices in the demand support index set, whereas for individual privacy some information about this correlation may be leaked, and hence is a weaker notion of privacy. The PLT problem can be viewed as an interesting extension of the PIR problem and the Private Linear Computation (PLC) problem in which the goal is to privately compute one linear combination of a subset of D messages in the database. The PLT problem can be motivated by several practical scenarios such as linear transformation technique applied for dimensionality reduction in Machine Learning (ML) applications (see [50]).

In this thesis, our goal is to develop novel and efficient PIR and PLT algorithms for practical and realistic settings. In particular, we aim to characterize the minimum required download costs in each of the considered settings and develop novel and efficient algorithms that satisfy the privacy requirements and enable the user to retrieve the desired file while achieving the minimum download cost.

1.3 Fast Data Access in Distributed Systems

The second part of this thesis focuses on fast content access in distributed storage and computing systems. The past two decades have seen an explosive growth in the amount of data stored in the cloud data centers which was accompanied by a rapid increase in the volume of users accessing it. To handle these surging demands in a fast, reliable and efficient manner, chunks of a data object are stored redundantly over multiple storage nodes through either replication or erasure coding. Although replication has been typically preferred due its simplicity, it can be expensive in terms of storage. Erasure codes have been shown to be effective in achieving various goals in cloud systems such as providing reliability against node failures (see e.g., [51–54]), ensuring availability of stored content

during high demand (see e.g., [55–58]), enabling the recovery of a data object from multiple disjoint groups of nodes (see e.g., [59–61]), and providing fast content download (see e.g., [62–70]). Content files stored on cloud storage systems may be simultaneously requested by multiple users. Serving a large number of users simultaneously is a major concern in cloud systems and so is considered as one of the most significant considerations in the design of coded distributed systems.

In this thesis, we seek to understand how redundancy can be used to increase the total volume of requests that can be served concurrently by the system. In particular, we consider coded distributed systems where k different data objects (rather than k chunks of one object) are combined into n coded objects which are stored on n nodes. We consider *heterogeneous* requests to access these objects at rates $\lambda_1, \lambda_2, \ldots, \lambda_k$, respectively. Each of the n nodes can serve at most μ rate of requests. Thus, the total request rate allocated to each node must not exceed μ. Under these constraints, we aim to characterize the set of achievable vectors $(\lambda_1, \lambda_2, \ldots, \lambda_k)$, which we refer to as the *service rate region* of a coded distributed system. Since the nodes storing coded objects can be used to partially serve requests for any of the objects included in that coded combination, coded distributed systems are more flexible and can have a different (possibly more favorable) service rate region than an uncoded system with the same number of nodes. We illustrate this through the motivating example below.

Motivating Example. Consider an example shown in Figure 1.1, where two objects a and b are redundantly stored on 4 nodes. Figure 1.1 (left) shows 3 redundant storage schemes: replication, coding, and replication and coding combined. Given that each node can serve $\mu = 1$ request per second, we want to maximize λ_a and λ_b, the rate of requests for a and b that can be supported. Object a can be downloaded from the node storing a, or from two nodes that store coded combinations of a and b.

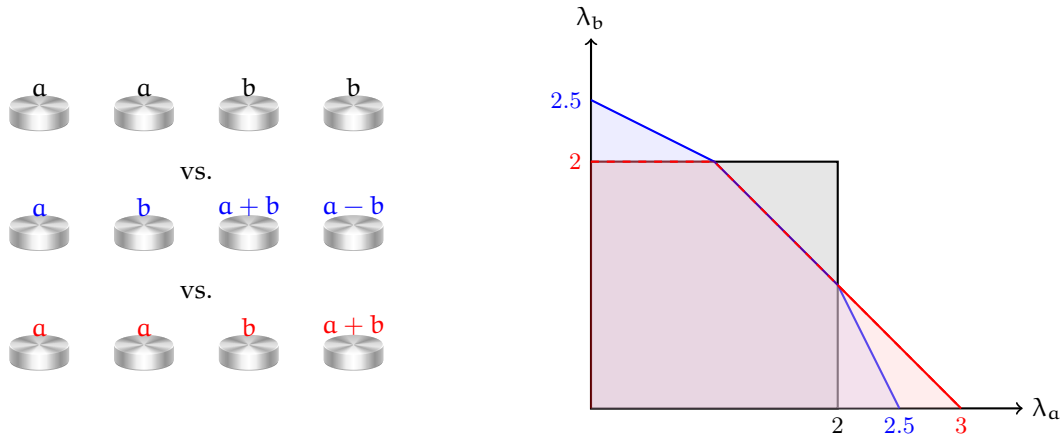

Figure 1.1: (left) Replicated, coded, and hybrid systems with $n = 4$ nodes storing $k = 2$ files. (right) Service rate regions of the three systems when the service capacity of each node is $\mu = 1$. The regions have the same areas. Coding can handle the skews in request arrival rates λ_a and λ_b for the two stored objects. Reprinted with permission from [1].

Figure 1.1 (right) shows the service rate regions of the 3 storage systems. The replicated system can achieve the square service rate region with $0 \leq \lambda_a, \lambda_b \leq 2$; this is because there are two copies of each object and each node can support $\mu = 1$ rate of requests. The coded system with two nodes storing $a + b$ and $a - b$ respectively instead of uncoded copies of a and b achieves the blue colored shaded service rate region. This system can handle skews in λ_a and λ_b better than the replicated system when one of the two objects a and b are more frequently accessed but both objects are unlikely to be popular simultaneously. The service rate region of a combined replication and coding system (shown in red) can better support asymmetries in the demands λ_a and λ_b and is the best choice when the request rate for a is expected to be larger than that of b.

The service rate region can be used as an important performance metric in the design and analysis of erasure coded distributed systems, which describes the set of all data access requests that can be simultaneously served by the system [1, 71–77]. Characterizing the service rate region of a coded distributed system gives us a clear picture of the collective

rate of requests that can be supported by the system as well as its robustness to heterogeneous request patterns where some objects are more frequently accessed than others. It is noteworthy that expanding the service rate region reduces the users' experienced latency, particularly in a high traffic regime, which is important for the delay-sensitive applications such as live streaming, where many users wish to get the same content at the same time.

In the context of using the service rate region as a metric to design erasure-coded distributed systems, two classes of problems arise: 1) characterizing the service rate region of a given storage scheme and finding the optimal request allocation (i.e., optimal policies to split incoming requests across the nodes in order to maximize the the volume of the achievable service rate region), and 2) designing the underlying erasure code to cover a given desired service rate region with minimum storage or to maximize the service rate region of a distributed system with given number of storage nodes. In this thesis, our goal is to address some of the problems within each of these two threads.

1.4 Our Contributions and Organization

This section summarizes the key contributions of the two parts of this thesis, and describes the organization of chapters within these parts.

Part I

In Part I of this thesis, we study the single-server and multi-server settings of the PIR problem in the scenarios where the user initially has a random linear combination of a subset of files in the database as side information, in Chapter 2 and Chapter 3, respectively. These settings can be motivated by several practical scenarios. For instance, the user may have obtained a coded side information via overhearing in a wireless network, or on-the-fly recording of a random linear combination of messages being broadcast by an information source, or from a trusted agent, e.g., an entity who makes profit by offering privacy to users, with limited knowledge about the database, or from the information which

is locally stored, e.g., using an erasure code, in the user's cache of limited size. We refer to this problem as PIR with coded side information (PIR-CSI) when only the identity of the demanded message must be protected, and PIR with private coded side information (PIR-PCSI) when the identities of both the demanded message and the messages participating in the side information must be protected. Toward our research goals, in Chapter 4, we also study the multi-server setting of the PLT problem under a strong privacy guarantee, referred to as joint privacy guarantee, for which the identity of the entire set of messages in the support set of the demanded linear combinations must be kept private (i.e., without leaking any information about the correlation between them).

In **Chapter 2** and **Chapter 3**, we study the single-server and multi-server settings of single-message (information-theoretic) PIR in the presence of a *coded side information*, respectively. In Sections 2.4 and 2.5, we study the single-server setting of the PIR-PCSI and PIR-CSI problems, respectively. We extend these settings to the multi-server settings of the PIR-PCSI and PIR-CSI problems in Sections 3.4 and 3.5, respectively.

We assume that the identities of the messages in the support set of the coded side information as well as the coding coefficients are initially unknown to the server. Depending on whether the support set of the user's coded side information includes the user's demand or not, we consider two different models for each of the PIR-PCSI and PIR-CSI problems. In the first model, referred to as *Model I*, the demand does not belong to the support set of the coded side information, whereas in the second model, referred to as *Model II*, the demand belongs to the support set of the coded side information. We refer to the PIR-PCSI (or PIR-CSI) problem under Model I and Model II as *PIR-PCSI–I* (or *PIR-CSI–I*) and *PIR-PCSI–II* (or *PIR-CSI–II*), respectively.

For each model and for each of the privacy requirements, we consider the problem of designing a protocol for generating the user's query and the servers' answers that enables the user to decode the message they need while satisfying the privacy requirement.

We establish fundamental bounds on the capacity of each setting, defined as the ratio of the number of information bits in a message to the minimum number of information bits downloaded from the server over all protocols that satisfy the privacy condition. Our converse proofs rely on new information-theoretic and algebraic arguments. We also develop novel and efficient privacy-preserving protocols for each of the considered settings.

In **Chapter 4**, we study the multi-server setting of the PLT problem with arbitrary number of servers $N \geq 1$. We focus on the setting in which the coefficient matrix of the required linear combinations generates a Maximum Distance Separable (MDS) code. This setting can be motivated by several practical scenarios. For instance, the user may have chosen the coefficient matrix randomly over the field of real numbers or a finite field of large size [50]. First, we show that the capacity of PLT problem for the case of $L = 1$, i.e., when the user wishes to compute one linear combination of D messages, is equal to $\Phi(1/N, K - D + 1)$, where $\Phi(A, B) = (1 + A + A^2 + \cdots + A^{B-1})^{-1}$. Moreover, we establish an upper bound on the capacity of PLT problem for any arbitrary parameters $N, K, D, L \geq 1$, and based on some known capacity results, we show the tightness of the provided upper bound for some special cases of the problem: (i) the case where there is a single server (i.e., $N = 1$), (ii) the case where $L = 1$, and (iii) the case where $L = D$.

Part II

In part II of this thesis, we first formulate the problem of characterizing the service rate region and finding optimal request splitting as a constrained optimization problem. However, it cannot be trivially solved using linear solvers because the number of optimization variables is large and the problem becomes computationally intractable. As contributions along this thread, we characterize the service rate region of some well-known classes of codes such as Simplex (also called Hadamard) codes and first-order Reed-Muller (RM) codes by introducing two different novel techniques. In particular, in Chapter 5, we use

fractional matching and vertex cover on graph representation of codes to find the service rate region of Simplex codes, and we use the geometric approach in Chapter 6 to find the rate region of simplex codes and first-order RM codes. These analyses provide insights into how the service rate region is affected by the length and the rate of the underlying code. Along the second thread of designing the underlying erasure code, in Chapter 7 we study the problem of covering a desired rate region with minimum storage.

In **Chapter 5**, first we introduce a novel technique for constructing a special graph representation of a linear code in Section 5.3.1. Then, using this approach we show the following results in Section 5.3.4: 1) equivalence between the service rate problem and the well-known fractional matching problem and 2) equivalence between the integral service rate problem and the matching problem. These equivalence results allow us to use techniques from the rich literature of the graph theory for solving the service rate problem. Leveraging these equivalence results, it is shown that the maximum sum rates that can be simultaneously served by the system equals the fractional matching number in the graph representation of the code, and thus is lower bounded and upper bounded by the matching number and the vertex cover number, respectively. This is of great interest because if the graph representation of a code is bipartite, then the derived upper bound and lower bound are equal which allows one to establish the maximum sum rate that can be served by the system. Leveraging this result, we characterize the service rate region of the binary simplex codes whose graph representation is bipartite, as we will show in Section 5.3.4. We also show in Section 5.4 that the notion of integral service rate region opens up interesting connections with batch codes, a class of codes designed for simultaneous access [78]. Specifically, we show that the service rate problem can be viewed as a generalization of the batch code problem, and the multiset primitive batch codes problem is a special case of the service rate problem when the portion of requests assigned to the recovery sets is restricted to be integral.

In **Chapter 6**, to study the service rates of codes problem, we introducing a novel geometric approach that provides a set of half-spaces whose intersection encompasses the service rate region of a given linear storage scheme. In other words, the geometric approach provides upper bounds on the sum of each subset of arrival rates in any demand vector $(\lambda_1, \cdots, \lambda_k)$ in the service rate region of a linear code in a more straightforward manner in comparison to other approaches. This technique is of great significance since it allows one to derive upper bounds on the service rates of linear codes without explicitly knowing the list of all possible recovery sets while other approaches such as waterfilling (see [1]) and combinatorial approaches rely on enumeration of all recovery sets that gets increasingly complex when the number of files k increases. Leveraging our novel geometric technique, we take initial steps towards deriving bounds on the service rates of some parametric classes of linear codes without explicitly listing the set of all possible recovery sets. In particular, in Sections 6.4 and 6.5, we derive upper bounds on the service rates of the first order Reed-Muller codes and Simplex codes, respectively, as two classes of codes which are most important in theory as well as in practice. It is worth mentioning that only the cardinality of the recovery sets matters in deriving upper bounds on the service rate of linear codes using the geometric approach. Subsequently, we show how the derived upper bounds can be achieved. Moreover, utilizing the proposed geometric technique, we show that given the service rate region of a code, a lower bound on the minimum distance of the code can be derived.

In **Chapter 7**, we focus on designing the underlying erasure code for covering a given service rate region with minimum storage. In particular, we consider a practical setting of designing a coded distributed storage system where we wish to store k files redundantly across multiple storage nodes in a distributed storage system. Also, we are given a bounded subset $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$ as a desired service rate region for this storage system. Our goal is: 1) to find the minimum number of storage nodes $n(\mathcal{R})$ (or a lower bound on

$n(\mathcal{R})$) required for serving all demand vectors $\boldsymbol{\lambda}$ inside the desired service rate region $\mathcal{R}$, and 2) to design the most storage-efficient redundancy scheme with the service rate region that covers the set $\mathcal{R}$. We formulate the problem of minimizing the number of storage nodes required for covering a desired service rate region as an integer linear programming (ILP). We show that the corresponding ILP cannot be solved trivially using linear solvers especially when the number of files K increases. The reason is that in order to list the constraints of the corresponding ILP, one needs to explicitly know all recovery sets which becomes increasingly complex when the number of files k increases. By using a novel geometric technique, we propose three different general lower bounds for $n(\mathcal{R})$ in Section 7.3.2. Also, we show that for $k = 2$, these bounds are tight and we design an efficient storage scheme that achieves the desired service rate region while minimizing the storage in Section 7.3.3.

Finally, in **Chapter 8**, we conclude this thesis by summarizing our results and presenting some interesting further directions.

Part I

Private Data Access in Distributed Systems

2. SINGLE-SERVER PRIVATE INFORMATION RETRIEVAL (PIR)*

2.1 Introduction

In Private Information Retrieval (PIR) problem, there is a user that wishes to privately download a single or multiple messages belonging to a database stored on a single or multiple (non-colluding or colluding) servers. There are two types of PIR in the literature: *computational* PIR and *information-theoretic* PIR. In the computational PIR (see, e.g., [3]), the identity of the requested message(s) must be protected from the server(s), assuming that the server(s) is computationally bounded. Aside from the computational PIR is the information-theoretic PIR, introduced by Chor *et al.* in [4], where no such assumption is made on the computational power of the server(s), and the identity of the requested message(s) need to be protected in an information-theoretic sense. The drawback of this strong requirement is that in the single-server case, the user must download the entire database from the server [4]. This has led to an extensive body of work on multi-server information-theoretic PIR (see, e.g., [5–18]).

An information-theoretic PIR scheme seeks the most efficient solution (with minimum download cost) for a user to privately download a file from a database stored redundantly on a single or multiple servers, without revealing any information about the identity of the desired file to any individual server. As an application of PIR, consider the following scenario. Online service providers (such as Google, Facebook, YouTube, Netflix, etc.) collect significant amounts of user data (such as email addresses, phone numbers, etc.) for various purposes, most notably for building their advertising platforms through which advertisers can target platform users. This data that uniquely identify users may pose a severe threat

*Reprinted with permission from [36] "The Role of Coded Side Information in Single-Server Private Information Retrieval," by A. Heidarzadeh, F. Kazemi, and A. Sprintson, Jan 2021. IEEE Transactions on Information Theory, vol. 67, no. 1, pp. 25-44. Copyright © by IEEE.

to users' privacy if accessed by untrusted parties, and thus is a major concern for online users who may unknowingly reveal critical personal information (such as political proclivity, medical conditions, etc.) via daily online activities. As a result, protecting the privacy of user's accessed data is of paramount importance for the online service providers. This well-acknowledged concern has led to many interesting theoretical problems such as PIR problem, as demonstrated by prior studies [22]. In addition to its direct applications in providing private content access in cloud storage systems, PIR is intimately related to many fundamental problems in cryptography, coding theory, and network coding. Therefore, PIR represents an important focal point to tackle significant challenges across these fields.

Related Work

Initiated by the work of Kadhe *et al.* in [22] and [37], the information-theoretic PIR was recently extended to the settings wherein the user has a random subset of messages in the database as side information, and the identities of side information messages are unknown to the server(s) [21–24, 29, 32, 33, 37, 79]. (Some other types of side information, not closely related to our work, were also studied, see, e.g., [26–28, 80].) Three different notions of privacy were considered: (i) (W, S)-*privacy*, where both the identities of the requested messages (denoted by the index set W) and the identities of the side information messages (denoted by the index set S) must be protected [22–24, 29, 37, 79]; (ii) *joint* W-*privacy*, where only the identities of the requested messages (and not necessarily the identities of the side information messages) must be protected [21–23, 33, 37, 79]; and (iii) *individual* W-*privacy*, where the identity of each requested message must be protected individually (but not necessarily jointly) [32]. In single-message PIR, where the user wants to retrieve one message only, the notions of joint and individual W-privacy, referred to as W-*privacy* for brevity, are equivalent. The differences between these two notions of W-privacy in multi-message PIR were studied in [32].

Main Contributions

In this chapter, we focus on the single-server setting of single-message information-theoretic PIR in the presence of a *coded side information*. In Sec. 2.4 and 2.5, we study this problem for the cases in which W-privacy and (W, S)-privacy are required, respectively, where W denotes the index of the requested message, and S denotes the index set of the messages in the support set of the coded side information. In the next chapter, we extends these works to the multi-server setting in Sec. 3.4 and 3.5, respectively.

In this problem, there is a single server storing a database of K independently and uniformly distributed messages, and there is a user who is interested in retrieving a single message from the server. The user initially knows a linear coded combination of a subset of M messages in the database, where the identities of the messages in the support set of the user's coded side information as well as their coding coefficients are initially unknown to the server(s). This setting can be motivated by several practical scenarios. For instance, the user may have obtained a coded side information via overhearing in a wireless network, or on-the-fly recording of a random linear combination of messages being broadcast by an information source, or from a trusted agent, e.g., an entity who makes profit by offering privacy to users, with limited knowledge about the database, or from the information which is locally stored, e.g., using an erasure code, in the user's cache of limited size. Recently, inspired by [22], a group of researchers from Google in [81] used the idea of a coded side information in a new single-server PIR scheme, which leverages both the information-theoretic and computational PIR.

The problem is to design a protocol for generating the user's query and the server's answer which satisfy one of the following two privacy conditions: (W, S)-privacy, i.e., the privacy of both the requested message and the messages in the support set of the coded side information must be preserved, or W-privacy, i.e., only the privacy of the requested

message needs to be protected. We refer to this problem as *PIR with Private Coded Side Information (PIR-PCSI)* or *PIR with Coded Side Information (PIR-CSI)* when (W, S)-privacy or W-privacy is required, respectively.

Depending on whether the support set of the user's coded side information includes the user's demand or not, we consider two different models for each of the PIR-PCSI and PIR-CSI problems. In the first model, referred to as *Model I*, the demand does not belong to the support set of the coded side information, whereas in the second model, referred to as *Model II*, the demand belongs to the support set of the coded side information. We refer to the PIR-PCSI (or PIR-CSI) problem under Model I and Model II as *PIR-PCSI–I* (or *PIR-CSI–I*) and *PIR-PCSI–II* (or *PIR-CSI–II*), respectively.

For each of these settings, we define the *capacity* as the ratio of the number of information bits in a message to the minimum number of information bits downloaded from the server(s) over all protocols that satisfy the privacy condition. We similarly define the *scalar-linear capacity* of each setting, except when the minimum is taken over all scalar-linear protocols that satisfy the privacy condition. Note that in a scalar-linear protocol, the user only downloads a number of scalar-linear combinations (i.e., linear combinations with scalar coding coefficients) of the database messages. In contrast, in a general protocol, the user may download arbitrary (non-linear) functions of the database messages.

In this work, our goal is to characterize the capacity and the scalar-linear capacity of each of the PIR-PCSI and PIR-CSI settings, and design a capacity-achieving protocol for each of these settings. We focus on the settings in which (i) the parameter M is fixed and known to the server; (ii) the indices of the M messages in the support set of the coded side information are chosen uniformly at random; (iii) the index of the demand message is chosen uniformly at random from either outside (Model I) or within (Model II) the index set of messages in the coded side information; and (iv) the (nonzero) coding coefficients in the coded side information are chosen uniformly at random.

The capacity results for the single-server PIR-PCSI and single-server PIR-CSI are respectively summarized in Tables 2.1 and 2.2. The main contributions of this work in each of these settings are summarized as follows.

Single-Server PIR-PCSI

For the single-server PIR-PCSI–I setting, we prove that the capacity and the scalar-linear capacity are both given by $(K-M)^{-1}$ for any $1 \leq M \leq K-1$. This is interesting because, as shown in [22, Theorem 2], the capacity of PIR when M randomly chosen messages are available at the user as side information and the (W,S)-privacy is required, referred to as PIR with Private Side Information (PIR-PST), is equal to $(K-M)^{-1}$. This shows that for achieving (W,S)-privacy, even *one* random linear coded combination of a random subset of M messages is as efficient as M randomly chosen messages separately, as side information. The converse proof is based on the same argument.

For the single-server PIR-PCSI–II setting, we prove that the scalar-linear capacity for any value of $2 \leq M \leq K$ and the capacity for any value of $\frac{K+1}{2} < M \leq K$ are given by $(K-M+1)^{-1}$, whereas the capacity for any value of $2 \leq M \leq \frac{K+1}{2}$ remains open. This shows that when the user knows only *one* random linear coded combination whose support set consists of the requested message along with $M-1$ other randomly chosen messages, achieving (W,S)-privacy is no more costly than that when the user knows $M-1$ randomly chosen (uncoded) messages, different from the requested message.

Single-Server PIR-CSI

For the single-server PIR-CSI–I setting, we prove that the capacity and the scalar-linear capacity are given by $\lceil \frac{K}{M+1} \rceil^{-1}$ for any $0 \leq M < K$. Interestingly, this is the same as the capacity of PIR with M randomly chosen messages as side information [22, Theorem 1]. For the PIR-CSI–II setting, we prove that the capacity and the scalar-linear capacity are equal to 1 for $M=2$ and $M=K$, and are equal to $\frac{1}{2}$ for any $3 \leq M \leq K-1$. This

Privacy Condition	(W, S)-Privacy	
Model	$W \notin S$ (PIR-PCSI–I)	$W \in S$ (PIR-PCSI–II)
Parameters	$1 \leq M \leq K-1$	$2 \leq M \leq K$
Capacity	$(K-M)^{-1}$	$(K-M+1)^{-1}$ for $M > \frac{K+1}{2}$ Open for $M \leq \frac{K+1}{2}$
Scalar-Linear Capacity		$(K-M+1)^{-1}$
Achievability Scheme	Specialized GRS Code	Modified Specialized GRS Code

Table 2.1: Summary of our main results for single-server PIR-PCSI

Privacy Condition	W-Privacy	
Model	$W \notin S$ (PIR-CSI–I)	$W \in S$ (PIR-CSI–II)
Parameters	$1 \leq M \leq K-1$	$2 \leq M \leq K$
Capacity	$\lceil \frac{K}{M+1} \rceil^{-1}$	1 for $M = 2$, $M = K$ $\frac{1}{2}$ for $3 \leq M \leq K-1$
Scalar-Linear Capacity		
Achievability Scheme	Modified Partition-and-Code	Randomized Selection-and-Code

Table 2.2: Summary of our main results for single-server PIR-CSI

result is particularly interesting because, unlike the previous settings, the gap between the capacity and the trivial capacity upper bound 1 is a constant, regardless of the size of support set of the side information (M).

The converse proofs are based on new information-theoretic arguments. These arguments are tailored to the setting of single-server PIR and are different from the proof techniques being commonly used in the multi-server PIR settings. In particular, the main ingredients in the proofs are a necessary condition for (W, S)-privacy and a necessary condition for W-privacy, which reveal the combinatorial nature of the problem of single-

server PIR in the presence of (uncoded or coded) side information. In addition, our converse proofs for the PIR-PCSI–I and PIR-CSI–I settings serve as alternative information-theoretic proofs for the results in [22] which were proven using index coding arguments.

The achievability proofs are based on novel scalar-linear PIR-PCSI and PIR-CSI protocols. In particular, the proposed PIR-PCSI–I and PIR-PCSI–II protocols, termed the *Specialized GRS Code protocol* and the *Modified Specialized GRS Code protocol*, rely on the Generalized Reed-Solomon (GRS) codes that contain a specific codeword, depending on the index of the requested message as well as the indices of the messages in the support set of the coded side information and their coding coefficients.

The proposed protocol for the PIR-CSI–I setting, termed the *Modified Partition-and-Code (MPC) protocol*, is inspired by recently proposed Partition-and-Code with Interference Alignment protocol in [82] for single-server private computation with uncoded side information. The MPC protocol also generalizes the Partition-and-Code protocol of [22] for single-server PIR with uncoded side information. It is noteworthy that we originally introduced a different PIR-CSI–I protocol in [30], termed *Randomized Partitioning (RP) protocol*, which is also capacity-achieving.

For the PIR-CSI–II setting, we propose a protocol, termed the *Randomized Selection-and-Code protocol*, which is based on the idea of randomizing the structure of the user's query and the server's answer (instead of always using a fixed structure for query/answer). We introduced this idea in [30] for the first time, and Tian *et al.*, concurrently and independently, used a similar idea in [83] for multi-server PIR without side information.

2.2 Problem Setup and Formulation

2.2.1 Basic Notation

Throughout this work, we denote random variables by bold-face letters and their realizations by regular letters. The functions $\mathbb{P}(\cdot)$, $\mathbb{P}(\cdot|\cdot)$, $H(\cdot)$, $H(\cdot|\cdot)$, and $I(\cdot;\cdot|\cdot)$ denote

probability, conditional probability, (Shannon) entropy, conditional entropy, and conditional mutual information, respectively. Let $\mathbb{F}_q$ be a finite field for a prime power q, and let $\mathbb{F}_q^\times \triangleq \mathbb{F}_q \setminus \{0\}$ be the multiplicative group of $\mathbb{F}_q$. Let $\mathbb{F}_{q^l}$ be an extension field of $\mathbb{F}_q$ for an integer $l \geq 1$, and let $L \triangleq l \log_2 q$. The parameters q and l are referred to as the *base-field size* and the *field-extension degree*, respectively. Let $K \geq 1$ and $1 \leq M \leq K$ be two integers. Let $\mathcal{K} \triangleq \{1, \ldots, K\}$. We denote by $\mathcal{S}$ the set of all M-subsets (i.e., all subsets of size M) of $\mathcal{K}$, and denote by $\mathcal{C}$ the set of all sequences of size M (i.e., all length-M sequences) with elements from $\mathbb{F}_q^\times$. Note that $|\mathcal{S}| = \binom{K}{M}$ and $|\mathcal{C}| = (q-1)^M$.

2.2.2 Setup and Assumptions

There is a server that stores a set of K messages $X_1, \ldots, X_K$, denoted by $X_\mathcal{K} \triangleq \{X_1, \ldots, X_K\}$, where $\mathbf{X}_i$'s are independently and uniformly distributed over $\mathbb{F}_{q^l}$, i.e., $H(\mathbf{X}_i) = L$ for $i \in \mathcal{K}$ and $H(\mathbf{X}_\mathcal{K}) = KL$, where $\mathbf{X}_\mathcal{K} \triangleq \{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$.

There is a user who wants to retrieve a message X_W for some $W \in \mathcal{K}$ from the server, and knows a linear combination $Y^{[S,C]} \triangleq \sum_{i \in S} c_i X_i$ on the messages $X_S \triangleq \{X_i : i \in S\}$, for some $S \triangleq \{i_1, \ldots, i_M\} \in \mathcal{S}$ and $C \triangleq \{c_{i_1}, \ldots, c_{i_M}\} \in \mathcal{C}$. We refer to X_W as the *demand*, W as the *demand index*, X_S as the *side information support set*, S as the *side information support index set*, M as the *side information support size*, and $Y^{[S,C]}$ as the *(coded) side information*.

We assume that $\mathbf{S}$ and $\mathbf{C}$ are uniformly distributed over $\mathcal{S}$ and $\mathcal{C}$, respectively. Also, we consider two different models for the conditional distribution of $\mathbf{W}$ given $\mathbf{S} = S$:

Model I: $\mathbf{W}$ is uniformly distributed over $\mathcal{K} \setminus S$,

$$\mathbb{P}(\mathbf{W} = W | \mathbf{S} = S) = \begin{cases} \frac{1}{K-M}, & W \in \mathcal{K} \setminus S, \\ 0, & \text{otherwise;} \end{cases}$$

Model II: $\mathbf{W}$ is uniformly distributed over S,

$$\mathbb{P}(\mathbf{W} = W|\mathbf{S} = S) = \begin{cases} \frac{1}{M}, & W \in S, \\ 0, & \text{otherwise.} \end{cases}$$

For both Models I and II, $\mathbf{W}$ is distributed uniformly over $\mathcal{K}$.

Let $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}$ be an indicator random variable such that that $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 0$ if $\mathbf{W} \notin \mathbf{S}$, and $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 1$ otherwise. Note that $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 0$ in Model I, and $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 1$ in Model II.

We assume that the server knows the underlying model (i.e., whether $\mathbf{W} \notin \mathbf{S}$ or $\mathbf{W} \in \mathbf{S}$), the side information support size M, the distributions of $\mathbf{S}$ and $\mathbf{C}$, and the conditional distribution of $\mathbf{W}$ given $\mathbf{S}$, in advance; whereas the realizations W, S, C are unknown to the servers in advance.

2.2.3 Privacy and Recoverability Conditions

For any given W, S, C, to retrieve X_W, the user sends to the server the query $Q^{[W,S,C]}$, which is a (potentially stochastic) function of W, S, C.* For simplifying the notation, we denote $\mathbf{Q}^{[\mathbf{W},\mathbf{S},\mathbf{C}]}$ by $\mathbf{Q}$. The query must satisfy one of the following two privacy conditions:

(i) both the user's demand index and side information support index set must be protected from the servers;

(ii) only the user's demand index (and not necessarily the side information support index set) must be protected from the servers.

The condition (i) is referred to as the (W, S)-*privacy condition*, and the condition (ii) is referred to as the W-*privacy condition*. (Note that (W, S)-privacy is a stronger condition than W-privacy.)

*In general, the query may also depend on the content of the side information—notwithstanding, in this work we focus on queries that are "universal" in the sense that any such query achieves privacy for all realizations of the messages.

The (W,S)-privacy condition implies that $(\mathbf{W},\mathbf{S})$ and $\mathbf{Q}$ must be conditionally independent given $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}$, that is,

$$I(\mathbf{W},\mathbf{S};\mathbf{Q}|\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}})=0.$$

The W-privacy condition implies that $\mathbf{W}$ and $\mathbf{Q}$ must be conditionally independent given $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}$,

$$I(\mathbf{W};\mathbf{Q}|\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}})=0.$$

Equivalently, for a given $\theta\in\{0,1\}$, when (W,S)-privacy is required, it must hold that

$$\begin{aligned}&\mathbb{P}(\mathbf{W}=W^*,\mathbf{S}=S^*|\mathbf{Q}=Q^{[W,S,C]},\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}=\theta)\\&=\mathbb{P}(\mathbf{W}=W^*,\mathbf{S}=S^*|\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}=\theta)\end{aligned}$$

for all $W^*\in\mathcal{K}$ and $S^*\in\mathcal{S}$, and when W-privacy is required, it must hold that

$$\begin{aligned}&\mathbb{P}(\mathbf{W}=W^*|\mathbf{Q}=Q^{[W,S,C]},\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}=\theta)\\&=\mathbb{P}(\mathbf{W}=W^*|\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}=\theta)\end{aligned}$$

for all $W^*\in\mathcal{K}$. The mutual information based definitions of the (W,S)-privacy and W-privacy conditions will be used in the converse proofs, whereas their probability based counterparts will be used in the achievability proofs.

Upon receiving $Q^{[W,S,C]}$, the server sends to the user an answer $A^{[W,S,C]}$, which is a (deterministic) function of the query $Q^{[W,S,C]}$, the indicator variable $\mathbb{1}_{\{W\in S\}}$, and the messages in $X_{\mathcal{K}}$. For simplifying the notation, we denote $\mathbf{A}^{[\mathbf{W},\mathbf{S},\mathbf{C}]}$ by $\mathbf{A}$. Note that $(\mathbf{W},\mathbf{S},\mathbf{C})\rightarrow(\mathbf{Q},\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}},\mathbf{X}_{\mathcal{K}})\rightarrow\mathbf{A}$ forms a Markov chain, and $H(\mathbf{A}|\mathbf{Q},\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}},\mathbf{X}_{\mathcal{K}},\mathbf{W},\mathbf{S},\mathbf{C})=0$ holds.

The answer $A^{[W,S,C]}$ along with $Q^{[W,S,C]}$, $\mathbb{1}_{\{W\in S\}}$, $Y^{[S,C]}$, and W, S, C must enable the user to retrieve the demand X_W. That is, it must hold that

$$H(\mathbf{X_W}|\mathbf{A}, \mathbf{Q}, \mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}, \mathbf{Y}^{[\mathbf{S},\mathbf{C}]}, \mathbf{W}, \mathbf{S}, \mathbf{C}) = 0.$$

We refer to this condition as the *recoverability condition*.

2.2.4 PIR-PCSI and PIR-CSI Problems

For each type of privacy and for each model, the problem is to design a protocol for generating a query $Q^{[W,S,C]}$ (and the corresponding answer $A^{[W,S,C]}$, given $Q^{[W,S,C]}$, $\mathbb{1}_{\{W\in S\}}$, and $X_{\mathcal{K}}$) for any given W, S, C, such that both the privacy and recoverability conditions are satisfied. Note that the protocol is assumed to be known at the server. When (W, S)-privacy is required, we refer to this problem as *Private Information Retrieval (PIR) with Private Coded Side Information (PIR-PCSI)*, and when W-privacy is required we refer to this problem as *PIR with Coded Side Information (PIR-CSI)*.

The PIR-PCSI problem under Model I (or Model II) is referred to as the *PIR-PCSI–I* (or *PIR-PCSI–II*) setting; and the PIR-CSI problem under Model I (or Model II) is referred to as the *PIR-CSI–I* (or *PIR-CSI–II*) setting. A protocol for generating queries/answers for the PIR-PCSI–I (or PIR-PCSI–II) setting is referred to as a *PIR-PCSI–I* (or *PIR-PCSI–II*) *protocol*. A *PIR-CSI–I* (or *PIR-CSI–II*) *protocol* is defined similarly. A protocol is said to be *scalar-linear* if the server's answer to the user's query consists only of the scalar-linear combinations of the messages in $X_{\mathcal{K}}$. This is in contrast to general protocols where the user can download arbitrary (non-linear) functions of the messages in $X_{\mathcal{K}}$ from the server.

2.2.5 Capacity and Scalar-Linear Capacity

The *rate of a PIR-PCSI–I (or PIR-PCSI–II) protocol* is defined as the ratio of the entropy of a message, i.e., L, to the conditional entropy of $\mathbf{A}^{[\mathbf{W},\mathbf{S},\mathbf{C}]}$ given that $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 0$

(or $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 1$). The *rate of a PIR-CSI–I (or PIR-CSI–II) protocol* is defined similarly.

The *capacity of PIR-PCSI–I (or PIR-PCSI–II) setting* is defined as the supremum of rates over all PIR-PCSI–I (or PIR-PCSI–II) protocols and over all base-field sizes q and all field-extension degrees l; and the *capacity of PIR-CSI–I (or PIR-CSI–II) setting* is defined similarly.

The *scalar-linear capacity of PIR-PCSI–I (or PIR-PCSI–II) setting* is defined as the supremum of rates over all scalar-linear PIR-PCSI–I (or PIR-PCSI–II) protocols and over all q and l. The *scalar-linear capacity of PIR-CSI–I (or PIR-CSI–II) setting* is defined similarly.[†]

2.2.6 Problem Statement

In this work, our goal is to derive upper bounds on the capacity and the scalar-linear capacity of the single-server PIR-PCSI–I, PIR-PCSI–II, PIR-CSI–I, and PIR-CSI–II settings, and to design protocols that achieve the corresponding upper-bounds.

2.3 Necessary Conditions

The following two lemmas give a necessary condition for (W, S)-privacy and W-privacy, respectively. These simple but powerful lemmas are the key components in the converse proofs of our main results.

Lemma 1. *For (W, S)-privacy, for a given $\theta \in \{0, 1\}$, for any $W^* \in \mathcal{K}$ and $S^* \in \mathcal{S}$ with $\mathbb{1}_{\{W^*\in S^*\}} = \theta$, there must exist $C^* \in \mathcal{C}$ such that*

$$H(\mathbf{X}_{W^*}|\mathbf{A}, \mathbf{Q}, \mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = \theta, \mathbf{Y}^{[S^*,C^*]}) = 0.$$

[†]Although our definitions of capacity and scalar-linear capacity are independent of the base-field size q and the field-extension degree l, these quantities may depend on q and l in general. In this work, we show that the capacity and the scalar-linear capacity of the PIR-PCSI settings are achievable so long as $q \geq K$ and $l \geq 1$; and depending on the parameters K, M and the model (I or II), the capacity and the scalar-linear capacity of the PIR-CSI settings are achievable so long as $q \geq 2$ or $q \geq 3$ and $l \geq 1$.

Proof. The proof is by the way of contradiction. For a given $\theta \in \{0,1\}$, consider an arbitrary $W^* \in \mathcal{K}$ and an arbitrary $S^* \in \mathcal{S}$ such that $\mathbb{1}_{\{W^* \in S^*\}} = \theta$. Suppose that there does not exist any $C^* \in \mathcal{C}$ such that $H(\mathbf{X}_{W^*}|\mathbf{A}, \mathbf{Q}, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = \theta, \mathbf{Y}^{[S^*, C^*]}) = 0$. If W^* and S^* are respectively the user's demand index and side information support index set, no matter what the user's side information $Y^{[S^*,\cdot]}$ is, the user cannot recover X_{W^*} given the answer, query, and the side information $Y^{[S^*,\cdot]}$. This violates the recoverability condition. Thus, W^* and S^* cannot be the user's demand index and side information support index set, respectively. This obviously violates the (W, S)-privacy condition, because given the query, every $W^* \in \mathcal{K}$ and every $S^* \in \mathcal{S}$ such that $\mathbb{1}_{\{W^* \in S^*\}} = \theta$ must be equally likely to be the user's demand index and side information support index set, respectively. □

Lemma 2. *For W-privacy, for a given $\theta \in \{0,1\}$, for any $W^* \in \mathcal{K}$, there must exist $S^* \in \mathcal{S}$ with $\mathbb{1}_{\{W^* \in S^*\}} = \theta$ and $C^* \in \mathcal{C}$ such that*

$$H(\mathbf{X}_{W^*}|\mathbf{A}, \mathbf{Q}, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = \theta, \mathbf{Y}^{[S^*, C^*]}) = 0.$$

Proof. The proof is similar to the proof of Lemma 1 except that the W-privacy condition is used instead of the (W, S)-privacy condition, and is omitted for brevity. □

2.4 Single-Server PIR with Private Coded Side Information (PIR-PCSI)

In this section, we present our main results for the single-server PIR-PCSI–I and single-server PIR-PCSI–II settings in Section 2.4.1 and Section 2.4.2, respectively. The capacity and the scalar-linear capacity of the single-server PIR-PCSI–I setting (for all $1 \leq M \leq K-1$) are characterized in Theorem 1, and the capacity (for all $\frac{K+1}{2} < M \leq K$) and the scalar-linear capacity (for all $2 \leq M \leq K$) of the single-server PIR-PCSI–II setting are characterized in Theorem 2. For any $2 \leq M \leq \frac{K+1}{2}$, the capacity of the PIR-PCSI–II setting, which we conjecture to be the same as the scalar-linear capacity, remains open. The proofs are provided.

2.4.1 Single-Server PIR-PCSI-I

Theorem 1. *For the single-server PIR-PCSI–I setting with K messages and side information support size M, the capacity and the scalar-linear capacity are given by $(K-M)^{-1}$ for all $1 \leq M \leq K-1$.*

The converse follows directly from the result of [22, Theorem 2], which was proven using an index coding argument, for single-server single-message PIR with (uncoded) side information when (W, S)-privacy is required. In this work, we provide an alternative proof of converse by upper bounding the rate of any PIR-PCSI–I protocol using the information-theoretic arguments (see Section 2.4.1.1). The key component of the proof is the necessary condition for (W, S)-privacy, stated in Lemma 1.

The achievability proof relies on a new scalar-linear PIR-PCSI–I protocol, termed the *Specialized GRS Code protocol*, which achieves the rate $(K-M)^{-1}$ (see Section 2.4.1.2). This protocol is based on the Generalized Reed-Solomon (GRS) codes that contain a specific codeword depending on W, S, C.

Remark 1. As shown in [22], when there is a single server storing K independent and identically distributed messages, and there is a user that knows M randomly chosen (uncoded) messages as their side information and demands a single message not in their side information, in order to guarantee (W, S)-privacy, the minimum download cost is $(K-M)L$, where L is the entropy of a message. Surprisingly, this result matches the result of Theorem 1. This shows that, when compared to having M random messages separately as side information, for achieving (W, S)-privacy there will be no additional loss in capacity even if only *one* random linear coded combination of M random messages is known by the user.

Proof of Theorem 1

2.4.1.1 Converse

As shown in [22] using an index-coding argument, when (W, S)-privacy is required, the capacity of PIR with M uncoded messages as side information is given by $(K-M)^{-1}$. The capacity of the PIR-PCSI–I setting is upper bounded by this quantity. This proves the converse for Theorem 1. In this section, we present an alternative information-theoretic proof for the general case, which also proves the converse for the scalar-linear case.

Lemma 3. *For any $1 \leq M \leq K-1$, the (scalar-linear) capacity of the PIR-PCSI–I setting is upper bounded by $(K-M)^{-1}$.*

Proof. In the following, all entropies are conditional on the event $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 0$, and we remove this event from the conditions everywhere, for the ease of notation. We need to show that $H(\mathbf{A}) \geq (K-M)L$.

Take arbitrary W, S, C (and $\mathbf{Y} \triangleq \mathbf{Y}^{[S,C]}$) such that $W \notin S$. Then, we have

$$\begin{aligned} H(\mathbf{A}) &\geq H(\mathbf{A}|\mathbf{Q},\mathbf{Y}) && (2.1)\\ &= H(\mathbf{A}|\mathbf{Q},\mathbf{Y}) + H(\mathbf{X}_W|\mathbf{A},\mathbf{Q},\mathbf{Y}) && (2.2)\\ &= H(\mathbf{A},\mathbf{X}_W|\mathbf{Q},\mathbf{Y}) && (2.3)\\ &= H(\mathbf{X}_W|\mathbf{Q},\mathbf{Y}) + H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W) && (2.4)\\ &= H(\mathbf{X}_W) + H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W) && (2.5) \end{aligned}$$

where (2.1) follows since conditioning does not increase the entropy; (2.2) holds because by the recoverability condition, $H(\mathbf{X}_W|\mathbf{A},\mathbf{Q},\mathbf{Y}) = 0$; (2.3) and (2.4) follow from the chain rule of entropy; and (2.5) follows from $H(\mathbf{X}_W|\mathbf{Q},\mathbf{Y}) = H(\mathbf{X}_W)$ since $\mathbf{X}_W$ is independent of $(\mathbf{Q},\mathbf{Y})$ (noting that $W \notin S$).

For the case that $W \cup S = \mathcal{K}$ (i.e., $M = K - 1$), we have $H(\mathbf{A}) \geq H(\mathbf{X}_W) = L$ (by using the first term in (2.5)), as was to be shown. For the case that $W \cup S \neq \mathcal{K}$, we proceed by lower bounding the second term in (2.5), $H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W)$. By Lemma 1, for each $i \in \mathcal{K} \setminus (W \cup S)$, there exists $C_i \in \mathcal{C}$ (and $\mathbf{Y}_i \triangleq \mathbf{Y}^{[S,C_i]}$) such that $H(\mathbf{X}_i|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_i) = 0$. Let I be a maximal subset of $\mathcal{K} \setminus (W \cup S)$ such that $\mathbf{Y}$ and $\mathbf{Y}_I \triangleq \{\mathbf{Y}_i\}_{i \in I}$ are linearly independent. (Note that $|I| \leq |S| - 1 = M - 1$.) Let $\mathbf{X}_I \triangleq \{\mathbf{X}_i\}_{i \in I}$. Then, we have

$$\begin{aligned}
H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) &\geq H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I) \\
&\geq H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I) \\
&\quad + H(\mathbf{X}_I|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I) \qquad (2.6) \\
&= H(\mathbf{A}, \mathbf{X}_I|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I) \\
&= H(\mathbf{X}_I|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I) \\
&\quad + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I, \mathbf{X}_I) \\
&= H(\mathbf{X}_I) \\
&\quad + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I, \mathbf{X}_I) \qquad (2.7)
\end{aligned}$$

where (2.6) holds because $H(\mathbf{X}_i|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_i) = 0$ for all $i \in I$ (by assumption); and (2.7) holds since $\mathbf{X}_I$ is independent of $(\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I)$ by construction (noting that I and $W \cup S$ are disjoint). The first term in (2.7), $H(\mathbf{X}_I)$, is lower bounded by $|I|L \geq 0$. Thus, in order to further lower bound $H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W)$, we need to lower bound the second term in (2.7), $H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_I, \mathbf{X}_I)$. By the maximality of I, for each $j \in J \triangleq \mathcal{K} \setminus (W \cup S \cup I)$, there exists $C_j \in \mathcal{C}$ (and $\mathbf{Y}_j \triangleq \mathbf{Y}^{[S,C_j]}$, which is linearly dependent on $\mathbf{Y}$ and $\mathbf{Y}_I$) such that $H(\mathbf{X}_j|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_j) = 0$, and as a consequence, $H(\mathbf{X}_j|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{Y}_I) = 0$. (Note that $|J| = K - M - 1 - |I|$.) Let $\mathbf{X}_J \triangleq \{X_j\}_{j \in J}$. Then, we can write

$$
\begin{aligned}
&H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I)\\
&\quad= H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I)\\
&\qquad+ H(\mathbf{X}_J|\mathbf{A},\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I) \qquad (2.8)\\
&\quad= H(\mathbf{A},\mathbf{X}_J|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I)\\
&\quad= H(\mathbf{X}_J|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I)\\
&\qquad+ H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I,\mathbf{X}_J)\\
&\quad\geq H(\mathbf{X}_J) \qquad (2.9)
\end{aligned}
$$

where (2.8) holds since $H(\mathbf{X}_j|\mathbf{A},\mathbf{Q},\mathbf{Y},\mathbf{Y}_I)=0$ for all $j\in J$ (by assumption); and (2.9) holds because $\mathbf{X}_J$ and $(\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_I,\mathbf{X}_I)$ are independent by construction (noting that J and $W\cup S\cup I$ are disjoint). Putting (2.5), (2.7), and (2.9) together, $H(\mathbf{A})\geq H(\mathbf{X}_W)+H(\mathbf{X}_I)+H(\mathbf{X}_J)=L+|I|L+(K-M-1-|I|)L=(K-M)L$, as was to be shown.□

2.4.1.2 Achievability

In this section, we propose a scalar-linear PIR-PCSI–I protocol that achieves the rate $(K-M)^{-1}$. The proposed protocol requires a base-field size $q\geq K$ (and arbitrary field-extension degree $l\geq 1$) where the messages X_i's are elements from $\mathbb{F}_{q^l}$.

It is noteworthy that the rate $(K-M)^{-1}$ is not necessarily achievable for $q<K$, and for the special case of scalar-linear schemes, the achievability of this rate is conditional upon the existence of a $(K,K-M)$ maximum distance seperable (MDS) code over $\mathbb{F}_q$ that has a codeword with support $W\cup S$ such that the jth code symbol is non-zero for $j=W$ and it is equal to c_j for each $j\in S$ where c_j is the coefficient of the message X_j in the coded side information $Y^{[S,C]}$. In the following, we show how to design such a code for any $q\geq K$.

Specialized GRS Code Protocol: This protocol consists of three steps as follows:

Step 1: First, the user arbitrarily chooses K distinct elements $\omega_1, \ldots, \omega_K$ from $\mathbb{F}_q$, and constructs a polynomial

$$p(x) \triangleq \prod_{i \in \mathcal{K} \setminus (W \cup S)} (x - \omega_i).$$

Then, the user constructs $K - M$ sequences $Q_1, \ldots, Q_{K-M}$, each of size K, defined as

$$Q_i = \{v_1 \omega_1^{i-1}, \ldots, v_K \omega_K^{i-1}\},$$

where the parameters v_j's are chosen as follows. For each $j \in S$, $v_j = \frac{c_j}{p(\omega_j)}$ where c_j is the coefficient of X_j in $Y^{[S,C]}$; and for each $j \notin S$, v_j is chosen at random from $\mathbb{F}_q^{\times}$.

The user then sends to the server the query $Q^{[W,S,C]} = \{Q_1, \ldots, Q_{K-M}\}$.

Note that the jth element in the set Q_i can be thought of as the entry (i, j) of a $(K - M) \times K$ matrix $G \triangleq [g_1^T, \ldots, g_{K-M}^T]^T$, which generates a $(K, K - M)$ GRS code with distinct parameters $\omega_1, \ldots, \omega_K$ and non-zero multipliers $v_1, \ldots, v_K$ [84]. This construction ensures that such a GRS code has a specific codeword, namely $\sum_{i=1}^{K-M} p_i g_i$ where p_i is the coefficient of x^{i-1} in the expansion of the polynomial $p(x) = \sum_{i=1}^{K-M} p_i x^{i-1}$, with support $W \cup S$ such that the jth code symbol is non-zero for $j = W$, and it is equal to c_j for each $j \in S$. This observation is the chief idea in the proof of the recoverability condition for the proposed protocol.

Step 2: By using Q_i's, the server computes A_i's, defined as $A_i = \sum_{j=1}^{K} v_j \omega_j^{i-1} X_j$, and it sends the answer $A^{[W,S,C]} = \{A_1, \ldots, A_{K-M}\}$ to the user.

Note that A_i's are the parity check equations of a (K, M) GRS code which is the dual code of the GRS code generated by the matrix G defined earlier.

Step 3: Upon receiving the answer, the user retrieves X_W by subtracting off the contribution of the side information $Y^{[S,C]}$ from $\sum_{i=1}^{K-M} p_i A_i = c_W X_W + \sum_{i \in S} c_i X_i$.

Example 1. Consider a scenario where the server has $K = 4$ messages $X_1, \ldots, X_4 \in \mathbb{F}_5$, and the user demands the message X_1 and has a coded side information $Y = X_2 + X_3$ with support size $M = 2$. For this example, $W = 1$, $S = \{2, 3\}$, and $C = \{c_2, c_3\} = \{1, 1\}$. First, the user chooses $K = 4$ distinct elements from $\mathbb{F}_5$, say $(\omega_1, \omega_2, \omega_3, \omega_4) = (0, 1, 2, 3)$.

Then, the user constructs the polynomial

$$p(x) = \prod_{i \notin W \cup S} (x - \omega_i) = x - \omega_4 = x + 2.$$

Note that $p(x) = p_1 + p_2 x = 2 + x$. The user then computes v_j for $j \in S$, i.e., v_2 and v_3, by setting $v_2 = \frac{c_2}{p(\omega_2)} = 2$ and $v_3 = \frac{c_3}{p(\omega_3)} = 4$, and chooses v_j for $j \notin S$, i.e., v_1 and v_4, at random (from $\mathbb{F}_5^{\times}$). Suppose that the user chooses $v_1 = 1$ and $v_4 = 2$. Then, the user constructs $K - M = 2$ sequences $Q_1 = \{v_1, \ldots, v_4\} = \{1, 2, 4, 2\}$ and $Q_2 = \{v_1\omega_1, \ldots, v_4\omega_4\} = \{0, 2, 3, 1\}$. The user sends the query $Q = \{Q_1, Q_2\}$ to the server.

The server computes $A_1 = \sum_{j=1}^{4} v_j X_j = X_1 + 2X_2 + 4X_3 + 2X_4$ and $A_2 = \sum_{j=1}^{4} v_j \omega_j X_j = 2X_2 + 3X_3 + X_4$, and sends the answer $A = \{A_1, A_2\}$ back to the user. Then, the user computes $\sum_{i=1}^{2} p_i A_i = 2A_1 + A_2 = 2X_1 + X_2 + X_3$, and recovers X_1 by subtracting off $Y = X_2 + X_3$.

For this example, the rate of the proposed protocol is $1/2$.

Note that the server knows the protocol, including the parameters $\omega_1, \ldots, \omega_4$, and can compute the multipliers $v_1, \ldots, v_4$, given the query. Since the side information coefficients c_2 and c_3 are uniformly distributed, the server finds each of the polynomials $x - \omega_1 = x$, $x - \omega_2 = 4 + x$, $x - \omega_3 = 3 + x$, and $x - \omega_4 = 2 + x$ equally likely to be the polynomial $p(x) = p_1 + p_2 x$, constructed in Step 1 of the protocol. Since the server knows that by the protocol the user requires the linear combination $p_1 A_1 + p_2 A_2$ to recover the demand, from the server's perspective, each of the linear combinations $Z_1 = A_2$, $Z_2 = 4A_1 + A_2$, $Z_3 = 3A_1 + A_2$, $Z_4 = 2A_1 + A_2$, i.e., $Z_1 = 2X_2 + 3X_3 + X_4$, $Z_2 = 4X_1 + 4X_3 + 4X_4$, $Z_3 =$

$3X_1 + 3X_2 + 2X_4$, $Z_4 = 2X_1 + X_2 + X_3$, are equally likely to be the linear combination required by the user. Note, also, that, for each candidate demand index (e.g., $\{1\}$) and each candidate side information support index set (e.g., $\{2, 3\}$), there exists exactly one of the linear combinations $Z_1, \ldots, Z_4$ (e.g., Z_4) from which the candidate demand (e.g., X_1) can be recovered, given some linear combination (e.g., $X_2 + X_3$) of the messages in the candidate side information support set (e.g., X_2, X_3). By these arguments, the server finds every index $i \in \{1, \ldots, 5\}$ and every pair of indices $\{i_1, i_2\}$ such that $i \notin \{i_1, i_2\}$ equally likely to be the user's demand index and side information support index set, respectively. This confirms that the proposed protocol achieves the (W, S)-privacy requirement in this example.

Lemma 4. *The Specialized GRS Code protocol is a scalar-linear PIR-PCSI–I protocol, and it achieves the rate* $(K - M)^{-1}$.

Proof. Since the matrix G, defined in Step 1 of the Specialized GRS Code protocol, generates a $(K, K - M)$ GRS code which is an MDS code, the rows of G are linearly independent. Accordingly, $\mathbf{A}_1, \ldots, \mathbf{A}_{K-M}$, defined in Step 2, are linearly independent combinations of the messages in $\mathbf{X}_{\mathcal{K}}$, which are themselves independently and uniformly distributed over $\mathbb{F}_{q^l}$. This implies that $\mathbf{A}_1, \ldots, \mathbf{A}_{K-M}$ are independently and uniformly distributed over $\mathbb{F}_{q^l}$. Since $H(\mathbf{X}_j) = L$ for all $j \in \mathcal{K}$, then $H(\mathbf{A}_i) = L$ for all $i \in \{1, \ldots, K - M\}$. Thus, for all $W \in \mathcal{K}, S \in \mathcal{S}, C \in \mathcal{C}$ such that $W \notin S$, we have $H(\mathbf{A}^{[W,S,C]}) = H(\mathbf{A}_1, \ldots, \mathbf{A}_{K-M}) = \sum_{i=1}^{K-M} H(\mathbf{A}_i) = (K - M)L$. (Note that $H(\mathbf{A}^{[W,S,C]}) = (K - M)L$ does not depend on the realizations W, S, C.) Given that $\mathbf{W} \notin \mathbf{S}$, $\mathbf{W}$ and $\mathbf{S}$ are jointly distributed uniformly, and $\mathbf{C}$ is distributed uniformly (and independently from $(\mathbf{W}, \mathbf{S})$). Thus, $H(\mathbf{A}^{[\mathbf{W},\mathbf{S},\mathbf{C}]}|\mathbf{W} \notin \mathbf{S}) = H(\mathbf{A}^{[W,S,C]}) = (K - M)L$, implying that the rate of the Specialized GRS Code protocol is equal to $L/((K - M)L) = (K - M)^{-1}$.

The scalar-linearity of $\mathbf{A}_i$'s in the messages $\mathbf{X}_j$'s confirms that the Specialized GRS Code protocol is scalar-linear. From the construction, it should also be obvious that the recoverability condition is satisfied. The proof of (W, S)-privacy relies on two facts: (i) the $(K, K-M)$ GRS code, generated by the matrix G, is an MDS code, and hence the minimum (Hamming) weight of a codeword is $K-(K-M)+1 = M+1$; and (ii) there exist the same number of minimum-weight codewords for any support of size $M+1$ [84].

From (i) and (ii), it follows that for any $W^* \in \mathcal{K}, S^* \in \mathcal{S}$ such that $W^* \notin S^*$ (note that $|W^* \cup S^*| = M+1$), the dual code, whose parity check matrix is G, contains the same number of parity check equations with support $W^* \cup S^*$ (i.e., the messages $\{X_i\}_{i \in W^* \cup S^*}$ have non-zero coefficients and the rest of the messages all have zero coefficients). Thus, from the perspective of the server, given the query, every pair (W^*, S^*) is equally likely to be the pair of the user's demand index and the user's side information support index set. This proves the (W, S)-privacy of the Specialized GRS Code protocol. □

2.4.2 Single-Server PIR-PCSI-II

Theorem 2. *For the single-server PIR-PCSI–II setting with K messages and side information support size M, the capacity is given by $(K-M+1)^{-1}$ for all $\frac{K+1}{2} < M \leq K$, and it is lower bounded by $(K-M+1)^{-1}$ for all $2 \leq M \leq \frac{K+1}{2}$. Moreover, the scalar-linear capacity is given by $(K-M+1)^{-1}$ for all $2 \leq M \leq K$.*

The proof of converse for the scalar-linear case is based on a mix of algebraic and information-theoretic arguments (see Section 2.4.2.1), and the converse proof of the general case relies on different information-theoretic arguments. The main ingredient of the proofs is the result of Lemma 1.

The proof of achievability is based on a novel scalar-linear protocol, referred to as the *Modified Specialized GRS Code protocol*—a modified version of the Specialized GRS Code protocol, which achieves the rate $(K-M+1)^{-1}$ (see Section 2.4.2.2).

Remark 2. Interestingly, comparing the results of [22, Theorem 2] and Theorem 2, one can see that when the side information is composed of $M-1$ randomly chosen messages (different from the requested message), (W,S)-privacy cannot be achieved more efficiently than the case in which the side information is only *one* random linear coded combination of M randomly chosen messages including the demand.

Proof of Theorem 2

2.4.2.1 Converse

First, we prove the converse for the scalar-linear case of Theorem 2 for all $2 \leq M \leq K$. The proof is based on a combination of algebraic and information-theoretic arguments.

Lemma 5. *For any $2 \leq M \leq K$, the scalar-linear capacity of the PIR-PCSI–II setting is upper bounded by $(K-M+1)^{-1}$.*

Proof. In the following, all the entropies are conditional on the event $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}=1$, and for simplifying the notation, we remove this event from the conditions. We need to show that $H(\mathbf{A}) \geq (K-M+1)L$.

Let I be the set of all $i \in \mathcal{K}$ such that $H(\mathbf{X}_i|\mathbf{A},\mathbf{Q})=0$. (Note that $0 \leq |I| \leq K$). Let $\mathbf{X}_I \triangleq \{\mathbf{X}_i\}_{i\in I}$. By assumption, $\mathbf{X}_I$ and $\mathbf{Q}$ are independent and $H(\mathbf{X}_I|\mathbf{A},\mathbf{Q})=0$. Then, we have

$$\begin{aligned} H(\mathbf{A}) &\geq H(\mathbf{A}|\mathbf{Q}) \\ &= H(\mathbf{A}|\mathbf{Q}) + H(\mathbf{X}_I|\mathbf{A},\mathbf{Q}) \\ &= H(\mathbf{A},\mathbf{X}_I|\mathbf{Q}) \\ &= H(\mathbf{X}_I|\mathbf{Q}) + H(\mathbf{A}|\mathbf{Q},\mathbf{X}_I) \\ &= H(\mathbf{X}_I) + H(\mathbf{A}|\mathbf{Q},\mathbf{X}_I). \end{aligned} \tag{2.10}$$

If $|I| \geq K - M + 1$, the first term in (2.10), $H(\mathbf{X}_I)$, is lower bounded by $(K - M + 1)L$, and hence, $H(\mathbf{A}) \geq (K - M + 1)L$, as was to be shown. If $0 \leq |I| \leq K - M$, the second term in (2.10), $H(\mathbf{A}|\mathbf{Q}, \mathbf{X}_I)$, can be further lower bounded as follows.

Assume, w.l.o.g., that $I = \{1, \ldots, |I|\}$. (Note that $I = \emptyset$ for $|I| = 0$.) Let $J \triangleq \{1, \ldots, K - M - |I| + 1\}$, and let $S_j \triangleq \{|I| + 1, |I| + j + 1, \ldots, |I| + j + M - 1\}$ for $j \in J$. By Lemma 1, for each $j \in J$, there exists $C_j \in \mathcal{C}$ (and $\mathbf{Y}_j \triangleq \mathbf{Y}^{[S_j, C_j]}$) such that $H(\mathbf{X}_{|I|+1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_j) = 0$. Let $\mathbf{Z}_j \triangleq \mathbf{Y}_j - c_j \mathbf{X}_{|I|+1}$ where c_j is the coefficient of $\mathbf{X}_{|I|+1}$ in $\mathbf{Y}_j$. For any scalar-linear protocol where the answer consists only of scalar-linear combinations of messages in $X_{\mathcal{K}}$, it is easy to see that for each $j \in J$, (i) $H(\mathbf{Z}_j|\mathbf{A}, \mathbf{Q}) = 0$, or (ii) $H(\mathbf{Z}_j + c\mathbf{X}_{|I|+1}|\mathbf{A}, \mathbf{Q}) = 0$ for some $c \in \mathbb{F}_q^\times \setminus \{c_j\}$. (Otherwise, the server learns that W and S cannot be $|I| + 1$ and S_j, respectively. This obviously violates the (W, S)-privacy condition.) In either case (i) or (ii), one can see that $H(\mathbf{Z}_j|\mathbf{A}, \mathbf{Q}, \mathbf{X}_{|I|+1}) = 0$. (Note that this observation, which is the key in the proof of Lemma 5, holds for all scalar-linear schemes, but not necessarily for all vector-linear or non-linear schemes in general. This implies the need for a different proof technique for the general schemes, and an example of such a technique is used in the proof of Lemma 6.) Let $\mathbf{Z}_J \triangleq \{\mathbf{Z}_j\}_{j \in J}$. Then, we have

$$
\begin{aligned}
H(\mathbf{A}|\mathbf{Q}, \mathbf{X}_I) &\geq H(\mathbf{A}|\mathbf{Q}, \mathbf{X}_I, \mathbf{X}_{|I|+1}) \\
&= H(\mathbf{A}|\mathbf{Q}, \mathbf{X}_I, \mathbf{X}_{|I|+1}) \\
&\quad + H(\mathbf{Z}_J|\mathbf{A}, \mathbf{Q}, \mathbf{X}_I, \mathbf{X}_{|I|+1}) \qquad (2.11) \\
&= H(\mathbf{A}, \mathbf{Z}_J|\mathbf{Q}, \mathbf{X}_I, \mathbf{X}_{|I|+1}) \\
&= H(\mathbf{Z}_J|\mathbf{Q}, \mathbf{X}_I, \mathbf{X}_{|I|+1}) \\
&\quad + H(\mathbf{A}|\mathbf{Q}, \mathbf{X}_I, \mathbf{X}_{|I|+1}, \mathbf{Z}_J) \\
&\geq H(\mathbf{Z}_J) \qquad (2.12)
\end{aligned}
$$

where (2.11) holds as $H(\mathbf{Z}_j|\mathbf{A},\mathbf{Q},\mathbf{X}_{|I|+1})=0$ for all $j\in J$ (by assumption); and (2.12) follows because $\mathbf{Z}_J$ is independent of $(\mathbf{Q},\mathbf{X}_I,\mathbf{X}_{|I|+1})$ by construction, noting that $\mathbf{Z}_J$, $\mathbf{X}_I$, and $\mathbf{X}_{|I|+1}$ are linearly independent. By the linear independence of $\mathbf{Z}_j$'s for all $j\in J$, it follows that $H(\mathbf{Z}_J)=(K-M-|I|+1)L$. By (2.10) and (2.12), we get $H(\mathbf{A})\geq H(\mathbf{X}_I)+H(\mathbf{Z}_J)=|I|L+(K-M-|I|+1)L=(K-M+1)L$, as was to be shown.

□

Next, we give an information-theoretic proof of converse for the general case of Theorem 2 for all $\frac{K+1}{2}<M\leq K$. For any $2\leq M\leq\frac{K+1}{2}$, the converse proof remains open.

Lemma 6. *For any $\frac{K+1}{2}<M\leq K$, the capacity of the PIR-PCSI–II setting is upper bounded by $(K-M+1)^{-1}$.*

Proof. Similar to the proof of Lemma 5, for the ease of notation in the following we remove the event $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}}=1$ from the conditions of all the entropies. We need to show that $H(\mathbf{A})\geq(K-M+1)L$.

Let $J\triangleq\{1,\ldots,K-M+1\}$ and $S_j\triangleq\{j,\ldots,j+M-1\}$ for $j\in J$. By Lemma 1, for each $j\in J$, there exists $C_j\in\mathcal{C}$ (and $\mathbf{Y}_j\triangleq\mathbf{Y}^{[S_j,C_j]}$) such that $H(\mathbf{X}_j|\mathbf{A},\mathbf{Q},\mathbf{Y}_j)=0$. Let $\mathbf{X}_J\triangleq\{\mathbf{X}_j\}_{j\in J}$. (Note that $|J|=K-M+1<M$ when $M>\frac{K+1}{2}$). Then, we have

$$H(\mathbf{X}_J,\mathbf{Y}_J|\mathbf{Q})=H(\mathbf{X}_J,\mathbf{Y}_J) \tag{2.13}$$

$$=2(K-M+1)L, \tag{2.14}$$

where (2.13) holds since $\mathbf{Q}$ is independent of $(\mathbf{X}_J,\mathbf{Y}_J)$ (by assumption); and (2.14) follows because $\mathbf{X}_J$ and $\mathbf{Y}_J$ are independent by construction. (Note that $\mathbf{X}_J$ and $\mathbf{Y}_J$ are not necessarily independent for $|J|=K-M+1\geq M$, and a different technique which remains open, is required for converse proof when $2\leq M\leq\frac{K+1}{2}$.) Moreover, we have

$$
\begin{aligned}
H(\mathbf{X}_J, \mathbf{Y}_J|\mathbf{A}, \mathbf{Q}) &\leq \sum_{j \in J} H(\mathbf{X}_j, \mathbf{Y}_j|\mathbf{A}, \mathbf{Q}) && (2.15)\\
&= \sum_{j \in J} H(\mathbf{Y}_j|\mathbf{A}, \mathbf{Q}) \\
&\quad + \sum_{j \in J} H(\mathbf{X}_j|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_j) \\
&= \sum_{j \in J} H(\mathbf{Y}_j|\mathbf{A}, \mathbf{Q}) && (2.16)\\
&\leq \sum_{j \in J} H(\mathbf{Y}_j) \\
&= (K - M + 1)L, && (2.17)
\end{aligned}
$$

where (2.15) follows from the chain rule of entropy; (2.16) holds because $H(\mathbf{X}_j|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_j) = 0$ for $j \in J$ (by assumption); and (2.17) holds because $\mathbf{Y}_j$'s for all $j \in J$ are independent by construction, and $\mathbf{Y}_j$ for each $j \in J$ is a scalar-linear combination of $\mathbf{X}_j, \mathbf{X}_{j+1}, \ldots, \mathbf{X}_{j+M-1}$.

Using (2.14) and (2.17), we can bound $H(\mathbf{X}_J, \mathbf{Y}_J, \mathbf{A}|\mathbf{Q})$ from below and above. On the one hand, we have

$$
\begin{aligned}
H(\mathbf{X}_J, \mathbf{Y}_J, \mathbf{A}|\mathbf{Q}) &\geq H(\mathbf{X}_J, \mathbf{Y}_J|\mathbf{Q}) \\
&= 2(K - M + 1)L, && (2.18)
\end{aligned}
$$

where (2.18) follows from (2.14). On the other hand, we have

$$
\begin{aligned}
H(\mathbf{X}_J, \mathbf{Y}_J, \mathbf{A}|\mathbf{Q}) &= H(\mathbf{A}|\mathbf{Q}) + H(\mathbf{X}_J, \mathbf{Y}_J|\mathbf{A}, \mathbf{Q}) \\
&\leq H(\mathbf{A}|\mathbf{Q}) + (K - M + 1)L, && (2.19)
\end{aligned}
$$

where (2.19) follows from (2.17). Now, combining (2.18) and (2.19), we have $H(\mathbf{A}|\mathbf{Q}) \geq (K-M+1)L$, and as a consequence, $H(\mathbf{A}) \geq H(\mathbf{A}|\mathbf{Q}) \geq (K-M+1)L$, as was to be shown. □

2.4.2.2 Achievability

In this section, we propose a scalar-linear PIR-PCSI–II protocol, termed the *Modified Specialized GRS Code protocol*, that achieves the rate $(K-M+1)^{-1}$. For this protocol, the requirements for the parameters q and l are the same as those for the Specialized GRS Code protocol.

Modified Specialized GRS Code Protocol: This protocol consists of three steps, where the steps 2-3 are the same as Steps 2-3 in the Specialized GRS Code protocol (Section 2.4.1.2), when the parameter M is replaced with $M-1$ everywhere. The step 1 of this protocol is as follows:

Step 1: For K arbitrarily chosen distinct elements $\omega_1, \ldots, \omega_K$ from $\mathbb{F}_q$, the user constructs a polynomial

$$p(x) = \sum_{i=1}^{K-M+1} p_i x^{i-1} \triangleq \prod_{i \in \mathcal{K} \setminus S} (x - \omega_i),$$

and constructs $K-M+1$ sequences $Q_1, \ldots, Q_{K-M+1}$, each of length K, defined as

$$Q_i = \{v_1 \omega_1^{i-1}, \ldots, v_K \omega_K^{i-1}\},$$

where v_j's are chosen as follows. For each $j \in S \setminus W$, $v_j = \frac{c_j}{p(\omega_j)}$ where c_j is the coefficient of X_j in $Y^{[S,C]}$; $v_W = \frac{c}{p(\omega_W)}$ for a randomly chosen element c from $\mathbb{F}_q^\times \setminus \{c_W\}$ where c_W is the coefficient of X_W in $Y^{[S,C]}$; and for each $j \notin S$, v_j is chosen at random from $\mathbb{F}_q^\times$.

The user then sends to the server the query $Q^{[W,S,C]} = \{Q_1, \ldots, Q_{K-M+1}\}$.

Example 2. Consider a scenario where the server has $K = 4$ messages $X_1, \ldots, X_4 \in \mathbb{F}_5$, and the user demands the message X_1 and has a coded side information $Y = X_1 + X_2$ with support size $M = 2$. For this example, $W = 1$, $S = \{1, 2\}$, and $C = \{c_1, c_2\} = \{1, 1\}$.

First, the user chooses $K = 4$ distinct elements from $\mathbb{F}_5$, $(\omega_1, \omega_2, \omega_3, \omega_4) = (0, 1, 2, 3)$. Then, the user constructs the polynomial

$$p(x) = \prod_{i \notin S} (x - \omega_i) = (x - \omega_3)(x - \omega_4) = (x + 3)(x + 2).$$

Note that $p(x) = p_1 + p_2 x + p_3 x^2 = 1 + x^2$. The user then computes v_j for $j \in S \setminus W$, i.e., v_2, by setting $v_2 = \frac{c_2}{p(\omega_2)} = 3$; computes v_W, i.e., v_1, for a randomly chosen element c, say $c = 4$, from $\mathbb{F}_5^\times \setminus \{c_1 = 1\}$ by setting $v_1 = \frac{c}{p(\omega_1)} = 4$; and chooses v_j for $j \notin S$, i.e., v_3 and v_4, at random (from $\mathbb{F}_5^\times$).

Suppose the user chooses $v_3 = 1$, $v_4 = 3$. Then, the user constructs $K - M + 1 = 3$ sequences $Q_1 = \{v_1, \ldots, v_4\} = \{4, 3, 1, 3\}$, $Q_2 = \{v_1\omega_1, \ldots, v_4\omega_4\} = \{0, 3, 2, 4\}$, and $Q_3 = \{v_1\omega_1^2, \ldots, v_4\omega_4^2\} = \{0, 3, 4, 2\}$. The user sends $Q = \{Q_1, Q_2, Q_3\}$ to the server.

Then, the server computes $A_1 = \sum_{j=1}^4 v_j X_j = 4X_1 + 3X_2 + X_3 + 3X_4$, $A_2 = \sum_{j=1}^4 v_j \omega_j X_j = 3X_2 + 2X_3 + 4X_4$, and $A_3 = \sum_{j=1}^4 v_j \omega_j^2 X_j = 3X_2 + 4X_3 + 2X_4$, and sends the answer $A = \{A_1, A_2, A_3\}$ back to the user. Then, the user computes $\sum_{i=1}^3 p_i A_i = A_1 + A_3 = 4X_1 + X_2$, and recovers X_1 by subtracting off $Y = X_1 + X_2$. For this example, the rate of the proposed protocol is $1/3$.

The proof of (W, S)-privacy for the proposed protocol in this example is similar to the proof of (W, S)-privacy for the Specialized GRS Code protocol in Example 1.

Lemma 7. *The Modified Specialized GRS Code protocol is a scalar-linear PIR-PCSI–II protocol, and achieves the rate* $(K - M + 1)^{-1}$.

Proof. The proof, omitted to avoid repetition, follows from the same lines as in the proof of Lemma 4. □

2.5 Single-Server PIR with Coded Side Information (PIR-CSI)

We present our main results for the PIR-CSI–I setting and PIR-CSI–II setting in Section 2.5.1 and Section 2.5.2, respectively. The capacity and the scalar-linear capacity of the PIR-CSI–I setting (for all $1 \leq M \leq K-1$) and the capacity and the scalar-linear capacity of the PIR-CSI–II setting (for all $2 \leq M \leq K$) are characterized in Theorems 3 and 4, respectively.

2.5.1 Single-Server PIR-CSI-I

Theorem 3. *For the single-server PIR-CSI–I with K messages and side information support size M, the capacity and the scalar-linear capacity are given by $\lceil \frac{K}{M+1} \rceil^{-1}$ for all $1 \leq M \leq K-1$.*

The proof consists of two parts. In the first part, using information-theoretic arguments, we give an upper bound on the rate of any PIR-CSI–I protocol (see Section 2.5.1.1). The proofs rely primarily on the necessary condition for W-privacy, stated in Lemma 2. In the second part, we construct a new scalar-linear PIR-CSI–I protocol, termed the *Modified Partition-and-Code (MPC) protocol*, which achieves this rate upper-bound (see Section 2.5.1.2). The proposed protocol is inspired by recently proposed Partition-and-Code with Interference Alignment protocol in [82] for single-server private computation with uncoded side information.

Remark 3. Interestingly, the capacity of PIR with (uncoded) side information [22] is also equal to $\lceil \frac{K}{M+1} \rceil^{-1}$ where M is the number of (uncoded) messages known to the user in advance as side information. This shows that there will be no loss in capacity, when compared to the case that the user knows M randomly chosen messages separately, even if the user knows only *one* random linear coded combination of M randomly chosen messages.

Remark 4. When (W, S)-privacy is required, the result of Theorem 1 shows that the capacity of single-server PIR with a coded side information with support size M that does not include the demand is equal to $(K-M)^{-1}$. Note that $\lceil \frac{K}{M+1} \rceil < K - M$ for all $1 \leq M \leq K-2$. This implies that the capacity of the PIR-CSI–I setting is strictly greater than that of the PIR-PCSI–I setting for any $1 \leq M \leq K-2$. This is expected because W-privacy is a weaker notion of privacy when compared to (W, S)-privacy. However, for the extremal case of $M = K-1$, as can be seen (W, S)-privacy comes at no extra cost compared to W-privacy.

Proof of Theorem 3

2.5.1.1 Converse

The capacity of the PIR-CSI–I setting is naturally upper bounded by the capacity of PIR with uncoded side information [22] where M uncoded messages are available at the user as side information. As shown in [22], the capacity of this problem is equal to $\lceil \frac{K}{M+1} \rceil^{-1}$, and the proof of this result relies on an index coding argument. In this section, we present an alternative converse proof for the case of general PIR-CSI–I protocols, by using information-theoretic arguments. Obviously, this proof also serves for the special case of scalar-linear PIR-CSI–I protocols.

Lemma 8. *For any $1 \leq M \leq K-1$, the (scalar-linear) capacity of the PIR-CSI–I setting is upper bounded by $\lceil \frac{K}{M+1} \rceil^{-1}$.*

Proof. In the following, all entropies are conditional on the event $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = 0$, and this event is removed from the conditions for the ease of notation. We need to show that $H(\mathbf{A}) \geq \lceil \frac{K}{M+1} \rceil L$.

Take arbitrary W, S, C (and $\mathbf{Y} \triangleq \mathbf{Y}^{[S,C]}$) such that $W \notin S$. Similar to the proof of

Lemma 3, it can be shown that

$$H(\mathbf{A}) \geq H(\mathbf{X}_W) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W). \tag{2.20}$$

There are two cases: (i) $W \cup S = \mathcal{K}$, and (ii) $W \cup S \neq \mathcal{K}$. In the case (i), $M = K - 1$, and so, $\lceil \frac{K}{M+1} \rceil L = L$. Since $H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) \geq 0$, then $H(\mathbf{A}) \geq H(\mathbf{X}_W) = L$ (by (2.20)), as was to be shown. In the case (ii), we proceed by lower bounding $H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W)$ as follows.

We arbitrarily choose a message, say $\mathbf{X}_{W_1}$, for some $W_1 \notin W \cup S$. By Lemma 2, there exist $S_1 \in \mathcal{S}$ with $W_1 \notin S_1$ and $C_1 \in \mathcal{C}$ (and $\mathbf{Y}_1 = \mathbf{Y}^{[S_1, C_1]}$) so that $H(\mathbf{X}_{W_1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_1) = 0$. Since conditioning does not increase the entropy, then $H(\mathbf{X}_{W_1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) = 0$. Thus, we have

$$\begin{aligned}
H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) &\geq H\big(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1\big) \\
&= H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) \\
&\quad + H(\mathbf{X}_{W_1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) \\
&= H\big(\mathbf{A}, \mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1\big) \\
&= H(\mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) \\
&\quad + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}) \\
&= H(\mathbf{X}_{W_1}) \\
&\quad + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}) \qquad (2.21)
\end{aligned}$$

where (2.21) holds because $\mathbf{X}_{W_1}$ and $(\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1)$ are independent (noting that $W_1 \notin W \cup S \cup S_1$), and hence, $H(\mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) = H(\mathbf{X}_{W_1})$.

Let $n \triangleq \lceil \frac{K}{M+1} \rceil$. Similarly as above, it can be shown that for all $1 \leq i \leq n-1$ there

exist $W_1, \ldots, W_i \in \mathcal{K}$ and $S_1, \ldots, S_i \in \mathcal{S}$ with $W_j \notin S_j$ for all $1 \leq j \leq i$ and $W_i \notin \cup_{j=1}^{i-1}(W_j \cup S_j) \cup (W \cup S)$, and $C_1, \ldots, C_i \in \mathcal{C}$ (and $\mathbf{Y}_1 = \mathbf{Y}^{[S_1,C_1]}, \ldots, \mathbf{Y}_i = \mathbf{Y}^{[S_i,C_i]}$), such that

$$H(\mathbf{X}_{W_i}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}, \ldots, \mathbf{Y}_{i-1}, \mathbf{X}_{W_{i-1}}, \mathbf{Y}_i) = 0.$$

Note that by construction, $\left|\cup_{j=1}^{i-1}(W_j \cup S_j) \cup (W \cup S)\right| \leq (M+1)i$ for all $1 \leq i \leq n-1$. Repeating an argument similar to the one being used for lower bounding $H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W)$ as in (2.21), it can be shown that

$$\begin{aligned} &H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}, \ldots, \mathbf{Y}_{i-1}, \mathbf{X}_{W_{i-1}}) \\ &\quad \geq H(\mathbf{X}_{W_i}) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}, \ldots, \mathbf{Y}_i, \mathbf{X}_{W_i}) \end{aligned}$$

for all $1 \leq i \leq n-1$. Combining these lower bounds for all $1 \leq i \leq n-1$, we have

$$\begin{aligned} H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) &\geq \sum_{i=1}^{n-1} H(\mathbf{X}_{W_i}) \\ &= (n-1)L. \end{aligned} \tag{2.22}$$

Putting (2.20) and (2.22) together, we get $H(\mathbf{A}) \geq nL = \lceil \frac{K}{M+1} \rceil L$. □

2.5.1.2 Achievability

This section presents a scalar-linear PIR-CSI–I protocol for arbitrary $1 \leq M \leq K-1$. This protocol, termed *Modified Partition-and-Code (MPC)*, is inspired by our recently proposed Partition-and-Code with Interference Alignment protocol in [82] for private computation with uncoded side information. The MPC protocol does not make any assumption on the base-field size q and the degree of field-extension l, and is applicable for arbitrary $q \geq 2$ and $l \geq 1$.

It should be noted that the Partition-and-Code protocol of [22] is only applicable to

the PIR-CSI–I setting when $M+1$ divides K. Otherwise, when $M+1$ is not a divisor of K, the Partition-and-Code protocol will generate one part of size less than $M+1$. This immediately results in a violation of the W-privacy condition. This is because the user's demand cannot be any of the messages pertaining to this part, noting that (i) the support set of the user's side information has size M, and (ii) all messages in the user's side information support set need to be combined with the user's demand.

Modified Partition-and-Code (MPC) Protocol: This protocol consists of 3 steps as follows:

Step 1: Let us first define $n \triangleq \lceil \frac{K}{M+1} \rceil$. For $1 \leq i \leq n-1$, we define $I_i \triangleq \{(i-1)(M+1)+1, \ldots, i(M+1)\}$, and $I_n \triangleq \{(n-1)(M+1)+1, \ldots, K, 1, \ldots, n(M+1)-K\}$. (Note that $I_n = \{(n-1)(M+1)+1, \ldots, K\}$ when $M+1$ divides K.)

First, the user constructs a random permutation π on $\mathcal{K} = \{1, \ldots, K\}$ as follows. The user randomly chooses an index j^* from $\mathcal{K}$, and assigns the demand index W to $\pi(j^*)$, i.e., $\pi(j^*) = W$. Let $i^* \triangleq \lceil \frac{j^*}{M+1} \rceil$ be the smallest index $i \in \{1, \ldots, n\}$ such that $j^* \in I_i$. Then, the user randomly assigns the side information support indices in S to $\{\pi(j) : j \in I_{i^*} \setminus \{j^*\}\}$ and randomly assigns the (not-yet-assigned) indices in $\mathcal{K} \setminus (W \cup S)$ to $\{\pi(j) : j \in \mathcal{K} \setminus I_{i^*}\}$.

Next, the user constructs n sequences $U_1, \ldots, U_n$, each of size $M+1$, defined as $U_i = \{\pi(j) : j \in I_i\}$; and constructs a sequence V, defined as $V = \{c_{\pi(j)} : j \in I_{i^*}\}$ where $c_{\pi(j)}$ for $j \in I_{i^*} \setminus \{j^*\}$ is the coefficient of message $X_{\pi(j)}$ in the side information $Y^{[S,C]}$, and $c_{\pi(j^*)} = c_W$ is a randomly chosen element from $\mathbb{F}_q^\times$.

The user then constructs $Q_i = (U_i, V)$ for each $1 \leq i \leq n$, and sends to the server the query $Q^{[W,S,C]} = \{Q_1, \ldots, Q_n\}$.

Step 2: By using $Q_i = (U_i, V)$'s, the server computes A_i's as $A_i = \sum_{j=1}^{M+1} c_{i_j} X_{i_j}$ where $U_i = \{i_1, \ldots, i_{M+1}\}$ and $V = \{c_{i_1}, \ldots, c_{i_{M+1}}\}$, and sends back to the user the

answer $A^{[W,S,C]} = \{A_1, \ldots, A_n\}$.

Step 3: Upon receiving the answer from the server, the user retrieves X_W by subtracting off the contribution of the side information $Y^{[S,C]}$ from $A_{i^*} = c_W X_W + \sum_{i \in S} c_i X_i$.

Example 3. Consider a scenario where the server has $K = 5$ messages $X_1, \ldots, X_5 \in \mathbb{F}_3$, and the user demands the message X_1 and has a coded side information $Y = X_2 + 2X_3$ with support size $M = 2$. For this example, $W = 1$, $S = \{2, 3\}$, $C = \{c_2, c_3\} = \{1, 2\}$.

The parameters of the MPC protocol for this example are as follows: $n = \lceil \frac{K}{M+1} \rceil = 2$, $I_1 = \{1, 2, 3\}$, and $I_2 = \{4, 5, 1\}$.

First, the user constructs a permutation π of $\{1, \ldots, 5\}$ as follows. The user randomly chooses an index j^* from $\{1, \ldots, 5\}$, say 4, and assigns the index $W = 1$ to $\pi(j^*) = \pi(4)$, i.e., $\pi(4) = 1$. Note that, in this case, $i^* \triangleq \lceil \frac{j^*}{M+1} \rceil = 2$, and $I_{i^*} = I_2 = \{4, 5, 1\}$. The user then randomly assigns the indices in S, i.e., 2 and 3, to $\{\pi(j) : j \in I_{i^*} \setminus \{j^*\}\} = \{\pi(5), \pi(1)\}$, say $\pi(5) = 3$ and $\pi(1) = 2$; and randomly assigns the (not-yet-assigned) indices 4 and 5 to $\{\pi(j) : j \in \{1, \ldots, 5\} \setminus I_{i^*}\} = \{\pi(2), \pi(3)\}$, say $\pi(2) = 4$ and $\pi(3) = 5$. Thus, the permutation π maps $\{1, 2, 3, 4, 5\}$ to $\{2, 4, 5, 1, 3\}$.

Next, the user constructs $n = 2$ sequences U_1, U_2, each of length $M + 1 = 3$, defined as $U_1 = \{\pi(j) : j \in I_1\} = \{2, 4, 5\}$ and $U_2 = \{\pi(j) : j \in I_2\} = \{1, 3, 2\}$; and constructs a sequence V, defined as $V = \{c_{\pi(j)} : j \in I_2\} = \{c_1, c_3, c_2\}$ where $c_2 = 1$ and $c_3 = 2$ are the coefficients of X_2 and X_3 in the side information Y, and c_1 is a randomly chosen element from $\mathbb{F}_3^\times = \{1, 2\}$, say $c_1 = 2$. Thus, $V = \{2, 2, 1\}$.

The user constructs both $Q_1 = (U_1, V) = (\{2, 4, 5\}, \{2, 2, 1\})$ and $Q_2 = (U_2, V) = (\{1, 3, 2\}, \{2, 2, 1\})$, and sends the query $Q = \{Q_1, Q_2\}$ to the server. The server then computes $A_1 = 2X_2 + 2X_4 + X_5$ and $A_2 = 2X_1 + 2X_3 + X_2$, and sends the answer $A = \{A_1, A_2\}$ back to the user. Then, the user subtracts off the contribution of $Y = X_2 + 2X_3$ from $A_{i^*} = A_2 = 2X_2 + X_2 + 2X_3$, and recovers X_1.

For this example, the rate of the MPC protocol is $1/2$. Note that the rate of the Specialized GRS Code protocol—which achieves (W, S)-privacy and hence W-privacy, for the scenario of this example is $(K - M)^{-1} = 1/3$.

From the perspective of the server, who knows the model and the parameters as well as the protocol, the messages $X_1, \ldots, X_5$ are equally likely to be the user's demand. This is because, given the query, for each candidate demand, the server finds a unique potential side information. In particular, by the protocol, there must exist a linear combination A_i in the answer $A = \{A_1, \ldots, A_n\}$ (i.e., $\{A_1, A_2\}$ in this example) which is a function of the demand and the side information, and not a function of any other message. For example, given that the candidate demand is X_1, the server finds $X_2 + 2X_3$ as the only potential side information, noting that only $A_2 = 2X_1 + X_2 + 2X_3$ is a linear combination of X_1 and $M = 2$ other messages (i.e., X_2 and X_3).

As an another example, consider the message X_2. Given that the candidate demand is X_2, there exist two linear combinations A_1 and A_2, each of which is a function of X_2 and $M = 2$ other messages. However, by the protocol, among all linear combinations A_i that are functions of the candidate demand and M other messages, *only the linear combination A_i with the smallest index i is a function of the demand and the side information.* Thus, for the candidate demand X_2, the server finds $2X_4 + X_5$ as the only potential side information, noting that among A_1 and A_2—which are both functions of X_2 and $M = 2$ other messages, the linear combination $A_1 = 2X_2 + 2X_4 + X_5$ has the smallest index. Similarly, for each of the other candidate demands X_3, X_4, X_5, the server finds a unique potential side information. Moreover, the side information support index set is uniformly distributed and the demand index is conditionally distributed uniformly given the side information support index set. Putting these arguments together, one can see that given the query each message is equally likely to be the user's demand. This confirms that the MPC protocol satisfies the W-privacy condition for this example. It is worth noting that the existence of a *unique*

potential side information for each candidate demand, which ensures W-privacy, results in the violation of the (W, S)-privacy condition. For instance, in this example, given the query, for the candidate demand index 1 the only potential side information support index set is $\{2, 3\}$; and no other pair of indices in $\{2, \ldots, 5\}$ can be a potential side information support index set for the demand index 1.

Lemma 9. *The Modified Partition-and-Code (MPC) protocol is a scalar-linear PIR-CSI–I protocol, and achieves the rate* $\lceil \frac{K}{M+1} \rceil^{-1}$.

Proof. By the construction of the Modified Partition-and-Code (MPC) protocol (see Steps 1-2), $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are linearly independent combinations of the messages in $\mathbf{X}_{\mathcal{K}}$. Using a similar argument as the one in the proof of Lemma 4, it can be shown that $H(\mathbf{A}^{[W,S,C]}) = H(\mathbf{A}_1, \ldots, \mathbf{A}_n) = nL$ for all $W \in \mathcal{K}, S \in \mathcal{S}, C \in \mathcal{C}$ such that $W \notin S$, and $H(\mathbf{A}^{[\mathbf{W},\mathbf{S},\mathbf{C}]}|\mathbf{W} \notin \mathbf{S}) = H(\mathbf{A}^{[W,S,C]}) = nL$. This implies that the rate of the MPC protocol is equal to $L/nL = \lceil \frac{K}{M+1} \rceil^{-1}$.

The scalar-linearity of the MPC protocol follows from the construction. The recoverability condition is also obviously satisfied (see Step 3).

To prove that the MPC protocol satisfies the W-privacy condition, we need to show that for any query Q generated by the protocol,

$$\mathbb{P}(\mathbf{W} = W|\mathbf{Q} = Q, \mathbf{W} \notin \mathbf{S}) = \mathbb{P}(\mathbf{W} = W|\mathbf{W} \notin \mathbf{S})$$

for all $W \in \mathcal{K}$, or in turn, $\mathbb{P}(\mathbf{W} = W|\mathbf{Q} = Q, \mathbf{W} \notin \mathbf{S})$ does not depend on W. (Note that by construction, $\mathbf{Q}$ is independent of the messages in $\mathbf{X}_{\mathcal{K}}$.)

By Step 1 of the protocol, for any given $W \in \mathcal{K}$, there exist a unique $S_W \in \mathcal{S}$ (with $W \notin S_W$) and a unique $C_W \in \mathcal{C}$ such that the triple (W, S_W, C_W) complies with the query Q, i.e., given that X_W and $Y^{[S_W,C_W]}$ are the user's demand and side information,

respectively, the protocol could potentially generate the query Q. Then, we have

$$\begin{aligned}&\mathbb{P}(\mathbf{W} = W|\mathbf{Q} = Q, \mathbf{W} \notin \mathbf{S})\\ &= \mathbb{P}(\mathbf{W} = W, \mathbf{S} = S_W, \mathbf{C} = C_W|\mathbf{Q} = Q, \mathbf{W} \notin \mathbf{S}).\end{aligned}$$

Since the conditional distribution of $(\mathbf{W}, \mathbf{S}, \mathbf{C})$ given $\mathbf{W} \notin \mathbf{S}$ is uniform, by applying the Bayes' rule one can see that $\mathbb{P}(\mathbf{W} = W, \mathbf{S} = S_W, \mathbf{C} = C_W|\mathbf{Q} = Q, \mathbf{W} \notin \mathbf{S})$ does not depend on W so long as $\mathbb{P}(\mathbf{Q} = Q|\mathbf{W} = W, \mathbf{S} = S_W, \mathbf{C} = C_W)$ does not depend on W. By the design of the protocol, we have

$$\begin{aligned}&\mathbb{P}(\mathbf{Q} = Q|\mathbf{W} = W, \mathbf{S} = S_W, \mathbf{C} = C_W)\\ &= \frac{1}{K!}\binom{K-1}{M}(q-1)^{-1}\end{aligned}$$

for all $W \in \mathcal{K}$, and hence $\mathbb{P}(\mathbf{W} = W|\mathbf{Q} = Q, \mathbf{W} \notin \mathbf{S})$ does not depend on W. □

2.5.2 Single-Server PIR-CSI-II

Theorem 4. *For the single-server PIR-CSI–II setting with K messages and side information support size M, the capacity and the scalar-linear capacity are equal to 1 for $M = 2, K$, and $1/2$ for all $3 \leq M \leq K-1$.*

For each range of values of M, the proof consists of two parts. In the first part, we use information-theoretic arguments—based on the result of Lemma 2, so as to upper bound the rate of any PIR-CSI–II protocol (see Section 2.5.2.1). In the second part, we construct novel scalar-linear PIR-CSI–II protocols, collectively termed the *Randomized Selection-and-Code (RSC) protocols*, for different ranges of values of M. The proposed protocols rely on probabilistic techniques, and achieve the corresponding rate upper-bounds (see Section 2.5.2.2).

Remark 5. Theorem 4 shows that when W-privacy is required, no matter what the size of support set of the side information is, the user can privately retrieve any message belonging to the support set of their coded side information, with a download cost at most twice the cost of downloading the message directly—which obviously does not preserve the privacy of the requested message.

Remark 6. As shown in Theorem 2, when (W, S)-privacy is required, the (scalar-linear) capacity of single-server PIR with a coded side information whose support set includes the requested message is equal to $(K - M + 1)^{-1}$, where M is the side information support size. The result of Theorem 4 matches this result for the cases of $M = K$ and $M = K-1$, and hence, (W, S)-privacy and W-privacy are attainable at the same cost in these cases; whereas for the other cases of M, achieving (W, S)-privacy is much more costly than achieving W-privacy.

Proof of Theorem 4

2.5.2.1 Converse

In this section, we give an information-theoretic proof of converse for the case of general PIR-CSI–II protocols, which also serves as a converse proof for the special case of scalar-linear PIR-CSI–II protocols.

Lemma 10. *For $M = 2$ and $M = K$, the (scalar-linear) capacity of the PIR-CSI–II setting is upper bounded by 1, and for any $3 \leq M \leq K - 1$, the (scalar-linear) capacity of the PIR-CSI–II setting is upper bounded by $1/2$.*

Proof. In the following, all entropies are conditional on the event $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 1$, and for simplifying the notation, we remove this event from the conditions everywhere.

Take arbitrary W, S, C (and $\mathbf{Y} \triangleq \mathbf{Y}^{[S,C]}$) such that $W \in S$. For the cases of $M = 2$ and $M = K$, it suffices to show that $H(\mathbf{A}) \geq L$. Note that $H(\mathbf{A}) \geq H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}) = H(\mathbf{A}, \mathbf{X}_W|\mathbf{Q}, \mathbf{Y})$, where the equality follows from the recoverability condition.

Moreover, $H(\mathbf{A}, \mathbf{X}_W|\mathbf{Q}, \mathbf{Y}) = H(\mathbf{X}_W|\mathbf{Q}, \mathbf{Y}) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) \geq H(\mathbf{X}_W)$, where the inequality follows from the independence of $\mathbf{X}_W$ and $(\mathbf{Q}, \mathbf{Y})$ by assumption. Putting these arguments together, $H(\mathbf{A}) \geq H(\mathbf{X}_W) = L$.

For the cases of $3 \leq M \leq K-1$, we need to show that $H(\mathbf{A}) \geq 2L$. By the above arguments, we have

$$H(\mathbf{A}) \geq H(\mathbf{X}_W) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W). \tag{2.23}$$

Consider an arbitrary index $W_1 \in S$. By the result of Lemma 2, there exist $S_1 \in \mathcal{S}$ with $W_1 \in S_1$ and $C_1 \in \mathcal{C}$ (and $\mathbf{Y}_1 = \mathbf{Y}^{[S_1, C_1]}$) so that $H(\mathbf{X}_{W_1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}_1) = 0$. As conditioning does not increase the entropy, then $H(\mathbf{X}_{W_1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) = 0$. Then, we can write

$$\begin{aligned}
H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) &\geq H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) \\
&= H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) + H(\mathbf{X}_{W_1}|\mathbf{A}, \mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) \\
&= H(\mathbf{A}, \mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) \\
&= H(\mathbf{X}_{\mathbf{W}_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}) \\
&\geq H(\mathbf{X}_{\mathbf{W}_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1).
\end{aligned} \tag{2.24}$$

Noting that $\mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1, \mathbf{X}_{W_1}$ are linear functions of the messages in $\mathbf{X}_\mathcal{K}$, and $\mathbf{Q}$ is independent of $\mathbf{X}_\mathcal{K}$, there are two possible cases: (i) $H(\mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) = H(\mathbf{X}_{W_1})$, i.e., $\mathbf{X}_{W_1}$ is independent of $(\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1)$, or (ii) $H(\mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) = 0$, i.e., $\mathbf{X}_{W_1}$ can be recovered from $\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1$.

In the case (i), $H(\mathbf{X}_{W_1}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W, \mathbf{Y}_1) = H(\mathbf{X}_{W_1})$ by assumption. Rewriting (2.24),

$$H(\mathbf{A}|\mathbf{Q}, \mathbf{Y}, \mathbf{X}_W) \geq H(\mathbf{X}_{W_1}). \tag{2.25}$$

By (2.23) and (2.25), $H(\mathbf{A}) \geq H(\mathbf{X}_W) + H(\mathbf{X}_{W_1}) = 2L$.

In the case (ii), by assumption, $H(\mathbf{X}_{W_1}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_1)=0$. Again, by the linearity of $\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_1,\mathbf{X}_{W_1}$, it must hold that $\mathbf{Y}=c_W\mathbf{X}_W+c_{W_1}\mathbf{X}_{W_1}+\mathbf{Z}$ and $\mathbf{Y}_1=c'_W\mathbf{X}_W+c'_{W_1}\mathbf{X}_{W_1}+c'\mathbf{Z}$ for some $c'_W,c'_{W_1},c'\in\mathbb{F}_q^\times$, where $\mathbf{Z}=\sum_{i\in S\setminus\{W,W_1\}}c_i\mathbf{X}_i$. Unlike the previous case, this time we turn to an arbitrary index $W_2\notin S$. Again, by the result of Lemma 2, there exist $S_2\in\mathcal{S}$ with $W_2\in S_2$ and $C_2\in\mathcal{C}$ (and $\mathbf{Y}_2=\mathbf{Y}^{[S_2,C_2]}$) such that $H(\mathbf{X}_{W_2}|\mathbf{A},\mathbf{Q},\mathbf{Y}_2)=0$. Similar to (2.24), it can be shown that

$$\begin{aligned}H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W)\geq{}& H(\mathbf{X}_{W_2}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_2)\\&+H(\mathbf{A}|\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_2,\mathbf{X}_{W_2}).\end{aligned}\tag{2.26}$$

If $\mathbf{X}_{W_2}$ is independent of $(\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_2)$, similarly as in the case (i) we can show that $H(\mathbf{A})\geq H(\mathbf{X}_W)+H(\mathbf{X}_{W_2})=2L$. If $\mathbf{X}_{W_2}$ is recoverable from $(\mathbf{Q},\mathbf{Y},\mathbf{X}_W,\mathbf{Y}_2)$, it must hold that $\mathbf{Y}_2=c''_{W_2}\mathbf{X}_{W_2}+c''(c_{W_1}\mathbf{X}_{W_1}+\mathbf{Z})$ for some $c''_{W_2},c''\in\mathbb{F}_q^\times$. Note that $\mathbf{X}_{W_2}$ is independent of $(\mathbf{Q},\mathbf{Y}_1,\mathbf{X}_{W_1},\mathbf{Y}_2)$ since by construction, $\mathbf{X}_{W_2}$ cannot be recovered from $c'_W\mathbf{X}_W+c'\mathbf{Z}$ and $c''_{W_2}\mathbf{X}_{W_2}+c''\mathbf{Z}$, or in turn, from $\mathbf{Y}_1$ and $\mathbf{Y}_2$ given $\mathbf{X}_{W_1}$. Also, $\mathbf{X}_{W_1}$ is independent of $(\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2)$ as $\mathbf{X}_{W_1}$ cannot be recovered from $\mathbf{Y}_1$ and $\mathbf{Y}_2$. So, we have

$$\begin{aligned}H(\mathbf{A})&\geq H(\mathbf{A}|\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2)\\&=H(\mathbf{A},\mathbf{X}_{W_1},\mathbf{X}_{W_2}|\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2)&(2.27)\\&\geq H(\mathbf{X}_{W_1}|\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2)+H(\mathbf{X}_{W_2}|\mathbf{Q},\mathbf{X}_{W_1},\mathbf{Y}_1,\mathbf{Y}_2)\\&=H(\mathbf{X}_{W_1})+H(\mathbf{X}_{W_2})&(2.28)\end{aligned}$$

where (2.27) holds as $H(\mathbf{X}_{W_1}|\mathbf{A},\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2)=0$ and $H(\mathbf{X}_{W_2}|\mathbf{A},\mathbf{Q},\mathbf{Y}_1,\mathbf{X}_{W_1},\mathbf{Y}_2)=0$, noting that by assumption, $H(\mathbf{X}_{W_1}|\mathbf{A},\mathbf{Q},\mathbf{Y}_1)=0$, $H(\mathbf{X}_{W_2}|\mathbf{A},\mathbf{Q},\mathbf{Y}_2)=0$, and (2.28) holds since as was shown earlier, $\mathbf{X}_{W_1}$ and $\mathbf{X}_{W_2}$ are independent of $(\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2)$ and $(\mathbf{Q},\mathbf{Y}_1,\mathbf{X}_{W_1},\mathbf{Y}_2)$, respectively. By (2.28), we get $H(\mathbf{A})\geq 2L$. □

2.5.2.2 *Achievability*

In this section, we propose a scalar-linear PIR-CSI–II protocol for each of the following cases: (Case 1) $M = 2$; (Case 2) $3 \leq M \leq \frac{K}{2} + 1$; (Case 3) $\frac{K+1}{2} \leq M \leq K - 1$; and (Case 4) $M = K$. (Note that Cases 2 and 3 are overlapping at $M = \frac{K}{2} + 1$ or $M = \frac{K+1}{2}$ when K is even or odd, respectively. In these scenarios, either of the proposed protocols for Cases 2 and 3 applies.) It should be noted that the proposed protocols for Cases 1 and 2 are applicable for any base-field size $q \geq 2$ and any degree of the field-extension $l \geq 1$; whereas the proposed protocols for Cases 3 and 4 are applicable for any $q \geq 3$ and any $l \geq 1$.

The proposed protocols rely on the idea of randomizing the *structure* of query/answer, and are referred to as the *Randomized Selection-and-Code (RSC) protocols*. In particular, in these protocols, for any given instance of the problem, there exist multiple different query/answer structures, each of which satisfies the recoverability condition; and one of these structures will be chosen at random according to a probability distribution, which is carefully designed to ensure the W-privacy condition.

For example, consider a scenario of Case 1 where the server stores $X_1, X_2, \ldots, X_K$, and the user's demand and side information are X_1 and $X_1 + X_2$, respectively. The RSC protocol for Case 1 has two different (query/answer) structures: (i) the user queries X_1, which is the user's demand, and the server sends X_1 back to the user; or (ii) the user queries X_2, which is the other message in the user's side information, and the server sends back X_2 to the user. (Note that neither of these structures depend on the other messages $X_3, \ldots, X_K$.) The RSC protocol for Case 1 randomly generates one of the two structures (i) and (ii), according to a probability distribution—specified shortly in the description of the protocol, designed in order to guarantee W-privacy (i.e., given the query, each message in $X_1, \ldots, X_K$ is equally likely to be the user's demand.) Using either of the two structures

(i) and (ii), the user can recover X_1. The RSC protocols for Cases 2-4 use a similar idea.

W.l.o.g., we assume that $W = \{1\}$, $S = \{1, \ldots, M\}$, and $C = \{c_1, \ldots, c_M\}$.

Randomized Selection-and-Code (RSC) Protocols: The RSC protocol for each case consists of three steps, where the Steps are the same as Steps 2-3 in the MPC protocol (Section 2.5.1.2). The Step 1 of the RSC protocols are as follows:

Case 1: The user randomly selects the index W (i.e., 1) with probability $\frac{1}{K}$, or the other index in S (i.e., 2) with probability $\frac{K-1}{K}$, and constructs two sets $U = \{i\}$ and $V = \{1\}$, where i is the selected index by the user.

The user then constructs $Q = (U, V)$, and sends the query $Q^{[W,S,C]} = Q$ to the server.

Example 4. Consider a scenario where the server has $K = 6$ messages $X_1, \ldots, X_6 \in \mathbb{F}_3$, and the user demands the message X_1 and has a side information $Y = 2X_1 + X_2$ with support size $M = 2$. For this example, $W = 1$, $S = \{1, 2\}$, and $C = \{c_1, c_2\} = \{2, 1\}$.

The user randomly selects an index i from $S = \{1, 2\}$, where the probability of selecting the index $i = 1$ is $\frac{1}{K} = \frac{1}{6}$, and the probability of selecting the index $i = 2$ is $\frac{K-1}{K} = \frac{5}{6}$. Suppose that the user selects the index $i = 2$. Then, the user requests the server for the message $X_i = X_2$. Subtracting off X_2 from $Y = 2X_1 + X_2$, the user then recovers X_1.

Here, the rate of the RSC protocol is 1. Note that the Modified Specialized GRS Code protocol which yields (W, S)-privacy and hence W-privacy, achieves the rate $(K - M + 1)^{-1} = 1/5$ for the scenario of this example.

From the server's perspective, the probability that the message X_2 is the user's demand is $\frac{1}{6}$, and the probability that one of the messages $X_1, X_3, \ldots, X_6$ is the user's demand is $\frac{5}{6}$. Since these messages are equally likely to be the demand, the probability of any of them to be the user's demand is $\frac{5}{6} \times \frac{1}{5} = \frac{1}{6}$. This guarantees the W-privacy.

Now, suppose that the user selects $i = 1$. In this case, the user requests their demand X_1 from the server, and the server responds by sending X_1 back to the user. Again, from

the perspective of the server, the probability that the message X_1 is the user's demand is $\frac{1}{6}$, and the probability of any of the messages $X_2, \ldots, X_6$ to be the user's demand is $\frac{5}{6} \times \frac{1}{5} = \frac{1}{6}$. This again ensures the W-privacy.

Case 2: The user constructs two sequences U_1, U_2, each of size $M-1$, with elements from the indices in $\mathcal{K}$, and an sequences V of size $M-1$ with elements from $\mathbb{F}_q^\times$. The constructions of U_1, U_2, V are as follows.

First, the user chooses an integer $r \in \{M-2, M-1\}$ by sampling from a probability distribution given by

$$\mathbb{P}(\mathbf{r} = r) = \begin{cases} \frac{2M-2}{K}, & r = M-2, \\ 1 - \frac{2M-2}{K}, & r = M-1. \end{cases}$$

If $r = M-1$ is chosen, the user randomly selects $M-1$ indices from $\mathcal{K} \setminus S$; otherwise, if $r = M-2$ is chosen, the user selects the index W along with $M-2$ randomly chosen indices from $\mathcal{K} \setminus S$. Denote by $\{i_1, \ldots, i_{M-1}\}$ the sequence of the $M-1$ selected indices (in increasing order). Then, the user constructs $U_1 = \{2, \ldots, M\}$ (i.e., the set of elements in $S \setminus W$ in increasing order) and $U_2 = \{i_1, \ldots, i_{M-1}\}$.

Next, the user constructs the sequence $V = \{c_2, \ldots, c_M\}$ (i.e., the sequence of elements in C excluding the element c_W).

The user then constructs $Q_i = (U_i, V)$ for each $i \in \{1, 2\}$, and for a randomly chosen permutation $\sigma : \{1, 2\} \mapsto \{1, 2\}$, sends the query $Q^{[W,S,C]} = \{Q_{\sigma(1)}, Q_{\sigma(2)}\}$ to the server.

Example 5. Consider a scenario where the server has $K = 6$ messages $X_1, \ldots, X_6 \in \mathbb{F}_3$, and the user demands the message X_1 and has a coded side information $Y = 2X_1 + X_2 + 2X_3$ with support size $M = 3$. For this example, $W = 1$, $S = \{1, 2, 3\}$, and $C = \{c_1, c_2, c_3\} = \{2, 1, 2\}$.

First, the user randomly chooses an integer $r \in \{M-2=1, M-1=2\}$, where the probability of choosing $r=1$ is $\frac{2}{3}$, and the probability of choosing $r=2$ is $\frac{1}{3}$. Suppose that the user chooses $r=1$. The user then selects the index $W=1$ along with $r=1$ randomly chosen index from $\{1,\ldots,6\} \setminus \{1,2,3\} = \{4,5,6\}$, say the index 4. Then, the user constructs two sequences $U_1 = \{2,3\}$ and $U_2 = \{1,4\}$, and the sequence $V = \{c_2, c_3\} = \{1,2\}$.

The user builds $Q_1 = (U_1, V) = (\{2,3\}, \{1,2\})$ and $Q_2 = (U_2, V) = (\{1,4\}, \{1,2\})$. For a randomly chosen permutation σ on $\{1,2\}$, say $\sigma(1)=2$ and $\sigma(2)=1$, the user constructs the query $Q = \{Q_{\sigma(1)}, Q_{\sigma(2)}\} = \{Q_2, Q_1\}$, and sends it to the server. The server computes $A_i = \sum_{j=1}^{M-1} c_{i_j} X_{i_j}$ for each $i \in \{1,2\}$ where $Q_i = (\{i_1, i_2\}, \{c_{i_1}, c_{i_2}\})$. For this example, $A_1 = X_2 + 2X_3$ and $A_2 = X_1 + 2X_4$. Then, the server sends the answer $A = \{A_{\sigma(1)}, A_{\sigma(2)}\} = \{A_2, A_1\}$ back to the user. Subtracting off A_1 from $Y = 2X_1 + X_2 + 2X_3$, the user recovers X_1.

For this example, the rate of the RSC protocol is $1/2$; whereas the rate of the Modified Specialized GRS Code protocol for the scenario of this example is $(K-M+1)^{-1} = 1/4$.

From the server's perspective, $U_1 = \{2,3\}$ and $U_2 = \{1,4\}$ are equally likely to be the index set of the user's side information support set (excluding the demand index). Let us refer to the event that X_2 and X_3 (or X_1 and X_4) are the two messages in the user's side information support set as E1 (or E2). Then, E1 (or E2) has probability $\frac{1}{2}$. Note also that, given E1 (or E2), X_2 and X_3 (or X_1 and X_4) have zero probability to be the user's demand.

Given E1, (i) with probability $\frac{1}{3}$, the user's demand is neither X_1 nor X_4, or (ii) with probability $\frac{2}{3}$, the user's demand is either X_1 or X_4. Given E1-(i), X_5 and X_6 are equally likely to be the user's demand. That is, given E1, X_5 (or X_6) is the user's demand with probability $\frac{1}{3} \times \frac{1}{2} = \frac{1}{6}$. Given E1-(ii), X_1 and X_4 are equally likely to be the user's demand. Then, given E1, X_1 (or X_4) is the user's demand with probability $\frac{2}{3} \times \frac{1}{2} = \frac{1}{3}$.

Given E2, (i) with probability $\frac{1}{3}$, the user's demand is neither X_2 nor X_3, or (ii) with probability $\frac{2}{3}$, the user's demand is either X_2 or X_3. Given E2-(i), either of X_5 and X_6 is the user's demand with probability $\frac{1}{2}$. Then, given E2, X_5 (or X_6) is the user's demand with probability $\frac{1}{3} \times \frac{1}{2} = \frac{1}{6}$. Given E2-(ii), either of X_2 and X_3 is the user's demand with probability $\frac{1}{2}$. Then, given E1, X_2 (or X_3) is the user's demand with probability $\frac{2}{3} \times \frac{1}{2} = \frac{1}{3}$.

From the above arguments, it is easy to see that given the query, each message X_i is equally likely to be the user's demand, and hence the W-privacy condition is satisfied. For example, X_1 has probability $\frac{1}{3}$ (or 0) to be the user's demand given E1 (or E2). Since E1 and E2 each have probability $\frac{1}{2}$, the probability of X_1 to be the user's demand is $\frac{1}{2} \times \frac{1}{3} + \frac{1}{2} \times 0 = \frac{1}{6}$. As an another example, consider X_5. Given either of E1 or E2, X_5 has probability $\frac{1}{6}$ to be the user's demand. Thus, the probability of X_5 to be the user's demand is $\frac{1}{2} \times \frac{1}{6} + \frac{1}{2} \times \frac{1}{6} = \frac{1}{6}$.

Case 3: The user constructs two sequences U_1, U_2, each of size M, with elements from the indices in $\mathcal{K}$, and a sequence V of size M with elements from $\mathbb{F}_q^{\times}$. The constructions of U_1, U_2, V are as follows.

The user chooses an integer $s \in \{2M - K - 1, 2M - K\}$ by sampling from a probability distribution given by

$$\mathbb{P}(\mathbf{s} = s) = \begin{cases} 1 - \frac{2K-2M}{K}, & s = 2M - K - 1, \\ \frac{2K-2M}{K}, & s = 2M - K. \end{cases}$$

If $s = 2M-K$ is chosen, the user randomly selects $2M-K$ indices from $S \backslash W$; otherwise, if $s = 2M - K - 1$ is chosen, the user selects the index W together with $2M - K - 1$ randomly chosen indices from $S \backslash W$. Denote by $\{i_1, \ldots, i_M\}$ the sequence of the $2M-K$ selected indices and the $K - M$ indices in $\mathcal{K} \setminus S$ (in increasing order). The user constructs $U_1 = \{1, \ldots, M\}$ (i.e., the set of indices in S in increasing order) and $U_2 = \{i_1, \ldots, i_M\}$.

Next, the user constructs the sequence $V = \{c, c_2, \ldots, c_M\}$ (i.e., the sequence of the elements in C, except when the element c_W is replaced by the element c) where c is randomly chosen from $\mathbb{F}_q^\times \setminus \{c_1\}$ (i.e., $\mathbb{F}_q^\times \setminus \{c_W\}$).

The user then constructs $Q_i = (U_i, V)$ for each $i \in \{1, 2\}$, and for a randomly chosen permutation $\sigma : \{1, 2\} \mapsto \{1, 2\}$, sends the query $Q^{[W,S,C]} = \{Q_{\sigma(1)}, Q_{\sigma(2)}\}$ to the server.

Case 4: The user creates two sequences $U = \{1, \ldots, K\}$ and $V = \{c, c_2, \ldots, c_K\}$ (i.e., the sequence of elements in C, except when the element c_W is replaced by the element c) where c is randomly chosen from $\mathbb{F}_q^\times \setminus \{c_1\}$ (i.e., $\mathbb{F}_q^\times \setminus \{c_W\}$).

The user then constructs $Q = (U, V)$, and sends the query $Q^{[W,S,C]} = Q$ to the server.

Lemma 11. *The Randomized Selection-and-Code (RSC) protocols for $M = 2$, $3 \leq M \leq \frac{K}{2} + 1$, $\frac{K+1}{2} \leq M \leq K - 1$, and $M = K$ are scalar-linear PIR-CSI–II protocols, and achieve the rates 1, $1/2$, $1/2$, and 1, respectively.*

Proof. The proofs for the rates of the RSC protocols follow the same line as in the proof of the rate of the MPC protocol in Lemma 9, and hence omitted. From the construction, it should also be obvious that the RSC protocols are scalar-linear. Moreover, it should not be hard to see from the description of these protocols that the recoverability condition is satisfied.

To prove that the RSC protocols satisfy the W-privacy condition, we need to show that

$$\mathbb{P}(\mathbf{W} = W | \mathbf{Q} = Q, \mathbf{W} \in \mathbf{S}) = \mathbb{P}(\mathbf{W} = W | \mathbf{W} \in \mathbf{S})$$

for all $W \in \mathcal{K}$. Alternatively, by the Bayes' rule, it suffices to show that $\mathbb{P}(\mathbf{Q} = Q | \mathbf{W} = W, \mathbf{W} \in \mathbf{S})$ does not depend on W.

Recall that $Q = (U, V)$ for Cases 1 and 4, and $Q = \{Q_1, Q_2\} = \{(U_1, V), (U_2, V)\}$ for Cases 2 and 3. For simplifying the notation, let us denote $\{U_1, U_2\}$ by U for Cases 2 and 3.

By the construction of the RSC protocols and the model assumptions, given $\mathbf{W} \in \mathbf{S}$, the following two observations hold:

(i) $\mathbf{U}$ and $\mathbf{V}$ are conditionally independent given $\mathbf{W}$, and

(ii) $\mathbf{V}$ and $\mathbf{W}$ are independent.

The observation (i) should be obvious, and the observation (ii) holds because $\mathbf{V}$ is uniformly distributed over all possible choices of V for each case. (For example, for Case 1, $\mathbf{V} = \{1\}$; and for Case 2, $\mathbf{V} = \{\mathbf{c}_i : i \in \mathbf{S} \setminus \mathbf{W}\}$—where $\mathbf{c}_i$'s are uniformly distributed over $\mathbb{F}_q^\times$, is uniformly distributed over all sequences of size $M-1$ with elements from $\mathbb{F}_q^\times$.) Using (i) and (ii), we have

$$\begin{aligned}\mathbb{P}(\mathbf{Q} = Q|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) &= \mathbb{P}(\mathbf{U} = U, \mathbf{V} = V|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\ &= \mathbb{P}(\mathbf{V} = V|\mathbf{W} \in \mathbf{S}) \times \mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}).\end{aligned}$$

As $\mathbb{P}(\mathbf{V} = V|\mathbf{W} \in \mathbf{S})$ does not depend on W, instead of showing that $\mathbb{P}(\mathbf{Q} = Q|\mathbf{W} = W, \mathbf{W} \in \mathbf{S})$ is not a function of W, it suffices to show that $\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S})$ does not depend on W. In the following, we prove this claim for the RSC protocol for each case separately.

With a slight abuse of notation, hereafter for the ease of exposition, we treat the sequences U_1, U_2 as (unordered) sets.

Case 1: For an arbitrary $i \in \mathcal{K}$, consider $U = \{i\}$. Take an arbitrary $W \in \mathcal{K}$. There are two cases as follows: (i) $W = i$, and (ii) $W \neq i$.

In the case (i), we have

$$\begin{aligned}&\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\ &= \sum_{j \in \mathcal{K} \setminus W} \mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = \{W, j\}) \times \mathbb{P}(\mathbf{S} = \{W, j\}|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}). \quad (2.29)\end{aligned}$$

By the model assumption, we have

$$\mathbb{P}(\mathbf{S} = \{W, j\} | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) = \frac{1}{K-1} \tag{2.30}$$

for all $j \in \mathcal{K} \setminus W$. Moreover, given that $\mathbf{W} = W$ and $\mathbf{S} = \{W, j\}$, the protocol constructs $U = \{W\}$ with probability $\frac{1}{K}$. This implies that

$$\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = \{W, j\}) = \frac{1}{K} \tag{2.31}$$

for all $j \in \mathcal{K} \setminus W$. Substituting (2.30) and (2.31) into (2.29),

$$\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) = \frac{1}{K}. \tag{2.32}$$

In the case (ii), we have

$$\begin{aligned} &\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\ &= \mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = \{W, i\}) \times \mathbb{P}(\mathbf{S} = \{W, i\} | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\ &= \frac{1}{K}, \end{aligned} \tag{2.33}$$

noting that by the model assumption,

$$\mathbb{P}(\mathbf{S} = \{W, i\} | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) = \frac{1}{K-1},$$

and by the design of the protocol,

$$\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = \{W, i\}) = \frac{K-1}{K}.$$

From (2.32) and (2.33), we infer that $\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{W} \in \mathbf{S})$ does not depend on W.

Case 2: Consider an arbitrary $U = \{U_1, U_2\}$. (Recall that $|U_1| = |U_2| = M - 1$.) Take an arbitrary $W \in \mathcal{K}$. There are two cases: (i) $W \in U_1 \cup U_2$, and (ii) $W \notin U_1 \cup U_2$.

In the case (i), w.l.o.g., assume that $W \in U_1$. Note that $\mathbf{W} = W$ and $W \in U_1$ together imply that $\mathbf{S} = W \cup U_2$ (by the design of the protocol). Then, we have

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = W \cup U_2) \\
&\quad \times \mathbb{P}(\mathbf{S} = W \cup U_2 | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}).
\end{aligned} \tag{2.34}$$

By the model assumption, we have

$$\mathbb{P}(\mathbf{S} = W \cup U_2 | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) = \binom{K-1}{M-1}^{-1}. \tag{2.35}$$

Moreover, given that $\mathbf{W} = W$ and $\mathbf{S} = W \cup U_2$, the protocol constructs U_1 with probability $(\frac{2M-2}{K}) \times \binom{K-M}{M-2}^{-1}$, noting that $W \in U_1$. (The protocol selects the demand index W to be one of the elements in U_1 with probability $\frac{2M-2}{K}$, and selects the set of other $M-2$ elements in U_1 from the set of $K - M$ indices in $\mathcal{K} \setminus S$ with probability $\binom{K-M}{M-2}^{-1}$.) This implies that

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = W \cup U_2) \\
&= 2 \left(\frac{M-1}{K} \right) \binom{K-M}{M-2}^{-1}.
\end{aligned} \tag{2.36}$$

Substituting (2.35) and (2.36) into (2.34),

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= 2 \left(\frac{M-1}{K} \right) \binom{K-M}{M-2}^{-1} \binom{K-1}{M-1}^{-1}.
\end{aligned} \tag{2.37}$$

In the case (ii), we have

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = W \cup U_1) \\
&\quad \times \mathbb{P}(\mathbf{S} = W \cup U_1|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&\quad + \mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = W \cup U_2) \\
&\quad \times \mathbb{P}(\mathbf{S} = W \cup U_2|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= 2\left(1 - \frac{2M-2}{K}\right)\binom{K-M}{M-1}^{-1}\binom{K-1}{M-1}^{-1}, \qquad (2.38)
\end{aligned}$$

noting that

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = W \cup U_1) \\
&= \mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = W \cup U_2) \\
&= \left(1 - \frac{2M-2}{K}\right)\binom{K-M}{M-1}^{-1},
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}(\mathbf{S} = W \cup U_1|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \mathbb{P}(\mathbf{S} = W \cup U_2|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \binom{K-1}{M-1}^{-1}.
\end{aligned}$$

Now, it is easy to verify that

$$\left(\frac{M-1}{K}\right)\binom{K-M}{M-2}^{-1} = \left(1 - \frac{2M-2}{K}\right)\binom{K-M}{M-1}^{-1}.$$

This shows that (2.37) and (2.38) are equal, completing the proof that $\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S})$ does not depend on W.

Case 3: Consider an arbitrary query $U = \{U_1, U_2\}$. (Recall that $|U_1| = |U_2| = M$.) Take an arbitrary $W \in \mathcal{K}$. There are two cases as follows: (i) $W \in U_1 \cap U_2$, and (ii) $W \notin U_1 \cap U_2$.

In the case (i), we have

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = U_1) \\
&\quad \times \mathbb{P}(\mathbf{S} = U_1 | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&\quad + \mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = U_2) \\
&\quad \times \mathbb{P}(\mathbf{S} = U_2 | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= 2 \binom{2M-K}{K} \binom{M-1}{2M-K-1}^{-1} \binom{K-1}{M-1}^{-1},
\end{aligned} \tag{2.39}$$

noting that

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = U_1) \\
&= \mathbb{P}(\mathbf{U} = U | \mathbf{W} = W, \mathbf{S} = U_2) \\
&= \binom{2M-K}{K} \binom{M-1}{2M-K-1}^{-1},
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}(\mathbf{S} = U_1 | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \mathbb{P}(\mathbf{S} = U_2 | \mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \binom{K-1}{M-1}^{-1}.
\end{aligned}$$

In the case (ii), w.l.o.g., assume that $W \in U_1$. Then, we have

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= \mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = U_1) \\
&\quad \times \mathbb{P}(\mathbf{S} = U_1|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) \\
&= 2\left(\frac{K-M}{K}\right)\binom{M-1}{2M-K}^{-1}\binom{K-1}{M-1}^{-1}, \qquad (2.40)
\end{aligned}$$

noting that

$$\begin{aligned}
&\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{S} = U_1) \\
&= 2\left(\frac{K-M}{K}\right)\binom{M-1}{2M-K}^{-1},
\end{aligned}$$

and

$$\mathbb{P}(\mathbf{S} = U_1|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) = \binom{K-1}{M-1}^{-1}.$$

It is easy to verify that

$$\begin{aligned}
&\left(\frac{2M-K}{K}\right)\binom{M-1}{2M-K-1}^{-1} \\
&= \left(\frac{K-M}{K}\right)\binom{M-1}{2M-K}^{-1}.
\end{aligned}$$

This shows that (2.39) and (2.40) are equal, completing the proof that $\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S})$ does not depend on W.

Case 4: The protocol implies that $U = \mathcal{K}$, and hence $\mathbb{P}(\mathbf{U} = U|\mathbf{W} = W, \mathbf{W} \in \mathbf{S}) = 1$ for all W. □

3. MULTI-SERVER PRIVATE INFORMATION RETRIEVAL (PIR)*

3.1 Introduction

In the Private Information Retrieval (PIR) problem, there is a user who wishes to download a single or multiple messages belonging to a database with copies stored on a single or multiple servers, while protecting the identity of the demanded message(s) from every individual server [4,8,9,20]. To retrieve the desired message(s), the user generates one query for each server. Upon receiving the user's query, each server will return an answer, which depends on the stored messages and the received query. To ensure that each server learns nothing about the identity of the user's demanded message(s), in an information theoretic sense, each query must be marginally independent of the desired message(s) index.

In a single-server setting or a multi-server setting when all servers can fully collude, the user must download the whole database to achieve privacy in the information-theoretic sense [4]. However, when the user has some side information (unknown to the server(s)) about the messages in the database [21–30, 32, 33] or when the servers do not fully collude [8, 9, 15], the privacy can be achieved more efficiently in terms of the download cost (i.e., the amount of information downloaded from the server(s)).

For the PIR problem in the presence of side information, two types of privacy requirements can be considered: (i) W*-privacy*, in which the identity of the message(s) demanded by the user needs to be protected, and (ii) (W, S)*-privacy* in which the identities of both the message(s) demanded by user and the message(s) in the support set of the user's side information needs to be protected, where W denotes the index set of the user's requested

*Reprinted with permission from [31] "Private Information Retrieval with Private Coded Side Information: The Multi-Server Case," by F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, 2019. In Proceedings of 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1098-1104, Sept 2019 and [35] "Multi-Server Private Information Retrieval with Coded Side Information," by F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, 2019. In Proceedings of 2019 16th Canadian Workshop on Information Theory (CWIT), pp. 1-6, June 2019. Copyright © by IEEE.

message(s), and S denotes the index set of the messages in the support set of the user's side information. Based on the definitions, it is clear that the (W, S)-privacy requirement is a stronger privacy requirement compared to the W-privacy requirement.

In this work, there is a user who is interested in retrieving a single message from a database of K independently and uniformly distributed messages, with copies stored on N non-colluding servers. The user initially knows a linear coded combination of a subset of M messages in the database, where the identities of the messages in the support set of the user's coded side information as well as their coding coefficients are initially unknown to the servers. This type of side information is motivated by several practical scenarios. For instance, the side information could have been obtained in advance from a trusted server with limited knowledge about the database, or via overhearing in a wireless network, or from the information locally stored in the user's cache.

The problem is to design a protocol for generating the user's query and the servers' answers which satisfy one of the following two privacy conditions: (W, S)-privacy or W-privacy. We refer to this problem as *PIR with Private Coded Side Information (PIR-PCSI)* when (W, S)-privacy is required and *PIR with Coded Side Information (PIR-CSI)* when W-privacy is required.

Related Work: The setting in which the side information is a random subset of messages is referred to as *PIR with Side Information (PIR-SI)* or *PIR with Private Side Information (PIR-PSI)* when W-privacy or (W, S)-privacy is required, respectively. The single-server settings of these problems were studied in [21–23], and their multi-server settings were studied in [24, 25, 29]. In Chapter 2, we studied the single-server setting of a related problem in which the side information is a random linear combination of a subset of messages. In particular, Section 2.4 and Section 2.5 studied the single-server setting of PIR-CSI and PIR-PCSI problems, respectively. In this chapter, we study the extension of these works to the multi-server settings in Section 3.4 and Section 3.5, respectively.

Our Contributions

Depending on whether the support set of the user's coded side information includes the user's demand or not, we consider two different models for each of the PIR-PCSI and PIR-CSI problems. In the first model, referred to as *Model I*, the demand does not belong to the support set of the coded side information, whereas in the second model, referred to as *Model II*, the demand belongs to the support set of the coded side information. We refer to the PIR-PCSI (or PIR-CSI) problem under Model I and Model II as *PIR-PCSI–I* (or *PIR-CSI–I*) and *PIR-PCSI–II* (or *PIR-CSI–II*), respectively.

For each of these settings, we define the capacity as the ratio of the number of information bits in a message to the minimum number of information bits downloaded from the servers over all protocols that satisfy the privacy condition. We also define the server-symmetric capacity for the PIR-CSI problem similarly, except when the minimum is taken over all server-symmetric protocols (the protocols in which the user's query and the servers' answers are symmetric in structure) that satisfy the privacy condition.

In this work, our goal is to characterize the capacity (or server-symmetric capacity) of the multi-server settings of the PIR-PCSI-I, PIR-PCSI-II, PIR-CSI-I, and PIR-CSI-II problems, and to design a capacity-achieving protocol for each of these settings. The capacity results for the multi-server PIR-PCSI and multi-server PIR-CSI, are respectively summarized in Table 3.1 and Table 3.2. To present our capacity results, we define the function $\Phi(A,B) \triangleq (1+A+A^2+\cdots+A^{B-1})^{-1}$ for a positive real number A and positive integer number B. The main contributions of this work in each of these settings are summarized below.

Multi-Server PIR-PCSI

For multi-server PIR-PCSI–I setting, we prove that the capacity is $\Phi(1/N, K-M)$. This result is interesting because the capacity of the multi-server PIR-PSI is equal to

$\Phi(1/N, K-M)$, as shown in [24]. This result shows that there is no loss in capacity due to restricting the user's side information to *one* random linear combination of M messages, instead of M uncoded messages. The converse proof follows from the fact that the capacity of this setting is upper-bounded by the capacity of the multi-server PIR-PSI which is given by $\Phi(1/N, K-M)$ (see [24, Theorem 1]).

For the achievability proof, we devise a new protocol that builds upon two existing achievability schemes for two different problems: (i) the Private Computation (PC) scheme of [85] for multi-server private computation where a user wishes to privately retrieve one arbitrary linear combination of the messages replicated at multiple servers, and (ii) our Specialized GRS Code scheme proposed for single-server PIR-PCSI in Sec. 2.4.

The main ideas of our achievability scheme are as follows. First, the user utilizes the Specialized GRS Code scheme of [34] for single-server PIR-PCSI to construct $K-M$ super-messages which are some linearly independent combinations of the original messages, to play the role of the original messages in a multi-server private computation problem. Then, the user and all the N servers leverage the PC scheme of [85] for the constructed $K-M$ super-messages in such a way that the user can privately download one of $\binom{K}{M+1}$ linear combinations of the $K-M$ super-messages where the support of each linear combination is a distinct subset of $[K]$ of size $M+1$.

For the multi-server PIR-PCSI–II setting, we show that the capacity is lower-bounded by $\Phi(1/N, K-M+1)$. The proof is based on a new achievability scheme that leverages the PC scheme of [85] for multi-server private computation, combined with our Modified Specialized GRS Code scheme proposed in [34] for single-server PIR-PCSI.

Multi-Server PIR-CSI

For the multi-server PIR-CSI–I setting, we show that, for any $1 \leq M \leq K-1$, the capacity is equal to $\Phi(1/N, \lceil \frac{K}{M+1} \rceil)$. Interestingly, the capacity in this case is the same

Privacy Condition	(W, S)-Privacy	
Model	$W \notin S$ (PIR-PCSI–I)	$W \in S$ (PIR-PCSI–II)
Parameters	$1 \leq M \leq K-1$	$2 \leq M \leq K$
Capacity	$\left(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-M-1}}\right)^{-1}$	Lower bound: $\left(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-M}}\right)^{-1}$
Achievability Scheme	PC + Specialized GRS Code	PC + Modified Specialized GRS Code

Table 3.1: Summary of our main results for multi-server PIR-PCSI

Privacy Condition	W-Privacy	
Model	$W \notin S$ (PIR-CSI–I)	$W \in S$ (PIR-CSI–II)
Parameters	$1 \leq M \leq K-1$	$2 \leq M \leq K$
Capacity	$\left(1+\frac{1}{N}+\cdots+\frac{1}{N^{\left\lceil \frac{K}{M+1} \right\rceil -1}}\right)^{-1}$	1 for $M=2$, $M=K$ Open for $3 \leq M \leq K-1$
Server-Symmetric Capacity		1 for $M=2$, $M=K$ $\frac{N}{N+1}$ for $3 \leq M \leq K-1$
Achievability Scheme	Sun-Jafar Scheme [8] + Modified Partition-and-Code	Sun-Jafar Scheme [8]+ Randomized Selection-and-Code

Table 3.2: Summary of our main results for multi-server PIR-CSI

as that of multi-server PIR-SI as shown in [25]. This result shows that by knowing only one linear combination of M messages as side information, the privacy requirement can be satisfied as efficiently as (in terms of download cost) knowing M (uncoded) messages separately. Moreover, comparing this result with the capacity of multi-server PIR without side information [8], one can see that having a coded side information (which is not a function of the demanded message) of size M reduces the effective number of messages from K to $\lceil K/(M+1) \rceil$.

For the multi-server PIR-CSI–II setting, we show that the capacity is equal to 1 for $M = 2$ and $M = K$, and the server-symmetric capacity is equal to $N/(N+1)$ for any $3 \leq M \leq K-1$. Again, a comparison of these results with the capacity of multi-server PIR without side information reveals that having a coded side information (which is a function of the demanded message) of size $M \in \{2, K\}$ and $M \in \{3, \ldots, K-1\}$ reduces the effective number of messages from K to 1 and 2, respectively.

Our converse proofs rely on new information-theoretic arguments, and the achievability schemes are inspired by our proposed scheme in [30] for single-server PIR-CSI as well as the Sun-Jafar scheme of [8] for multi-server PIR.

3.2 Problem Setup and Formulation

In this section, we generalize the problem formulation in Section 2.2 to the multi-server setting.

3.2.1 Basic Notation

Throughout this chapter, we denote random variables by bold-face letters and their realizations by regular letters. The functions $\mathbb{P}(\cdot)$, $\mathbb{P}(\cdot|\cdot)$, $H(\cdot)$, $H(\cdot|\cdot)$, and $I(\cdot;\cdot|\cdot)$ denote probability, conditional probability, entropy, conditional entropy, and conditional mutual information, respectively.

Let $\mathbb{F}_q$ be a finite field for a prime power q, and let $\mathbb{F}_q^{\times} \triangleq \mathbb{F}_q \setminus \{0\}$ be the multiplicative group of $\mathbb{F}_q$. Let $\mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$ for an integer $m \geq 1$, and let $L \triangleq m \log_2 q$. The parameters q and m are referred to as the *base-field size* and the *field-extension degree*, respectively.

For an integer $i \geq 1$, let $[i] \triangleq \{1, \ldots, i\}$. Let $K \geq 1$ and $1 \leq M \leq K$ be two integers, and let $\mathcal{K} \triangleq \{1, \ldots, K\}$. We denote by $\mathcal{S}$ the set of all M-subsets (i.e., all subsets of size M) of $\mathcal{K}$, and denote by $\mathcal{C}$ the set of all sequences of size M (i.e., all length-M sequences) with elements from $\mathbb{F}_q^{\times}$. Note that $|\mathcal{S}| = \binom{K}{M}$ and $|\mathcal{C}| = (q-1)^M$.

3.2.2 Setup and Assumptions

There are N non-colluding servers, each of which stores an identical copy of a database consists of K messages $X_1, \ldots, X_K$, denoted by $X_{\mathcal{K}} \triangleq \{X_1, \ldots, X_K\}$, where $\mathbf{X}_i$'s are independently and uniformly distributed over $\mathbb{F}_{q^m}$, that is, $H(\mathbf{X}_i) = L$ for all $i \in \mathcal{K}$ and $H(\mathbf{X}_{\mathcal{K}}) = KL$, where $\mathbf{X}_{\mathcal{K}} \triangleq \{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$.

There is a user who wishes to retrieve a message X_W for some $W \in \mathcal{K}$ from the servers, and knows a linear combination $Y^{[S,C]} \triangleq \sum_{i \in S} c_i X_i$ on the messages $X_S \triangleq \{X_i : i \in S\}$, for some $S \triangleq \{i_1, \ldots, i_M\} \in \mathcal{S}$ and $C \triangleq \{c_{i_1}, \ldots, c_{i_M}\} \in \mathcal{C}$. We refer to X_W as the *demand*, W as the *demand index*, X_S as the *side information support set*, S as the *side information support index set*, M as the *side information support size*, and $Y^{[S,C]}$ as the *(coded) side information*.

We assume that $\mathbf{S}$ and $\mathbf{C}$ are uniformly distributed over $\mathcal{S}$ and $\mathcal{C}$, respectively. Also, we consider two different models for the conditional distribution of $\mathbf{W}$ given $\mathbf{S} = S$:

Model I: $\mathbf{W}$ is uniformly distributed over $\mathcal{K} \setminus S$,

$$\mathbb{P}(\mathbf{W} = W | \mathbf{S} = S) = \begin{cases} \frac{1}{K-M}, & W \in \mathcal{K} \setminus S, \\ 0, & \text{otherwise;} \end{cases}$$

Model II: $\mathbf{W}$ is uniformly distributed over S,

$$\mathbb{P}(\mathbf{W} = W | \mathbf{S} = S) = \begin{cases} \frac{1}{M}, & W \in S, \\ 0, & \text{otherwise.} \end{cases}$$

For both Models I and II, $\mathbf{W}$ is distributed uniformly over $\mathcal{K}$.

Let $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}$ be an indicator random variable such that that $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = 0$ if $\mathbf{W} \notin \mathbf{S}$, and $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = 1$ otherwise. Note that $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = 0$ in Model I, and $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = 1$ in Model II.

We assume that the servers know the underlying model (i.e., whether $\mathbf{W} \notin \mathbf{S}$ or $\mathbf{W} \in \mathbf{S}$), the side information support size M, the distributions of $\mathbf{S}$ and $\mathbf{C}$, and the conditional distribution of $\mathbf{W}$ given $\mathbf{S}$, in advance; whereas the realizations W, S, C are unknown to the servers in advance.

3.2.3 Privacy and Recoverability Conditions

For any W, S, C, to retrieve X_W, the user generates N queries $Q_1^{[W,S,C]}, \ldots, Q_N^{[W,S,C]}$, and sends to the n-th server the query $Q_n^{[W,S,C]}$. Each query $Q_n^{[W,S,C]}$ for $n \in [N]$ is assumed to be a (potentially stochastic) function of W, S, C. For simplifying the notation, we denote $\mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]}$ by $\mathbf{Q}_n$ for all $n \in [N]$. The query must satisfy one of the following two privacy conditions:

(i) both the user's demand index and side information support index set must be protected from the servers;

(ii) only the user's demand index (and not necessarily the side information support index set) must be protected from the servers.

The condition (i) is referred to as the (W, S)-*privacy condition*, and the condition (ii) is referred to as the W-*privacy condition*. (Note that (W, S)-privacy is a stronger condition than W-privacy.) The (W, S)-privacy condition implies that $(\mathbf{W}, \mathbf{S})$ and $\mathbf{Q}_n$ for all $n \in [N]$, must be conditionally independent given $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}$, that is,

$$I(\mathbf{W}, \mathbf{S}; \mathbf{Q}_n | \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}) = 0. \quad \forall n \in [N]$$

The W-privacy condition implies that $\mathbf{W}$ and $\mathbf{Q}_n$ for all $n \in [N]$, must be conditionally independent given $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}$, that is,

$$I(\mathbf{W}; \mathbf{Q}_n | \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}) = 0 \quad \forall n \in [N].$$

Equivalently, for a given $\theta \in \{0, 1\}$, when (W, S)-privacy is required, it must hold that

$$\begin{aligned}&\mathbb{P}(\mathbf{W} = W^*, \mathbf{S} = S^* | \mathbf{Q}_n = Q_n^{[W,S,C]}, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = \theta) \\ &= \mathbb{P}(\mathbf{W} = W^*, \mathbf{S} = S^* | \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = \theta)\end{aligned}$$

for all $n \in [N]$, $W^* \in \mathcal{K}$ and $S^* \in \mathcal{S}$, and when W-privacy is required, it must hold that

$$\begin{aligned}&\mathbb{P}(\mathbf{W} = W^* | \mathbf{Q}_n = Q_n^{[W,S,C]}, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = \theta) \\ &= \mathbb{P}(\mathbf{W} = W^* | \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = \theta)\end{aligned}$$

for all $n \in [N]$ and $W^* \in \mathcal{K}$.

Upon receiving $Q_n^{[W,S,C]}$, the n-th server sends to the user an answer $A_n^{[W,S,C]}$, which is a (deterministic) function of the query $Q_n^{[W,S,C]}$, the indicator variable $\mathbb{1}_{\{W \in S\}}$, and the messages in $X_{\mathcal{K}}$. For simplifying the notation, we denote $\mathbf{A}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]}$ by $\mathbf{A}_n$ for all $n \in [N]$. It should be noted that $(\mathbf{W}, \mathbf{S}, \mathbf{C}) \to (\mathbf{Q}_n, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}, \mathbf{X}_{\mathcal{K}}) \to \mathbf{A}_n$ forms a Markov chain, and for all $n \in [N]$, it holds that $H(\mathbf{A}_n | \mathbf{Q}_n, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}, \mathbf{X}_{\mathcal{K}}, \mathbf{W}, \mathbf{S}, \mathbf{C}) = 0$.

The answers from all servers $A_1^{[W,S,C]}, \ldots, A_N^{[W,S,C]}$ along with $Q_1^{[W,S,C]}, \ldots, Q_N^{[W,S,C]}$, $\mathbb{1}_{\{W \in S\}}$, $Y^{[S,C]}$, and W, S, C must enable the user to retrieve the demand X_W. That is, it must hold that

$$H(\mathbf{X}_{\mathbf{W}} | \mathbf{A}, \mathbf{Q}, \mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}}, \mathbf{Y}^{[\mathbf{S},\mathbf{C}]}, \mathbf{W}, \mathbf{S}, \mathbf{C}) = 0.$$

where $\mathbf{A} \triangleq \{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$ and $\mathbf{Q} \triangleq \{\mathbf{Q}_1, \ldots, \mathbf{Q}_N\}$. We refer to this condition as the *recoverability condition*.

3.2.4 PIR-PCSI and PIR-CSI Problems

For each type of privacy requirement and for each model, the problem is to design a protocol for generating queries $Q_1^{[W,S,C]}, \ldots, Q_N^{[W,S,C]}$ (and the corresponding answers

$A_1^{[W,S,C]}, \ldots, A_N^{[W,S,C]}$, given $Q_1^{[W,S,C]}, \ldots, Q_N^{[W,S,C]}$, $\mathbb{1}_{\{W \in S\}}$, $X_{\mathcal{K}}$) for any given W, S, C, such that both the privacy and recoverability conditions are satisfied. Note that the protocol is assumed to be known at the servers. When (W, S)-privacy is required, we refer to this problem as *Private Information Retrieval (PIR) with Private Coded Side Information (PIR-PCSI)*, and when W-privacy is required we refer to this problem as *PIR with Coded Side Information (PIR-CSI)*.

The PIR-PCSI problem under Model I (or Model II) is referred to as the *PIR-PCSI–I* (or *PIR-PCSI–II*) setting; and the PIR-CSI problem under Model I (or Model II) is referred to as the *PIR-CSI–I* (or *PIR-CSI–II*) setting. A protocol for generating queries/answers for the PIR-PCSI–I (or PIR-PCSI–II) setting is referred to as a *PIR-PCSI–I* (or *PIR-PCSI–II*) *protocol*. A *PIR-CSI–I* (or *PIR-CSI–II*) *protocol* is defined similarly.

In this work, for the PIR-CSI problem, we focus on *server-symmetric* protocols in which the user's queries and the servers' answers are symmetric in structure. In particular, we refer to a PIR-CSI–I protocol (or respectively, PIR-CSI–II protocol) as *server-symmetric* if

$$(\mathbf{Q}_n^{[W,S,C]}, \mathbf{A}_n^{[W,S,C]}, \mathbf{X}_{\mathcal{K}}) \sim (\mathbf{Q}_{n'}^{[W,S,C]}, \mathbf{A}_{n'}^{[W,S,C]}, \mathbf{X}_{\mathcal{K}})$$

holds for all $n, n' \in [N]$ and for any $W \in \mathcal{K}, S \in \mathcal{S}, C \in \mathcal{C}$ under Model I (or respectively, Model II), where the relation $\mathbf{U} \sim \mathbf{V}$ means that $\mathbf{U}$ and $\mathbf{V}$ have identical distributions. It should be noted that most of the existing multi-server PIR protocols (with information-theoretic guarantees) are server-symmetric. Server-symmetric protocols are particularly of interest because the symmetry of queries/answers across the servers makes the implementation quite simple in practice.

3.2.5 Capacity and Server-Symmetric Capacity

The *rate of a PIR-PCSI–I (or PIR-PCSI–II) protocol* is defined as the ratio of the entropy of a message, i.e., L, to the conditional entropy of $\mathbf{A}^{[\mathbf{W},\mathbf{S},\mathbf{C}]}$ given that $\mathbb{1}_{\{\mathbf{W} \in \mathbf{S}\}} = 0$

(or $\mathbb{1}_{\{\mathbf{W}\in\mathbf{S}\}} = 1$). The *rate of a PIR-CSI–I (or PIR-CSI–II) protocol* is defined similarly.

The *capacity of PIR-PCSI–I (or PIR-PCSI–II) setting* is defined as the supremum of rates over all PIR-PCSI–I (or PIR-PCSI–II) protocols and over all base-field sizes q and all field-extension degrees m; and the *capacity of PIR-CSI–I (or PIR-CSI–II) setting* is defined similarly.

The *server-symmetric capacity of PIR-CSI–I (or PIR-CSI–II) setting* is defined as the supremum of rates over all server-symmetric PIR-PCSI–I (or PIR-PCSI–II) protocols and all q and m.

3.2.6 Problem Statement

In this work, our goal is to derive upper bounds on the capacity (server-symmetric capacity) of the multi-server settings of the PIR-PCSI–I, PIR-PCSI–II, PIR-CSI–I, and PIR-CSI–II problems, and to design protocols that achieve the corresponding upper-bounds.

3.3 Necessary Condition

The following lemma renders a necessary condition for any server-symmetric PIR-CSI–I (or PIR-CSI–II) protocol that satisfies the W-privacy condition. This lemma plays a key role in the converse proof of our main result for the PIR-CSI–II problem.

Lemma 12. *Any server-symmetric PIR-CSI–I (or PIR-CSI–II) protocol satisfies the following condition: for any $W, W' \in \mathcal{K}, S \in \mathcal{S}, C \in \mathcal{C}$ with $W \notin S$ (or $W \in S$), there exist $S' \in \mathcal{S}, C' \in \mathcal{C}$ with $W' \notin S'$ (or $W' \in S'$), such that*

$$(\mathbf{Q}_n^{[W,S,C]}, \mathbf{A}_n^{[W,S,C]}, \mathbf{X}_{\mathcal{K}}) \sim (\mathbf{Q}_n^{[W',S',C']}, \mathbf{A}_n^{[W',S',C']}, \mathbf{X}_{\mathcal{K}})$$

holds for all $n \in [N]$.

Proof. The proof is by the way of contradiction, and based on the definitions of W-privacy and server-symmetry. To protect the user's privacy, for different demands, the strate-

gies (queries and answers) must be indistinguishable (identically distributed) from the perspective of each server. In particular, for each $n \in [N]$, it must hold that for any $W \in \mathcal{K}, S \in \mathcal{S}, C \in \mathcal{C}$ with $W \notin S$ (or $W \in S$), and any candidate demand $W' \in \mathcal{K}$, there exist $S_n \in \mathcal{S}, C_n \in \mathcal{C}$ with $W' \notin S_n$ (or $W' \in S_n$) that satisfy the condition of the lemma for the server n. Otherwise, if there do not exist such $S_n \in \mathcal{S}, C_n \in \mathcal{C}$ that satisfy the condition of the lemma for some server n, then the privacy condition is violated. Moreover, by the server-symmetry assumption, for any candidate demand $W' \in \mathcal{K}$, there must exist $S' \in \mathcal{S}, C' \in \mathcal{C}$ (independent of n) with $W' \notin S'$ (or $W' \in S'$) that make the strategies indistinguishable from the perspective of each server $n \in [N]$. That is, there must exist $S' \in \mathcal{S}, C' \in \mathcal{C}$ with $W' \notin S'$ (or $W' \in S'$) that satisfy the condition of the lemma for all servers. Otherwise, the server-symmetry assumption is violated. □

3.4 Multi-Server PIR with Private Coded Side Information (PIR-PCSI)

In this section, Theorem 5 characterizes the capacity of the PIR-PCSI–I, denoted by $C_{(W,S)-I}$, and Theorem 6 presents a lower-bound on the capacity of the PIR-PCSI–II problem, denoted by $C_{(W,S)-II}$. The proofs of theorems 5 and 6 are given in sections 3.4.1 and 3.4.2, respectively.

3.4.1 Multi-Server PIR-PCSI-I

Theorem 5. *The capacity of the multi-server PIR-PCSI–I problem with N servers, K messages, and side information size $1 \leq M \leq K-1$ is given by*

$$C_{(W,S)-I} = \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-M-1}}\right)^{-1}.$$

Interestingly, this result indicates that the capacity of multi-server PIR-PCSI–I, i.e., $C_{(W,S)-I}$, is equal to the capacity of the multi-server PIR-PSI [24] where M uncoded messages are available at the user as side information.

Proof of Theorem 5

3.4.1.1 Converse

Note that having only a random linear combination of M messages as side information instead of M uncoded messages, cannot increase the capacity which implies the converse. Thus, to complete the proof of Theorem 5, we only need to prove the achievability (see Section 3.4.1.2). Notably, our results show that having only one random linear coded message instead of multiple uncoded messages does not decrease the capacity, either.

3.4.1.2 Achievability

We complete the proof of Theorem 5 by proposing a scheme for any arbitrary N, $K \geq 1$ and $0 \leq M \leq K-1$ that achieves the rate $\left(1 + 1/N + \cdots + 1/N^{K-M-1}\right)^{-1}$. The proposed protocol, referred to as the *Multi-Server PIR-PCSI–I protocol*, is a non-trivial combination of the Specialized GRS Code scheme of [34] for single-server PIR-PCSI and the Private Computation (PC) scheme of [85] for multi-server PC problem. For the proposed protocol, we assume that $q \geq K$, and each message X_i consists of $m = N^{\binom{K}{M+1}}$ symbols over $\mathbb{F}_q$.

Multi-Server PIR-PCSI–I protocol: The protocol consists of the following five steps:

Step 1: The user utilizes the Specialized GRS Code protocol proposed in [34] to first construct a polynomial

$$p(x) = \sum_{i=0}^{K-M-1} p_i x^i \triangleq \prod_{i \notin S \cup W} (x - \omega_i)$$

where $\omega_1, \ldots, \omega_K$ are K arbitrarily chosen distinct elements from $\mathbb{F}_q$, and then construct $r \triangleq K - M$ vectors $\underline{u}_1, \ldots, \underline{u}_r$, each of length K, such that $\underline{u}_i = [\beta_1 \omega_1^{i-1}, \ldots, \beta_K \omega_K^{i-1}]$ for $i \in [r]$, where $\beta_j = \frac{c_j}{p(\omega_j)}$ for $j \in S$, and β_j is a randomly chosen element from $\mathbb{F}_q^{\times}$ for $j \notin S$.

Step 2: Let $\hat{X}_i \triangleq \sum_{j=1}^{K} \beta_j \omega_j^{i-1} X_j$ for $i \in [r]$. Each $\hat{X}_i$ is referred to as a *coded message*. Note that the vector $\underline{u}_i$ (constructed in Step 1) is the vector of coefficients of the messages $\{X_i\}_{i\in[K]}$ in the coded message $\hat{X}_i$. Let $F \triangleq \binom{K}{M+1}$, and let $J_1, J_2, \ldots, J_F$ be the collection of all $(M+1)$-subsets of $[K]$ in a lexicographical order. The structure of the Specialized GRS Code protocol [34] ensures that for each J_f, $f \in [F]$, there exist exactly $q-1$ linear combinations $Z_f^1, Z_f^2, \ldots, Z_f^{q-1}$ of the messages $\{X_i\}_{i\in J_f}$ with (non-zero) coefficients from $\mathbb{F}_q^{\times}$, such that for every $k \in [q-1]$, Z_f^k can be written as a linear combination of the coded messages $\hat{X}_1, \ldots, \hat{X}_r$. Let $\underline{v}_f^k \triangleq [v_{f,1}^k, \ldots, v_{f,r}^k]$ be a vector of length r such that $Z_f^k = \sum_{i=1}^{r} v_{f,i}^k \hat{X}_i$.

Note that, for each $f \in [F]$, $Z_f^1, Z_f^2, \ldots, Z_f^{q-1}$ are the same up to a scalar multiple, i.e., for each $k \in [q-1]$, $Z_f^k = \alpha_k Z_f^1$, or equivalently, $\underline{v}_f^k = \alpha_k \underline{v}_f^1$, for some distinct $\alpha_k \in \mathbb{F}_q^{\times}$. For each $f \in [F]$, let $i_f \triangleq \min(J_f)$. Note also that for every $f \in [F]$, there exists a unique $k_f \in [q-1]$ such that the coefficient of the message X_{i_f} in the linear combination $Z_f^{k_f}$ is equal to 1. The user then constructs F vectors $\underline{v}_1, \ldots, \underline{v}_F$, each of length r, such that $\underline{v}_f = \underline{v}_f^{k_f}$. (Note that the above procedure dictates a specific choice of the coefficient vectors $\underline{v}_f$. However, for each $f \in [F]$, the vector $\underline{v}_f$ can be chosen arbitrarily from the set of vectors $\{\underline{v}_f^k\}_{k\in[q-1]}$.) Let $Z_f \triangleq Z_f^{k_f}$ for $f \in [F]$. Each Z_f is referred to as a (linear) function. Note that $\underline{v}_f$ is the vector of coefficients of the coded messages $\{\hat{X}_i\}_{i\in[r]}$ in the function Z_f.

Step 3: The user sends to all servers the vectors $\underline{u}_1, \ldots, \underline{u}_r$ (associated with coded messages $\hat{X}_1, \ldots, \hat{X}_r$), and the vectors $\underline{v}_1, \ldots, \underline{v}_F$ (associated with functions $Z_1, \ldots, Z_F$). It is noteworthy that the user needs only to send the vectors $\{\underline{u}_i\}_{i\in[r]}$ to all servers, and each server can construct the vectors $\{\underline{v}_f\}_{f\in[F]}$ by using $\{\underline{u}_i\}_{i\in[r]}$ (according to the procedure described in Step 2).

Step 4: The user and the servers leverage the PC scheme of [85] with r (independent) messages and F (linear) functions of these messages in order for the user to privately re-

trieve one of these functions. In particular, the $r = K - M$ coded messages $\{\hat{X}_i\}_{i\in[r]}$ and the F functions $\{Z_f\}_{f\in[F]}$ play the role of the original messages and the functions in the PC scheme, respectively, and the user is interested in retrieving the function Z_{f^*} privately, where Z_{f^*} is an $\mathbb{F}_q^\times$-linear combination (i.e., a linear combination with non-zero coefficients only) of the messages $\{X_i\}_{i\in W\cup S}$. (By the construction, there exists one (and only one) function Z_f among $Z_1, \ldots, Z_F$ such that Z_f is an $\mathbb{F}_q^\times$-linear combination of the messages $\{X_i\}_{i\in W\cup S}$.) To be more specific, each server first constructs the coded messages $\{\hat{X}_i\}_{i\in[r]}$ by using the coefficient vectors $\{\underline{u}_i\}_{i\in[r]}$ (defined in Step 3), and then constructs the functions $\{Z_f\}_{f\in[F]}$ by using the coded messages $\{\hat{X}_i\}_{i\in[r]}$ and the coefficient vectors $\{\underline{v}_f\}_{f\in[F]}$ (defined in Step 3). Note that each function Z_f for $f \in [F]$ consists of $m = N^F$ $\mathbb{F}_q$-symbols where N is the number of servers. Then, each server sends to the user $m(1/N + 1/N^2 + \cdots + 1/N^{K-M})$ carefully designed linear combinations of all $\mathbb{F}_q$-symbols associated with all functions $\{Z_f\}_{f\in[F]}$. The details of the design of the user's query to each server as well as the linear combinations transmitted by each server (which also depend on the query of the user) can be found in [85, Section 4].

Example 6. Assume that there are $N = 2$ servers, $K = 4$ messages from $\mathbb{F}_{5^{16}}$, and $M = 2$. Note that each message consists of $m = N^{\binom{K}{M+1}} = 16$ symbols from $\mathbb{F}_5$. Suppose that the user demands the message X_1 and has a coded side information $Y = X_2 + X_3$, i.e., $W = 1$, $S = \{2, 3\}$, and $C = \{1, 1\}$ (i.e., $c_2 = 1, c_3 = 1$).

First, the user picks $K = 4$ distinct elements $\omega_1, \ldots, \omega_4$ from $\mathbb{F}_5$. Suppose that the user chooses $\omega_1 = 0$, $\omega_2 = 1$, $\omega_3 = 2$, $\omega_4 = 3$. Then, the user constructs the polynomial

$$p(x) = \prod_{i\notin S\cup W} (x - \omega_i) = x - \omega_4 = x - 3.$$

The user then computes β_j for $j \in S$, i.e., β_2 and β_3, by setting $\beta_2 = \frac{c_2}{p(\omega_2)} = 2$ and $\beta_3 = \frac{c_3}{p(\omega_3)} = 4$, and chooses β_j for $j \notin S$, i.e., β_1 and β_4, at random (from $\mathbb{F}_5^\times$). Assume

that the user chooses $\beta_1 = 1$ and $\beta_4 = 2$. Then, the user constructs $r = K - M = 2$ vectors $\underline{u}_1$ and $\underline{u}_2$, each of length $K = 4$, such that $\underline{u}_i = [\beta_1 \omega_1^{i-1}, \ldots, \beta_K \omega_K^{i-1}]$ for $i \in \{1, 2\}$. That is, the user constructs $\underline{u}_1 = [1, 2, 4, 2]$ and $\underline{u}_2 = [0, 2, 3, 1]$. For set $J_1 = \{1, 2, 3\}$, there exist exactly $q - 1 = 4$ vectors $\underline{v}_1^k = [k, 3k]$ for $k \in \{1, \ldots, 4\}$ such that $k\underline{u}_1 + 3k\underline{u}_2 = k[1, 3, 3, 0]$.

It should be noted that there exists no other vector $\underline{v} = [v_1, v_2]$ such that the support of the vector $v_1\underline{u}_1 + v_2\underline{u}_2$ is $J_1 = \{1, 2, 3\}$. Note that the coefficient of the message $X_{i_1} = X_1$ (i.e., $i_1 = \min(J_1) = 1$) in the function Z_1 is equal to 1 when $k = 1$. Thus, the user constructs the vector $\underline{v}_1 = \underline{v}_1^1 = [1, 3]$. Similarly, the user constructs the vectors $\underline{v}_2 = [1, 2]$, $\underline{v}_3 = [1, 4]$ and $\underline{v}_4 = [0, 3]$. Then, the user sends to all servers the vectors $\underline{u}_1$ and $\underline{u}_2$ (associated with the coded messages $\hat{X}_1$ and $\hat{X}_2$), and the vectors $\underline{v}_1, \ldots, \underline{v}_4$ (associated with the functions $Z_1, \ldots, Z_4$). Using the coefficient vectors $\underline{u}_1$ and $\underline{u}_2$, each server first constructs the following two coded messages

$$\hat{X}_1 = X_1 + 2X_2 + 4X_3 + 2X_4 \quad \text{and} \quad \hat{X}_2 = 2X_2 + 3X_3 + X_4.$$

Then, the user constructs the functions $Z_1, \ldots, Z_4$ using the coded messages $\hat{X}_1$ and $\hat{X}_2$ and the coefficient vectors $\underline{v}_1, \ldots, \underline{v}_4$ as follows:

$$\begin{aligned}
Z_1 &= \hat{X}_1 + 3\hat{X}_2 = X_1 + 3X_2 + 3X_3 \\
Z_2 &= \hat{X}_1 + 2\hat{X}_2 = X_1 + X_2 + 4X_4 \\
Z_3 &= \hat{X}_1 + 4\hat{X}_2 = X_1 + X_3 + X_4 \\
Z_4 &= 3\hat{X}_2 = X_2 + 4X_3 + 3X_4
\end{aligned}$$

Finally, the user and the servers apply the PC scheme of [85] for two coded messages $\hat{X}_1$, $\hat{X}_2$ in order for the user to privately retrieve the function Z_1. (Note that among the functions $Z_1, \ldots, Z_4$, only Z_1 is an $\mathbb{F}_5^\times$-linear combination of the messages $\{X_i\}_{i \in W \cup S} =$

$\{X_1, X_2, X_3\}$.) The details of the PC scheme for this example are as follows. Let $\pi : [16] \to [16]$ be a randomly chosen permutation. Let

$$u_f(i) \triangleq \sigma_i Z_f(\pi(i))$$

for $f \in [4]$ and $i \in [16]$, where $Z_f(\pi(i))$ is the $\pi(i)$-th $\mathbb{F}_5$-symbol of Z_f, and σ_i is a randomly chosen element from $\{-1, +1\}$. For simplifying the notation, we define the following: $(a_i, b_i, c_i, d_i) = (u_1(i), u_2(i), u_3(i), u_4(i))$ for $\forall i \in [16]$.

Then, from each of the two servers (S1 and S2), the user queries 15 carefully designed linear combinations of the symbols $\{\{a_i\}_{i\in[16]}, \{b_i\}_{i\in[16]}, \{c_i\}_{i\in[16]}, \{d_i\}_{i\in[16]}\}$, as given in Table 3.3 [85].

As shown in [85], among the 15 symbols queried from S1 (or S2), based on the information obtained from S2 (or S1), 3 symbols are redundant. For instance, consider the 15 symbols queried from S1. (Similar observations can be made regarding the queries from S2.) Among the 4 symbols $\{a_1, b_1, c_1, d_1\}$, any 2 symbols suffice to recover the other 2 symbols. For example, c_1 and d_1 can be obtained from a_1 and b_1. (Note that Z_3 and Z_4 can be written as a linear combination of Z_1 and Z_2.)

Thus, the server S1 needs to send two arbitrary symbols from $\{a_1, b_1, c_1, d_1\}$. In addition, given any 2 symbols from $\{a_2, b_2, c_2, d_2\}$, any 5 symbols among the 6 symbols $\{a_3 - b_2,\ a_4 - c_2,\ a_5 - d_2,\ b_4 - c_3,\ b_5 - d_3,\ c_5 - d_4\}$ queried from S1 would suffice to recover the remaining symbol. For example, $c_5 - d_4$ can be obtained from the symbols $\{a_3 - b_2, a_4 - c_2, a_5 - d_2, b_4 - c_3, b_5 - d_3, b_2, d_2\}$ (for the details, see [85, Section 5.1]). Thus, each of the two servers S1 and S2 needs to send to the user only 12 symbols.

In particular, the servers S1 transmits 2 arbitrary symbols from $\{a_1, b_1, c_1, d_1\}$, 5 arbitrary symbols from $\{a_3 - b_2, a_4 - c_2, a_5 - d_2, b_4 - c_3, b_5 - d_3, c_5 - d_4\}$, and all the 4 symbols $\{a_9 - b_7 + c_6,\ a_{10} - b_8 + d_6,\ a_{11} - c_8 + d_7,\ b_{11} - c_{10} + d_9\}$, and the symbol

Table 3.3: The queries of the PC protocol for $N = 2$ servers, 2 coded messages, and $F = 4$ functions, when the user demands Z_1.

S1	S2
a_1, b_1, c_1, d_1	a_2, b_2, c_2, d_2
$a_3 - b_2$	$a_6 - b_1$
$a_4 - c_2$	$a_7 - c_1$
$a_5 - d_2$	$a_8 - d_1$
$b_4 - c_3$	$b_7 - c_6$
$b_5 - d_3$	$b_8 - d_6$
$c_5 - d_4$	$c_8 - d_7$
$a_9 - b_7 + c_6$	$a_{12} - b_4 + c_3$
$a_{10} - b_8 + d_6$	$a_{13} - b_5 + d_3$
$a_{11} - c_8 + d_7$	$a_{14} - c_5 + d_4$
$b_{11} - c_{10} + d_9$	$b_{14} - c_{13} + d_{12}$
$a_{15} - b_{14} + c_{13} - d_{12}$	$a_{16} - b_{11} + c_{10} - d_9$

$\{a_{15} - b_{14} + c_{13} - d_{12}\}$; and S2 transmits 2 arbitrary symbols from $\{a_2, b_2, c_2, d_2\}$, 5 arbitrary symbols from $\{a_6 - b_1, a_7 - c_1, a_8 - d_1, b_7 - c_6, b_8 - d_6, c_8 - d_7\}$, and the 4 symbols $\{a_{12} - b_4 + c_3, a_{13} - b_5 + d_3, a_{14} - c_5 + d_4, b_{14} - c_{13} + d_{12}\}$, and the symbol $\{a_{16} - b_{11} + c_{10} - d_9\}$.

From the answers by the servers, the user obtains all 16 symbols $a_1, \ldots, a_{16}$, and accordingly, all 16 symbols of Z_1. (Note that $a_i = u_1(i) = \sigma_i Z_1(\pi(i))$ for $i \in [16]$.) From Z_1 $(= X_1 + 3X_2 + 3X_3)$, the user can decode the desired message X_1 by subtracting off the contribution of their side information $X_2 + X_3$.

In order to retrieve X_1 which consists of 16 symbols (over $\mathbb{F}_5$), according to the proposed protocol, the user downloads 24 symbols (over $\mathbb{F}_5$) from both servers, and hence the rate of the proposed protocol is $16/24 = 2/3$.

Note that for every 3-subset $\{X_{j_1}, X_{j_2}, X_{j_3}\}$ of the messages $\{X_i\}_{i\in[4]}$, in the proposed protocol there exists one (and only one) linear combination Z_f for some $f \in [4]$ of the messages $X_{j_1}, X_{j_2}, X_{j_3}$. On the other hand, the PC scheme guarantees that no server can obtain any information about the index (f) of the linear combination Z_f being requested by the user. Thus, the proposed scheme satisfies the (W, S)-privacy condition, as it was desired in this example.

Lemma 13. *The Multi-Server PIR-PCSI–I protocol satisfies the recoverability and (W,S)-privacy conditions, and achieves the rate* $C_{(W,S)-I} = \left(1 + 1/N + \cdots + 1/N^{K-M-1}\right)^{-1}$.

Proof. Since the messages in the $\mathbf{X}_{\mathcal{K}}$ are uniformly and independently distributed over $\mathbb{F}_{q^m}$, and $\{\hat{X}_1, \ldots, \hat{X}_r\}$ are linearly independent combinations of the messages in $X_{[K]}$, thus $\{\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_r\}$ are uniformly and independently distributed over $\mathbb{F}_{q^m}$ as well, i.e., $H(\hat{\mathbf{X}}_1) = \cdots = H(\hat{\mathbf{X}}_r) = m \log q = L$. Hence, the rate of the Multi-Server PIR-PCSI–I protocol is the same as the rate of the PC protocol for N servers and $K - M$ messages, given by $\left(1 + 1/N + \cdots + 1/N^{K-M-1}\right)^{-1}$ (see [85, Theorem 1]).

From the step 4 of the Multi-Server PIR-PCSI–I protocol, it is evident that the recoverability condition is satisfied. The proof of the (W, S)-privacy of the proposed protocol is as follows. The PC protocol protects the privacy of the function (linear combination) requested by the user. That is, given the query, no server can obtain any information about the index of the function requested by the user. Consider an arbitrary server $n \in [N]$, and an arbitrary query Q_n to server n, generated by the proposed protocol. Thus, given $\mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n$, from the perspective of server n, every function Z_f for $f \in [F]$ is equally likely to include the demanded message. We denote the support of Z_f by $\mathcal{Z}_f$, i.e., $\mathcal{Z}_f$ is the set of all indices $i \in [K]$ such that X_i has a non-zero coefficient in the linear combination

Z_f. Thus, for all $f \in [F]$, we have

$$\Pr(\mathbf{W} \in \mathcal{Z}_f | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) = \frac{1}{\binom{K}{M+1}}, \tag{3.1}$$

noting that $F = \binom{K}{M+1}$. Note that any given index $W' \in [K]$ is in the support of exactly $\binom{K-1}{M}$ functions Z_f, $f \in [F]$. For any given $f \in [F]$, given $\mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n$ and $\mathbf{W} \in \mathcal{Z}_f$, from the perspective of server n, every index $W' \in \mathcal{Z}_f$ is equally likely to be the demand index. That is, for all $f \in [F]$, we have

$$\begin{aligned} &\Pr(\mathbf{W} = W' | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n, \mathbf{W} \in \mathcal{Z}_f) \\ &= \begin{cases} \frac{1}{M+1}, & W' \in \mathcal{Z}_f, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.2}$$

Furthermore, for any given $f \in [F]$ and $W' \in \mathcal{Z}_f$, we have

$$\begin{aligned} &\Pr(\mathbf{S} = S' | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n, \mathbf{W} \in \mathcal{Z}_f, \mathbf{W} = W') \\ &= \begin{cases} 1, & S' = \mathcal{Z}_f \setminus \{W'\}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.3}$$

Consider arbitrary $W' \in [K]$ and $S' \subset [K] \setminus \{W'\}, |S'| = M$. Let $f' \in [F]$ be the (unique) index such that $\mathcal{Z}_{f'} = W' \cup S'$. It is easy to see that

$$\Pr(\mathbf{W} = W', \mathbf{S} = S', \mathbf{W} \in \mathcal{Z}_f | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) = 0$$

for all $f \in [F], f \neq f'$. Thus, by using (3.1)-(3.3), we can write

$$\begin{aligned}
&\Pr(\mathbf{W} = W', \mathbf{S} = S' | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) \\
&= \sum_{f \in [F]} \Pr(\mathbf{W} = W', \mathbf{S} = S', \mathbf{W} \in \mathcal{Z}_f | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) \\
&= \Pr(\mathbf{W} = W', \mathbf{S} = S', \mathbf{W} \in \mathcal{Z}_{f'} | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) \\
&= \Pr(\mathbf{W} \in \mathcal{Z}_{f'} | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) \\
&\quad \times \Pr(\mathbf{W} = W' | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n, \mathbf{W} \in \mathcal{Z}_{f'}) \\
&\quad \times \Pr(\mathbf{S} = S' | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n, \mathbf{W} \in \mathcal{Z}_{f'}, \mathbf{W} = W') \\
&= \frac{1}{\binom{K}{M+1}} \times \frac{1}{M+1} \times 1 \\
&= \frac{M!(K-M-1)!}{K!}
\end{aligned} \tag{3.4}$$

On the other hand, we have

$$\begin{aligned}
&\Pr(\mathbf{W} = W', \mathbf{S} = S') \\
&= \Pr(\mathbf{W} = W') \times \Pr(\mathbf{S} = S' | \mathbf{W} = W') \\
&= \frac{1}{K} \times \frac{1}{\binom{K-1}{M}} \\
&= \frac{M!(K-M-1)!}{K!}.
\end{aligned} \tag{3.5}$$

From (3.4) and (3.5), for any $W' \in [K]$ and $S' \subset [K] \setminus \{W'\}$, $|S'| = M$, we have

$$\begin{aligned}
&\Pr(\mathbf{W} = W', \mathbf{S} = S' | \mathbf{Q}_n^{[\mathbf{W},\mathbf{S},\mathbf{C}]} = Q_n) \\
&= \Pr(\mathbf{W} = W', \mathbf{S} = S').
\end{aligned}$$

This completes the proof of (W, S)-privacy of the proposed protocol. □

3.4.2 Multi-Server PIR-PCSI-II

Theorem 6. *The capacity of the multi-server PIR-PCSI–II problem with N servers, K messages, and side information size $2 \leq M \leq K$ is lower-bounded by*

$$C_{(W,S)-II} \geq \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-M}}\right)^{-1}.$$

This result is interesting because it shows that the lower-bound on the capacity of the multi-server PIR-PCSI–II is the same as the capacity of multi-server PIR-SI when the size of side information is $M-1$. That is, having a side information which is only a random linear combination of M messages including the demanded message would be at least as effective as knowing $M-1$ messages separately in terms of minimizing the download cost. For the proof, we construct a PIR-PCSI–II protocol that achieves the capacity lower-bound of Theorem 6. It should be noted that the tightness of this lower bound remains open in general.

Proof of Theorem 6

3.4.2.1 Achievability

In this section, we prove the result of Theorem 2 by constructing a PIR-PCSI–II protocol, referred to as the *Multi-Server PIR-PCSI–II protocol*, for arbitrary N, $K \geq 2$ and $2 \leq M \leq K$ that achieves the rate $\left(1 + 1/N + \cdots + 1/N^{K-M}\right)^{-1}$.

For the proposed protocol, we assume that $q \geq K$, and each message is comprised of $m = N^{\binom{K}{M}}$ symbols over $\mathbb{F}_q$.

Multi-Server PIR-PCSI–II protocol: The protocol consists of four steps, where the steps 2-4 are the same as the steps 2-4 in the Multi-Server PIR-PCSI–I protocol, except that M is replaced with $M-1$ everywhere. The step 1 of the proposed protocol is as follows:

Step 1: The user utilizes the Modified Specialized GRS Code protocol proposed in [34] to first construct the polynomial $p(x)$ as follows

$$p(x) = \sum_{i=0}^{K-M} p_i x^i \triangleq \prod_{i \notin S} (x - \omega_i)$$

where $\omega_1, \ldots, \omega_K$ are K arbitrarily chosen distinct elements from $\mathbb{F}_q$, and then construct $r \triangleq K - M + 1$ vectors $\underline{u}_1, \ldots, \underline{u}_r$, each of length K, such that $\underline{u}_i = [\beta_1 \omega_1^{i-1}, \ldots, \beta_K \omega_K^{i-1}]$ for $i \in [r]$, where $\beta_j = \frac{c_j}{p(\omega_j)}$ for $j \in S \setminus W$, $\beta_W = \frac{c}{p(\omega_W)}$ where c is chosen uniformly at random from $\mathbb{F}_q^\times \setminus \{c_W\}$, and β_j is a randomly chosen element from $\mathbb{F}_q^\times$ for $j \notin S$.

Lemma 14. *The Multi-Server PIR-PCSI–II protocol satisfies the recoverability and (W,S)-privacy conditions, and achieves the rate* $\left(1 + 1/N + \cdots + 1/N^{K-M}\right)^{-1}$.

Proof. The proof is similar to the proof of Lemma 13, and omitted to avoid repetition. □

3.5 Multi-Server PIR with Coded Side Information (PIR-CSI)

We present our results for the multi-server PIR-CSI-I and multi-server PIR-CSI-II in Section 3.5.1 and Section 3.5.2, respectively.

3.5.1 Multi-Server PIR-CSI-I

Theorem 7. *The capacity of PIR-CSI–I problem with N servers, K messages, and side information size $0 \leq M \leq K - 1$ is given by*

$$\mathrm{C}_{W-I} = \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{\lceil \frac{K}{M+1} \rceil - 1}}\right)^{-1}.$$

Interestingly, the capacity of multi-server PIR with (uncoded) side information [25] is also equal to $\lceil \frac{K}{M+1} \rceil^{-1}$ where M is the number of (uncoded) messages known to the user in advance as side information. This shows that there will be no loss in capacity, when compared to the case that the user knows M randomly chosen messages separately,

even if the user knows only *one* random linear coded combination of M randomly chosen messages.

Proof of Theorem 7

3.5.1.1 Converse

The capacity of the multi-server PIR-CSI–I setting is upper bounded by the capacity of multi-server setting of the PIR problem with uncoded side information where M uncoded messages are available at the user as side information. As shown in [25], the capacity of this problem is equal to $\mathrm{C}_{W-I} = \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{\lceil \frac{K}{M+1} \rceil - 1}}\right)^{-1}$. In what follows, we provide an alternative converse proof for the *server-symmetric* protocols.

Without loss of generality, suppose the user wishes to retrieve X_W for a given $W \in [K]$, and has a side information $Y \triangleq Y^{[S,C]}$ for given $S \in \mathcal{S}, C \in \mathcal{C}$ such that $W \notin S$. The user sends to the nth server a query $Q_n^{[W,S,C]}$, and the nth server responds to the user with an answer $A_n^{[W,S,C]}$. We want to show that the total entropy of the answers from all servers, denoted by D, is lower bounded by $(1 + 1/N + \cdots + 1/N^{\lceil \frac{K}{M+1} \rceil - 1})L$. The proof proceeds as follows:

$$
\begin{aligned}
D &\geq H(\mathbf{A}^{[W,S,C]} | \mathbf{Q}^{[W,S,C]}, \mathbf{Y}) \\
&= H(\mathbf{A}^{[W,S,C]}, \mathbf{X}_W | \mathbf{Q}^{[W,S,C]}, \mathbf{Y}) && (3.6) \\
&= L + H(\mathbf{A}^{[W,S,C]} | \mathbf{Q}^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) && (3.7) \\
&\geq L + H(\mathbf{A}_1^{[W,S,C]} | \mathbf{Q}^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \\
&= L + H(\mathbf{A}_1^{[W,S,C]} | \mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) && (3.8)
\end{aligned}
$$

where (3.6) follows from $H(\mathbf{X}_W | \mathbf{A}^{[W,S,C]}, \mathbf{Q}^{[W,S,C]}, \mathbf{Y}) = 0$ (by the recoverability condition); (3.7) holds since $\mathbf{X}_W$ is independent of $(\mathbf{Q}^{[W,S,C]}, \mathbf{Y})$, and $H(\mathbf{X}_W | \mathbf{Q}^{[W,S,C]}, \mathbf{Y}) = H(\mathbf{X}_W) = L$; and (3.8) holds because $\mathbf{A}_1^{[W,S,C]}$ only depends on $(\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_{[K]})$, and is

conditionally independent of $\mathbf{Q}_n^{[W,S,C]}$ for all $n \neq 1$, given $(\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y})$.

We will consider the following two cases separately: (i) $\lceil K/(M+1) \rceil = 1$ (i.e., $K = M+1$), and (ii) $\lceil K/(M+1) \rceil > 1$ (i.e., $K > M+1$). In the case (i), we need to show that D is lower bounded by L. Since $H(\mathbf{A}_1^{[W,S,C]}|\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \geq 0$, then $D \geq L$ (by (3.8)).

In the case (ii), in order to continue lower bounding (3.8), we arbitrarily choose a message, say X_{W_1}, such that $W_1 \notin W \cup S$. (Note that such W_1 exists because $|W \cup S| = M+1 < K$.) Based on Lemma 12, there exist $S_1 \in \mathcal{S}$, $C_1 \in \mathcal{C}$ with $W_1 \notin S_1$, and accordingly $Y_1 \triangleq Y^{[S_1,C_1]}$, such that

$$H(\mathbf{A}_1^{[W,S,C]}|\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) = H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}).$$

Then, we can write

$$\begin{aligned} D &\geq L + H(\mathbf{A}_1^{[W,S,C]}|\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \\ &= L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\ &\geq L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}). \end{aligned}$$

Similarly, by the server-symmetry assumption we have

$$D \geq L + H(\mathbf{A}_n^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y})$$

for all $n \in [N]$. Combining all of these inequalities, we get

$$\begin{aligned} D &\geq L + \frac{1}{N}\sum_{n=1}^{N} H(\mathbf{A}_n^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\ &\geq L + \frac{1}{N} H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}). \end{aligned} \tag{3.9}$$

To further lower bound (3.9), we can write

$$
\begin{aligned}
&H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\
&\geq H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) \\
&= H(\mathbf{A}^{[W_1,S_1,C_1]}, \mathbf{X}_{W_1}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) \qquad (3.10) \\
&= L + H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \qquad (3.11) \\
&\geq L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \\
&= L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \qquad (3.12)
\end{aligned}
$$

where (3.10) holds since X_{W_1} is retrievable from $A^{[W_1,S_1,C_1]}, Q^{[W_1,S_1,C_1]}, Y_1, W_1, S_1, C_1$; and (3.11) holds because $\mathbf{X}_{W_1}$ is independent of $(\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1)$.

We consider two cases separately: (ii.1) $\lceil K/(M+1) \rceil = 2$, and (ii.2) $\lceil K/(M+1) \rceil > 2$. In the case (ii.1), from (3.9) and (3.12) it follows that $D \geq L + L/N$.

In the case (ii.2), to continue lower bounding (3.12), we pick a message, say X_{W_2}, such that $W_2 \notin W \cup S \cup W_1 \cup S_1$. (Note that such W_2 exists since $|W \cup S \cup W_1 \cup S_1| \leq 2(M+1) < K$.) According to Lemma 12, there exist $S_2 \in \mathcal{S}$, $C_2 \in \mathcal{C}$ with $W_2 \notin S_2$, and accordingly, $Y_2 = Y^{[S_2,C_2]}$, such that

$$
H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) = H(\mathbf{A}_1^{[W_2,S_2,C_2]}|\mathbf{Q}_1^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1).
$$

Thus, we have

$$
\begin{aligned}
&H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\
&\geq L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \\
&= L + H(\mathbf{A}_1^{[W_2,S_2,C_2]}|\mathbf{Q}_1^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \\
&\geq L + H(\mathbf{A}_1^{[W_2,S_2,C_2]}|\mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1).
\end{aligned}
$$

Similarly, by the server-symmetry assumption, we have

$$\begin{aligned} &H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]},\mathbf{X}_W,\mathbf{Y}) \\ &\geq L + H(\mathbf{A}_n^{[W_2,S_2,C_2]}|\mathbf{Q}^{[W_2,S_2,C_2]},\mathbf{X}_W,\mathbf{Y},\mathbf{X}_{W_1},\mathbf{Y}_1) \end{aligned}$$

for all $n \in [N]$. Combining all of these inequalities, we get

$$\begin{aligned} &H(\mathbf{A}^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]},\mathbf{X}_W,\mathbf{Y}) \\ &\geq L + \frac{1}{N}H(\mathbf{A}^{[W_2,S_2,C_2]}|\mathbf{Q}^{[W_2,S_2,C_2]},\mathbf{X}_W,\mathbf{Y},\mathbf{X}_{W_1},\mathbf{Y}_1). \end{aligned} \tag{3.13}$$

Putting (3.9) and (3.13) together, we get

$$D \geq L + \frac{L}{N} + \frac{1}{N^2}H(\mathbf{A}^{[W_2,S_2,C_2]}|\mathbf{Q}^{[W_2,S_2,C_2]},\mathbf{X}_W,\mathbf{Y},\mathbf{X}_{W_1},\mathbf{Y}_1).$$

By choosing the messages X_{W_i} for the indices $i \in \{3,\ldots,\lceil K/(M+1)\rceil\}$ (similarly as X_{W_1} and X_{W_2}) recursively and using the same lower bounding technique, it can be shown that

$$D \geq L + L/N + \cdots + L/N^{\lceil \frac{K}{M+1}\rceil - 1}.$$

3.5.1.2 Achievability

This section proposes a PIR-CSI–I protocol that achieves a rate equal to C_{W-I}. The proposed protocol employs the *Randomized Partitioning (RP)* scheme which we proposed in [30] for single-server PIR-CSI (under Model I) as well as the Sun-Jafar scheme of [8] for multi-server PIR.

We assume that each message consists of $N^{\lceil K/(M+1)\rceil}$ symbols over $\mathbb{F}_q$.

Multi-Server PIR-CSI–I Protocol:

Step 1: The user utilizes the RP scheme of [30] to construct $r \triangleq \lceil \frac{K}{M+1} \rceil$ sequences $I_1, \ldots, I_r$ from indices in $[K]$, each of length $M+1$, and r sequences $I'_1, \ldots, I'_r$ with elements in $\mathbb{F}_q^{\times}$, each of length $M+1$. In particular, $I_1 = \{W, S\}$ and $I'_1 = \{c, C\}$ where C is the sequence of coefficients in the user's side information $Y^{[S,C]}$, and c is randomly chosen from $\mathbb{F}_q^{\times}$. (For more details, see [30, Section IV-B].)

Step 2: The user then creates $\tilde{I}_i$ and $\tilde{I}'_i$ for each $i \in [r]$ by reordering the elements of both I_i and I'_i with the same randomly picked permutation $\pi_i : [M+1] \to [M+1]$, and constructs $I^*_i = (\tilde{I}_i, \tilde{I}'_i)$. Then, the user sends $\{I^*_{\sigma(i)}\}_{i\in[r]}$ to all servers, for a randomly chosen permutation $\sigma : [r] \to [r]$. Note that in the RP scheme, $\{I_i\}_{i\in[r]}$ and $\{I'_i\}_{i\in[r]}$ are designed in such a way that given $\{I^*_{\sigma(1)}, \ldots, I^*_{\sigma(r)}\}$, any index in $[K]$ is equally likely to be the user's demand index.

Step 3: Using $I^*_{\sigma(i)} = (\tilde{I}_{\sigma(i)}, \tilde{I}'_{\sigma(i)})$ for all $i \in [r]$, the user and all the servers form r *super-messages* $\hat{X}_1, \ldots, \hat{X}_r$ such that $\hat{X}_i = \sum_{j=1}^{M+1} c_{i_j} X_{i_j}$ for all $i \in [r]$, where $\tilde{I}_{\sigma(i)} = \{i_1, \ldots, i_{M+1}\}$ and $\tilde{I}'_{\sigma(i)} = \{c_{i_1}, \ldots, c_{i_{M+1}}\}$.

Step 4: The user and the servers then utilize the Sun-Jafar protocol with r super-messages $\hat{X}_1, \ldots, \hat{X}_r$ in such a way that the user can privately download the super-message $\hat{X}_{\sigma^{-1}(1)} = cX_W + Y^{[S,C]}$; and subsequently, subtracting off $Y^{[S,C]}$ from $\hat{X}_{\sigma^{-1}(1)}$, the user recovers X_W.

Remark 7. *Note that the proposed Multi-Server PIR-CSI–I protocol is a server-symmetric protocol since as explained in Step* 4 *of this protocol, it builds upon the Sun-Jafar protocol that enforces symmetry across servers [8].*

Example 7. Assume that there are $N = 2$ servers, $K = 9$ messages from $\mathbb{F}_{3^8}$ (i.e., each message has 8 symbols over $\mathbb{F}_3$), and $M = 3$. Suppose that the user demands the message X_1 and has a side information $X_2 + 2X_3 + X_4$. Note that for this example, $W = 1$, $S = \{2, 3, 4\}$, and $C = \{1, 2, 1\}$.

First, the user labels $r = \lceil \frac{K}{M+1} \rceil = 3$ sequences as I_1, I_2, I_3, each of length $M+1 = 4$. For creating these sequences, the user needs to have 12 indices, but at the beginning the user has 9 indices. For selecting the remaining 3 required indices, following the RP scheme of [30], the user selects $w \in \{0,1\}$, $s \in \{0,1,2,3\}$, and $t \in \{0,1,\ldots,5\}$ randomly chosen indices from $W = \{1\}$, $S = \{2,3,4\}$, and $T = \{5,6,7,8,9\}$, respectively, according to a carefully designed probability distribution (ensuring W-privacy of the RP scheme) on all (w,s,t) such that $w+s+t = 3$. For this example, the probability distribution is given by

$$p(w,s,t) \triangleq \begin{cases} \frac{14}{171}, & w=0, s=3, t=0 \\ \frac{60}{171}, & w=0, s=2, t=1 \\ \frac{36}{171}, & w=0, s=1, t=2 \\ \frac{4}{171}, & w=0, s=0, t=3 \\ \frac{21}{171}, & w=1, s=2, t=0 \\ \frac{30}{171}, & w=1, s=1, t=1 \\ \frac{6}{171}, & w=1, s=0, t=2 \end{cases}$$

Suppose that the user chooses $w=1, s=1, t=1$, and selects the 3 indices $\{1,2,5\}$. Following the RP protocol, the user forms the sequence $I_1 = \{W,S\} = \{1,2,3,4\}$. In the remaining 8 indices, there is one repetitive index, 5. For forming the other two sequences, I_2 and I_3, the user places the repetitive index 5 into both I_2 and I_3. Next, the user randomly partitions the remaining 6 indices, $\{1,2,6,7,8,9\}$, into I_2 and I_3. For this example, suppose that $I_2 = \{5,1,7,8\}$ and $I_3 = \{5,2,6,9\}$.

The user then labels $r = 3$ sequences as I_1', I_2', I_3', each of length 4. For this example, suppose that the user creates $I_1' = I_2' = I_3' = \{1,1,2,1\}$. Then, the user randomly reorders the elements of I_i and I_i', and constructs

$$\tilde{I}_1 = \{2,4,1,3\}, \quad \tilde{I}'_1 = \{1,1,1,2\}$$
$$\tilde{I}_2 = \{7,5,1,8\}, \quad \tilde{I}'_2 = \{2,1,1,1\}$$
$$\tilde{I}_3 = \{2,9,6,5\}, \quad \tilde{I}'_3 = \{1,1,2,1\}.$$

Next, the user sends a uniform random permutation of $\{I_1^*, I_2^*, I_3^*\}$, say $\{I_1^*, I_3^*, I_2^*\}$, to both servers, where $I_i^* = (\tilde{I}_i, \tilde{I}'_i)$.

The user and the servers then form three super-messages as follows:

$$\hat{X}_1 = X_2 + X_4 + X_1 + 2X_3$$
$$\hat{X}_2 = X_2 + X_9 + 2X_6 + X_5$$
$$\hat{X}_3 = 2X_7 + X_5 + X_1 + X_8.$$

Finally, the user and the servers run the Sun-Jafar protocol as follows for the three super-messages $\hat{X}_1, \hat{X}_2, \hat{X}_3$ in such a way that the user can privately download $\hat{X}_1$. For each $\hat{X}_i$, let $[\hat{X}_{i,1}, \ldots, \hat{X}_{i,8}]$ be an independent and uniform random permutation of the 8 symbols (over $\mathbb{F}_3$) of $\hat{X}_i$. The user requests 7 symbols from the first server and 7 symbols from the second server as listed in Table 3.4 [8], where the requested symbols are carefully designed linear combinations of symbols $\{\hat{X}_{i,j}\}_{i\in[3],j\in[8]}$. From the servers' answers, the

Table 3.4: The queries/answers of Sun-Jafar protocol for 2 servers and 3 messages $\hat{X}_1, \hat{X}_2, \hat{X}_3$, when the user demands $\hat{X}_1$.

Server 1	Server 2
$\hat{X}_{1,1}, \hat{X}_{2,1}, \hat{X}_{3,1}$	$\hat{X}_{1,2}, \hat{X}_{2,2}, \hat{X}_{3,2}$
$\hat{X}_{1,3} + \hat{X}_{2,2}$	$\hat{X}_{1,5} + \hat{X}_{2,1}$
$\hat{X}_{1,4} + \hat{X}_{3,2}$	$\hat{X}_{1,6} + \hat{X}_{3,1}$
$\hat{X}_{2,3} + \hat{X}_{3,3}$	$\hat{X}_{2,4} + \hat{X}_{3,4}$
$\hat{X}_{1,7} + \hat{X}_{2,4} + \hat{X}_{3,4}$	$\hat{X}_{1,8} + \hat{X}_{2,3} + \hat{X}_{3,3}$

user first obtains the super-message $\hat{X}_1 = X_2 + X_4 + X_1 + 2X_3$, and then recovers the desired message X_1 by subtracting off the side information $X_2 + 2X_3 + X_4$. For this example, the proposed protocol requires to download a total of 14 symbols (over $\mathbb{F}_3$), achieving the rate of $8/14 = 4/7$.

Lemma 15. *The Multi-Server PIR-CSI–I protocol satisfies the recoverability and W-privacy conditions, and achieves the rate* $(1 + 1/N + \cdots + 1/N^{\lceil \frac{K}{M+1} \rceil - 1})^{-1}$.

Proof. Since the messages $\mathbf{X}_1, \ldots, \mathbf{X}_K$ are uniformly and independently distributed over $\mathbb{F}_{q^m}$, and the messages $\hat{X}_1, \ldots, \hat{X}_r$ are linearly independent combinations of $X_1, \ldots, X_K$ over $\mathbb{F}_q$, then $\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_r$ are independently and uniformly distributed over $\mathbb{F}_{q^m}$, i.e., $H(\hat{\mathbf{X}}_i) = m \log_2 q = L$ for all $i \in [r]$. Thus, the proposed protocol achieves the same rate as the Sun-Jafar protocol for N servers and $\lceil K/(M+1) \rceil$ identically and independently distributed messages, i.e., $(1 + 1/N + \cdots + 1/N^{\lceil \frac{K}{M+1} \rceil - 1})^{-1}$ (see [8, Theorem 1]).

From the step 4 of the proposed protocol, it can be easily confirmed that the recoverability condition is satisfied. The proof of W-privacy is as follows. By the design of the protocol, all servers are fully aware of how the super-messages $\hat{X}_1, \ldots, \hat{X}_r$ have been formed. From the perspective of each server, according to the RP protocol, each super-message $\hat{X}_i$ has a certain probability to be the super-message needed by the user, i.e., the super-message from which the user can recover the demanded message. On the other hand, the Sun-Jafar protocol guarantees that given their query, no server can obtain any information about which super-message is being requested by the user. That is, given their query, from each server's perspective the probability of any super-message $\hat{X}_i$ to be the super-message needed by the user remains the same as that in the RP protocol. Moreover, the W-privacy of the RP protocol ensures that given their query, each server finds every message in $X_{\mathcal{K}}$ equally likely to be the user's demand. This proves the W-privacy of the proposed protocol. □

3.5.2 Multi-Server PIR-CSI-II

Theorem 8. *For the PIR-CSI–II problem with N servers, K messages, and side information size M, when $M = 2$ or $M = K$, the capacity of is 1, and when $3 \leq M \leq K-1$, the server-symmetric capacity is given by $\mathrm{C}_{W-II} = \frac{N}{N+1}$, that is*

$$\mathrm{C}_{W-II} = \begin{cases} 1, & M = 2, K, \\ \frac{N}{N+1}, & 3 \leq M \leq K-1. \end{cases}.$$

This result shows that for the two corner cases of $M = 2$ and $M = K$, the cost of retrieving one message privately is no more than that of downloading the message directly. For the cases of $3 \leq M \leq K-1$, full privacy can be achieved for only an additional download cost of L/N.

Proof of Theorem 8

3.5.2.1 Converse

Without loss of generality, suppose the user wishes to retrieve X_W for a given $W \in [K]$, and has a side information $Y \triangleq Y^{[S,C]}$ for given $S \in \mathcal{S}, C \in \mathcal{C}$ such that $W \in S$. We need to show that the maximum total entropy of the answers from all servers, denoted by D, is lower bounded by L when $M = 2$ or $M = K$, and is lower bounded by $(1 + 1/N)L$ when $3 \leq M \leq K-1$.

The proof proceeds as follows:

$$\begin{aligned} D &\geq H(\mathbf{A}^{[W,S,C]}|\mathbf{Q}^{[W,S,C]}, \mathbf{Y}) \\ &= H(\mathbf{A}^{[W,S,C]}, \mathbf{X}_W|\mathbf{Q}^{[W,S,C]}, \mathbf{Y}) \qquad (3.14) \\ &= L + H(\mathbf{A}^{[W,S,C]}|\mathbf{Q}^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \qquad (3.15) \end{aligned}$$

where (3.14) holds because of the recoverability condition, and (3.15) holds because $\mathbf{X}_W$ is independent of $(\mathbf{Q}^{[W,S,C]}, \mathbf{Y})$. By the non-negativity of the entropy, (3.15) yields $D \geq L$, which completes the proof for the cases of $M = 2$ and $M = K$. For the cases of $3 \leq M \leq K-1$, we continue lower bounding (3.15) as follows:

$$\begin{aligned} D &\geq L + H(\mathbf{A}^{[W,S,C]}|\mathbf{Q}^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \\ &\geq L + H(\mathbf{A}_1^{[W,S,C]}|\mathbf{Q}^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \\ &= L + H(\mathbf{A}_1^{[W,S,C]}|\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) \end{aligned} \tag{3.16}$$

where (3.16) holds because given $(\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y})$, $\mathbf{A}_1^{[W,S,C]}$ is conditionally independent of $\mathbf{Q}_n^{[W,S,C]}$ for all $n \neq 1$. In order to continue lower bounding (3.16), we choose an arbitrary message, say X_{W_1}, such that $W_1 \in S \setminus W$. According to Lemma 12, there exist $S_1 \in \mathcal{S}$, $C_1 \in \mathcal{C}$ with $W_1 \in S_1$, and accordingly $Y_1 \triangleq Y^{[S_1,C_1]}$, such that

$$H(\mathbf{A}_1^{[W,S,C]}|\mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) = H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}).$$

Rewriting (3.16),

$$\begin{aligned} D &\geq L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\ &\geq L + H(\mathbf{A}_1^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}). \end{aligned}$$

Similarly, by the server-symmetry assumption, we can write

$$D \geq L + H(\mathbf{A}_n^{[W_1,S_1,C_1]}|\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y})$$

for all $n \in [N]$. Combining all of these inequalities, we get

$$D \geq L + \frac{1}{N} H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}). \tag{3.17}$$

To further lower bound $H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y})$, we can write

$$\begin{aligned}
&H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\
&\geq H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) \\
&= H(\mathbf{A}^{[W_1,S_1,C_1]}, \mathbf{X}_{W_1} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) \qquad (3.18)
\end{aligned}$$

where (3.18) holds as X_{W_1} is recoverable from $A^{[W_1,S_1,C_1]}, Q^{[W_1,S_1,C_1]}, Y_1, W_1, S_1, C_1$.

We consider two cases separately: (i) $\mathbf{X}_{W_1}$ is independent of $(\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1)$, and (ii) $\mathbf{X}_{W_1}$ and $(\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1)$ are not independent.

In the case (i), as mentioned before $\mathbf{X}_{W_1}$ and $(\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1)$ are independent. This means that the following holds

$$H(\mathbf{X}_{W_1} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) = H(\mathbf{X}_{W_1}) = L.$$

Then, we can continue lower bounding (3.18) as follows:

$$\begin{aligned}
&H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}) \\
&\geq H(\mathbf{A}^{[W_1,S_1,C_1]}, \mathbf{X}_{W_1} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) \\
&= H(\mathbf{X}_{W_1} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1) \\
&\quad + H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \\
&= L + H(\mathbf{A}^{[W_1,S_1,C_1]} | \mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \\
&\geq L. \qquad (3.19)
\end{aligned}$$

By (3.17) and (3.19), $D \geq L + L/N$, as was to be shown.

In the case (ii), due to the dependence of $\mathbf{X}_{W_1}$ and $(\mathbf{Q}^{[W_1,S_1,C_1]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_1)$ and the linearity of X_W, Y, X_{W_1}, Y_1, it must hold that $Y = c_W X_W + c_{W_1} X_{W_1} + Z$ and $Y_1 = c'_W X_W + c'_{W_1} X_{W_1} + c' Z$ for some $c'_W, c'_{W_1}, c' \in \mathbb{F}_q^\times$, where $Z = \sum_{i \in S \setminus \{W, W_1\}} c_i X_i$, and c_i's are the elements in the sequence C (i.e., the coefficients of the messages in the side information Y). We proceed by lower bounding (3.16), when W, S, C, Y are replaced by W_1, S_1, C_1, Y_1. To this end, we choose an arbitrary message, say X_{W_2}, such that $W_2 \notin S$. Based on Lemma 12, there exist $S_2 \in \mathcal{S}$, $C_2 \in \mathcal{C}$ with $W_2 \in S_2$, and accordingly $Y_2 \triangleq Y^{[S_2,C_2]}$, such that

$$H(\mathbf{A}_1^{[W,S,C]} | \mathbf{Q}_1^{[W,S,C]}, \mathbf{X}_W, \mathbf{Y}) = H(\mathbf{A}_1^{[W_2,S_2,C_2]} | \mathbf{Q}_1^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}).$$

We consider two cases: (ii.1) $\mathbf{X}_{W_2}$ is independent of $(\mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_2)$, and (ii.2) $\mathbf{X}_{W_2}$ depends on $(\mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_2)$. In the case (ii.1), the proof follows the exact same line as in the proof of case (i), and hence not repeated.

In the case (ii.2), X_{W_2} must be recoverable from $Q^{[W_2,S_2,C_2]}, X_W, Y, Y_2$ since $\mathbf{X}_{W_2}$ depends on $(\mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_W, \mathbf{Y}, \mathbf{Y}_2)$.Thus, $Y_2 = c''_{W_2} X_{W_2} + c''(c_{W_1} X_{W_1} + Z)$ for some $c''_{W_2}, c'' \in \mathbb{F}_q^\times$. It is also easy to verify that X_{W_2} is not recoverable from X_{W_1}, Y_1, Y_2, and $\mathbf{X}_{W_2}$ is independent of $(\mathbf{X}_{W_1}, \mathbf{Y}_1, \mathbf{Y}_2)$. On the other hand, we have

$$\begin{aligned} D &\geq L + H(\mathbf{A}_1^{[W_1,S_1,C_1]} | \mathbf{Q}_1^{[W_1,S_1,C_1]}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \qquad (3.20) \\ &= L + H(\mathbf{A}_1^{[W_2,S_2,C_2]} | \mathbf{Q}_1^{[W_2,S_2,C_2]}, \mathbf{X}_{W_1}, \mathbf{Y}_1). \end{aligned}$$

where (3.20) follows from (3.16) that holds for W_1, S_1, C_1 (and Y_1), because D is defined as the total entropy of answers from all servers over all $W' \in [K], S' \in \mathcal{S}, C' \in \mathcal{C}$ such that

$W' \in S'$. Similarly as before, by the server-symmetry assumption it can also be shown that

$$\begin{aligned} D &\geq L + \frac{1}{N} H(\mathbf{A}^{[W_2,S_2,C_2]} | \mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_{W_1}, \mathbf{Y}_1) \\ &\geq L + \frac{1}{N} H(\mathbf{A}^{[W_2,S_2,C_2]} | \mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_{W_1}, \mathbf{Y}_1, \mathbf{Y}_2). \end{aligned} \tag{3.21}$$

As $\mathbf{X}_{W_2}$ is independent of $(\mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_{W_1}, \mathbf{Y}_1, \mathbf{Y}_2)$, and X_{W_2} is recoverable from $A^{[W_2,S_2,C_2]}$, $Q^{[W_2,S_2,C_2]}$, and Y_2, a simple application of the chain rule of entropy yields

$$H(\mathbf{A}^{[W_2,S_2,C_2]} | \mathbf{Q}^{[W_2,S_2,C_2]}, \mathbf{X}_{W_1}, \mathbf{Y}_1, \mathbf{Y}_2) \geq L. \tag{3.22}$$

By (3.21) and (3.22), $D \geq L + L/N$, as was to be shown.

3.5.2.2 Achievability

In this section, we propose a server-symmetric PIR-CSI–II protocol for each $2 \leq M \leq K-1$ that achieves a rate equal to C_{W-II} for the corresponding M.

For $3 \leq M \leq K-1$, we assume that each message consists of N^2 symbols over $\mathbb{F}_q$. For $M = 2$ and $M = K$, each message can be as short as one $\mathbb{F}_q$-symbol.

Multi-Server PIR-CSI–II Protocols:

Case of $M = 2$: The user randomly selects one of the two indices, say i, in S as follows: $i = W$ with probability $1/K$, and $i = S \setminus W$ with probability $(K-1)/K$. Then, the user requests the message X_i from a randomly chosen server.

Case of $3 \leq M \leq K-1$: The proposed scheme for this case consists of four steps. In the first step, given W, S, C (and $Y^{[S,C]}$), the user utilizes the scheme of [30] for single-server PIR-CSI (under Model II), which we refer to as *Modified Randomized Partitioning (MRP)*, to construct two sequences I_1, I_2 of indices in $[K]$, each of length $M-1$, and two sequences I_1', I_2' of elements in $\mathbb{F}_q^{\times}$, each of length $M-1$. (For details, see [30, Section V-B].) Next, the user and the servers follow the steps 2-4 of the Multi-Server PIR-CSI–I

protocol.

Case of $M = K$: Assume, w.l.o.g., that $W = 1$. The user randomly chooses an element c_1' from $\mathbb{F}_q^\times \setminus \{c_1\}$, where c_1 is the coefficient of X_1 in the side information $Y^{[S,C]}$. Then, the user requests the linear combination $c_1'X_1 + c_2X_2 + \cdots + c_KX_K$ from a randomly chosen server, where c_i is the coefficient of X_i in the side information $Y^{[S,C]}$.

Example 8. (Case of $3 \leq M \leq \frac{K}{2} + 1$) Assume that there are $N = 2$ servers, $K = 10$ messages from $\mathbb{F}_{3^4}$ (i.e., each message has 4 symbols over $\mathbb{F}_3$), and $M = 4$. Suppose that the user demands the message X_1 and has a coded side information $X_1 + X_2 + 2X_3 + X_4$. Note that, for this example, $W = 1$, $S = \{1, 2, 3, 4\}$, and $C = \{1, 1, 2, 1\}$.

First, the user labels 2 sequences as I_1, I_2, each of length $M - 1 = 3$. For creating these sequences, the user selects $w \in \{0, 1\}$ and $t \in \{2, 3\}$ randomly chosen indices from $W = \{1\}$ and $T = \{5, 6, 7, 8, 9, 10\}$, respectively, according to a carefully designed probability distribution (ensuring W-privacy of the MRP scheme) on all (w, t) such that $w + t = 3$. For this example, the probability distribution is given by

$$p(w,t) \triangleq \begin{cases} 0.4, & w = 0, t = 3 \\ 0.6, & w = 1, t = 2 \end{cases}$$

Suppose that the user chooses $w = 1, t = 2$, and selects the 3 indices $\{1, 6, 10\}$. Following the MRP protocol, the user forms the sequence $I_1 = S \setminus W = \{2, 3, 4\}$ and $I_2 = \{1, 6, 10\}$.

The user then labels 2 sequences as I_1', I_2', each of length 3. For this example, suppose that the user creates $I_1' = \{1, 2, 1\}, I_2' = \{1, 1, 1\}$. Then, the user randomly reorders the elements of I_i and I_i', and constructs

$$\tilde{I}_1 = \{3, 2, 4\}, \quad \tilde{I}_1' = \{2, 1, 1\}$$
$$\tilde{I}_2 = \{1, 10, 6\}, \quad \tilde{I}_2' = \{1, 1, 1\}.$$

Table 3.5: The queries/answers of Sun-Jafar protocol for 2 servers and 2 messages $\hat{X}_1, \hat{X}_2$, when the user demands $\hat{X}_2$.

Server 1	Server 2
$\hat{X}_{1,1}$	$\hat{X}_{1,2}$
$\hat{X}_{2,1}$	$\hat{X}_{2,2}$
$\hat{X}_{2,3} + \hat{X}_{1,2}$	$\hat{X}_{2,4} + \hat{X}_{1,1}$

Next, the user sends a uniform random permutation of $\{I_1^*, I_2^*\}$, say $\{I_2^*, I_1^*\}$, to both servers, where $I_i^* = (\tilde{I}_i, \tilde{I}_i')$. The user and the servers form two super-messages as follows:

$$\hat{X}_1 = X_1 + X_{10} + X_6$$

$$\hat{X}_2 = 2X_3 + X_2 + X_4.$$

Finally, the user and the servers run the Sun-Jafar protocol for the two super-messages $\hat{X}_1, \hat{X}_2$ such that the user can privately download $\hat{X}_2$. For each $\hat{X}_i$, let $[\hat{X}_{i,1}, \ldots, \hat{X}_{i,4}]$ be an independent and uniform random permutation of the 4 symbols (over $\mathbb{F}_3$) of $\hat{X}_i$. The user requests 3 symbols from the first server and 3 symbols from the second server as listed in Table 3.5 [8], where the requested symbols are carefully designed linear combinations of symbols $\{\hat{X}_{i,j}\}_{i\in[2],j\in[4]}$. From the servers' answers, the user first obtains the super-message $\hat{X}_2 = 2X_3+X_2+X_4$, and then recovers the desired message X_1 by subtracting off $\hat{X}_2$ from the side information $X_1+X_2+2X_3+X_4$. For this example, the proposed protocol requires to download a total of 6 symbols (over $\mathbb{F}_3$), achieving the rate of $4/6 = 2/3$.

Example 9. (Case of $\frac{K}{2} \leq M \leq K-1$) Assume that there are $N = 2$ servers, $K = 5$ messages from $\mathbb{F}_{3^4}$ (i.e., each message has 4 symbols over $\mathbb{F}_3$), and $M = 4$. Suppose that the user demands the message X_1 and has coded a side information $X_1 + X_2 + 2X_3 + X_4$. Note that, for this example, $W = 1$, $S = \{1, 2, 3, 4\}$, and $C = \{1, 1, 2, 1\}$.

First, the user labels 2 sequences as I_1, I_2, each of length $M = 4$. For creating these

sequences, the user selects $w \in \{0,1\}$ and $t \in \{2,3\}$ randomly chosen indices from $W = \{1\}$ and $T = \{2,3,4\}$, respectively, according to a carefully designed probability distribution (ensuring W-privacy of the MRP scheme) on all (w,t) such that $w+t=3$. For this example, the probability distribution is given by

$$p(w,t) \triangleq \begin{cases} 0.4, & w=0, t=3 \\ 0.6, & w=1, t=2 \end{cases}$$

Suppose that the user chooses $w=1, t=2$, and selects the 3 indices $\{1,2,4\}$. Following the MRP protocol, the user forms the sequence $I_1 = S = \{1,2,3,4\}$ and $I_2 = \{5,1,2,4\}$.

The user then labels 2 sequences as I_1', I_2', each of length 4. For this example, suppose that the user creates $I_1' = \{2,1,2,1\}, I_2' = \{1,2,1,1\}$. Then, the user randomly reorders the elements of I_i and I_i', and constructs

$$\tilde{I}_1 = \{1,4,2,3\}, \quad \tilde{I}_1' = \{2,1,1,2\}$$
$$\tilde{I}_2 = \{1,5,2,4\}, \quad \tilde{I}_2' = \{2,1,1,1\}.$$

Next, the user sends a uniform random permutation of $\{I_1^*, I_2^*\}$, say $\{I_2^*, I_1^*\}$, to both servers, where $I_i^* = (\tilde{I}_i, \tilde{I}_i')$. The user and the servers form two super-messages as follows:

$$\hat{X}_1 = 2X_1 + X_5 + X_2 + X_4$$
$$\hat{X}_2 = 2X_1 + X_4 + X_2 + 2X_3.$$

Finally, the user and the servers run the Sun-Jafar protocol as explained in the previous example for the two super-messages $\hat{X}_1, \hat{X}_2$ in such a way that the user can privately download $\hat{X}_2$. The user requests 3 symbols from the first server and 3 symbols from the

second server as listed in Table 3.5 [8]. From the servers' answers, the user first obtains the super-message $\hat{X}_2 = 2X_1 + X_4 + X_2 + 2X_3$, and then recovers the desired message X_1 by subtracting off the side information $X_1 + X_2 + 2X_3 + X_4$ from $\hat{X}_2$. For this example, the proposed protocol requires to download a total of 6 symbols (over $\mathbb{F}_3$), achieving the rate of $4/6 = 2/3$.

Lemma 16. *The Multi-Server PIR-CSI–II protocols for the cases of $M = 2$, $3 \leq M \leq K - 1$, and $M = K$ are server-symmetric protocols that satisfy the recoverability and the W-privacy conditions, and achieve the rates 1, $N/(N + 1)$, and 1, respectively.*

Proof. The proof is similar to the proof of Lemma 15, and omitted to avoid repetition. □

4. PRIVATE LINEAR TRANSFORMATION*

4.1 Introduction

This work studies the Private Linear Transformation (PLT) problem, recently introduced in [49, 50], in which an identical copy of a database consisting of K independent messages are stored over N servers. There is a user who wishes to compute L independent linear combinations of a subset of D messages in the database, without revealing any information to the servers about the identities of the D messages required for the computation, while downloading the minimum possible amount of information from the servers.

The PLT problem can be viewed as an interesting extension of the Private Information Retrieval (PIR) (see e.g., [8, 10–12, 19–21, 30–37, 86–88]) and Private Linear Computation (PLC) (see e.g., [82, 85, 89, 90]) problems, which have been extensively studied in the literature. To be more specific, for $L = D$, the PLT problem reduces to the multi-message PIR problem in which the goal is to privately retrieve a subset of D messages in the database. Moreover, for $L = 1$, the PLT problem reduces to the PLC problem in which the goal is to privately compute one linear combination of a D-subset of messages. The PLT problem can be motivated by several practical scenarios such as linear transformation technique applied for dimensionality reduction in Machine Learning (ML) applications (see [50]).

Related Work: In the classical PIR problem, a user wants to privately download a message from N replicated non-colluding servers. The capacity of the information-theoretic PIR was derived in [8]. Then, the PIR problem has been extended in various directions, such as coded PIR (see e.g., [12, 19, 20]), multi-message PIR (see e.g., [10, 11, 21, 32]), and PIR with side information (see e.g., [30–37]).

*Reprinted with permission from [38] "Multi-Server Private Linear Transformation with Joint Privacy," by F. Kazemi and A. Sprintson, 2021. In Proceedings of 2021 XVII International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY), pp. 182-187, Oct 2021. Copyright © by IEEE.

The problem of Private Computation (PC), initially introduced in [85], is an interesting generalization of the PIR problem, in which the user wishes to compute one arbitrary linear combination of the messages in the database, while revealing no information about the identities and the coefficients of these messages to any server. Several variants of the PC problem were also studied in [82, 89–96]. In [90], a variation of the PC problem was considered in which it is only required to protect the identities of the messages in the demanded linear combination, while the coefficients used to construct the linear combination do not need to be hidden from the server.

The most related to this work is the PLT problem, recently introduced in [49,50], which is also closely related to the PIR and PLC problems. Indeed, a naive protocol for the PLT problem is to privately retrieve all the D messages required for the computation using a multi-message PIR scheme, and then compute the required linear combinations. Another simple approach for the PLT problem is to compute each required linear combination separately using a PLC protocol.

Although there is a significant body of literature on the PIR and PLC problems, there are only a few studies on the PLT problem. In particular, the PLT problem was studied in the single-server setting by considering the following two privacy requirements: (i) the individual privacy, where the identity of each individual message in the support set of the demanded linear combinations needs to be kept private [49]; and (ii) the joint privacy, in which the identity of the entire set of messages in the support set of the demanded linear combinations must be kept private [50]. All variants of the PIR and PC problems, can also be considered for the PLT that opens several interesting directions for future work.

In [50], Heidarzadeh et al. recently proved that the capacity of the PLT with a single server and joint privacy is $L/(K-D+L)$. However, the capacity of the PLT in the multi-server scenario was left open in [50]. Remarkably, neither a general achievability scheme nor a converse was known in this case. This work is motivated by this open problem.

Our Contributions

In this chapter, we consider the multi-server setting of the PLT problem with an arbitrary number of servers $N \geq 1$. We focus on the setting in which the coefficient matrix of the required linear combinations generates a Maximum Distance Separable (MDS) code. This setting can be motivated by several practical scenarios. For instance, the user may have chosen the the coefficient matrix randomly over the field of real numbers or a finite field of large size [50]. The first contribution of this work is to show that the capacity of PLT problem for the case of $L = 1$, i.e., when the user wishes to compute one linear combination of D messages, is equal to $\Phi(1/N, K - D + 1)$, where $\Phi(A, B) = (1 + A + A^2 + \cdots + A^{B-1})^{-1}$. This result establishes the capacity of the PLC problem for an arbitrary number of servers N, thus settling the open problem mentioned above for the case of $L = 1$. Moreover, we establish an upper bound on the capacity of PLT problem for any arbitrary parameters $N, K, D, L \geq 1$, and based on some known capacity results, we show the tightness of the provided upper bound for some special cases of the problem: (i) the case where there is a single server (i.e., $N = 1$), (ii) the case where $L = 1$, and (iii) the case where $L = D$.

4.2 Problem Formulation

4.2.1 Basic Notation

Throughout this chapter, we denote random variables by bold letters and their realizations by regular letters. The functions $\mathbb{P}(\cdot)$, $\mathbb{P}(\cdot|\cdot)$, $\mathbb{H}(\cdot)$, $\mathbb{H}(\cdot|\cdot)$, and $I(\cdot;\cdot|\cdot)$ denote probability, conditional probability, entropy, conditional entropy, and conditional mutual information, respectively. Let $\mathbb{Z}_{\geq 0}$ and $\mathbb{N}$ denote the set of non-negative integers and the set of positive integers, respectively. For any $i \in \mathbb{N}$, let $[i] \triangleq \{1, \ldots, i\}$. Let $\mathbb{F}_q$ be a finite field for some prime q, $\mathbb{F}_q^{\times} \triangleq \mathbb{F}_q \setminus \{0\}$ be the multiplicative group of $\mathbb{F}_q$, and $\mathbb{F}_q^S$ be the S-dimensional vector space over $\mathbb{F}_q$ for some integer $S \geq 1$. Let $B \triangleq S \log_2 q$. Let

$K, D, L \geq 1$ be integers such that $L \leq D \leq K$. Let $\mathcal{K} \triangleq [K]$. Let $\mathbb{W}$ denote the set of all D-subsets (i.e., subsets of size D) $\mathcal{W}$ of $\mathcal{K}$, and $\mathbb{V}$ denote the set of all MDS matrices V of dimension $L \times D$ with entries in $\mathbb{F}_q$ (i.e., every $L \times L$ submatrix of V is full-rank). We denote the cardinality of a set $\mathcal{S}$ by $|\mathcal{S}|$. For a positive real number A and a positive integer number B, let $\Phi(A, B) = (1 + A + A^2 + \cdots + A^{B-1})^{-1}$.

4.2.2 Setup and Assumptions

Consider N non-colluding servers, each stores an identical copy of a database consisting of K messages, $X_{\mathcal{K}} = \{X_1, \ldots, X_K\}$, where each message X_i is a row vector of length S. Let $X \triangleq [X_1^\top, \cdots, X_K^\top]^\top$ be a matrix of dimension $K \times S$. For some $\mathcal{R} \triangleq \{i_1, \ldots, i_r\} \subset \mathcal{K}$, let $X_{\mathcal{R}}$ be the submatrix of X of size $|\mathcal{R}| \times S$, restricted to its rows indexed by the set $\mathcal{R}$, i.e., $X_{\mathcal{R}} = [X_{i_1}^\top, \cdots, X_{i_r}^\top]^\top$.

Suppose that there is a user who wishes to compute L linear combinations of D messages $\{X_i : i \in \mathcal{W}\}$, as $V_1 X_{\mathcal{W}}, \cdots, V_L X_{\mathcal{W}}$, where $\mathcal{W} \in \mathbb{W}$ is the index set of the D messages required for the computation, and V_ℓ, $\ell \in [L]$, denoting the coefficient vector of the ℓth desired linear combination, is the ℓth row of an $L \times D$ MDS matrix V with entries in $\mathbb{F}_q$, i.e., $V = [V_1^\top, \cdots, V_L^\top]^\top$, $V \in \mathbb{V}$. In other words, the user wants to compute the $L \times S$ matrix $Z^{[\mathcal{W},V]} \triangleq V X_{\mathcal{W}}$ whose rows are the L required linear combinations. We refer to $Z^{[\mathcal{W},V]}$ as the *demand*, $\mathcal{W}$ as the *demand's index set*, V as the *demand's coefficient matrix*, L as the *demand's dimension*, and D as the *demand's support size*.

We assume that $\mathbf{X}_1, \cdots, \mathbf{X}_K$ are independently and uniformly distributed over $\mathbb{F}_q^S$, that is, $H(\mathbf{X}_i) = B$ for $i \in \mathcal{K}$. Thus, $H(\mathbf{X}) = KB$, $H(\mathbf{X}_{\mathcal{R}}) = |\mathcal{R}|B$ for every $\mathcal{R} \subset \mathcal{K}$, and $H(\mathbf{Z}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]}) = LB$. We also assume that $\boldsymbol{\mathcal{W}}$, $\mathbf{V}$, and $\mathbf{X}$ are independent random variables such that $\boldsymbol{\mathcal{W}}$ and $\mathbf{V}$ are uniformly distributed over $\mathbb{W}$ and $\mathbb{V}$, respectively. Moreover, we assume that the servers initially know the distributions of $\boldsymbol{\mathcal{W}}$ and $\mathbf{V}$, whereas the servers have no information about the realizations $\mathcal{W}$ and V in advance.

4.2.3 Privacy and Recoverability Conditions

To retrieve the demand $Z^{[\mathcal{W},V]}$ for any given $\mathcal{W}$ and V, user generates N queries $\{Q_n^{[\mathcal{W},V]}\}_{n\in[N]}$, and sends the query $Q_n^{[\mathcal{W},V]}$ to the n-th server. Note that server n just receives $Q_n^{[\mathcal{W},V]}$ without having any access to other queries (non-colluding servers assumption). Each query $Q_n^{[\mathcal{W},V]}$ is a (potentially stochastic) function of $\mathcal{W}$ and V. For clarity, we denote $Q^{[\mathcal{W},V]} \triangleq \{Q_n^{[\mathcal{W},V]}\}_{n\in[N]}$ and $\mathbf{Q}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]} \triangleq \{\mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]}\}_{n\in[N]}$.

Once the n-th server receives the query $Q_n^{[\mathcal{W},V]}$, it responds back to the user with an answer $A_n^{[\mathcal{W},V]}$. The answer $A_n^{[\mathcal{W},V]}$ is a (deterministic) function of the query $Q_n^{[\mathcal{W},V]}$ and X, i.e., $H(\mathbf{A}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]}|\mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]},\mathbf{X}) = 0$. For clarity, we denote $A^{[\mathcal{W},V]} \triangleq \{A_n^{[\mathcal{W},V]}\}_{n\in[N]}$ and $\mathbf{A}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]} \triangleq \{\mathbf{A}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]}\}_{n\in[N]}$.

Recoverability Condition: The answers $A^{[\mathcal{W},V]}$ from all the servers along with the queries $Q^{[\mathcal{W},V]}$, and the realizations $\mathcal{W}, V$ must enable the user to retrieve the demand $Z^{[W,V]}$. This condition is referred to as the *recoverability condition*, as formally stated in the following

$$H(\mathbf{Z}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]}|\mathbf{A}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]},\mathbf{Q}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]},\boldsymbol{\mathcal{W}},\mathbf{V}) = 0,$$

Privacy Condition: The queries $Q^{[\mathcal{W},V]}$ should be designed such that the servers infer no information about the user's demand index set $\mathcal{W}$. This condition is referred to as the *joint privacy condition*, formally stated as follows

$$I(\boldsymbol{\mathcal{W}};\mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]},\mathbf{A}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]},\mathbf{X}_{\mathcal{K}}) = 0 \quad \forall n \in [N].$$

Equivalently, from the perspective of each server, every D-subset of indices $\mathcal{K}$ must be equally likely to be the demand's index set, i.e., for any given $\tilde{\mathcal{W}} \in \mathbb{W}$, it must hold that

$$\mathbb{P}(\boldsymbol{\mathcal{W}} = \tilde{\mathcal{W}}|\mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]} = Q_n^{[\mathcal{W},V]}) = \mathbb{P}(\boldsymbol{\mathcal{W}} = \tilde{\mathcal{W}}) \quad \forall n \in [N].$$

4.2.4 Problem Statement

The problem is to design a protocol for generating queries $\{Q_n^{[\mathcal{W},V]}\}_{n\in[N]}$ and their corresponding answers $\{A_n^{[\mathcal{W},V]}\}_{n\in[N]}$ (for any given $\mathcal{W}$ and V) such that both the privacy and recoverability conditions are satisfied. We refer to this problem as *Private Linear Transformation (PLT)*. A protocol for generating queries/answers for PLT is referred to as a *PLT protocol*.

The *rate* of a PLT protocol is defined as the ratio of the entropy of demand , i.e., $H(\mathbf{Z}^{[\boldsymbol{\mathcal{W}},\mathbf{V}]}) = LB$, to the total entropy of answers from the servers, i.e., $\Sigma_{n=1}^{N} H(\mathbf{A}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]})$. The *capacity* of the PLT problem, denoted by $C^{PLT}(N, K, L, D)$, is defined as the supremum of rates over all PLT protocols, i.e.,

$$C^{PLT}(N, K, L, D) \triangleq \sup \frac{LB}{\Sigma_{n=1}^{N} H(\mathbf{A}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]})}$$

In this work, our goal is to characterize (or derive non-trivial bounds on) the capacity of the PLT problem, i.e., $C^{PLT}(N, K, L, D)$, and to design a PLT protocol that is capacity-achieving.

4.3 Main Results

In this section, we present our main results. Theorem 9 establishes an upper bound on the capacity of the PLT problem for all parameters $N, K, L, D \geq 1$. Leveraging some known capacity results, we show that the presented upper bound is tight in the following regimes: (i) the case where there is a single server (i.e., $N = 1$), (ii) the case where $L = 1$, and (iii) the case where $L = D$. Theorem 10 characterizes the capacity of the PLT problem for all parameters $N, K, D \geq 1$ and $L = 1$, i.e., the case where the user wishes to privately compute *one* linear combination of D messages in the database. The proofs of theorems 9 and 10 are given in sections 4.4 and 4.5, respectively.

Theorem 9. *The capacity of PLT problem with N non-colluding and replicated servers, K messages, demand's support size D, and demand's dimension L,*
(i) if $\frac{K-D}{L} \leq 1$, is upper bounded by

$$C^{PLT}(N,K,L,D) \leq \left(1+\frac{K-D}{LN}\right)^{-1},$$

(ii) and if $\frac{K-D}{L} \geq 1$, is upper bounded by

$$C^{PLT}(N,K,L,D) \leq \left(\frac{1-\left(\frac{1}{N}\right)^{\lfloor\theta\rfloor}}{1-\frac{1}{N}}+\frac{(\theta-\lfloor\theta\rfloor)}{N^{\lfloor\theta\rfloor}}\right)^{-1}.$$

where $\theta \triangleq \frac{K-D+L}{L}$.

The converse proof is provided in Section 4.4, which is based on a reduction argument and leverages the capacity result for multi-message PIR with private side information problem, introduced in [88].

Corollary 1. *If $\frac{K-D}{L} \in \mathbb{Z}_{\geq 0}$, the capacity upper bounds provided in Theorem 9, can be written as*

$$C^{PLT}(N,K,L,D) \leq \left(1+\frac{1}{N}+\cdots+\frac{1}{N^{\frac{K-D}{L}}}\right)^{-1} = \Phi(\frac{1}{N},\frac{K-D+L}{L}).$$

Remark 8. The capacity upper bounds in Theorem 9 are tight for the case when $N=1$ (i.e., when there is a single server), which is equal to $L/(K-D+L)$ as was shown in [50, Theorem 2]. Moreover, in Theorem 10, we prove the tightness of this upper bound for the case of $L=1$.

Remark 9. Notably, for the case of $L=D$, where the user wishes to privately compute D independent linear combinations of D-subset of messages in the database (that is

equivalent to privately retrieving these D messages), the capacity upper bound in Theorem 9, i.e., (i) $(1+(K-D)/DN)^{-1}$ if $K/D \leq 2$, and (ii) $\Phi(1/N, K/D)$ if $K/D \geq 2$ and $K/D \in \mathbb{N}$, is tight as was shown in [11]. Note that in this case, an optimal capacity-achieving multi-message PIR protocol proposed in [11, Theorems 1, 2] is an optimal protocol that achieves the capacity upper bound in Theorem 9.

Theorem 10. *The capacity of the PLT problem with N non-colluding and replicated servers, K messages, demand's support size D, and demand's dimension $L = 1$, is given by*

$$C^{PLT}(N, K, 1, D) = \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-D}}\right)^{-1} = \Phi\left(\frac{1}{N}, K - D + 1\right).$$

The converse proof follows directly from the result of Theorem 9 for $L = 1$. Also, an alternative proof of converse, similar to that of Theorem 9, is provided in Section 4.5. For the achievability proof, we design a PLT protocol that achieves the proposed upper bound provided by converse, and is inspired by both our recently proposed scheme of [34] for the single-server PIR with private coded side information problem, and the scheme proposed in [85] for the private computation problem.

Remark 10. The result of Theorem 10 generalizes the previous finding reported in [50] for the PLT problem with a single server, without any prior side information, when joint privacy is required, and $L = 1$. As was shown in [50], the capacity of this setting is equal to $K - D + 1$, which is consistent with the result of Theorem 10 for $N = 1$. Also, evidently it can be observed that for the case of $D = 1$, the result of Theorem 10 reduces to the known capacity result of [8] for the classical PIR problem where the user wants to privately download one message in the database, which is $\Phi\left(1/N, K\right)$.

Remark 11. It is worthwhile to compare the result of Theorem 10 with the capacity result of [85] for the related PC problem where the user wishes to compute one arbitrary lin-

ear combination of K independent messages in a database replicated at N non-colluding servers, while hiding both the identities and the coefficients of the messages participating in the demand. As was shown in [85], the capacity of this setting is equal to $\Phi\left(1/N, K\right)$. Unlike the privacy requirements in the private computation problem introduced in [85], in the PLT problem, the goal is to hide only the identities of the D messages participating in the user's demand and not necessarily the values of their coefficients, which based on the result of Theorem 10, it can be fulfilled more efficiently with much higher rate, i.e., $\Phi\left(1/N, K-D+1\right)$. This is interesting since this type of access privacy are motivated by many practical scenarios such as linear transformation technique used for dimensionality reduction in Machine Learning (ML) applications (see, e.g. [50, 97] and references therein). By comparing the capacity results of these two problems, one can readily conclude that the advantage of PLT protocols over the a repeated use of a PC protocol becomes more tangible when the demand's support size D increases.

Remark 12. It is noteworthy that for* $D \geq 2$, a trivial PLT protocol for $L = 1$ would be privately retrieving the D messages required for the linear computation using an optimal multi-message PIR scheme satisfying privacy of demand messages jointly, introduced in [11], and then computing the required linear combination. As was shown in [11, Theorems 1, 2], the optimal rate that can be achieved leveraging this approach, is upper bounded by $D^{-1} \leq 1/2$. The result of Theorem 10 indicates that the PLT problem in general can be addressed much more efficiently with the rate of $\Phi(1/N, K-D+1) \geq 1/2$.

Remark 13. Interestingly, in the PLT problem, a simple approach of computing each of the required linear combinations separately through applying an optimal PLT scheme introduced in Theorem 10, cannot achieve the capacity upper bound presented in Theorem 9 for all parameters N, K, L, D.

*Note that for the case of $D = 1$, the PLT problem reduces to the classical single-message PIR problem introduced in [8].

4.4 Proof of Theorem 9

Converse

The proof of converse follows from the capacity result for the problem of multi-message PIR with private side information, referred to as M-PIR-PSI, introduced in [88, Theorem 1]. In this problem, there is a database of K independent messages whose copies are replicated across N servers, and there is a user who has access to M messages from the database as side information. The user wishes to retrieve P messages from the database while leaking no information about the the identities of both the desired messages and the side information messages, to any individual server. As was shown in [88, Theorem 1], the capacity of this setting, denoted by $C^{MPIR-PSI}(N, K, P, M)$,
(i) if $\frac{K-M}{P} \leq 2$ is given by

$$C^{MPIR-PSI}(N, K, P, M) = \left(1 + \frac{K - M - P}{PN}\right)^{-1}, \tag{4.1}$$

(ii) if $\frac{K-M}{P} \geq 2$ is upper bounded by

$$C^{MPIR-PSI}(N, K, P, M) \leq \left(\frac{1 - \left(\frac{1}{N}\right)^{\lfloor \rho \rfloor}}{1 - \frac{1}{N}} + \frac{(\rho - \lfloor \rho \rfloor)}{N^{\lfloor \rho \rfloor}}\right)^{-1}, \tag{4.2}$$

where $\rho \triangleq \frac{K-M}{P}$. In case (ii), as was shown [88, Corollary 1], if $\frac{K-M}{P} \in \mathbb{N}$, the capacity is given by

$$C^{MPIR-PSI}(N, K, P, M) = \Phi(\frac{1}{N}, \frac{K - M}{P}). \tag{4.3}$$

In the following, we want to show that any PLT protocol designed for the problem with N servers, K messages, demand's support size D, and demand's dimension L, can be used as a protocol that satisfies both the recoverability and the privacy conditions of the M-PIR-PSI problem with demand size $P = L$ and side information size $M = D - L$.

Specifically, for a given instance of the M-PIR-PSI problem with the set of demand indices $\mathcal{P}$ of size L, (i.e., $P = L$), and the set of side information indices $\mathcal{S}$ of size $D - L$, (i.e., $M = D - L$), the user can construct a random $L \times D$ MDS matrix V and forms the set $\mathcal{W} = \mathcal{P} \cup \mathcal{S}$. Then, for the given $\mathcal{W}$ and V, the user and the servers can apply a PLT protocol for generating queries $Q^{[\mathcal{W},V]}$ and their corresponding answers $A^{[\mathcal{W},V]}$, such that the user can privately compute L MDS coded linear combinations of the D messages indexed by the set $\mathcal{W}$ (i.e., union of demands and side information messages). The user can then retrieve the L desired messages by subtracting off the contribution of the $D - L$ side information messages from the computed L linear combinations.

Now, we need to prove that the PLT-based protocol described above satisfies both the recoverability and the joint privacy conditions of the M-PIR-PSI problem. It should be noted that since the PLT protocol enables the user to compute L MDS coded linear combinations of D messages, based on the property of MDS codes[†], one can readily verify that the user can always retrieve the L desired messages by subtracting off the contribution of $D - L$ side information messages from the L computed linear equations, and solving the resulting system of L linear equations with L unknowns. Thus, the recoverability condition is satisfied.

It is easy to verify that by applying the PLT protocol, the identities of all the D messages (i.e., the union of the demand messages and side information messages) participating in the L linear combinations, will be jointly protected from each server as a result of the privacy guarantees of the PLT protocol. Indeed, from the perspective of each server, every D-subset of K messages is equally likely to be the union of the demand messages and side information messages. Moreover, due to the property of MDS codes, within each D-subset of messages, every subset of size L can be considered as the set of demand messages (i.e., the remaining $D - L$ as the set of side information messages) with equal probability.

[†]Every $L \times L$ submatrix of an $L \times D$ MDS matrix is invertible.

This ensures that the described PLT-based protocol satisfies the privacy condition in the M-PIR-PSI problem.

Thus, we conclude that any achievable rates in the PLT problem with N servers, K messages, demand's support size D, and demand's dimension L, would be also achievable (using the PLT-based protocol) in the M-PIR-PSI problem with N servers, K messages, demand size $P = L$, and side information size $M = D - L$. Thus, the capacity of PLT problem with parameters N, K, D, L, i.e., $C^{PLT}(N, K, L, D)$, is upper bounded by the capacity of the M-PIR-PSI problem with parameters $N, K, P = L, M = D - L$, i.e., $C^{MPIR-PSI}(N, K, L, D - L)$. Thus, substituting P with L, and M with $D - L$ in equations 4.1, 4.2 completes the proof. Also, in case (ii), if $\frac{K-M}{P} = \frac{K-D+L}{L} \in \mathbb{N}$ or equivalently $\frac{K-D}{L} \in \mathbb{Z}_{\geq 0}$, we have

$$C^{PLT}(N, K, L, D) \leq C^{MPIR-PSI}(N, K, L, D - L) = \Phi(\frac{1}{N}, \frac{K - D + L}{L}).$$

4.5 Proof of Theorem 10

We prove the converse by showing that the capacity for the case of $L = 1$, that is $C^{PLT}(N, K, 1, D)$, is upper bounded by the capacity of PIR with private side information problem, referred to as PIR-PSI, in which a database of K independent messages is replicated across N servers, and the user has access to M messages from the database as side information. The user wants to retrieve one message from the database while hiding jointly the identities of the desired message and the side information messages, from any individual server. This problem was introduced by Chen et al. [87]. As was shown in [87, Theorem 1], the capacity of PIR-PSI problem, denoted by $C^{PIR-PSI}(N, K, M)$, is equal to $\Phi(\frac{1}{N}, K - M)$.

Any PLT protocol designed for the problem with N servers, K messages, demand's support size D, and demand's dimension $L = 1$, enables the user to compute one linear

combination of a subset of D messages while hiding the identities of these messages from any server. So, based on a similar reasoning used in the converse proof of Theorem 9, one can easily confirm that such PLT protocol would also be a protocol satisfying the recoverability and the privacy conditions in the PIR-PSI problem with side information size $M = D - 1$. Thus, any achievable rate in the PLT problem with N servers, K messages, demand's support size D, and demand's dimension $L = 1$, can be also achieved for the PIR-PSI problem with N servers, K messages, and side information size $M = D - 1$. Thus, we have

$$C^{PLT}(N, K, 1, D) \leq C^{PIR-PSI}(N, K, D-1) = \Phi(\frac{1}{N}, K - D + 1).$$

Achievability

In this section, we complete the proof of Theorem 10 by designing a PLT protocol for the setting with N servers, K messages, demand's support size D, and demand's dimension $L = 1$, such that it achieves the upper bound provided by converse on the rate of any such PLT protocols, i.e., $\Phi(1/N, K - D + 1)$. The proposed protocol, referred to as the *Modified GRS Code*, leverages ideas from a modified version of the Specialized GRS Code Protocol proposed for the problem of single-server PIR with private coded side information in [34], as well as the PC scheme proposed for the PC problem in [85].

Modified GRS Code protocol: Assume $q \geq K$, and let each message consists of $S = N^{\binom{K}{D}}$ symbols from $\mathbb{F}_q$. Suppose the user wishes to privately compute one linear combination of D messages indexed by a set $\mathcal{W}$, as $V_1 X_{\mathcal{W}} = \sum_{i \in \mathcal{W}} v_i X_i$ where V_1 is a row vector of length D. This protocol consists of four steps as follows:

Step 1: By using the Modified Specialized GRS Code protocol proposed in [34], the user first constructs a polynomial $p(x) = \sum_{i=0}^{K-D} p_i x^i \triangleq \prod_{i \notin \mathcal{W}} (x - \omega_i)$ where $\omega_1, \ldots, \omega_K$ are K distinct arbitrarily chosen elements from $\mathbb{F}_q$. The user then constructs $r \triangleq K - D +$

1 vectors $Q_1, \ldots, Q_r$, each of length K, such that $Q_i = [\alpha_1 \omega_1^{i-1}, \ldots, \alpha_K \omega_K^{i-1}]$, $i \in [r]$, where $\alpha_j = \frac{v_j}{p(\omega_j)}$ for any $j \in \mathcal{W}$, and α_j is chosen randomly from $\mathbb{F}_q^\times$ for any $j \notin \mathcal{W}$.

Step 2: Let $\hat{X}_i \triangleq \sum_{j=1}^{K} \alpha_j \omega_j^{i-1} X_j$ for $i \in [r]$. We refer to $\hat{X}_i$ as a *super-message*. Note that the vector Q_i, constructed in Step 1, is the vector of coefficients of the messages $\{X_i\}_{i \in \mathcal{K}}$ in the super-message $\hat{X}_i$. Let $F \triangleq \binom{K}{D}$, and let $W_1, W_2, \ldots, W_F$ be the collection of all D-subsets of $\mathcal{K}$ in a lexicographical order. The structure of the Specialized GRS Code protocol [34] ensures that for each W_f, $f \in [F]$, there exist exactly $q-1$ linear combinations $Y_f^1, Y_f^2, \ldots, Y_f^{q-1}$ of the messages $\{X_i\}_{i \in W_f}$ with (non-zero) coefficients from $\mathbb{F}_q^\times$, such that for every $k \in [q-1]$, Y_f^k can be written as a linear combination of the super-messages $\hat{X}_1, \ldots, \hat{X}_r$. Let $\beta_f^k \triangleq [\beta_{f,1}^k, \ldots, \beta_{f,r}^k]$ be a vector of length r such that $Y_f^k = \sum_{i=1}^{r} \beta_{f,i}^k \hat{X}_i$. It should be noted that, for each $f \in [F]$, $Y_f^1, Y_f^2, \ldots, Y_f^{q-1}$ are the same up to a scalar multiple, i.e., for each $k \in [q-1]$, $Y_f^k = \delta_k Y_f^1$, or equivalently, $\beta_f^k = \delta_k \beta_f^1$, for some distinct $\delta_k \in \mathbb{F}_q^\times$. The user then constructs F vectors $\beta_1, \ldots, \beta_F$, each of length r, such that $\beta_f = \beta_f^{k_f}$ for $f \in [F]$, is chosen arbitrarily from the set of vectors $\{\beta_f^k\}_{k \in [q-1]}$. Let $Y_f \triangleq Y_f^{k_f}$ for $f \in [F]$. Each Y_f is referred to as a (linear) *function*. Note that β_f is the vector of coefficients of the super-messages $\{\hat{X}_i\}_{i \in [r]}$ in the function Y_f.

Step 3: The user then sends to all servers the vectors $Q_1, \ldots, Q_r$, associated with the super-messages $\hat{X}_1, \ldots, \hat{X}_r$, and the vectors $\beta_1, \ldots, \beta_F$, associated with the functions $Y_1, \ldots, Y_F$.

Step 4: Then, the user and the servers leverage the PC scheme of [85] with r (independent) messages and F (linear) functions of these messages such that the user can privately retrieve one of these functions. Indeed, the $r = K - D + 1$ super-messages $\{\hat{X}_i\}_{i \in [r]}$ and the F functions $\{Y_f\}_{f \in [F]}$, respectively, play the role of the original messages and the functions in the PC scheme, and the user is interested in retrieving the function Y_{f^*} privately, where Y_{f^*} is a linear combination with non-zero coefficients of the messages $\{X_i\}_{i \in \mathcal{W}}$. Note that by construction, there exists only one function Y_{f^*} among $Y_1, \ldots, Y_F$ such that

Y_{f^*} is a linear combination (with only non-zero coefficients) of the messages $\{X_i\}_{i\in\mathcal{W}}$, and the user's demand is an scalar multiple of Y_{f^*}. More specifically, each server first constructs the super-messages $\{\hat{X}_i\}_{i\in[r]}$ by using the coefficient vectors $\{Q_i\}_{i\in[r]}$ as described in Step 2, and then constructs the functions $\{Y_f\}_{f\in[F]}$ by utilizing the super-messages $\{\hat{X}_i\}_{i\in[r]}$ and the coefficient vectors $\{\beta_f\}_{f\in[F]}$ as explained in Step 2. Note that each function Y_f for $f \in [F]$ consists of $S = N^F$ symbols (from $\mathbb{F}_q$) where N is the number of servers. Then, each server sends to the user $S(1/N + 1/N^2 + \cdots + 1/N^{K-D+1})$ carefully designed linear combinations of all symbols associated with all functions $\{Y_f\}_{f\in[F]}$. The details of the design of the user's query to each server and each server's transmitted linear combinations (which also depend on the query of the user) can be found in [85, Section 4].

Example 1. (Modified GRS Code protocol) Assume that $K = 4$ independent messages from $\mathbb{F}_5^{16}$ are replicated over $N = 2$ servers, and the user wishes to compute one linear combination of $D = 3$ messages as $2X_1 + X_2 + X_3$, i.e., $\mathcal{W} = \{1, 2, 3\}$ and $V_1 = [2, 1, 1]$ (i.e., $v_1 = 2$, $v_2 = 1$, and $v_3 = 1$). Note that each message consists of $S = N^{\binom{K}{D}} = 16$ symbols from $\mathbb{F}_5$.

First, the user chooses $K = 4$ distinct elements $\omega_1, \ldots, \omega_4$ from $\mathbb{F}_5$. Suppose that the user picks $\omega_1 = 0$, $\omega_2 = 1$, $\omega_3 = 2$, $\omega_4 = 3$, and then constructs the polynomial $p(x) = \prod_{i\notin\mathcal{W}}(x - \omega_i) = x - \omega_4 = x - 3$. Then, the user computes α_j for $j \in \mathcal{W}$, as follows; $\alpha_1 = \frac{v_1}{p(\omega_1)} = 1$, $\alpha_2 = \frac{v_2}{p(\omega_2)} = 2$ and $\alpha_3 = \frac{v_3}{p(\omega_3)} = 4$, and chooses α_j for $j \notin \mathcal{W}$, i.e., α_4, randomly from $\mathbb{F}_5^{\times}$. Assume that the user chooses $\alpha_4 = 2$.

Then, the user constructs $r = K - D + 1 = 2$ vectors Q_1 and Q_2, each of length $K = 4$, such that $Q_i = [\alpha_1\omega_1^{i-1}, \ldots, \alpha_K\omega_K^{i-1}]$ for $i \in \{1, 2\}$, i.e., the user constructs $Q_1 = [1, 2, 4, 2]$ and $Q_2 = [0, 2, 3, 1]$. Note that for the set $W_1 = \{1, 2, 3\}$, there exist exactly $q - 1 = 4$ vectors $\beta_1^k = [2k, k]$ for $k \in [4]$ such that $2kQ_1 + kQ_2 = k[2, 1, 1, 0]$.

Then, the user arbitrarily chooses the vector β_1 from the set of vectors $\{\beta_1^k = [2k, k]\}_{k\in[4]}$. Suppose that the user chooses $\beta_1 = \beta_1^2 = [4, 2]$. Similarly, the user picks the vectors

$\beta_2 = [3, 1]$, $\beta_3 = [1, 4]$ and $\beta_4 = [0, 3]$. Then, the user sends to all servers the vectors Q_1 and Q_2 (associated with the super-messages $\hat{X}_1$ and $\hat{X}_2$), and the vectors $\beta_1, \ldots, \beta_4$ (associated with the functions $Y_1, \ldots, Y_4$). Using the coefficient vectors Q_1 and Q_2, each server first constructs the two super-messages $\hat{X}_1 = X_1 + 2X_2 + 4X_3 + 2X_4$ and $\hat{X}_2 = 2X_2 + 3X_3 + X_4$, and then constructs the functions $Y_1, \ldots, Y_4$ using the super-messages $\hat{X}_1$ and $\hat{X}_2$ and the coefficient vectors $\beta_1, \ldots, \beta_4$ as follows:

$$
\begin{aligned}
Y_1 &= 4\hat{X}_1 + 2\hat{X}_2 = 4X_1 + 2X_2 + 2X_3 \\
Y_2 &= 3\hat{X}_1 + \hat{X}_2 = 3X_1 + 3X_2 + 2X_4 \\
Y_3 &= \hat{X}_1 + 4\hat{X}_2 = X_1 + X_3 + X_4 \\
Y_4 &= 3\hat{X}_2 = X_2 + 4X_3 + 3X_4
\end{aligned}
$$

Finally, the user and the servers apply the PC scheme of [85] for two super-messages $\hat{X}_1$, $\hat{X}_2$ in order for the user to privately retrieve the function Y_1. It should be noted that among the functions $Y_1, \ldots, Y_4$, only Y_1 is a linear combination of the messages $\{X_i\}_{i \in \mathcal{W}} = \{X_1, X_2, X_3\}$, and the user's demand, i.e., $2X_1 + X_2 + X_3$ is equal to $3Y_1$. The details of the PC scheme for this example are as follows. Let $\pi : [16] \to [16]$ be a randomly chosen permutation. Let $u_f(i) \triangleq \sigma_i Y_f(\pi(i))$ for $f \in [4]$ and $i \in [16]$, where $Y_f(\pi(i))$ is the $\pi(i)$-th $\mathbb{F}_5$-symbol of Y_f, and σ_i is a randomly chosen element from $\{-1, +1\}$. For simplifying the notation, let $(a_i, b_i, c_i, d_i) = (u_1(i), u_2(i), u_3(i), u_4(i))$ for all $i \in [16]$. The user then queries 15 carefully designed linear combinations of the symbols $\{\{a_i\}_{i \in [16]}, \{b_i\}_{i \in [16]}, \{c_i\}_{i \in [16]}, \{d_i\}_{i \in [16]}\}$, as given in Table 4.1 [85], from each of the servers (S1 and S2).

As shown in [85], among the 15 symbols queried from S1 (or S2), 3 symbols are redundant (based on the information obtained from S2 (or S1)). For example, consider the 15 symbols queried from S1. (Similar observations can be made regarding the queries

from S2.) Among the 4 symbols $\{a_1, b_1, c_1, d_1\}$, any 2 symbols suffice to recover the other 2 symbols. For example, c_1 and d_1 can be obtained from a_1 and b_1. (Note that Y_3 and Y_4 can be written as a linear combination of Y_1 and Y_2.) Thus, the server S1 needs to send two arbitrary symbols from $\{a_1, b_1, c_1, d_1\}$. In addition, given any 2 symbols from $\{a_2, b_2, c_2, d_2\}$, any 5 symbols from $\{a_3 - b_2, a_4 - c_2, a_5 - d_2, b_4 - c_3, b_5 - d_3, c_5 - d_4\}$ queried from S1 would suffice to recover the remaining symbol. For example, $c_5 - d_4$ can be obtained from the symbols $\{a_3 - b_2, a_4 - c_2, a_5 - d_2, b_4 - c_3, b_5 - d_3, b_2, d_2\}$ (for details, see [85, Section 5.1]). Thus, each of the servers S1 and S2 needs to send to the user only 12 symbols. In particular, S1 transmits 2 arbitrary symbols from $\{a_1, b_1, c_1, d_1\}$, 5 from $\{a_3 - b_2, a_4 - c_2, a_5 - d_2, b_4 - c_3, b_5 - d_3, c_5 - d_4\}$, and the 4 symbols $\{a_9 - b_7 + c_6, a_{10} - b_8 + d_6, a_{11} - c_8 + d_7, b_{11} - c_{10} + d_9\}$, and the symbol $\{a_{15} - b_{14} + c_{13} - d_{12}\}$; and S2 transmits 2 arbitrary symbols from $\{a_2, b_2, c_2, d_2\}$, 5 arbitrary symbols from $\{a_6 - b_1, a_7 - c_1, a_8 - d_1, b_7 - c_6, b_8 - d_6, c_8 - d_7\}$, and the 4 symbols $\{a_{12} - b_4 + c_3, a_{13} - b_5 + d_3, a_{14} - c_5 + d_4, b_{14} - c_{13} + d_{12}\}$, and the symbol $\{a_{16} - b_{11} + c_{10} - d_9\}$.

From the answers sent by the servers, the user obtains all 16 symbols $a_1, \ldots, a_{16}$, and accordingly, all 16 symbols of Y_1. (Note that $a_i = u_1(i) = \sigma_i Y_1(\pi(i))$ for $i \in [16]$.) Then, the user can compute the desired linear combination, i.e., $2X_1 + X_2 + X_3$ by computing $3Y_1$. In order to retrieve Y_1 which consists of 16 symbols (over $\mathbb{F}_5$), according to the proposed protocol, the user downloads 24 symbols (over $\mathbb{F}_5$) from both servers. Thus, the rate of the proposed protocol is $16/24 = 2/3$.

It should be noted that for every subset of size 3 of the messages $\{X_i\}_{i\in[4]}$, in the proposed protocol, there exists one (and only one) linear combination (with non-zero coefficients) of these messages, namely Y_{f^*} for some $f^* \in [4]$. Moreover, as a result of the privacy guarantees of the PC scheme, no server can infer any information about the index (f^*) of the function Y_{f^*} being requested by the user. Thus, the proposed scheme satisfies the required joint privacy condition of the PLT problem.

Table 4.1: The queries of the PC protocol for $N = 2$ servers, 2 super-messages, and $F = 4$ functions, when the user demands Y_1.

S1	S2
a_1, b_1, c_1, d_1	a_2, b_2, c_2, d_2
$a_3 - b_2$	$a_6 - b_1$
$a_4 - c_2$	$a_7 - c_1$
$a_5 - d_2$	$a_8 - d_1$
$b_4 - c_3$	$b_7 - c_6$
$b_5 - d_3$	$b_8 - d_6$
$c_5 - d_4$	$c_8 - d_7$
$a_9 - b_7 + c_6$	$a_{12} - b_4 + c_3$
$a_{10} - b_8 + d_6$	$a_{13} - b_5 + d_3$
$a_{11} - c_8 + d_7$	$a_{14} - c_5 + d_4$
$b_{11} - c_{10} + d_9$	$b_{14} - c_{13} + d_{12}$
$a_{15} - b_{14} + c_{13} - d_{12}$	$a_{16} - b_{11} + c_{10} - d_9$

Lemma 17. *The Modified GRS Code protocol is a PLT protocol, and achieves the rate* $(\frac{1}{N}, K - D + 1)$.

Proof. The messages $\mathbf{X}_{[K]}$ are uniformly and independently distributed over $\mathbb{F}_q^S$, and the messages $\{\hat{X}_1, \ldots, \hat{X}_r\}$ are linearly independent combinations of the messages in $X_{[K]}$, thus $\{\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_r\}$ are uniformly and independently distributed over $\mathbb{F}_q^S$ as well, i.e., $H(\hat{\mathbf{X}}_1) = \cdots = H(\hat{\mathbf{X}}_r) = S \log q = B$. Hence, the rate of the Modified GRS Code protocol is the same as the rate of the PC protocol for N servers and $K - D + 1$ messages, which is given by $\Phi(\frac{1}{N}, K - D + 1)$ (see [85, Theorem 1]).

From the step 4 of the Modified GRS Code protocol, it is evident that the recoverability condition is satisfied. For the joint privacy of the proposed protocol, the proof is as follows. The PC protocol protects the privacy of the function requested by the user (i.e., no server can infer any information about the index of the function requested by the user

upon receiving the query). Consider an arbitrary server $n \in [N]$, which receives an arbitrary query $Q_n^{[\mathcal{W},V]}$, generated by the proposed protocol. Given $\mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]} = Q_n^{[\mathcal{W},V]}$, from the perspective of server n, every function Y_f for $f \in [F]$, is equally likely to be the user's desired function. We denote the support of Y_f by $\mathcal{Y}_f$, i.e., $\mathcal{Y}_f$ is the set of all indices $i \in [K]$ such that X_i has a non-zero coefficient in the linear combination Y_f. Note that for any $\tilde{\mathcal{W}} \in \mathbb{W}$, in the proposed protocol, there exists only one function Y_{f^*} among $Y_1, \ldots, Y_F$ with $\mathcal{Y}_{f^*} = \tilde{\mathcal{W}}$. Thus, for any $\tilde{\mathcal{W}} \in \mathbb{W}$ and every $n \in [N]$, the following holds

$$\mathbb{P}(\boldsymbol{\mathcal{W}} = \tilde{\mathcal{W}} | \mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]} = Q_n^{[\mathcal{W},V]}) = \Pr(\boldsymbol{\mathcal{W}} = \mathcal{Y}_{f^*} | \mathbf{Q}_n^{[\boldsymbol{\mathcal{W}},\mathbf{V}]} = Q_n) = \tfrac{1}{F} = \tfrac{1}{\binom{K}{D}} = \mathbb{P}(\boldsymbol{\mathcal{W}} = \tilde{\mathcal{W}}).$$

This completes the proof. □

Part II

Fast Data Access in Distributed Systems

5. SERVICE RATE REGION USING COMBINATORIAL APPROACH*

5.1 Introduction

Providing reliability against failures, ensuring availability of stored content during high demand, providing fast content download and serving a large number of users simultaneously are major concerns in cloud storage systems. The service capacity has been recently recognized as an important performance metric. It has a wide relevance, and can be interpreted as a measure of the maximum number of users that can be simultaneously served by a coded storage system [71–74, 76, 98, 99]. Thus, maximizing the service capacity is of great significance for the emerging applications such as distributed learning. Moreover, maximizing the service capacity reduces the users' latency, particularly in a high traffic regime, which is important for the delay-sensitive applications such as live streaming.

The service rate problem is concerned with a distributed storage system in which k files $f_1, \ldots, f_k$ are stored across n servers using a linear $[n, k]_q$ code such that the requests to download file f_i arrive at rate λ_i, and the server l operates at rate μ_l. A goal of the service rate problem is to determine the service rate region of this coded storage system which is the set of all request arrival rates $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ that can be served by this system given the finite service rate of the servers. The service rate problem is generally formulated as a sequence of linear programs, that has been studied only in some limited cases [73, 74, 76]. In this work, we show that the service rate problem is equivalent to the fractional matching problem which were extensively studied in the context of graph theory. This equivalence result allows one to leverage the techniques in the rich literature of the graph theory for solving the service rate problem.

*Reprinted with permission from [75] "A Combinatorial View of the Service Rates of Codes Problem, its Equivalence to Fractional Matching and its Connection with Batch Codes," by F. Kazemi, E. Karimi, E. Soljanin, and A. Sprintson, 2020. In Proceedings of 2020 IEEE International Symposium on Information Theory (ISIT), pp. 646-651, June 2020. Copyright © by IEEE.

Related Work

Existing studies on data access pursue various directions. Many are focused on providing efficient maintenance of storage under possible failures of a subset of nodes accessed (see e.g., [51, 52, 99–101]). These studies typically assume infinite service rate (instantaneous service) for each storage node. Hence, they do not address the problem of serving a large number of users simultaneously.

Another important line of work is concerned with caching (see e.g., [56, 57, 102]), in which generally the limited capacity of the backhaul link is considered as the main bottleneck of the system, and the goal is usually to minimize the backhaul traffic by prefetching the popular contents at the storage nodes of limited size. Thus, these works do not address the scenarios where many users want to get the same content concurrently given the limited capacity of the access part of the network.

The other related body of work is concerned with minimizing the download latency (see e.g., [62–67, 69, 70, 103–106]). These papers assume that the storage nodes can serve the customers at some finite rate, and aim to compute the download latency for intractable queueing systems that appear in coded storage.

We note now and explain in detail later that because of the constraints on the service rate of servers, by maximizing the service capacity the load balancing is provided in the distributed storage system (see [98]). In that sense, the most relevant work to this work includes batch codes, switch codes and PIR codes (see e.g., [16, 107–110]). However, the problems considered in these papers, as we will show later, can be often seen as special cases of the service rate problem.

A connection between distributed storage allocation problems (see [99, 111] and references therein) and matching problems in hyper-graphs have been observed in computer science literature [112] (see also [113]). In particular, it was noted that the uniform model

of distributed storage allocation considered in [99] leads to a question which is asymptotically equivalent to the fractional version of a long standing conjecture by Erdős [114] on the maximum number of edges in a uniform hypergraph.

Our Contributions

We first introduce a novel graph representation of coding schemes, which we refer to as a recovery graph for the coding scheme, in Sec. 5.3.1. We then show the following results in Sec. 5.3.4: 1) equivalence between the service rate problem and the well-known fractional matching problem and 2) equivalence between the integral service rate problem and the matching problem. These equivalence results allow us to show that the service capacity of a code is equal to the fractional matching number in the recovery graph of a code, and thus is lower bounded and upper bounded by the matching number and the vertex cover number, respectively. This is beneficial because if the recovery graph of a code is a bipartite graph, then the upper bound and lower bound are equal, which allows us to establish the service capacity of the storage system. Leveraging this result, we determine the service capacity of the binary simplex codes whose recovery graph, as we will show, is bipartite. Furthermore, we show that the service rate problem can be viewed as a generalization of batch codes problem in Sec. 5.4. In particular, we show that the multiset primitive batch codes problem is a special case of the service rate problem in which the solution (i.e., the portion of requests assigned to the recovery sets of each file) is restricted to be integral.

5.2 Coded System and its Service Rate Region

Throughout this work, we use bold-face lower-case letters for vectors and bold-face capital letters for matrices. Let $\mathbb{N}$ denote the set of positive integers. $\mathbb{F}_q$ denotes the finite field with q elements. For $i \in \mathbb{N}$, $[i] \triangleq \{1, \ldots, i\}$. For $n \in \mathbb{N}$, $\mathbf{1}_n$ denotes the all-one vector of length n.

Consider a storage system where k files $f_1, \ldots, f_k$ are stored across n servers labeled $1, \ldots, n$, using an $[n, k]_q$ code with generator matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$. A set of stored symbols that can be used to recover file f_i is referred to as a recovery set for file f_i. Let $\mathbf{g}_j$ be the jth column of $\mathbf{G}$. The set $R \subseteq [n]$ is a recovery set for file f_i if there exists non-zero α_j's $\in \mathbb{F}_q$ such that $\sum_{j \in R} \alpha_j \mathbf{g}_j = \mathbf{e}_i$, where $\mathbf{e}_i$ denotes the ith unit vector. In other words, a set R is a recovery set for file f_i if the unit vector $\mathbf{e}_i$ can be recovered by a linear combination of the columns of $\mathbf{G}$ indexed by the set R.

Let $t_i \in \mathbb{N}$ denote the number of recovery sets for file f_i, and $\mathcal{R}_i = \{R_{i,1}, \ldots, R_{i,t_i}\}$ denote the set of recovery sets for file f_i. We assume w.l.o.g. that the time to download a file from server $l \in [n]$ is exponential with rate $\mu_l \in \mathbb{R}_{\geq 0}$, i.e., μ_l is the average rate at which server l resolves the received file requests. We denote the service rates of servers $1, \ldots, n$ by the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$. We further assume that the arrival of requests for file f_i is Poisson with rate λ_i, $i \in [k]$. We denote the request rates for files $1, \ldots, k$ by the vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$. We consider the class of scheduling strategies that assign a fraction of requests for a file to each of its recovery sets. Let $\lambda_{i,j}$ be the portion of requests for file f_i that are assigned to the recovery set $R_{i,j}$, $j \in [t_i]$.

The service rate problem seeks to determine the set of arrival rates $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ that can be served by a coded storage system with generator matrix $\mathbf{G}$ and service rate $\boldsymbol{\mu}$, referred to as service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu}) \subseteq \mathbb{R}_{\geq 0}^k$.

Definition 1. *An $(\mathbf{G}, \boldsymbol{\mu})$ system is a coded storage system in which k files are stored across n servers using a linear $[n, k]_q$ code with generator matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ such that file f_i for $i \in [k]$ has $t_i \in \mathbb{N}$ recovery sets denoted by $\mathcal{R}_i = \{R_{i,1}, \ldots, R_{i,t_i}\}$, and the service rate of servers in the system is $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$.*

Definition 2. *The service rate region of an $(\mathbf{G}, \boldsymbol{\mu})$ system, denoted by $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$, is the set*

of vectors $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ *for which there exist* $\lambda_{i,j}$ *satisfying the following constraints:*

$$\sum_{j=1}^{t_i} \lambda_{i,j} = \lambda_i, \qquad \textit{for all} \quad i \in [k] \tag{5.1a}$$

$$\sum_{i=1}^{k} \sum_{\substack{j \in [t_i] \\ l \in R_{i,j}}} \lambda_{i,j} \leq \mu_l, \quad \textit{for all} \quad l \in [n] \tag{5.1b}$$

$$\lambda_{i,j} \in \mathbb{R}_{\geq 0}, \qquad \textit{for all} \quad i \in [k],\ j \in [t_i] \tag{5.1c}$$

Note that the constraints (5.1a) ensure that the demands for all files are served, and the constraints (5.1b) guarantee that no node is sent requests in excess of its service rate.

Proposition 1. *[76, Lemma 1] The service rate region of an* $(\mathbf{G}, \boldsymbol{\mu})$ *system* $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ *is a non-empty, convex, closed, and bounded subset of the* $\mathbb{R}^k_{\geq 0}$.

The service capacity of an $(\mathbf{G}, \boldsymbol{\mu})$ system, $\lambda^\star(\mathbf{G}, \boldsymbol{\mu})$, is defined as the maximum sum of arrival rates that can be served simultaneously by the storage system. We define a maximum demand vector, denoted by $\boldsymbol{\lambda}^\star = (\lambda_1^\star, \ldots, \lambda_k^\star)$, as a vector in the service rate region for which $\sum_{i=1}^k \lambda_i^\star = \lambda^\star(\mathbf{G}, \boldsymbol{\mu})$. An instance of the maximum demand vector is obtained by solving the following linear programming (LP):

$$\max \ \sum_{i=1}^{k} \lambda_i \qquad \text{s.t. (5.1) holds.} \tag{5.2}$$

Definition 3. *The integral service rate region of an* $(\mathbf{G}, \boldsymbol{\mu})$ *system, denoted by* $\mathcal{S}_I(\mathbf{G}, \boldsymbol{\mu})$, *is the set of all vectors* $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ *for which there exist* $\lambda_{i,j} \in \mathbb{Z}_{\geq 0}$ *satisfying the sets of constraints* (5.1a), (5.1b).

Note that each demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the integral service rate region has integral coordinates, i.e., $\mathcal{S}_I(\mathbf{G}, \boldsymbol{\mu}) \subseteq \mathbb{Z}^k_{\geq 0}$. However, because of the fractional relaxation

of $\lambda_{i,j}$, it is not guaranteed that the vectors with integral coordinates in the service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ are also in the integral service rate region $\mathcal{S}_I(\mathbf{G}, \boldsymbol{\mu})$.

Remark 14. *In the integral setting of the service rate problem where $\lambda_{i,j}$ are non-negative integers, if each server can serve up to one request at a time, i.e., $\mu_l = 1$ for all servers $l \in [n]$, then one can easily conclude that $\lambda_{i,j}$ are binary and the recovery sets used for each file f_i, $i \in [k]$ are disjoint.*

5.3 Equivalence to Fractional Matching

We first introduce a graph representation of a coding scheme, referred to as a recovery graph, which is useful for characterizing the service capacity of a coded storage system through relating this problem with the well-known problem of finding the maximum fractional matching in a graph. In particular, we show that the service capacity of a code equals the fractional matching number in the recovery graph of the code. Another way of determining the service capacity of a coded storage system is providing tight bounds on the maximum sum of the arrival rates that can be served by the storage system. We show that the matching number and the vertex cover number in the recovery graph of a code, respectively are a lower bound and an upper bound on the service capacity of a code. Thus, if the recovery graph of a code is a bipartite graph, according to the Duality Theorem [115], the matching number and vertex cover number are identical, and we are able to determine the capacity. As an application of this result, we determine the service capacity of the binary simplex codes whose recovery graph, as we will show, is a bipartite graph. We next describe how to construct the recovery graph of a code, and then we present the interesting connections.

5.3.1 Graph Representation of Storage Schemes

Here, we introduce a graph representation of storage schemes. For simplicity, we consider linear codes, however, it is straightforward to generalize the notion for non-linear

codes. For the clarity of exposition, we focus on recovery sets of size one and two. In other words, a recovery set for each file is either a systematic symbol or a group of two symbols, as is the case when $k = 2$. As we discuss in Remark 15 later, the notions described can be easily extended to the general case of arbitrary sized recovery sets by considering hypergraphs in which each edge can be incident to an arbitrary number of vertices.

Consider an $[n, k]$ code with a $k \times n$ generator matrix $\mathbf{G}$. We define a graph $G(V, E)$ associated with the generator matrix $\mathbf{G}$, referred to as a *recovery graph* for the coding scheme G, where the vertices in V correspond to the n encoded data symbols (the servers of the storage system), and the edges in E correspond to the recovery sets of files. In $G(V, E)$, each self-loop represents a recovery set of size 1 for the vertex (file) that it is connected to, and each edge between two vertices represents a recovery set of size 2 for the file that can be recovered from these two vertices. Each edge is assigned a color such that the edges that correspond to the recovery sets of the same file are assigned the same color. In that sense, we have an edge-colored graph. It should be noted that a graph with self-loops can be simply converted to a graph without any self-loops by adding sufficient number of dummy vertices (servers). We assume that the label of all dummy servers is zero and thus we denote a systematic recovery set for file f_i by $\{0, r\}$ where r is the label of the systematic server storing file f_i. Section 5.3.3 provides an example that shows the recovery graph for the binary $[7, 3]$ simplex code.

5.3.2 Matching and Vertex Cover on Graphs

In this section, we with briefly review the notions of matching and vertex cover on graphs. For details, we refer the reader to standard texts on graph theory, e.g., [115, 116].

Definition 4. *A matching in a graph is a set of all pairwise non-adjacent edges.*

Alternatively, a matching in a graph $G(V, E)$ is an assignment of the values $\tilde{x}_e \in \{0, 1\}$ to the edges $e \in E$ in such a way that for each vertex $v \in V$, the sum of the values on the

incident edges is at most 1. All the edges $e \in E$ with value $\tilde{x}_e = 1$ are in the matching. Thus, a *matching vector* in a graph $G(V, E)$ can be defined as a vector $\tilde{\boldsymbol{x}} = (\tilde{x}_e : e \in E)$ satisfying the following conditions:

$$\sum_{\text{e incident to } v} \tilde{x}_e \leq 1, \qquad \text{for all } \ v \in V \tag{5.3a}$$

$$\tilde{x}_e \in \{0, 1\}, \qquad \text{for all } \ e \in E \tag{5.3b}$$

Definition 5. *A maximum matching in a graph is a matching that contains the largest number of edges. The maximum matching vector is denoted by* $\tilde{\boldsymbol{x}}^\star$.

The size of a maximum matching in a graph $G(V, E)$ is called matching number, denoted by $m(G)$. There may be several instances of maximum matchings in a graph. The problem of finding an instance of maximum matching can be formulated as follows:

$$\max \ \sum_{e \in E} \tilde{x}_e \qquad \text{s.t. (5.3) holds.} \tag{5.4}$$

Definition 6. *A fractional matching in a graph* $G(V, E)$ *is an assignment of the values* $x_e \in [0, 1]$ *to the edges* $e \in E$ *such that for each vertex* $v \in V$*, the sum of the values on the incident edges is at most* 1.

A *fractional matching vector* in a graph $G(V, E)$ can be defined as a vector $\boldsymbol{x} = (x_e : e \in E)$ satisfying the following constraints:

$$\sum_{\text{e incident to } v} x_e \leq 1, \qquad \text{for all } \ v \in V \tag{5.5a}$$

$$x_e \in [0, 1], \qquad \text{for all } \ e \in E \tag{5.5b}$$

Definition 7. *A maximum fractional matching, denoted by* $\boldsymbol{x}^\star$*, is a fractional matching vector in the graph that has the maximum value* $\sum_{e \in E} x_e$.

The value of a maximum fractional matching in a graph $G(V, E)$ is called the fractional matching number, denoted as $m_f(G)$. Finding an instance of maximum fractional matching in a graph can be formulated as the following LP:

$$\max \sum_{e \in E} x_e \qquad \text{s.t. (5.5) holds.} \tag{5.6}$$

Definition 8. *A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex in the set.*

Alternatively, a vertex cover of a graph $G(V, E)$ is an assignment of the values $y_v \in \{0, 1\}$ to the vertices $v \in V$ in such a way that for each edge $e \in E$, the sum of the values on the endpoint vertices is at least 1. All the vertices $v \in V$ with value $\tilde{y}_v = 1$ are in the vertex cover. Thus, a *vertex cover vector* of a graph $G(V, E)$ can be defined as a vector $\boldsymbol{y} = (y_v : v \in V)$ satisfying the following conditions:

$$\sum_{v \text{ incident to } e} y_v \geq 1, \qquad \text{for all } e \in E \tag{5.7a}$$

$$y_v \in \{0, 1\}, \qquad \text{for all } v \in V \tag{5.7b}$$

Definition 9. *A minimum vertex cover in a graph is a vertex cover with minimum number of vertices.*

The cardinality of a minimum vertex cover in a graph $G(V, E)$ is called vertex cover number, denoted by $v(G)$. There may be several instances of a minimum vertex cover in a graph. Finding an instance of minimum vertex cover in a graph can be formulated as the following integer LP:

$$\min \sum_{v \in V} y_v \qquad \text{s.t. (5.7) holds.} \tag{5.8}$$

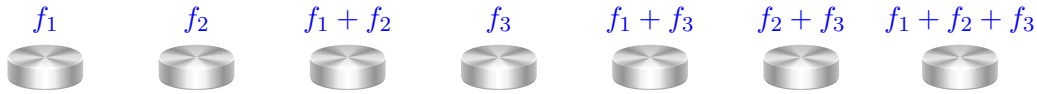

Figure 5.1: A distributed storage system consists of 7 servers storing files f_1, f_2, and f_3 using a binary $[7, 3]$ simplex code.

Proposition 2. *For an arbitrary graph G, it is known that $m(G) \leq m_f(G) \leq v(G)$. For a bipartite graph G, it holds that $m(G) = m_f(G) = v(G)$.*

In what follows, we assume that each server in the distributed storage system can serve up to one request at each moment, i.e., $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) = (1, \ldots, 1)$. Thus, $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ and $\lambda^\star(\mathbf{G}, \boldsymbol{\mu})$ only depend on the generator matrix $\mathbf{G}$ and are respectively denoted by $\mathcal{S}(\mathbf{G})$ and $\lambda^\star(\mathbf{G})$. Next, we present an example to show how the service rate of a code is connected to the matching and the vertex cover problems.

5.3.3 Example of Equivalence

Here, we present an example to give more intuition about the subsequent results and to provide a sketch of the proofs. Consider a distributed storage system in which files f_1, f_2, and f_3 are stored across 7 servers, labeled $1, \ldots, 7$, using a binary $[7, 3]_2$ simplex code with the service rate $\mu_l = 1$, $l \in [7]$. The generator matrix of this code is given by:

$$\mathbf{G} = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array},$$

where the number above each column shows the label of the corresponding column (server). Fig. 5.1 depicts this distributed storage system.

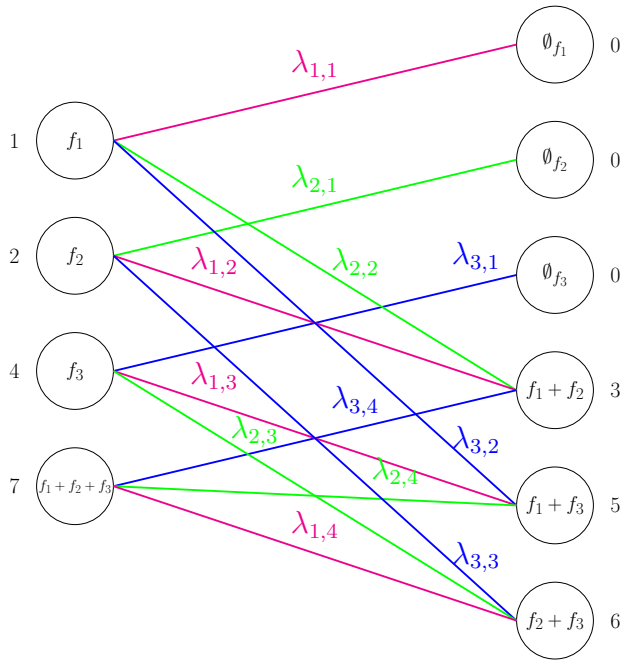

Figure 5.2: Recovery graph of the binary $[7, 3]$ Simplex code.

The recovery sets for each file are given by

$$\mathcal{R}_1 = \{R_{1,1}, \ldots, R_{1,4}\} = \{\{0,1\},\{2,3\},\{4,5\},\{6,7\}\}$$
$$\mathcal{R}_2 = \{R_{2,1}, \ldots, R_{2,4}\} = \{\{0,2\},\{1,3\},\{4,6\},\{5,7\}\}$$
$$\mathcal{R}_3 = \{R_{3,1}, \ldots, R_{3,4}\} = \{\{0,4\},\{1,5\},\{2,6\},\{3,7\}\}$$

The recovery graph of $[7, 3]_2$ simplex code is drawn in Fig. 5.2, which is a bipartite graph. The vertices $\emptyset_{f_1}$, $\emptyset_{f_2}$ and $\emptyset_{f_3}$ are the dummy vertices added to the graph for the purpose of removing the self-loops of systematic vertices f_1, f_2, and f_3, respectively. The edges with color magenta, green, and blue represent recovery sets for files f_1, f_2, and f_3, respectively. Moreover, the label $\lambda_{i,j}$ above an edge indicates the portion of requests for file f_i that is assigned to the recovery set $R_{i,j}$.

The service rate region $\mathcal{S}(\mathbf{G})$ of this system is the set of vectors $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ for which there exist $\lambda_{i,j}$'s, $i \in [3]$ and $j \in [4]$, satisfying the set of constraints (5.1) as follows:

$$\text{(5.1a)} \Rightarrow \begin{cases} \lambda_1 = \lambda_{1,1} + \lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4} \\ \lambda_2 = \lambda_{2,1} + \lambda_{2,2} + \lambda_{2,3} + \lambda_{2,4} \\ \lambda_3 = \lambda_{3,1} + \lambda_{3,2} + \lambda_{3,3} + \lambda_{3,4} \end{cases} \tag{5.9}$$

$$\text{(5.1b)} \Rightarrow \begin{cases} \lambda_{1,1} + \lambda_{2,2} + \lambda_{3,2} \leq 1 \\ \lambda_{2,1} + \lambda_{1,2} + \lambda_{3,3} \leq 1 \\ \lambda_{3,1} + \lambda_{1,3} + \lambda_{2,3} \leq 1 \\ \lambda_{3,4} + \lambda_{2,4} + \lambda_{1,4} \leq 1 \\ \lambda_{2,2} + \lambda_{1,2} + \lambda_{3,4} \leq 1 \\ \lambda_{3,2} + \lambda_{1,3} + \lambda_{2,4} \leq 1 \\ \lambda_{3,3} + \lambda_{2,3} + \lambda_{1,4} \leq 1 \end{cases} \tag{5.10}$$

$$\text{(5.1c)} \Rightarrow \begin{cases} \lambda_{i,j} \in \mathbb{R}_{\geq 0}, \quad \text{for all} \quad i \in [3],\ j \in [4] \end{cases} \tag{5.11}$$

Fig. 5.3 shows the service rate region $\mathcal{S}(\mathbf{G})$ of this coded storage system.

Based on (5.5), a fractional matching $\boldsymbol{x} = (\lambda_{1,1}, \ldots, \lambda_{1,4}, \lambda_{2,1}, \ldots, \lambda_{2,4}, \lambda_{3,1}, \ldots, \lambda_{3,4})$ of the graph depicted in Fig. 5.2, satisfies the constraints (5.10) and (5.11). Thus, according to Definition 2, a vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ obtained from $\boldsymbol{x}$ using (5.9) is in the service rate region of $[7, 3]_2$ simplex code. Conversely, for a vector $\boldsymbol{\lambda}$ in the service rate region of $[7, 3]_2$ simplex code, there exist $\lambda_{i,j}$'s, $i \in [3]$ and $j \in [4]$, satisfying the constraints (5.10) and (5.11), that define a fractional matching vector $\boldsymbol{x} = (\lambda_{i,j} : i \in [3] \text{ and } j \in [4])$ in the recovery graph of $[7, 3]_2$ simplex code drawn in Fig. 5.2.

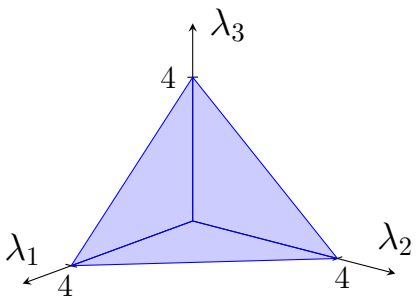

Figure 5.3: Service rate region of binary $[7, 3]$ Simplex code.

Based on (5.6), a maximum fractional matching vector $\boldsymbol{x}^\star$ is obtained by solving the following LP:

$$\max \sum_{i=1}^{3}\sum_{j=1}^{4} \lambda_{i,j} \qquad \text{s.t. (5.10) and (5.11) hold.} \tag{5.12}$$

We want to show that the vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ obtained from $\boldsymbol{x}^\star$ using (5.9) is in fact a maximum demand vector $\boldsymbol{\lambda}^\star$ in the service rate region of $[7, 3]_2$ simplex code. From (5.9), $\sum_{i=1}^{3}\sum_{j=1}^{4} \lambda_{i,j} = \lambda_1 + \lambda_2 + \lambda_3$. Thus, it can be easily verified that $\boldsymbol{x}^\star$ provides a solution for the following LP:

$$\max \ \lambda_1 + \lambda_2 + \lambda_3 \qquad \text{s.t. (5.9), (5.10), (5.11) hold.} \tag{5.13}$$

Moreover, according to (5.2), an instance of maximum demand vector is obtained by solving the LP in (5.13). Thus, the vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ obtained from $\boldsymbol{x}^\star$ using (5.9) is a maximum demand vector $\boldsymbol{\lambda}^\star$. On the other hand, for an instance of $\boldsymbol{\lambda}^\star$ in the service rate region of $[7, 3]_2$ simplex code obtained from (5.13), there exists a fractional matching vector $\boldsymbol{x}$ which according to the same reasoning, provides a solution for (5.12). Thus, the vector $\boldsymbol{x}$ is a maximum fractional matching vector $\boldsymbol{x}^\star$ in the recovery graph of $[7, 3]_2$ sim-

plex code in Fig. 5.2. Since a maximum demand vector $\boldsymbol{\lambda}^\star = (\lambda_1^\star, \lambda_2^\star, \lambda_3^\star)$ is obtained from a maximum fractional matching vector $\boldsymbol{x}^\star$ by (5.9), it follows that $\lambda_1^\star + \lambda_2^\star + \lambda_3^\star = \sum \lambda_{i,j}^\star$, where $\lambda_{i,j}^\star$'s are the elements of $\boldsymbol{x}^\star$. Hence, we have $\lambda^\star(\mathbf{G}) = m_f(G)$, and based on Proposition 2, $m(G) \leq \lambda^\star(\mathbf{G}) \leq v(G)$ holds.

We show that the service capacity of $[7, 3]_2$ simplex code is 4. The proof consists of two parts. First, we need to prove the converse by showing that the service capacity cannot be bigger than 4. It is easy to see that the set of vertices $\{f_1, f_2, f_3, f_1 + f_2 + f_3\}$ is a minimum vertex cover for the graph in Fig. 5.2. Thus, the vertex cover number of this graph is $v(G) = 4$ which indicates that $\lambda^\star(\mathbf{G}) \leq 4$. Next, we show the achievability proof by showing that there exists a demand vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ in the service rate region such that $\lambda_1 + \lambda_2 + \lambda_3 = 4$. For this purpose, one can consider the set of edges labeled by $\lambda_{1,1}$, $\lambda_{1,2}$, $\lambda_{1,3}$, and $\lambda_{1,4}$ as a matching in the graph. Corresponding to this matching, a demand vector $\boldsymbol{\lambda} = (4, 0, 0)$ is obtained by applying (5.9).

5.3.4 Equivalence Results

We first show an equivalence between the service rate problem and the fractional matching problem. This equivalence result allow us to derive bounds on the service capacity of a coded storage system and then to recover the service capacity of the binary simplex code whose recovery graph is bipartite.

Theorem 11. *Consider an $(\mathbf{G}, \boldsymbol{\mu})$ system with the service rate $\boldsymbol{\mu} = \mathbf{1}_n$. There exists a demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region of this system if and only if there exists a fractional matching vector $\boldsymbol{x} = (\lambda_{i,j} : i \in [k] \text{ and } j \in [t_i])$ in the recovery graph of $[n, k]_q$ code such that $\boldsymbol{\lambda}$ and $\boldsymbol{x}$ are related based on* (5.1a).

Proof. If a vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ is in the service rate region of this storage system, there exist $\lambda_{i,j}$'s, for $i \in [n]$ and $j \in [t_i]$, satisfying the set of constraints in (5.1a), (5.1b) and (5.1c). Based on the definition of the recovery graph of codes and the fact that

$\mu_l = 1$, $l \in [n]$, it is easy to observe that the set of constraints (5.1b) and (5.1c) are equivalent to the set of constraints (5.5). Thus, the vector $\boldsymbol{x} = (\lambda_{i,j} : i \in [k] \text{ and } j \in [t_i])$ is a fractional matching in the recovery graph of $[n, k]_q$ code. Now, assume that a vector $\boldsymbol{x} = (\lambda_{i,j} : i \in [k] \text{ and } j \in [t_i])$ is a fractional matching in the recovery graph of $[n, k]_q$ code. Hence, the vector $\boldsymbol{x}$ satisfies the sets of constraints (5.5), or equivalently, it satisfies the set of constraints (5.1b) and (5.1c). Based on Definition 2, a vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ obtained from $\boldsymbol{x}$ using (5.1a) is in the service rate region of $[n, k]_q$ code. □

Remark 15. In defining recovery graphs and in Theorem 11, we restricted our attention to linear coding schemes having recovery sets of size at most two. Extension to the general case of a code having recovery sets of arbitrary size is straightforward: we associate a hypergraph with the code's generator matrix. (A hypergraph is a generalization of a graph in which any subset of vertices may be joined by an edge, called a hyperedge, see, e.g., [117, Chapter 7].) We form a hypergraph $G(V, E)$ associated with G such that its vertices correspond to columns of G and hyperedges correspond to recovery sets. It is straightforward to generalize the hypergraph representation for non-linear codes.

Corollary 1. *Consider an* $(\mathbf{G}, \boldsymbol{\mu})$ *system with* $\boldsymbol{\mu} = \mathbf{1}_n$. *There exists a maximum demand vector* $\boldsymbol{\lambda}^\star = (\lambda_1^\star, \ldots, \lambda_k^\star)$ *in the service rate region* $S(\mathbf{G})$ *of this storage system if and only if there exists a maximum fractional matching vector* $\boldsymbol{x}^\star = (\lambda_{i,j}^\star : i \in [k]$ *and* $j \in [t_i])$ *in the recovery graph of* $[n, k]_q$ *code such that* $\boldsymbol{\lambda}^\star$ *and* $\boldsymbol{x}^\star$ *are related based on* (5.1a).

Proof. An instance of the maximum fractional matching vector in the recovery graph of an $[n, k]_q$ code can be obtained by solving the following LP according to (5.6).

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} \sum_{j=1}^{t_i} \lambda_{i,j} \\ \text{s.t.} \quad & \text{(5.1b), (5.1c)} \end{aligned}$$

According to the Theorem 11, there exist a demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region which is obtained from $\boldsymbol{x}^\star = (\lambda^\star_{i,j} : i \in [k] \text{ and } j \in [t_i])$ using (5.1a). We want to show that the vector $\boldsymbol{\lambda}$ is in fact a maximum demand vector $\boldsymbol{\lambda}^\star$. Using (5.1a), we have $\sum_{i=1}^{k} \sum_{j=1}^{t_i} \lambda^\star_{i,j} = \sum_{i=1}^{k} \lambda_i$. Thus, it can be easily verified that $\boldsymbol{x}^\star$ provides a solution for the following LP:

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} \lambda_i \\ \text{s.t.} \quad & \text{(5.1a), (5.1b), (5.1c)} \end{aligned}$$

Based on (5.2), an instance of the maximum demand vector is obtained by solving the above linear programming. Thus, the vector $\boldsymbol{\lambda}$ resulted from $\boldsymbol{x}^\star$ by (5.1a) is a maximum demand vector $\boldsymbol{\lambda}^\star$. On the other hand, for a maximum demand vector $\boldsymbol{\lambda}^\star$ in the service rate region which is obtained from (5.2), there exists a fractional matching vector $\boldsymbol{x}$ that based on a similar reasoning, provides a solution for (5.6). Thus, the vector $\boldsymbol{x}$ is a maximum fractional matching vector $\boldsymbol{x}^\star$ in the recovery graph of $[n, k]_q$ code. □

Theorem 12. *Consider an* $(\mathbf{G}, \boldsymbol{\mu})$ *system with the service rate* $\boldsymbol{\mu} = \mathbf{1}_n$*. The service capacity* $\lambda^\star(\mathbf{G})$ *of this system is lower bounded by the matching number and upper bounded by the vertex cover number of the recovery graph of* $[n, k]_q$ *code. i.e.,* $m(G) \leq \lambda^\star(\mathbf{G}) = m_f(G) \leq v(G)$.

Proof. According to Corollary 1, a maximum demand vector $\boldsymbol{\lambda}^\star = (\lambda^\star_1, \ldots, \lambda^\star_k)$ is obtained from a maximum fractional matching vector $\boldsymbol{x}^\star = (\lambda^\star_{i,j} : i \in [k] \text{ and } j \in [t_i])$ using (5.1a). It follows that

$$\sum_{i=1}^{k} \lambda^\star_i = \sum \lambda^\star_{i,j}$$

where $\lambda^\star_{i,j}$'s are the elements of $\boldsymbol{x}^\star$. Thus, $\lambda^\star(\mathbf{G}) = m_f(G)$. Thus, according to Proposition 2, we have $m(G) \leq \lambda^\star(\mathbf{G}) \leq v(G)$. □

It should be noted that if the recovery graph of a code is bipartite, Proposition 2 results $m(G) = \lambda^\star(\mathbf{G}) = v(G)$.

Theorem 13. *The recovery graph of* $[2^k - 1, k, 2^{k-1}]_2$ *simplex code, is a bipartite graph.*

Proof. Based on the definition of the bipartite graph, a graph $G(V, E)$ is a bipartite graph if the vertices V of the graph, can be divided into two disjoint and independent sets, say V_1 and V_2 such that every edge of the graph $e \in E$ connects a vertex in V_1 to one in V_2. Thus, in order to show that the recovery graph of the k-dimensional binary simplex code with generator matrix $\mathbf{G}$ is a bipartite graph, we need to determine the two disjoint sets of vertices, i.e., V_1 and V_2, in the recovery graph $G(V, E)$ of the $[2^k - 1, k]_2$ simplex code. Then, we have to prove that every edge $e \in E$ of the recovery graph connects a vertex in V_1 to one in V_2 or equivalently we have to prove that there is no edge between the vertices in V_1 as well as in V_2.

The set of vertices V of the recovery graph $G(V, E)$ of the $[2^k - 1, k]_2$ simplex code correspond to the files stored on the storage nodes or the columns of the generator matrix $\mathbf{G}$. The columns of the generator matrix $\mathbf{G}$ of the $[2^k - 1, k]_2$ simplex code are the set of all non-zero vectors of $\mathbb{F}_2^k$. Note that up to column permutations the generator matrix $\mathbf{G}$ of the $[2^k - 1, k]_2$ simplex code is unique. Now, we can partition the columns of $\mathbf{G}$ into two sets V_1 and V_2 such that V_1 is the set of all non-zero column vectors in $\mathbb{F}_2^k$ with odd number of ones and V_2 is the set of all non-zero column vectors in $\mathbb{F}_2^k$ with even number of ones. Thus, V_1 and V_2 are two disjoint sets of columns that partition the columns of $\mathbf{G}$. Moreover, the self-loops corresponding to the systematic recovery sets are removed from the recovery graph by adding dummy nodes. Consider each dummy node (column) as a zero vector in $\mathbb{F}_2^k$, denoted by $\mathbf{0}$. Thus, V_1 and $V_2' = \{V_2 \cup \mathbf{0}\}$ determine two disjoint sets of vertices partitioning V in the $G(V, E)$.

We want to prove that there is no edge between the vertices corresponding to the set

of columns V_1 and there is no edge between the vertices corresponding to the set V_2'. The proof is based on the contradiction approach. Let $\mathbf{x}, \mathbf{x}' \in V_1$. Assume that there is an edge between the vertices corresponding to the $\mathbf{x}, \mathbf{x}' \in V_1$. This means that $\{\mathbf{x}, \mathbf{x}'\}$ forms a recovery set for a file f_i, $i \in [k]$, i.e., $\mathbf{x} + \mathbf{x}' = \mathbf{e}_i$. Since both $\mathbf{x}$ and $\mathbf{x}'$ have an odd number of ones, their sum must have an even number of ones which is a contradiction. Thus, there is no edge between the vertices in V_1. The proof for V_2' is similar. Let $\mathbf{x}, \mathbf{x}' \in V_2'$. Since both $\mathbf{x}$ and $\mathbf{x}'$ have an even number of ones, their sum must have an even number of ones which shows that $\{\mathbf{x}, \mathbf{x}'\}$ cannot be a recovery set for any file f_i, $i \in [k]$. Thus, there is no edge between any $\mathbf{x}, \mathbf{x}' \in V_2'$. □

Corollary 2. *For an* $(\mathbf{G}, \boldsymbol{\mu})$ *system with* $[2^k - 1, k, 2^{k-1}]_2$ *simplex code and service rate* $\boldsymbol{\mu} = \mathbf{1}_n$*, the service capacity is given by* $m(G) = \lambda^\star(\mathbf{G}) = v(G) = 2^{k-1}$.

Proof. The proof consists of two parts. First, we need to prove the converse by showing that the service capacity cannot be bigger than 2^{k-1}. It can be easily seen that the set of all 2^{k-1} vertices corresponding to the columns of $\mathbf{G}$ with odd number of ones forms a minimum vertex cover in the recovery graph of the $[2^k - 1, k, 2^{k-1}]_2$ simplex code. The reason is that since the recovery graph of the this code, based on Theorem 13, is a bipartite graph, all the edges are covered by either one of the two partitions, i.e., V_1 and V_2' introduced in the proof of Theorem 13. Thus, the vertex cover number of this graph is $v(G) = 2^{k-1}$ which indicates that $\lambda^\star(\mathbf{G}) \leq 2^{k-1}$.

Next, we show the achievability proof by showing that a vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ exists in service rate region of the $[2^k - 1, k, 2^{k-1}]_2$ simplex code with $\sum_{i=1}^k \lambda_i = 2^{k-1}$. For this purpose, since the recovery graph of this code is a bipartite graph, we have $m(G) = v(G) = 2^{k-1}$ which means that there exists a matching of size 2^{k-1} in the recovery graph of the binary k-dimensional simplex code. For the $[2^k - 1, k, 2^{k-1}]_2$ simplex code which is a special subclass of availability codes, it is known that every file f_i for $i \in [k]$ can be

recovered from $2^{k-1}-1$ (availability) disjoint groups of two (locality) servers. Thus, by considering the systematic recovery set, for each file f_i, $i \in [k]$, there are exactly 2^{k-1} disjoint recovery sets. One can consider the set of edges $\{\lambda_{i,1}, \ldots, \lambda_{i,2^{k-1}}\}$, for every $i \in [k]$, as an instance of matching in the recovery graph. Corresponding to this matching, a demand vector $\boldsymbol{\lambda} = 2^{k-1}.\mathbf{e}_i$ for $i \in [k]$ is obtained by applying (5.1a). □

Corollary 3. *Consider a $(\mathbf{G}, \boldsymbol{\mu})$ system with $\boldsymbol{\mu} = \mathbf{1}_n$. A demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ exists in the integral service rate region $S_I(\mathbf{G})$ of this system if and only if there exists a matching vector $\tilde{\boldsymbol{x}} = (\lambda_{i,j} : i \in [k]$ and $j \in [t_i])$ in the recovery graph of $[n,k]_q$ code such that $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{x}}$ are related based on* (5.1a).

Proof. The proof is similar to the proof of Theorem 11. □

Corollary 4. *Consider an $(\mathbf{G}, \boldsymbol{\mu})$ system with $\boldsymbol{\mu} = \mathbf{1}_n$. There exists a maximum demand vector $\boldsymbol{\lambda}^\star = (\lambda_1^\star, \ldots, \lambda_k^\star)$ in the integral service rate region $S_I(\mathbf{G})$ of this storage system if and only if there exists a maximum matching vector $\tilde{\boldsymbol{x}}^\star = (\lambda_{i,j}^\star : i \in [k]$ and $j \in [t_i])$ in the recovery graph of $[n,k]_q$ code such that $\boldsymbol{\lambda}^\star$ and $\tilde{\boldsymbol{x}}^\star$ are related based on* (5.1a).

Proof. The proof is similar to the proof of Corollary 1. □

5.4 Generalization of Batch codes

In this section, we show how the service rate problem can be viewed as a generalization of the problem of batch codes. That further illustrates connections with PIR codes, switch codes and locally repairable codes which all can be seen as special cases of batch codes (see [108]).

5.4.1 Definitions of Batch Codes and PIR Codes

Definition 10. *[107] An (n,k,t,m,τ) batch code $\mathcal{C}$ over a finite alphabet $\sum$ encodes any string $\boldsymbol{x} = (x_1, \ldots, x_k)$ into m strings (buckets) $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ of total length n by an*

encoding mapping $\mathcal{C} : \sum^k \to \sum^n$, such that for each t-tuple (batch) of indices $i_1, \ldots, i_t \in [k]$, the entries $x_{i_1}, \ldots, x_{i_t}$ can be decoded by reading at most τ symbols from each bucket.

Definition 11. *[108] An (n, k, t) primitive batch code is an (n, k, t, m, τ) batch code, where each bucket contains exactly one symbol, i.e., $n = m$. Note that in this setting $\tau = 1$, i.e., at most one symbol can be recovered from each bucket.*

Definition 12. *An (n, k, t) multiset primitive batch code is an (n, k, t) primitive batch code where the information symbols $x_{i_1}, \ldots, x_{i_t}$ are requested by t distinct users such that the indices $i_1, \ldots, i_t$ are not necessarily distinct and in general they form a multiset. Moreover, the requested symbols can be reconstructed from the data read by t different users independently (i.e., x_{i_j} can be recovered by the user j) so that the sets of the symbols read by these users are disjoint.*

It should be noted that for the sake of simplicity, we refer to a linear (n, k, t) multiset primitive batch code over $\mathbb{F}_q$ as an $[n, k, t]_q$ batch code.

Proposition 3. *[109, Theorem 1] A linear $[n, k]_q$ code $\mathcal{C}$ with generator matrix $\mathbf{G}$ is an $[n, k, t]_q$ batch code if and only if there exist t non-intersecting sets $T_1, \ldots, T_t$ of indices of columns in the generator matrix $\mathbf{G}$ such that for each $j \in [t]$, there exists a linear combination of columns of $\mathbf{G}$ indexed by T_j which equals to the vector $\mathbf{e}_{i_j}$, for all $j \in [t]$ and $i_j \in [k]$.*

Definition 13. *[16] A linear $[n, k]_2$ code $\mathcal{C}$ with generator matrix $\mathbf{G}$ is called a t-server PIR code if for every $i \in [k]$, there exist t disjoint sets of columns of $\mathbf{G}$ that add up to $\mathbf{e}_i$.*

5.4.2 Connection with Batch Codes and PIR Codes

Theorem 14. *Given the integral service rate region $\mathcal{S}_I(G)$ of code $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ with service rate $\boldsymbol{\mu} = \mathbf{1}_n$, if all vectors in the set $S_t = \{\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k) | \sum_{i=1}^k \lambda_i = t, \lambda_i \in \mathbb{Z}_{\geq 0}\}$ are in the $\mathcal{S}_I(G)$, the code $\mathbf{G}$ is a linear $[n, k, t]_q$ batch code.*

Proof. The existence of all vectors in set $S_t = \{\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k) | \sum_i^k \lambda_i = t, \lambda_i \in \mathbb{Z}_{\geq 0}\}$ in the integral service rate region $\mathcal{S}_I(\mathbf{G})$ of code $\mathbf{G}$ indicates that for any multiset of indices $\{i_1, \ldots, i_t\}$, $i_j \in [k]$, the requests for the information symbols $f_{i_1}, f_{i_2}, \ldots, f_{i_t}$ can be served at the same time by the storage system. On the other hand, each server can serve up to one request at a time, i.e., $\mu_l = 1$ for all servers $l \in [n]$, which shows that $\lambda_{i,j}$ are binary. As a result, t disjoint recovery sets are used for satisfying each demand vector $\boldsymbol{\lambda} \in S_t$. This means that for every multiset of indices $\{i_1, \ldots, i_t\}$, $i_j \in [k]$, there exist t disjoint sets $T_1, \ldots, T_t$ of indices of columns in the generator matrix $\mathbf{G}$ such that for each $j \in [t]$, there exists a linear combination of columns of $\mathbf{G}$ indexed by T_j which equals to the vector $\mathbf{e}_{i_j}$. Therefore, based on Proposition 3, the code $\mathbf{G}$ is a $[n, k, t]_q$ batch code. □

Theorem 14 shows that the integral setting of the service rate problem where the solution (the portion of requests that are assigned to the recovery sets) is restricted to be integral, is the same as the setting of the multiset primitive batch code problem. Thus, the general setting of the service rate problem where a fractional solution is allowed, can be viewed as a generalization of the setting of the multiset primitive batch code problem.

Corollary 5. *Given the integral service rate region $\mathcal{S}_I(G)$ of code $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ with service rate $\boldsymbol{\mu} = \mathbf{1}_n$, if all vectors in the set $S_t = \{t.e_1 = (t, 0, \ldots, 0), \ldots, t.e_k = (0, \ldots, 0, t) | t \in \mathbb{N}\}$ are in the $\mathcal{S}_I(G)$, the code $\mathbf{G}$ is a t-server PIR code.*

Proof. The existence of the set $S_t = \{t.\mathbf{e}_1 = (t, 0, \ldots, 0), \ldots, t.\mathbf{e}_k = (0, \ldots, 0, t) | t \in \mathbb{N}\}$ in the integral service rate region $\mathcal{S}_I(\mathbf{G})$ of code $\mathbf{G}$ indicates that for every $i \in [k]$, t requests for file f_i can be served at the same time by the storage system. Since $\mu_l = 1$ for all servers $l \in [n]$ and $\lambda_{i,j}$ are binary, one can readily confirm that for each file f_i, $i \in [k]$, there exist t disjoint recovery sets which are used for satisfying the demand vector $t.\mathbf{e}_i \in S_t$. Thus, for every $i \in [k]$, there exist t disjoint sets of columns in the $\mathbf{G}$ that add up to $\mathbf{e}_i$. Thus, based on definition 13, the code $\mathbf{G}$ is an $[n, k]_2$ t-server PIR code. □

Next, we present an example regarding the application of Theorems 14 that shows a binary $[7,3]_2$ simplex code is a $[7,3,4]_2$ batch code.

Example 10. Consider a binary $[7,3]_2$ simplex code. In this example, utilizing the recovery graph and the integral service rate region of the code, we want to show that this code is a $[7,3,4]_2$ batch code. To this end, we need to show that each demand vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ with $\sum_{i=1}^{3} \lambda_i = 4$, is in the integral service rate region of the $[7,3]_2$ simplex code, i.e., for each of these vectors, there exists a matching in the recovery graph of $[7,3]_2$ binary simplex code. W.l.o.g we assume that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The 4 recovery sets of each file, say f_1, are known. As can be seen in Fig. 5.4, the four magenta edges corresponding to the recovery sets of file f_1, constructs a maximum matching. Using the Algorithm 1, we show that how one can start with a maximum matching corresponding to the vector $\boldsymbol{\lambda_a} = (4,0,0)$ and by following at most two steps find the maximum matching corresponding to any vector $\boldsymbol{\lambda_b} = (\lambda_1, \lambda_2, \lambda_3)$ with $\lambda_1 + \lambda_2 + \lambda_3 = 4$. For this purpose, in a nutshell, we start with the recovery sets of file f_1 and replace some of them with the recovery sets for files f_2 and f_3 as needed. Next, we define three steps, based on which we present the Algorithm 1 that can be generalized for any number of files k.

Step 1: Consider the systematic recovery set for file f_i, and add the corresponding edge to the matching set. Accordingly, remove the recovery set for file f_1, incident to the node f_i, from the matching set.

Step 2: Find $(\lambda_i - 1)/2$ number of loops, each of size 4, consisting of 2 recovery sets for file f_i and 2 recovery sets for file f_1, in the recovery graph of the code. Then, by considering each of the loops, replace the 2 recovery sets for file f_1 with the 2 recovery sets for file f_i in the matching.

Step 3: This step would be the same as step 2, except that here $(\lambda_i - 1)/2$ is replaced with $(\lambda_i)/2$.

Algorithm 1 Finding a Max Matching for any $(\lambda_1, \lambda_2, \lambda_3)$ with $\lambda_1 + \lambda_2 + \lambda_3 = 4$

Input: Max matching corresponding to $(4, 0, 0)$
for $i = 2 : 3$ **do**
if λ_i is odd
do Step 1;
do Step 2;
else
do Step 3;
end
Output: Max matching corresponding to $(\lambda_1, \lambda_2, \lambda_3)$

For instance, to find the corresponding matching for the demand vector $\boldsymbol{\lambda} = (2, 2, 0)$, we need to show that there are two recovery sets of file f_1 that can be used to form two recovery sets for file f_2. The 4 magenta edges (recovery sets of file f_1) form a maximum matching of size 4. In the graph representation, it is easy to find a loop of size 4 consisting of two magenta edges, $\lambda_{a,3}$ and $\lambda_{a,4}$, and two green edges (recovery sets of file f_2), $\lambda_{b,3}$ and $\lambda_{b,4}$. Therefore, simply we can replace these two magenta edges with the green edges and construct another maximum matching of size 4 which is a matching corresponding to the demand $\boldsymbol{\lambda} = (2, 2, 0)$.

Now, consider the demand vector $\boldsymbol{\lambda} = (2, 1, 1)$. Since λ_2 and λ_3 are odd, we need to find the recovery sets of file f_1 that can be used for building systematic recovery sets for files f_2 and f_3. It can be seen that the magenta edges connected to the nodes 2 and 4 of the recovery graph, i.e., $\lambda_{a,2}$ and $\lambda_{a,3}$, can be removed from the original maximum matching and be substituted by the green edge $\lambda_{b,1}$, and the blue edge $\lambda_{c,1}$, which represent systematic recovery sets for f_1 and f_2, respectively.

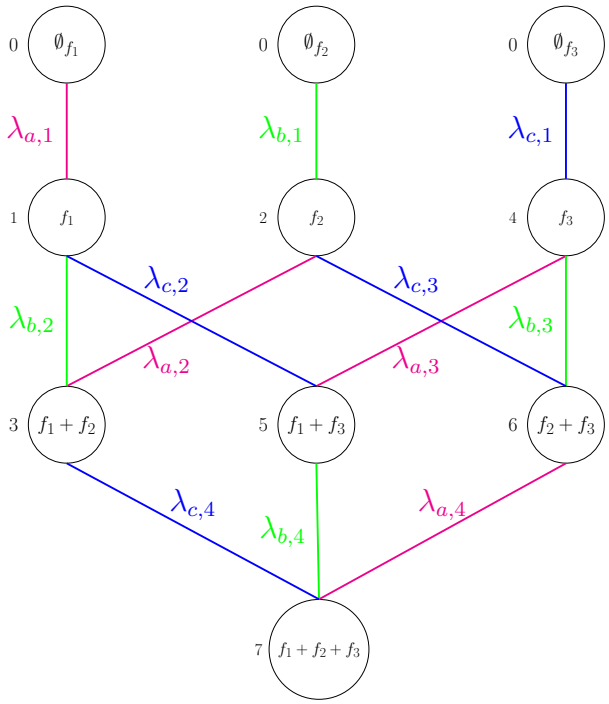

Figure 5.4: Graph representation of the binary $[7, 3]$ simplex code.

6. SERVICE RATE REGION USING GEOMETRIC APPROACH*

6.1 Introduction

One of the most significant considerations in the design of distributed storage systems is serving a large number of users concurrently. The service rate is an important performance metric that measures the number of users that can be simultaneously served by a storage system [71–75] that implements an erasure code. Maximizing the service rate reduces the latency experienced by users, particularly in a high traffic regime.

The service rate problem considers a distributed system where k files, $f_1, \ldots, f_k$ are encoded into n, and stored across n servers. File f_i can be recovered by reading data from a single or a set of storage nodes, referred to as a recovery set for file f_i. Requests to download file f_i arrive at rate λ_i, and can be split across its recovery sets. Server l can simultaneously serve multiple requests if their cumulative arrival rate does not exceed the maximum service rate μ_l of server l. The service rate problem seeks to determine the service rate region of the coded storage system which is defined as the set of all request arrival rate vectors $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ that can be served by this system.

The service rate problem has been studied only in some special cases: 1) for MDS codes when $n \geq 2k$ and binary simplex codes in [73] and 2) for systems with arbitrary n when $k = 2$ in [73] and $k = 3$ in [74]. The existing techniques for solving the problem require enumeration of all possible recovery sets, which becomes increasingly complex when the number of files k increases. Thus, introducing a technique which is not depending on the enumeration of recovery sets is of great significance. In this work, we introduce a novel geometric approach with this goal in mind.

*Reprinted with permission from [76] "A Geometric View of the Service Rates of Codes Problem and its Application to the Service Rate of the First Order Reed-Muller Codes," by F. Kazemi, S. Kurz, and E. Soljanin, 2020. In Proceedings of 2020 IEEE International Symposium on Information Theory (ISIT), pp. 66-71, June 2020. Copyright © by IEEE.

Related Work

The past two decades have seen an ever increasing interest in coding for storage and caching. Special codes that support efficient maintenance of storage under node failures have been proposed in e.g., [51,52,99–101]. The locality and availability of codes matter in such scenarios. This line of work mostly assumes infinite service rate (immediate service) for servers, and is primarily concerned with reliability of storage rather than with serving a large number of simultaneous users.

Another line of work is focused on caching (see e.g., [56, 57, 102]). In these work, the limited capacity of the backhaul link is considered as the main bottleneck, and the goal is usually to minimize its traffic by prefetching the popular contents at the storage nodes with limited size. Thus, these work do not address the scenarios such as live streaming, where many users wish to get the same content concurrently given the limited capacity of the access part of the network.

The most related to this work are papers concerned with content download from coded storage. Load balancing in such systems has recently been addressed in [98]. Memory allocation that maximizes the probability of successful content download under limited access to distributed storage was considered in e.g., [99, 111] and references therein. Minimizing the service rate in these scenarios was considered in [71, 72]. This problem is similar to ours but it assumes that all users request the same content which is encoded by an MDS code and stored on nodes with unlimited storage capacity. Fast content download from coded storage was considered in e.g., [64, 65], and references therein. These papers strive to compute the download latency for increasingly complex queueing systems that appear in coded storage [70, 105, 106]. The service rate problem is related to the stability region of such queues.

Our Contributions

We study the service rates of codes problem by introducing a novel geometric approach. This approach overcomes the main drawback of the previous work which are trying to solve this problem by formulating it as a sequence of linear programs (LP). There, one must exactly know all possible recovery sets to enumerate the constraints in each LP, and must also solve all the LPs.

Using our novel geometric technique, we take initial steps towards deriving bounds on the service rates of some parametric classes of linear codes without explicitly listing the set of all possible recovery sets. In particular, we derive upper bounds on the service rates of the first order Reed-Muller codes and simplex codes, as two classes of codes which are most important in theory as well as in practice. Subsequently, we show how the derived upper bounds can be achieved. Moreover, utilizing the proposed geometric technique, we show that given the service rate region of a code, a lower bound on the minimum distance of the code can be derived.

6.2 Problem Statement

6.2.1 Notation

Throughout this chapter, we denote vectors by bold-face small letters and matrices by bold-face capital letters. Let $\mathbb{N}$ denote the set of the non-negative integer numbers. Let $\mathbb{F}_q$ be a finite field for some prime power q, and $\mathbb{F}_q^n$ be the n-dimensional vector space over $\mathbb{F}_q$. Let us denote a q-ary linear code $\mathcal{C}$ of length n, dimension k and minimum distance d by $[n, k, d]_q$. We denote the Hamming weight of $\mathbf{x}$ by $\mathrm{w}(\mathbf{x})$. For a positive integer k, let $\mathbf{0}$ and $\mathbf{1}$ denote the all-zero and all-one column vectors of length k, respectively. Let $\mathbf{e}_i$ denote a unit vector of length k, having a one at position i and zeros elsewhere. For a positive integer i, define $[i] \triangleq \{1, \ldots, i\}$. Let us denote the cardinality of a set or multiset $\mathcal{S}$ by $\#\mathcal{S}$.

6.2.2 Service Rate of Codes

Consider a storage system in which k files $f_1, \ldots, f_k$ are stored over n servers, labeled $1, \ldots, n$, using a linear $[n, k]_q$ code with generator matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$. Let $\mathbf{g}_j$ denote the jth column of $\mathbf{G}$. A recovery set for the file f_i is a set of stored symbols which can be used to recover file f_i. With respect to $\mathbf{G}$, a set $R \subseteq [n]$ is a recovery set for file f_i if there exist α_j's $\in \mathbb{F}_q$ such that $\sum_{j \in R} \alpha_j \mathbf{g}_j = \mathbf{e}_i$, i.e., the unit vector $\mathbf{e}_i$ can be recovered by a linear combination of the columns of $\mathbf{G}$ indexed by the set R. W.l.o.g., we restrict our attention to the reduced recovery sets obtained by considering non-zero coefficients α_j's and linearly independent columns $\mathbf{g}_j$'s.

Let $\mathcal{R}_i = \{R_{i,1}, \ldots, R_{i,t_i}\}$ be the $t_i \in \mathbb{N}$ recovery sets for file f_i. Let $\mu_l \in \mathbb{R}_{\geq 0}$ be the average rate at which the server $l \in [n]$ resolves received file requests. We denote the service rates of servers $1, \ldots, n$ by a vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$. We further assume that the requests to download file f_i arrive at rate λ_i, $i \in [k]$. We denote the request rates for files $1, \ldots, k$ by the vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$. Let $\lambda_{i,j}$ be the portion of requests for file f_i that are assigned to the recovery set $R_{i,j}$, $j \in [t_i]$.

The service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu}) \subseteq \mathbb{R}_{\geq 0}^k$ is defined as the set of all request vectors $\boldsymbol{\lambda}$ that can be served by a coded storage system with generator matrix $\mathbf{G}$ and service rate $\boldsymbol{\mu}$. Alternatively, $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ can be defined as the set of all vectors $\boldsymbol{\lambda}$ for which there exist $\lambda_{i,j} \in \mathbb{R}_{\geq 0}$, $i \in [k]$ and $j \in [t_i]$, satisfying the following constraints:

$$\sum_{j=1}^{t_i} \lambda_{i,j} = \lambda_i, \qquad \text{for all} \quad i \in [k], \tag{6.1a}$$

$$\sum_{i=1}^{k} \sum_{\substack{j \in [t_i] \\ l \in R_{i,j}}} \lambda_{i,j} \leq \mu_l, \quad \text{for all} \quad l \in [n], \tag{6.1b}$$

$$\lambda_{i,j} \in \mathbb{R}_{\geq 0}, \qquad \text{for all} \quad i \in [k],\ j \in [t_i]. \tag{6.1c}$$

The constraints (6.1a) guarantee that the demands for all files are served, and constraints (6.1b) ensure that no node receives requests at a rate in excess of its service rate.

Lemma 18. *The service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ is a non-empty, convex, closed, and bounded subset of $\mathbb{R}^k_{\geq 0}$.*

Proof. It can be easily observed that for every service rate vector $\boldsymbol{\mu}$, setting $\lambda_{i,j} = 0$, where $i \in [k]$ and $j \in [t_i]$, satisfies the set of constraints in (6.1) for the all-zero demand vector of dimension k denoted by $\mathbf{0} = (0, \ldots, 0) \in \mathbb{R}^k$. Thus, $\mathbf{0}$ always belongs to the service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. It proves that the service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ is a non-empty subset of $\mathbb{R}^k_{\geq 0}$. Based on the definition of the convex set, we need to show that for all $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$ in $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ and for all $0 \leq \pi \leq 1$, all vectors $\pi\boldsymbol{\lambda} + (1-\pi)\tilde{\lambda}$ are in $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. Since $\boldsymbol{\lambda} \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$, there exist $\lambda_{i,j}$'s, where $i \in [k]$ and $j \in [t_i]$, that satisfy the set of constraints in (6.1) for the demand vector $\boldsymbol{\lambda}$ and the service rate vector $\boldsymbol{\mu}$. Also, since $\tilde{\lambda} \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$, there exist $\tilde{\lambda}_{i,j}$'s, where $i \in [k]$ and $j \in [t_i]$, that satisfy the set of constraints in (6.1) for the demand vector $\tilde{\lambda}$ and the service rate vector $\boldsymbol{\mu}$. One can easily confirm that $(\pi\lambda_{i,j} + (1-\pi)\tilde{\lambda}_{i,j})$'s, where $i \in [k]$ and $j \in [t_i]$, also satisfy the set of constraints in (6.1) for the demand vector $\pi\boldsymbol{\lambda} + (1-\pi)\tilde{\lambda}$ for all $0 \leq \pi \leq 1$, and the service rate vector $\boldsymbol{\mu}$. Thus, $\pi\boldsymbol{\lambda} + (1-\pi)\tilde{\lambda}$ belongs to $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ for all $0 \leq \pi \leq 1$. This completes the proof of convexity of the service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. Summing up the set of constraints in (6.1b) leads us to:

$$\sum_{l=1}^{n} \sum_{i=1}^{k} \sum_{\substack{j \in [t_i] \\ l \in R_{i,j}}} \lambda_{i,j} \leq \sum_{l=1}^{n} \mu_l$$

Changing the order of the sums and utilizing the fact that $\sum_{l=1}^{n} \sum_{\substack{j \in [t_i] \\ l \in R_{i,j}}} \lambda_{i,j} = \sum_{j=1}^{t_i} \lambda_{i,j}$, we obtain

$$\sum_{i=1}^{k} \sum_{j=1}^{t_i} \lambda_{i,j} \leq \sum_{l=1}^{n} \mu_l.$$

Using (6.1a), we rewrite the last inequality to

$$\sum_{i=1}^{k} \lambda_i \leq \sum_{l=1}^{n} \mu_l \tag{6.2}$$

The equation (6.2) indicates that the elements of every vector $\boldsymbol{\lambda} \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ are bounded. It also shows that all demand vectors $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ with $\sum_{i=1}^{k} \lambda_i > \sum_{l=1}^{n} \mu_l$ are not in $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. Hence, $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ is closed and bounded. □

Proposition 4. *[118] For any set $\mathcal{A} = \{\mathbf{v}_1, \ldots, \mathbf{v}_p\} \subseteq \mathbb{R}^k$, the convex hull of the set $\mathcal{A}$, denoted by $\mathrm{conv}(\mathcal{A})$, consists of all convex combinations of the elements of $\mathcal{A}$, i.e., all vectors of the form $\sum_{i=1}^{p} \gamma_i \mathbf{v}_i$, with $\gamma_i \geq 0$, $\sum_{i=1}^{p} \gamma_i = 1$.*

Corollary 6. *The service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu}) \subseteq \mathbb{R}^k_{\geq 0}$ forms a polytope which can be expressed in two forms: as the intersection of a finite number of half spaces or as the convex hull of a finite set of vectors (vertices of the polytope).*

Proof. Based on Lemma 18, the service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ is a convex and bounded subset of the $\mathbb{R}^k_{\geq 0}$, which indicates that $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ is a polytope. Thus, according to [119, Theorem 4], it can be described as the two mentioned forms, i.e., the intersection of a finite number of half spaces or the convex hull of a finite set of vectors (the vertices of the polytope). □

The service rate problem seeks to determine the service rate region $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ of a coded storage system with generator matrix $\mathbf{G}$ and service rate $\boldsymbol{\mu}$. Based on Corollary 6, the first algorithm for computing the service rate region that comes to mind is enumerating all vertices of the polytope $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ and then computing the convex hull of the resulting vertices. As we will show shortly, this problem can be formulated as an optimization problem consisting of a sequence of LPs.

Given that any $k-1$ request arrival rates, $\lambda_{i_1}, \ldots, \lambda_{i_{k-1}}$, are zeros, there exists a maximum value of λ_{i_k}, denoted by $\lambda^\star_{i_k}$, where $0 \leq \lambda^\star_{i_k} \leq \sum_{l=1}^{n} \mu_l$ (see the proof of Lemma 18) such that $\lambda^\star_{i_k}.\mathbf{e}_{i_k} \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ and all vectors $\lambda_{i_k}.\mathbf{e}_{i_k}$ with $\lambda_{i_k} > \lambda^\star_{i_k}$ are not in $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. These constrained optimization problems of finding the maximum value $\lambda^\star_{i_k}$ are all LPs. For $i \in [k]$, let $\mathbf{v}_i = \lambda^\star_i \mathbf{e}_i$. Since $\mathcal{J} = \{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\} \subseteq \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$, as an immediate consequence of Lemma 18 and Proposition 4, the set $\mathrm{conv}(\mathcal{J})$ is contained in $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. Starting with $\mathcal{J}$, one can iteratively enlarge $\mathcal{J}$ until the subsequent procedure stops. A facet H of $\mathrm{conv}(\mathcal{J})$ described by a vector $\mathbf{h} \in \mathbb{R}^k_{\geq 0}$ and $\eta \in \mathbb{R}_{\geq 0}$ is chosen as follows

$$H = \left\{\mathbf{x} \in \mathbb{R}^k_{\geq 0} \,:\, \mathbf{h}^\top \mathbf{x} = \eta\right\} \cap \mathrm{conv}(\mathcal{J}).$$

For the chosen facet H described by the vector $\mathbf{h} \in \mathbb{R}^k_{\geq 0}$ and $\eta \in \mathbb{R}_{\geq 0}$, one should solve $\max \mathbf{h}^\top \boldsymbol{\lambda}$, where $\boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0}$ satisfies the demand constraints (6.1a) and capacity constraints (6.1b). If the optimal target value is strictly larger than η, then the solution vector $\boldsymbol{\lambda}^\star$ is added to $\mathcal{J}$ and this procedure continues. For any $\mathbf{h} = (h_1, \ldots, h_k)$, the primal LP is given by

$$\max \sum_{i=1}^{k} h_i \lambda_i \qquad \text{s.t. (6.1) holds.} \tag{6.3}$$

and the corresponding dual LP is given by

$$\begin{aligned} \min \quad & \sum_{l=1}^{n} \gamma_l \mu_l && && (6.4)\\ \text{s.t.} \quad & h_i \leq \beta_i && \forall i \in [k] \\ & \beta_i \leq \sum_{l \in R_{i,j}} \gamma_l && \forall i \in [k], \forall j \in [t_i] \\ & \beta_i \in \mathbb{R} && \forall i \in [k] \\ & \gamma_l \in \mathbb{R}_{\geq 0} && \forall l \in [n] \end{aligned}$$

Based on the Duality Theorem [120], if both the primal LP and the corresponding dual LP have feasible solutions, then their optimal target values coincide. It can be easily confirmed that a feasible solution for the primal LP (6.3) can be given by $\lambda_{i,j} = 0$ and $\lambda_i = 0$, and a feasible solution for the dual LP (6.4) can be given by $\beta_i = h_i$ and $\gamma_l = \sum_{i=1}^{k} h_i$. Thus, given a generator matrix $\mathbf{G}$ of a linear code and a service rate $\boldsymbol{\mu}$ of the servers in a distributed coded storage system, the LP (6.3) can be utilized to compute the maximum value of $\eta = \sum_{i=1}^{k} h_i \lambda_i$, denoted by $\eta^\star$, for every vector $\mathbf{h} \in \mathbb{R}_{\geq 0}^{k}$. Having $\eta^\star$ at hand, it is known that all the vectors $\boldsymbol{\lambda} \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ satisfy $\sum_{i=1}^{k} h_i \lambda_i \leq \eta^\star$, which is a valid inequality for $\mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$. The downside of this approach is that we have to exactly know the set of all possible recovery sets for each file and also have to optimally solve all the LPs (6.3). Using the dual LP (6.4), we run into a similar problem since in order to list all the inequalities in (6.4), again we require to know the elements of all the recovery sets for each file, which becomes increasingly complex when the number of files k increases.

Thus, characterizing the exact service rate region of some parametric classes of linear codes or deriving some bounds on the service rate region of a storage scheme is a challenging problem, which we aim to address in this work. Towards this goal, we introduce a novel geometric approach. Leveraging our geometric approach, upper bounds on the sum of each subset of arrival rates in any demand vector that can be served by a linear code. That is, using this approach, one can obtain a finite set of half-spaces (upper bounds) whose intersection encompasses the service rate region of a given linear storage scheme. In this work, using our proposed geometric technique, we derive upper bounds on the service rates of the first order Reed-Muller codes and simplex codes.

6.2.3 Description of Storage Schemes

In this section, we briefly review the geometric description of linear codes. For more details, see [121–123].

Definition 14. *For a vector space $\mathcal{V}$ of dimension v over $\mathbb{F}_q$, ordered by inclusion, the set of all $\mathbb{F}_q$-subspaces of $\mathcal{V}$ forms a finite modular geometric lattice with meet $X \wedge Y = X \cap Y$, join $X \vee Y = X + Y$, and rank function $X \mapsto \dim(X)$. This subspace lattice of $\mathcal{V}$ is known as the projective geometry of $\mathcal{V}$, denoted by* $\mathrm{PG}(\mathcal{V})$.

For a vector space $\mathcal{V}$ of dimension v over $\mathbb{F}_q$, the 1-dimensional subspaces of $\mathcal{V}$ are the points of $\mathrm{PG}(\mathcal{V})$, the 2-dimensional subspaces of $\mathcal{V}$ are the lines of $\mathrm{PG}(\mathcal{V})$, and the $v-1$ dimensional subspaces of $\mathcal{V}$ are called the hyperplanes of $\mathrm{PG}(\mathcal{V})$. The projective geometry $\mathrm{PG}(\mathcal{V})$ is also denoted by $\mathrm{PG}(v-1,q)$, which is referred to as the $v-1$ dimensional projective space over $\mathbb{F}_q$. This notion makes sense because of the fact that, up to isomorphism, the projective geometry $\mathrm{PG}(\mathcal{V})$ only depends on the order q of the base field and the (*algebraic*) dimension v, which is justifying the notion $\mathrm{PG}(v-1,q)$ of (*geometric*) dimension $v-1$ over $\mathbb{F}_q$.

Let $\mathcal{V}$ be a vector space of dimension v over $\mathbb{F}_q$. The set of all k-dimensional subspaces of $\mathcal{V}$, referred to as *k-subspaces*, will be denoted by $\begin{bmatrix} \mathcal{V} \\ k \end{bmatrix}_q$. The cardinality of this set is given by the Gaussian binomial coefficient as follows

$$\begin{bmatrix} v \\ k \end{bmatrix}_q = \begin{cases} \frac{(q^v-1)(q^{v-1}-1)\cdots(q^{v-k+1}-1)}{(q^k-1)(q^{k-1}-1)\cdots(q-1)} & \text{if } 0 \leq k \leq v; \\ 0 & \text{otherwise.} \end{cases}$$

A multiset is a modification of the concept of a set that, unlike a set, allows for multiple instances for each of its elements. The positive integer number of instances, given for each element is called the multiplicity of this element in the multiset. More formally, a multiset $\mathcal{S}$ on a base set $\mathcal{X}$ can be identified with its characteristic function $\chi_{\mathcal{S}} : \mathcal{X} \to \mathbb{N}$, mapping $x \in \mathcal{X}$ to the multiplicity of x in $\mathcal{S}$. The *cardinality* of $\mathcal{S}$ is $\#\mathcal{S} = \sum_{x \in X} \chi_{\mathcal{S}}(x)$. $\mathcal{S}$ is also called *$\#\mathcal{S}$-multiset*.

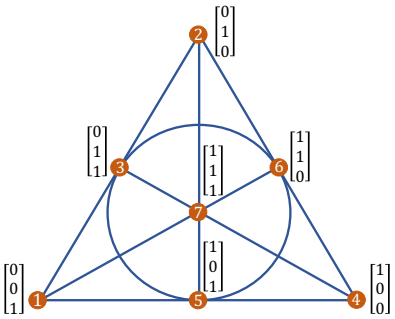

Figure 6.1: 7-multiset induced by binary $[7, 3]$ Simplex code (Fano plane).

Definition 15. *Let $\mathcal{V}$ be a vector space of dimension v over $\mathbb{F}_q$, $\mathcal{P}$ be a multiset of points p in $\mathrm{PG}(\mathcal{V})$ with characteristic function $\chi_{\mathcal{P}} : \mathrm{PG}(\mathcal{V}) \to \mathbb{N}$, and $\mathcal{H}$ denotes a hyperplane in $\mathrm{PG}(\mathcal{V})$. The restricted multiset $\mathcal{P} \cap \mathcal{H}$ is defined via its characteristic function as follows*

$$\chi_{\mathcal{P}\cap\mathcal{H}}(p) = \begin{cases} \chi_{\mathcal{P}}(p) & \text{if } p \in \begin{bmatrix}\mathcal{H}\\ 1\end{bmatrix}_q; \\ 0 & \text{otherwise.} \end{cases}$$

Then $\#(\mathcal{P} \cap \mathcal{H}) = \sum_{p\in\begin{bmatrix}\mathcal{H}\\ 1\end{bmatrix}_q} \chi_{\mathcal{P}}(p)$.

Let $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ be the generator matrix of a linear $[n, k]_q$ code $\mathcal{C}$, a k-subspace of the n-dimensional vector space $\mathbb{F}_q^n$. Let $\mathbf{g}_i \in \mathbb{F}_q^k$, $i \in [n]$ be the ith column of $\mathbf{G}$. Suppose that none of the $\mathbf{g}_i$'s is $\mathbf{0}$. (The code $\mathcal{C}$ is said to be of full length.) Then each $\mathbf{g}_i$ determines a point in the projective space $\mathrm{PG}(k-1, q)$, and $\mathcal{G} := \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n\}$ is a set of n points in $\mathrm{PG}(k-1, q)$ if the $\mathbf{g}_i$ happen to be pair-wise independent. When dependence occurs, $\mathcal{G}$ is interpreted as a multiset and each point is counted with the appropriate multiplicity. In general, $\mathcal{G}$ is called n-multiset induced by $\mathcal{C}$.

For example, consider the $[7, 3]_2$ Simplex code. The columns of its generator matrix are the seven non-zero vectors of $\mathbb{F}_2^3$, and the seven points in the projective space $\mathrm{PG}(2, 2)$. Figure 6.1 shows the corresponding 7-multiset, known as the Fano plane. Since $k = 3$, the 7 lines of the $\mathrm{PG}(2, 2)$ are also the hyperplanes of this 2-dimensional projective space.

Proposition 5. *Different generator matrices of a code yield projectively equivalent codes. In other words, there exist a bijective correspondence between the equivalence classes of full-length q-ary linear codes and the projective equivalence classes of multisets in finite projective spaces.*

It should be noted that the importance of this correspondence lies in the fact that it relates the coding-theoretic properties of $\mathcal{C}$ to the geometric or the combinatorial properties of multiset $\mathcal{G}$.

Proposition 6. *Let $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ be the generator matrix of a linear $[n, k, d]_q$ code $\mathcal{C}$, and $\mathcal{G}$ be the n-multiset induced by code $\mathcal{C}$. The minimum distance d of code $\mathcal{C}$ is given by*

$$d = n - \max \#(\mathcal{G} \cap \mathcal{H}),$$

where $\mathcal{H}$ runs through all the hyperplanes of $\mathrm{PG}(k-1, q)$.

Proof. For an arbitrary non-zero row vector $\mathbf{a} = [a_1, \cdots, a_k]$ of dimension k, the Hamming weight of codeword $\mathbf{aG} \in \mathcal{C}$ is given by

$$\mathrm{w}(\mathbf{aG}) = n - \#\{j \in [n]; \mathbf{ag}_j = 0\} = n - \#(\mathcal{G} \cap \mathcal{A}),$$

where $\mathcal{A}$ is a hyperplane in $\mathrm{PG}(k-1, q)$ with equation $a_1x_1 + \cdots + a_kx_k = 0$. Thus, the codeword with minimum Hamming weight is resulted from a hyperplane $\mathcal{H}$ in $\mathrm{PG}(k-1, q)$ with maximum $\#(\mathcal{G} \cap \mathcal{H})$. The proof is completed considering the fact that the minimum distance of a code is equal to the minimum Hamming weight of its nonzero codewords. □

Example 11. Consider the k-dimensional simplex code $\mathcal{C}$ over $\mathbb{F}_q$. In $\mathrm{PG}(k-1, q)$, the multiset $\mathcal{G}$ induced by code $\mathcal{C}$ has $\begin{bmatrix} k \\ 1 \end{bmatrix}_q$ points, and all hyperplanes contain $\begin{bmatrix} k-1 \\ 1 \end{bmatrix}_q$ points.

Thus, as an immediate consequence of Proposition 6 and its proof, every non-zero codeword of the corresponding linear code has a Hamming weight of q^{k-1}, which indicates that the minimum distance of code $\mathcal{C}$ is q^{k-1}. Let $\mathcal{H}$ be an arbitrary hyperplane in $\mathrm{PG}(k-1,q)$ and $\mathcal{P}$ be the set of all q^{k-1} points of $\mathbb{F}_q^k$ that are not contained in $\mathcal{H}$. The corresponding code of $\mathcal{P}$ is known as a k-dimensional first order Reed-Muller code or as an affine k-dimensional simplex code.

6.2.4 First Order Reed-Muller (RM) Codes

A k-dimensional binary first-order Reed-Muller code $\mathrm{RM}_2(1,k-1)$ with parameter $k \geq 2$, is a linear $[2^{k-1},k]$ code [124–126]. RM codes are important in both theory and practice.

For a given k, one way of obtaining this code is to evaluate all multilinear polynomials with the binary coefficients, $k-1$ variables and the total degree of one on the elements of $\mathbb{F}_2^{k-1}$. The encoding polynomial for $\mathrm{RM}_2(1,k-1)$ can be written as $c_1 + c_2 \cdot Z_1 + c_3 \cdot Z_2 + \cdots + c_k \cdot Z_{k-1}$ where $Z_1,\ldots,Z_{k-1}$ are the $k-1$ variables, and $c_1,\ldots,c_k$ are the binary coefficients of this polynomial. Indeed, the data symbols $f_1,\ldots,f_k$ are used as the coefficients of the encoding polynomial, and the codeword symbols are obtained by evaluating the encoding polynomial on all vectors $(Z_1,\ldots,Z_{k-1}) \in \mathbb{F}_2^{k-1}$.

Another way of describing k-dimensional binary first order Reed-Muller codes, i.e., $\mathrm{RM}_2(1,k-1)$ is based on the generator matrix which can be constructed as follows. Let us write the set of all $(k-1)$-dimensional binary vectors as $\mathcal{X} = \mathbb{F}_2^{k-1} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ where $n = 2^{k-1}$ and for $i \in [n]$, $\mathbf{x}_i = (x_{i_{k-1}},\ldots,x_{i_1})$ with $x_{i_j} \in \mathbb{F}_2$, $j \in [k-1]$. For any $\mathcal{A} \subseteq \mathcal{X}$, define the indicator vector $\mathbb{I}_{\mathcal{A}} \in \mathbb{F}_2^{k-1}$ as follows,

$$(\mathbb{I}_{\mathcal{A}})_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \in \mathcal{A}; \\ 0 & \text{otherwise.} \end{cases}$$

For the k rows of the generator matrix of $\text{RM}_2(1, k-1)$, define k row vectors of length 2^{k-1} as follows, $\mathbf{r}_0 = (1, \ldots, 1)$ and $\mathbf{r}_j = \mathbb{I}_{\mathcal{H}_j}$, $j \in [k-1]$, where $\mathcal{H}_j = \{\mathbf{x}_i \in \mathcal{X} \mid x_{i_j} = 0\}$. The set $\{\mathbf{r}_{k-1}, \ldots, \mathbf{r}_1, \mathbf{r}_0\}$ defines the rows of a non-systematic generator matrix of the $\text{RM}_2(1, k-1)$. For a systematic generator matrix of the $\text{RM}_2(1, k-1)$, the set of rows $\{\mathbf{r}_{k-1}, \ldots, \mathbf{r}_1, \sum_{i=0}^{k-1} \mathbf{r}_i\}$ can be considered.

Example 12. Consider $\text{RM}_2(1, 3)$ which is an $[8, 4, 4]_2$ code. We first define the set $\mathcal{X}$ as follows

$$\mathcal{X} = \mathbb{F}_2^3 = \{(0,0,0), (0,0,1), \ldots, (1,1,1)\} = \{\mathbf{x}_1, \ldots, \mathbf{x}_8\}$$

It then follows that $\mathcal{H}_3 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ that gives $\mathbf{r}_3 = (1,1,1,1,0,0,0,0)$, and $\mathcal{H}_2 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5, \mathbf{x}_6\}$ which gives $\mathbf{r}_2 = (1,1,0,0,1,1,0,0)$, and $\mathcal{H}_1 = \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_7\}$ which results $\mathbf{r}_1 = (1,0,1,0,1,0,1,0)$. Let $\mathbf{r}_0$ be the all-one row vector of dimension eight. The set $\{\mathbf{r}_3, \mathbf{r}_2, \mathbf{r}_1, \mathbf{r}_0\}$ defines the rows of a non-systematic generator matrix of the $\text{RM}_2(1, 3)$ as follows

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Also, $\sum_{i=0}^{3} \mathbf{r}_i = (0,1,1,0,1,0,0,1)$, and a systematic generator matrix of $\text{RM}_2(1, 3)$ is given by

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

6.3 Geometric View on Service Rate of Codes

This section uses the geometric description of linear codes. For a linear code $\mathcal{C}$ with generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$, we consider the n-multiset $\mathcal{G}$ induced by $\mathcal{C}$ in $\mathrm{PG}(k-1,q)$ with the characteristic function $\chi_{\mathcal{G}}$ as defined in the section 6.2.3. Thus, each point $p \in \mathrm{PG}(k-1,q)$ has a certain multiplicity $\chi_{\mathcal{G}}(p) \in \mathbb{N}$. In this language, the reduced recovery sets are subsets of $\mathcal{G}$ such that each point can be taken once in a reduced recovery set. Also, the service rate of each point p, denoted by $\mu(p)$, can be defined as the sum of the service rates of the nodes (columns of $\mathbf{G}$) corresponding to the point p. Based on this definition, $\mu(p) = \sum_{l\in\mathcal{L}_p} \mu_l$ where $\mathcal{L}_p$ is the set of nodes that correspond to the same point $p \in \mathrm{PG}(k-1,q)$. Since $\#\mathcal{L}_p = \chi_{\mathcal{G}}(p)$, if all nodes in the set $\mathcal{L}_p$ have the same service rate, say μ_p, then we have $\mu(p) = \chi_{\mathcal{G}}(p)\cdot\mu_p$.

Lemma 19. *Let $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ be the generator matrix of an $[n,k]_q$ code $\mathcal{C}$, and $\mathcal{G}$ be the n-multiset induced by code $\mathcal{C}$ with service rate $\mu(p)$ of each point $p \in \mathrm{PG}(k-1,q)$. If for some $i \in [k]$, $s\cdot\mathbf{e}_i \in \mathcal{S}(\mathbf{G},\boldsymbol{\mu})$ and a hyperplane $\mathcal{H}$ of $\mathrm{PG}(k-1,q)$ is not containing $\mathbf{e}_i$, then we have*

$$s \le \sum_{p\in\mathrm{PG}(k-1,q)\setminus\mathcal{H}} \mu(p).$$

Proof. Since $s\cdot\mathbf{e}_i \in \mathcal{S}(\mathbf{G},\boldsymbol{\mu})$, it means that the request rate of s for file f_i is satisfied by the coded storage system. Whatever the used recovery sets for file f_i are, some points outside of $\mathcal{H}$ have to be used since the points in $\mathcal{H}$ are not able to generate $\mathbf{e}_i$. Thus, replacing each recovery set in $\mathcal{R}_i$ by an arbitrary contained point outside of hyperplane $\mathcal{H}$, completes the proof. □

Corollary 7. *Let $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ be the generator matrix of a linear $[n,k,d]_q$ code $\mathcal{C}$ with service rate $\mu_l = 1$ of all nodes $l \in [n]$, and $\mathcal{G}$ be the n-multiset induced by code $\mathcal{C}$. If for all $i \in [k]$, $s\cdot\mathbf{e}_i \in \mathcal{S}(\mathbf{G},\boldsymbol{\mu})$, then the minimum distance d of code $\mathcal{C}$ is at least $\lceil s\rceil$.*

Proof. Since for all $i \in [k]$, $s \cdot \mathbf{e}_i \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ holds, this means that for all files f_i, $i \in [k]$, the request rate of s can be satisfied by the coded storage system. Thus, if we consider any hyperplane $\mathcal{H}$ in $\mathrm{PG}(k-1,q)$, it does not contain at least one of the $\mathbf{e}_i$'s for $i \in [k]$. In the special case of unit service rate of all servers, based on Lemma 19 results in

$$s \leq \#\left(\mathcal{G} \backslash \mathcal{H}\right) := \#\mathcal{G} - \#\left(\mathcal{G} \cap \mathcal{H}\right) = n - \#\left(\mathcal{G} \cap \mathcal{H}\right).$$

Since for every hyperplane $\mathcal{H}$ in $\mathrm{PG}(k-1,q)$, $s \leq n - \#\left(\mathcal{G} \cap \mathcal{H}\right)$ holds, according to the Proposition 6 and based on the fact that the minimum distance d is integer, we have $\lceil s \rceil \leq d$. □

Corollary 8. *Let $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ be the generator matrix of a linear $[n,k]_q$ code $\mathcal{C}$, and $\mathcal{G}$ be the n-multiset induced by code $\mathcal{C}$ with service rate $\mu(p)$ of each point $p \in \mathrm{PG}(k-1,q)$. Let $\mathcal{I} \subseteq [k]$. If for all $i \in \mathcal{I}$, there exist $s_i \in \mathbb{R}_{\geq 0}$ such that $\sum_{i \in \mathcal{I}} s_i \cdot \mathbf{e}_i \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ and a hyperplane $\mathcal{H}$ of $\mathrm{PG}(k-1,q)$ which is not containing $\mathbf{e}_i$ for all $i \in \mathcal{I}$, then we have*

$$s \leq \sum_{p \in \mathrm{PG}(k-1,q) \backslash \mathcal{H}} \mu(p).$$

where $s = \sum_{i \in \mathcal{I}} s_i$.

Proof. Since $\sum_{i \in \mathcal{I}} s_i \cdot \mathbf{e}_i \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$, based on Lemma 18, $s_i \cdot \mathbf{e}_i \in \mathcal{S}(\mathbf{G}, \boldsymbol{\mu})$ holds for all $i \in \mathcal{I}$. On the other hand, the hyperplane $\mathcal{H}$ of $\mathrm{PG}(k-1,q)$ does not contain any $\mathbf{e}_i$ for all $i \in \mathcal{I}$. Thus, by applying Lemma 19 for each $i \in \mathcal{I}$, we get $s_i \leq \sum_{p \in \mathrm{PG}(k-1,q) \backslash \mathcal{H}} \mu(p)$. Summing up all these inequalities gives

$$s = \sum_{i \in \mathcal{I}} s_i \leq \sum_{p \in \mathrm{PG}(k-1,q) \backslash \mathcal{H}} \mu(p).$$

□

It should be noted that Corollary 8 enables us to derive upper bounds on the service rate of the first order Reed-Muller codes and simplex codes. In what follows, without loss of generality, we assume that the service rate of all servers in the coded storage system is 1, i.e., $\mu_l = 1$ for all $l \in [n]$. Thus, by this assumption, the service rate region of a code only depends on the generator matrix $\mathbf{G}$ of the code and can be denoted by $\mathcal{S}(\mathbf{G})$.

6.4 Service Rate Region of Simplex Codes

In this section, by leveraging a novel geometric approach, we characterize the service rate region of the binary simplex codes which are special rate-optimal subclass of availability codes that are known as an important family of distributed storage codes. As we will show, the determined service rate region coincides with the region derived in [73, Theorem 1].

Theorem 15. *For each integer $k \geq 1$, the service rate region of the k-dimensional binary simplex code $\mathcal{C}$, which is a linear $[2^k - 1, k, 2^{k-1}]_2$ code with generator matrix $\mathbf{G}$ is given by*

$$\mathcal{S}(\mathbf{G}) = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0} \, : \, \sum_{i=1}^{k} \lambda_i \leq 2^{k-1} \right\}.$$

Proof. Note that the simplex code is projective. Since the projective space $\mathrm{PG}(k-1, 2)$ contains exactly $2^k - 1$ points, the generator matrix $\mathbf{G}$ consists of all non-zero vectors of $\mathbb{F}_2^k$. (Up to column permutations the generator matrix is unique.) Given an arbitrary $i \in [k]$, we partition the columns of $\mathbf{G}$ into $\mathbf{e}_i$ and $\{\mathbf{x}, \mathbf{x} + \mathbf{e}_i\}$ for all $2^{k-1} - 1$ non-zero vectors $\mathbf{x} \in \mathbb{F}_2^k$ with ith coordinate being equal to zero. Thus, for all $i \in [k]$, $2^{k-1} \cdot \mathbf{e}_i \in \mathcal{S}(\mathbf{G})$. Let $\mathbf{v}_i = 2^{k-1} \cdot \mathbf{e}_i$ for $i \in [k]$. Since $\mathcal{J} = \{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\} \subseteq \mathcal{S}(\mathbf{G})$, based on Lemma 18 and Proposition 4, the $\mathrm{conv}(\mathcal{J})$ is contained in $\mathcal{S}(\mathbf{G})$, i.e.,

$$\mathcal{S}(\mathbf{G}) \supseteq \left\{ \boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0} \, : \, \sum_{i=1}^{k} \lambda_i \leq 2^{k-1} \right\}$$

For the other direction, we consider the hyperplane $\mathcal{H}$ given by $\sum_{i=1}^{k} x_i = 0$, which does not contain any unit vector $\mathbf{e}_i$. Thus, for any demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region, the Corollary 8 results in $\sum_{i=1}^{k} \lambda_i \leq 2^{k-1}$. The reason is that half of the vectors in $\mathbb{F}_2^k$ which are the columns of $\mathbf{G}$ and so the elements of $\mathcal{G}$, are not contained in $\mathcal{H}$. □

6.5 Service Rate Region of Reed-Muller Codes

This section seeks to characterize the service rate region of the $\text{RM}_2(1, k-1)$ code with a non-systematic and a systematic generator matrix $\mathbf{G}$ constructed as described in section 6.2.4.

6.5.1 Non-Systematic First Order Reed-Muller Codes

Theorem 16. *For each integer $k \geq 2$, the service rate region of first order Reed-Muller code $RM_2(1, k-1)$ (or binary affine k-dimensional simplex code) with a non-systematic generator matrix $\mathbf{G}$ constructed as described in section 6.2.4, if $k \in \{2, 3\}$ is given by*

$$S(\mathbf{G}) = \left\{ \lambda \in \mathbb{R}^k_{\geq 0} \,:\, \sum_{i=1}^{k} \lambda_i \leq 2^{k-2} \right\} = \text{conv}\left(\{\mathbf{0}, \mathbf{v}_1, \ldots, \mathbf{v}_k\}\right)$$

and if $k \geq 4$, is given by

$$\begin{aligned} S(\mathbf{G}) &= \left\{ \lambda \in \mathbb{R}^k_{\geq 0} \,:\, \sum_{i=1}^{k} \lambda_i \leq 2^{k-2}, \sum_{i=1}^{k-1} \lambda_i + \frac{3}{2}\lambda_k - 1 \leq 2^{k-2} \right\} \\ &= \text{conv}\left(\{\mathbf{0}, \mathbf{v}_1, \ldots, \mathbf{v}_{k-1}, \mathbf{u}_k, \mathbf{w}_1, \ldots, \mathbf{w}_{k-1}\}\right), \end{aligned}$$

where $\mathbf{v}_i = 2^{k-2} \cdot \mathbf{e}_i$ for $i \in [k]$ and $\mathbf{w}_j = (2^{k-2} - 2) \cdot \mathbf{e}_j + 2 \cdot \mathbf{e}_k$ for $j \in [k-1]$. Also, $\mathbf{u}_k = \frac{2^{k-1}+2}{3} \cdot \mathbf{e}_k$.

Proof. The proof consists of a converse and an achievability.

Converse: The unit vector $\mathbf{e}_i$ for all $i \in [k-1]$ is not a column of $\mathbf{G}$ which means that file f_i does not have any systematic recovery set. Therefore, for file f_i, $i \in [k-1]$, all recovery sets have cardinality at least two, and the minimum system capacity utilized by λ_i, $i \in [k-1]$, is $2\lambda_i$. For file f_k, the cardinality of every reduced recovery set is odd since all columns of generator matrix $\mathbf{G}$ has one in the last row. Hence, for file f_k, the unit vector $\mathbf{e}_k$ that is a column of $\mathbf{G}$, forms a systematic recovery set of cardinality one, while all other recovery sets have cardinality at least three. Hence, the minimum capacity used by $\lambda_k \geq 1$ is $1 + 3(\lambda_k - 1)$. Since the system has 2^{k-1} servers, each of service rate (capacity) 1, based on the capacity constraints, the total capacity utilized by the requests for download must be less than 2^{k-1}. Thus, any vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region must satisfy the following valid constraint,

$$\sum_{i=1}^{k-1} \lambda_i + \frac{3}{2}\lambda_k - 1 \leq 2^{k-2} \tag{6.5}$$

Consider the hyperplane $\mathcal{H}$ given by $\sum_{i=1}^{k} x_i = 0$, that does not contain any unit vector $\mathbf{e}_i$. The columns of generator matrix $\mathbf{G}$ and so the elements of $\mathcal{G}$ which are not contained in $\mathcal{H}$, are the vectors in $\mathbb{F}_2^k$ with one in the last coordinate that satisfy $\sum_{i=1}^{k-1} x_i = 0$. It is easy to see that there are 2^{k-2} such vectors. Thus, applying Corollary 8 for hyperplane $\mathcal{H}$ impose another valid constraint as follows that any demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region must satisfy,

$$\sum_{i=1}^{k} \lambda_i \leq 2^{k-2} \tag{6.6}$$

Note that for $\lambda_k < 2$, the Inequality (6.6) is tighter than (6.5), while for $\lambda_k > 2$ Inequality (6.5) is tighter than (6.6). This means that for $k \in \{2, 3\}$ Inequality (6.5) is redundant.

Achievability: For the other direction, we will provide solutions (constructions) for the vertices of the corresponding polytope as follows. Let $\mathcal{R}' \subseteq \mathbb{F}_2^k$, $|\mathcal{R}'| = 2^{k-1}$ be

the set of columns of $\mathbf{G}$ with one in the last coordinate. For all $i \in [k-1]$, consider all the 2^{k-2} vectors $\mathbf{x} \in \mathcal{R}'$ with zero in the ith coordinate, then $\mathbf{x}+\mathbf{e}_i \in \mathcal{R}'$, and so $\{\mathbf{x}, \mathbf{x}+\mathbf{e}_i\}$ constitutes a recovery set of cardinality two for file f_i. Thus, for each file f_i, $i \in [k-1]$, the columns of $\mathbf{G}$ can be partitioned into 2^{k-2} pairs $\{\mathbf{x}, \mathbf{x}+\mathbf{e}_i\}$ which determines 2^{k-2} disjoint recovery sets for file f_i, $i \in [k-1]$. Therefore, the demand vectors $2^{k-2} \cdot \mathbf{e}_i$ for all $i \in [k-1]$ can be satisfied, i.e., $2^{k-2} \cdot \mathbf{e}_i \in S(\mathbf{G})$. For file f_k, there are exactly one systematic recovery set of cardinality one which is the column $\mathbf{e}_k$ of $\mathbf{G}$, and $(2^{k-1}-1).(2^{k-1}-2)/6$ recovery sets of cardinality three which are the sets $\{\mathbf{x}, \mathbf{x}', \mathbf{x}+\mathbf{x}'+\mathbf{e}_k\}$ for all pairs $\mathbf{x}, \mathbf{x}' \in \mathcal{R}' \setminus \mathbf{e}_k$. Note that for $k=2$, according to Inequality (6.6), one can readily confirm that $\lambda_k \leq 1$. Thus, for $k=2$ the systematic recovery set of file f_k can be utilized for satisfying the demand vector $1 \cdot \mathbf{e}_k$. For $k \geq 3$, it should be noted that that each column $\mathbf{x} \in \mathcal{R}' \setminus \mathbf{e}_k$ is contained in exactly $(2^{k-1}-2)/2$ recovery sets of file f_k of cardinality three. Since the capacity of each node is one, from each recovery set the request rate of $1/(2^{k-2}-1)$ can be satisfied without violating the capacity constraints. Thus, the demand vector $\frac{2^{k-1}+2}{3} \cdot \mathbf{e}_k$ can be satisfied. For the remaining part, we consider $k \geq 4$. Let $i, j \in [k-1]$ with $i \neq j$ be arbitrary. With this $\{\mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_k\}$ and $\{\mathbf{e}_j+\mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_j+\mathbf{e}_k\}$ are two of 2^{k-2} recovery sets of cardinality two for file f_i. Thus, the elements in $\mathcal{R}' \setminus \{\mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_k, \mathbf{e}_j+\mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_j+\mathbf{e}_k\}$ can be partitioned into $2^{k-2}-2$ recovery sets for file $f_i, i \in [k-1]$. Also, the sets $\{\mathbf{e}_k\}$ and $\{\mathbf{e}_i+\mathbf{e}_k, \mathbf{e}_j+\mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_j+\mathbf{e}_k\}$ can be utilized as two disjoint recovery sets for file f_k. Thus, the demand vector $\left(2^{k-2}-2\right) \cdot \mathbf{e}_i + 2 \cdot \mathbf{e}_k$ can be satisfied. □

6.5.2 Systematic First Order Reed-Muller Codes

Theorem 17. *For each integer $k \geq 2$, the service rate region of first order Reed-Muller code $RM_2(1, k-1)$ (or binary affine k-dimensional simplex code) with a systematic generator matrix $\mathbf{G}$ constructed as described in section* 6.2.4*, if $k=2$ is given by*

$$\mathcal{S}(\mathbf{G}) = \left\{\lambda \in \mathbb{R}^k_{\geq 0} \,:\, \lambda_1 \leq 1, \lambda_2 \leq 1\right\} = \mathrm{conv}\left(\mathbf{0}, \mathbf{e}_1 + \mathbf{e}_2\right)$$

if $k = 3$*, is given by*

$$\mathcal{S}(\mathbf{G}) = \left\{\lambda \in \mathbb{R}^k_{\geq 0} \,:\, -\lambda_i + \textstyle\sum_{j=1}^{3} \lambda_j \leq 2, \forall i \in [k]\right\} = \mathrm{conv}\left(\mathbf{0}, 2 \cdot \mathbf{e}_1, 2 \cdot \mathbf{e}_2, 2 \cdot \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3\right)$$

if $k = 4$*, is given by*

$$\mathcal{S}(\mathbf{G}) = \left\{\lambda \in \mathbb{R}^k_{\geq 0} \,: -\lambda_i + \sum_{j=1}^{k} \lambda_j \leq 4, 2\lambda_i + \sum_{j=1}^{k} \lambda_j \leq 10 \,\forall i \in [k]\right\}$$
$$= \mathrm{conv}\left(\mathbf{0}, \mathbf{p}_i \,\forall i \in [k], \mathbf{q}_{i,j} \,\forall i, j \in [k] \textit{ with } i \neq j, \tfrac{4}{3} \cdot \mathbf{1}\right)$$

and if $k \geq 5$*,* $\mathcal{S}(\mathbf{G})$ *lies inside the region given by*

$$\mathcal{S}(\mathbf{G}) \subseteq \left\{\lambda \in \mathbb{R}^k_{\geq 0} \,:\, \sum_{i \in [k] \setminus \mathcal{S}} \lambda_i + \sum_{j \in \mathcal{S}} (3\lambda_j - 2) \;\leq\; 2^{k-1} \,\forall \mathcal{S} \subseteq [k]\right\}.$$

where $\mathbf{p}_i = \frac{10}{3} \cdot \mathbf{e}_i$ *and* $\mathbf{q}_{i,j} = 3 \cdot \mathbf{e}_i + 1 \cdot \mathbf{e}_j$ *for* $i, j \in [k]$.

Proof. Based on the construction described in section 6.2.4 for a systematic generator matrix $\mathbf{G}$ of the $\mathrm{RM}_2(1, k-1)$, it can be seen that the number of ones in each column of $\mathbf{G}$ is odd, and the constructed systematic generator matrix, up to column permutations, is unique. Let the columns of $\mathbf{G}$ which are the set of vectors in $\mathbb{F}_2^k$ with odd number of ones, be denoted by $\mathcal{R}' \subseteq \mathbb{F}_2^k$, $|\mathcal{R}'| = 2^{k-1}$.

Converse: For an arbitrary file f_i, $i \in [k]$, the unit vector $\mathbf{e}_i$ is a column of $\mathbf{G}$ that forms a systematic recovery set of cardinality one, while all other recovery sets have cardinality at least three. The proof is based on the contradiction approach. Let $\mathbf{x}, \mathbf{x}' \in \mathcal{R}' \setminus \mathbf{e}_i$. Assume that $\{\mathbf{x}, \mathbf{x}'\}$ forms a recovery set of cardinality two for file f_i, i.e., $\mathbf{x} + \mathbf{x}' = \mathbf{e}_i$. Since both $\mathbf{x}$ and $\mathbf{x}'$ have an odd number of ones, their sum must have an even number of ones which is a contradiction. Indeed, for all pairs $\mathbf{x}, \mathbf{x}' \in \mathcal{R}' \setminus \mathbf{e}_i$, the set $\{\mathbf{x}, \mathbf{x}', \mathbf{x} + \mathbf{x}' + \mathbf{e}_i\}$

forms a recovery set of cardinality three for file f_i, $i \in [k]$. Thus, if $\lambda_i \leq 1$, the requests for file f_i can be fully satisfied by the systematic recovery set $\{\mathbf{e}_i\}$ and the system capacity utilized by λ_i is λ_i. However, for $\lambda_i \geq 1$, the system capacity utilized by λ_i is at least $1 + 3(\lambda_i - 1) = 3\lambda_i - 2$. Since the system has 2^{k-1} servers of capacity 1, any vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region must satisfy:

$$\sum_{i \in [k] \setminus \mathcal{S}} \lambda_i + \sum_{j \in \mathcal{S}} (3\lambda_j - 2) \leq 2^{k-1} \quad \forall \mathcal{S} \subseteq [k] \tag{6.7}$$

Applying Corollary 8 on all hyperplanes $\mathcal{H}_j$, $j \in [k]$, given by $\sum_{i \in [k] \setminus j} x_i = 0$, where each hyperplane $\mathcal{H}_j$, $j \in [k]$ does not contain any unit vectors $\mathbf{e}_i$, $i \in [k] \setminus j$, yields another set of valid constraints on any demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ in the service rate region as follows:

$$\sum_{i \in [k] \setminus j} \lambda_i \leq 2^{k-2} \quad \forall j \in [k] \tag{6.8}$$

Note that for $k \in \{2, 3\}$, Inequality (6.8) is tighter than (6.7). For $k = 2$, Inequality (6.8) gives $\lambda_1 \leq 1$ and $\lambda_2 \leq 1$. For $k = 3$, Inequality (6.8) gives $\sum_{i=1}^{3} \lambda_i - \lambda_i \leq 2$ for all $i \in [3]$. Summing up these three inequalities and dividing them by two, results $\sum_{i=1}^{3} \lambda_i \leq 3$. For $k = 4$, Inequality (6.8) yields $\sum_{i=1}^{4} \lambda_i - \lambda_i \leq 4$ for all $i \in [4]$. Summing up these four inequalities and dividing by three gives $\sum_{i=1}^{4} \lambda_i \leq \frac{16}{3}$. Also, for $k = 4$, Inequality (6.7) gives a set of constraints, among which the constraints $\sum_{i=1}^{4} \lambda_i + 2 \cdot \lambda_i \leq 10$ for all $i \in [4]$, are tighter than the ones already obtained from (6.8) in some region. For $k \geq 5$, Inequality (6.7) is always tighter than (6.8).

Achievability: For $k \leq 4$, we have to provide constructions for the vertices of the corresponding polytope. As discussed, for each file f_i, with $i \in [k]$, there are one systematic recovery set of cardinality one which is the column $\mathbf{e}_i$ of $\mathbf{G}$, and $(2^{k-1} - 1).(2^{k-1} - 2)/6$ recovery sets of cardinality three which are the sets of the form $\{\mathbf{x}, \mathbf{x}', \mathbf{x} + \mathbf{x}' + \mathbf{e}_i\}$ for all

pairs $\mathbf{x}, \mathbf{x}' \in \mathcal{R}' \setminus \mathbf{e}_i$. For $k = 2$, the two disjoint recovery sets $\{\mathbf{e}_1\}$ and $\{\mathbf{e}_2\}$, which are the only recovery sets for files f_1 and f_2, respectively, can be used to satisfy the demand vector $\mathbf{e}_1 + \mathbf{e}_2$. Now, consider $k \geq 3$. Since each column $\mathbf{x} \in \mathcal{R}' \setminus \mathbf{e}_i$ is contained in exactly $(2^{k-1} - 2)/2$ recovery sets of file f_i, $i \in [k]$ of cardinality three, and the capacity of each node is one, from each recovery set the request rate of $1/(2^{k-2} - 1)$ can be satisfied without violating the capacity constraints. Thus, the demand vector $\frac{2^{k-1}+2}{3} \cdot \mathbf{e}_i$ for all $i \in [k]$ can be satisfied. This means that for $k = 3$ and $k = 4$, respectively the demand vectors $2 \cdot \mathbf{e}_i$ for all $i \in [3]$, and $\frac{10}{3} \cdot \mathbf{e}_i$ for all $i \in [4]$ can be satisfied. Also, for $k = 3$, the demand vector $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$ can be achieved by the disjoint systematic recovery sets $\{\mathbf{e}_1\}$, $\{\mathbf{e}_2\}$, and $\{\mathbf{e}_3\}$. Now, assume $k \geq 4$. Let $i, j \in [k]$ with $i \neq j$ be arbitrary. The systematic recovery sets $\{\mathbf{e}_i\}$ and $\{\mathbf{e}_j\}$ can be used for files f_i and f_j, respectively. Additionally, consider all the $(2^{k-2} - 1).(2^{k-1} - 4)/3$ recovery sets $\{\mathbf{x}, \mathbf{x}', \mathbf{x} + \mathbf{x}' + \mathbf{e}_i\}$ of cardinality three for file f_i that do not contain $\mathbf{e}_j$, each of which can satisfy the request rate of $1/(2^{k-2} - 2)$ for file f_i without violating the capacity constraints. Thus, the demand vector $\frac{2^{k-1}+1}{3} \cdot \mathbf{e}_i + 1 \cdot \mathbf{e}_j$ can be achieved. Therefore, for $k = 4$ the demand vector $3 \cdot \mathbf{e}_i + 1 \cdot \mathbf{e}_j$ for all $i, j \in [k]$ with $i \neq j$ can be satisfied. For achieving the demand vector $\frac{4}{3} \cdot \mathbf{1}$, one can use all the systematic recovery sets $\{\mathbf{e}_1\}$, $\{\mathbf{e}_2\}$, $\{\mathbf{e}_3\}$, $\{\mathbf{e}_4\}$ with capacity 1. Moreover, the remaining four columns can be used to build up four recovery sets consisting of a unique recovery set of cardinality 3 for each file f_i, $i \in [4]$, and from each of these recovery sets the rate of $\frac{1}{3}$ can be satisfied. This completes the proof. □

6.6 Examples of Service Rate Region

6.6.1 Binary $[7, 3]$ Simplex code

In this section, as an example, we show how the service rate region of the $[7, 3]$ Simplex code is determined using the geometric approach. Consider a storage system using the $[7, 3]$ Simplex code. W.l.o.g, assume that $\mu = 1$. Let (x_1, x_2, x_3) denote a generic non-

zero vector in $\mathbb{F}_2^3$. Observe that the hyperplane $\mathcal{H}$ given by $\sum_{i=1}^{3} x_i = 0$ (namely, the hyperplane containing the points $(0,1,1)$, $(1,0,1)$ and $(1,1,0)$ in the Fano plane depicted in Figure 6.1) does not contain any unit vector $\mathbf{e}_i$, $i \in \{1,2,3\}$. Thus, for any demand vector $\boldsymbol{\lambda} = (\lambda_a, \lambda_b, \lambda_c)$ in the service rate region, applying Corollary 8 results in $\lambda_a + \lambda_b + \lambda_c \leq 4$. The reason is that the hyperplane $\mathcal{H}$ does not contain the points $(0,0,1)$, $(0,1,0)$, $(1,0,0)$ and $(1,1,1)$. Thus, so far we have shown that the service rate region is contained in the polytope $\mathcal{P} = \left\{\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^3 \,:\, \sum_{i=1}^{3} \lambda_i \leq 4\right\}$.

For the achievability proof, since the service rate region is a convex subset of $\mathbb{R}_{\geq 0}^3$, we only need to show that the vertices of the polytope $\mathcal{P}$, i.e., $(0,0,0)$, $(4,0,0)$, $(0,4,0)$ and $(0,0,4)$, are in the service rate region of this storage system. To see that, we observe that there are four disjoint recovery sets for each data object), and thus the request rate of 1 can be assigned to each of these recovery sets without violating the node capacity constraints. Thus, the service rate region of the $[7,3]$ Simplex code consists of all demand vectors $(\lambda_a, \lambda_b, \lambda_c)$ such that $\lambda_a + \lambda_b + \lambda_c \leq 4$.

6.6.2 Binary Non-Systematic $[8,4]$ First Order Reed-Muller code

Consider a system where four objects a, b, c, and d are stored across 8 servers using the first order Reed-Muller code $\text{RM}_2(1,3)$ with a non-systematic generator matrix as:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

which encodes $[a,b,c,d]$ into $[a{+}b{+}c{+}d, a{+}b{+}d, a{+}c{+}d, a{+}d, b{+}c{+}d, b{+}d, c{+}d, d]$. The recovery sets for objects a and d are shown in Figure 6.2 and Figure 6.3, respectively. The recovery sets for objects b and c can be obtained similarly to those for a.

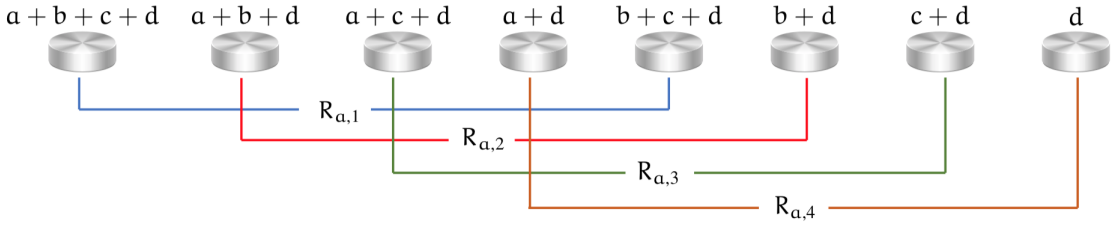

Figure 6.2: Recovery sets for data object a in the $[8, 4]$ Reed-Muller code.

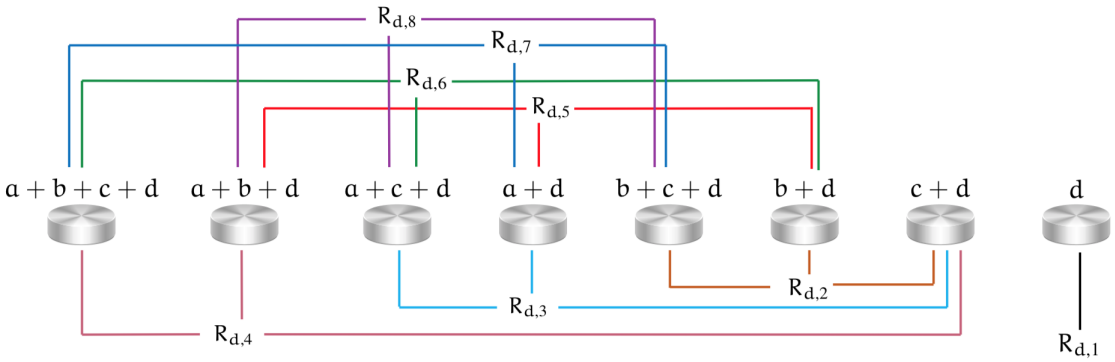

Figure 6.3: Recovery sets for data object d in the $[8, 4]$ Reed-Muller code.

Let $(x_1, \ldots, x_4)$ be a non-zero vector in $\mathbb{F}_2^4$. Observe that the hyperplane $\mathcal{H}$ given by $\sum_{i=1}^{4} x_i = 0$ does not contain any unit vector $\mathbf{e}_i$, $i \in [4]$. The hyperplane $\mathcal{H}$ does not contain the 4 column vectors $(1, 1, 0, 1)$, $(1, 0, 1, 1)$, $(0, 1, 1, 1)$ and $(0, 0, 0, 1)$ of the generator matrix. Thus, for any demand vector $\boldsymbol{\lambda} = (\lambda_a, \lambda_b, \lambda_c, \lambda_d)$ in the service rate region, applying the Corollary 8 results in the constraint below

$$\lambda_a + \lambda_b + \lambda_c + \lambda_d \leq 4. \tag{6.9}$$

On the other hand, we know that the unit vector $\mathbf{e}_i$ for all $i \in \{1, 2, 3\}$ is not a column of the generator matrix which means that files a, b, and c do not have any systematic recovery sets. Thus, for files a, b, and c, the cardinality of all recovery sets is at least two, and the minimum system capacity utilized by λ_i for $i \in \{a, b, c\}$ is $2\lambda_i$. For file d,

since all columns of the generator matrix have one in the last row, the cardinality of every recovery set is odd. Hence, for file d, the unit vector $\mathbf{e}_4$, which is a column of G, forms a recovery set of cardinality one, while all other recovery sets have cardinality at least three. Thus, the minimum system capacity utilized by λ_d for $\lambda_d \leq 1$ is λ_d and for $\lambda_d \geq 1$ is $1 + 3(\lambda_d - 1) = 3\lambda_d - 2$. Since the system has 8 servers, each of service capacity 1, based on the capacity constraints, the total capacity utilized by the requests for download must be at most 8. Thus, any vector $(\lambda_a, \lambda_b, \lambda_c, \lambda_d)$ in the service region must satisfy:

$$\begin{cases} 2(\lambda_a + \lambda_b + \lambda_c) + \lambda_d \leq 8 & \text{for } \lambda_d \leq 1 \\ 2(\lambda_a + \lambda_b + \lambda_c) + 3\lambda_d - 2 \leq 8 & \text{for } \lambda_d \geq 1 \end{cases} \tag{6.10}$$

So far, we showed that the service rate region lies inside the polytope $\mathcal{T}$ described as follows: $\mathcal{T} = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^4_{\geq 0} \, : \, \boldsymbol{\lambda} \text{ satisfies (6.5), (6.6)} \right\}$. Figure 6.4 depicts the service rate region of this storage scheme in the $\lambda_a - \lambda_d$ plane wherein (6.9) and (6.10) are respectively shown with the red line and the green line.

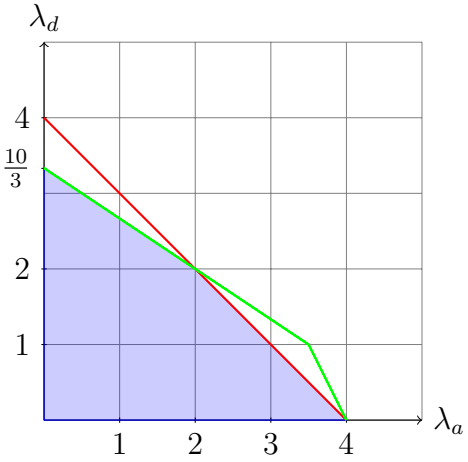

Figure 6.4: Service rate region of the $[8, 4]_2$ first order Reed-Muller code in $\lambda_a - \lambda_d$ plane with $\lambda_b = \lambda_c = 0$ where the constraints (6.9) and (6.10) are respectively shown with the red line and the green line.

For the achievability proof, one needs to provide constructions only for the vertices of polytope $\mathcal{T}$ in $\lambda_a - \lambda_d$ plane. The demand vector $(\lambda_a, \lambda_b, \lambda_c, \lambda_d) = (4, 0, 0, 0)$ can be achieved by assigning the request rate of 1 to each of the 4 disjoint recovery sets of file a shown in Figure 6.2. For the $(\lambda_a, \lambda_b, \lambda_c, \lambda_d) = (2, 0, 0, 2)$, the $\lambda_a = 2$ can be served by assigning the request rate of 1 to each of the recovery sets $(b + d, a + b + d)$ and $(b+c+d, a+b+c+d)$, and $\lambda_d = 2$ can be satisfied by assigning the request rate of 1 to the systematic recovery set (d), and the request rate 1 to the recovery set $(a+d, c+d, a+c+d)$ of file d. For the demand vector $(\lambda_a, \lambda_b, \lambda_c, \lambda_d) = (0, 0, 0, \frac{10}{3})$, the $\lambda_d = \frac{10}{3}$ can be served without violating the node capacity constraints by assigning the request rate of 1 to the systematic recovery set (d), and the request rate of $\frac{1}{3}$ to each of the 7 recovery sets of size 3 for file d, depicted in Figure 6.3.

7. STORAGE-EFFICIENT SCHEMES COVERING GIVEN RATE REGIONS*

7.1 Introduction

The explosive growth in the amount of data stored in the cloud data centers is accompanied by a rapid increase in the volume of users accessing it. A simple approach to handle these surging demands in a fast and reliable fashion is to replicate data at multiple storage nodes. However, replication can be expensive in terms of storage. Erasure codes have been shown to be effective in achieving various goals such as providing reliability against node failures (see e.g., [51–54]), ensuring availability of stored content during high demand (see e.g., [55–58]), enabling the recovery of a data object from multiple disjoint groups of nodes (see e.g., [59–61]), and providing fast content download (see e.g., [62–70]).

Serving a large number of users simultaneously is a major concern in cloud storage systems and so is considered as one of the most significant considerations in the design of coded distributed systems. The service rate region has been recently recognized as an important performance metric for coded distributed systems, which is defined as the set of all data access requests that can be simultaneously served by the system [1, 71–76]. It has been shown that erasure coding of data objects can increase the overall volume of the service rate region through handling skews in the request rates more flexibly [1, 73, 74].

The service rate problem considers a distributed storage system where k files $f_1, \ldots, f_k$ are stored across n servers using a linear $[n, k]_q$ code. The requests to download file f_i arrive at rate λ_i, and the service rate of each server is μ. A goal of the service rate problem is to determine the service rate region of this system which is the set of all request rates $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$ that can be handled by this system.

*Reprinted with permission from [77] "Efficient Storage Schemes for Desired Service Rate Regions," by F. Kazemi, S. Kurz, E. Soljanin, and A. Sprintson, 2020. In Proceedings of 2020 IEEE Information Theory Workshop (ITW), pp. 1-5, April 2021. Copyright © by IEEE.

Related Work

All the existing studies on the service rate problem focus on characterizing the service rate region of a given coded storage scheme and finding the optimal request allocation, that is, the optimal policies for splitting incoming requests across the nodes to maximize the service rate region (see [1]). In [73], the service rate region was characterized for MDS codes when $n \geq 2k$, binary simplex codes and systems with arbitrary n when $k = 2$. The service rate region of the systems with arbitrary n when $k = 3$ was determined in [74]. A connection between the service rate problem and the fractional matching problem is established in [75]. Also, it has been shown that the service rate problem can be viewed as a generalization of the multiset primitive batch codes problem. In [75], we characterized the service rate regions of the binary first order Reed-Muller codes and binary simplex codes using a novel geometric technique. Also, we showed that given the service rate region of a code, a lower bound on the minimum distance of the code can be derived.

Our Contributions

In this work, we consider a practical setting of designing a coded distributed storage system where we are asked to store k files redundantly across some number of storage nodes in the system. Also, we are given a bounded subset $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$ as a desired service rate region for this distributed storage system. Our goal is: 1) to find the minimum number of storage nodes $n(\mathcal{R})$ (or a lower bound on $n(\mathcal{R})$) required for serving all demand vectors $\boldsymbol{\lambda}$ inside the desired service rate region $\mathcal{R}$, and 2) to design the most storage-efficient redundancy scheme with the service rate region covering the set $\mathcal{R}$. In this work, we focus on designing the underlying erasure codes that cover a given service rate region with minimum storage. Towards this goal, we propose three different general lower bounds for $n(\mathcal{R})$. Also, we show that for $k = 2$, these bounds are tight and we design an efficient storage scheme that achieves the desired service rate region while minimizing the storage.

7.2 Problem Setup and Formulation

7.2.1 Basic Notation

Throughout this chapter, we denote vectors by bold-face lower-case letters and matrices by bold-face capital letters. Let $\mathbb{Z}_{\geq 0}$ and $\mathbb{N}$, respectively, denote the set of non-negative integers, and the set of positive integers. For $k \in \mathbb{N}$, let $\mathbf{0}_k$ and $\mathbf{1}_k$, respectively, denote the all-zero and all-one column vectors of length k. Let $\mathbf{e}_i$ be a unit vector of length k, having a one at position i and zeros elsewhere. For any $i \in \mathbb{N}$, we define $[i] \triangleq \{1, \ldots, i\}$. Let $\mathbb{F}_q$ be the finite field of order q, and $\mathbb{F}_q^n$ be the n-dimensional vector space over $\mathbb{F}_q$. Let $[n, k]_q$ denote a q-ary linear code $\mathcal{C}$ of length n and dimension k. We denote the cardinality of a set or multiset $\mathcal{S}$ by $\#\mathcal{S}$. Let $\langle \mathcal{S} \rangle$ and $\mathrm{conv}(\mathcal{S})$, respectively, denote the span and the convex hull of the set $\mathcal{S}$ of vectors. For two vectors $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$, let $\mathbf{x} \leq \mathbf{y}$ define $x_i \leq y_i$ for all $i \in [k]$.

7.2.2 Coded Storage System

Consider a coded storage system wherein k files $f_1, \ldots, f_k$ are stored redundantly across n servers using a linear code of length n and dimension k over $\mathbb{F}_q$ with generator matrix $\mathbf{G}$. Suppose all files are of the same size, and all servers have a storage capacity of one file. A set Y is a recovery set for file f_i if the unit vector $\mathbf{e}_i$ can be recovered through a linear combination of the columns of $\mathbf{G}$ indexed by the set Y, i.e., if there exist coefficients α_j's $\in \mathbb{F}_q$ such that $\sum_{j \in Y} \alpha_j \mathbf{g}_j = \mathbf{e}_i$ where $\mathbf{g}_j$ denotes the jth column of $\mathbf{G}$. For each file, w.l.o.g. we consider reduced recovery sets defined as the recovery sets that are not a proper superset of any other recovery sets for that file. In other words, the reduced recovery sets are obtained by considering non-zero coefficients α_j's and linearly independent columns $\mathbf{g}_j$'s. Let $\mathcal{Y}_i = \{Y_{i,1}, \ldots, Y_{i,t_i}\}$ denote the t_i recovery sets for file f_i.

We assume that the service rate of each server is μ, i.e., each server can resolve the received requests at the average rate μ. We further assume that the requests to download

file f_i arrive at rate λ_i, $i \in [k]$. The request arrival rates for the k files are denoted by the demand vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$. We consider the class of scheduling strategies that assign a fraction of requests for a file to each of its recovery sets. Let $\lambda_{i,j}$ be the portion of requests for file f_i that is assigned to the recovery set $Y_{i,j}$, $j \in [t_i]$.

7.2.3 Service Rate Region

The demand vector $\boldsymbol{\lambda}$ can be served by a coded distributed storage system with generator matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ and service rate μ iff there exists a set $\{\lambda_{i,j} : i \in [k], j \in [t_i]\}$, referred to as a valid allocation, that satisfies the following constraints:

$$\sum_{j=1}^{t_i} \lambda_{i,j} = \lambda_i, \qquad \text{for all} \quad i \in [k], \tag{7.1a}$$

$$\sum_{i=1}^{k} \sum_{\substack{j \in [t_i] \\ \ell \in Y_{i,j}}} \lambda_{i,j} \leq \mu, \quad \text{for all} \quad \ell \in [n], \tag{7.1b}$$

$$\lambda_{i,j} \in \mathbb{R}_{\geq 0}, \qquad \text{for all} \quad i \in [k],\ j \in [t_i]. \tag{7.1c}$$

The constraints (7.1a) guarantee that the demands for all files are served, and (7.1b) ensure that the total rate assigned to each server does no exceed its service rate.

The *service rate region* of an erasure coded storage system with the generator matrix $\mathbf{G}$ and service rate μ, denoted by $\mathcal{S}(\mathbf{G}, \mu) \subseteq \mathbb{R}_{\geq 0}^k$, is defined as the set of all demand vectors $\boldsymbol{\lambda}$ that can be served by the system. In what follows, w.l.o.g. we assume that $\mu = 1$ and abbreviate $\mathcal{S}(\mathbf{G}, 1)$ as $\mathcal{S}(\mathbf{G})$.

Note that there are several generator matrices that span the same linear code, i.e., whenever the row spans of two matrices $\mathbf{G}$ and $\mathbf{G}'$ coincide, they span the same code. However, the service rate regions of generator matrices $\mathbf{G}$ and $\mathbf{G}'$ of the same linear code might not be the same, i.e., $\mathcal{S}(\mathbf{G}) \neq \mathcal{S}(\mathbf{G}')$.

7.2.4 Geometric Description of Linear Codes

Here, we briefly review some preliminaries regarding the notions of projective space, multiset, and projective multisets induced by linear codes that we will use in Sec.7.2.5. For more details, see [121–123].

Definition 16. *For a vector space $\mathcal{V}$ of dimension v over $\mathbb{F}_q$, the projective space of $\mathcal{V}$, denoted as $\mathrm{PG}(\mathcal{V})$, is the set of equivalence classes of $\mathcal{V} \setminus \{\mathbf{0}_v\}$ under the equivalence relation $\sim$ defined as $x \sim y$ if there is a non-zero element $\alpha \in \mathbb{F}_q$ such that $x = \alpha y$.*

We remark that the 1-dimensional subspaces of $\mathcal{V}$ are the points of the projective space $\mathrm{PG}(\mathcal{V})$. The 2-dimensional subspaces of $\mathcal{V}$ are the lines of $\mathrm{PG}(\mathcal{V})$ and the $v-1$ dimensional subspaces of $\mathcal{V}$ are called the hyperplanes of $\mathrm{PG}(\mathcal{V})$.

For a vector space $\mathcal{V}$ of (*geometric*) dimension v over $\mathbb{F}_q$, the projective space $\mathrm{PG}(\mathcal{V})$ is also denoted by $\mathrm{PG}(v-1, q)$, referred to as the projective space of (*algebraic*) dimension $v-1$ over $\mathbb{F}_q$. This notion makes sense since up to isomorphism, the $\mathrm{PG}(\mathcal{V})$ only depends on the order q of the base field and the dimension v of the vector space $\mathcal{V}$. Thus, $\mathrm{PG}(v-1, q)$ can be defined as the set of v-tuples of elements of $\mathbb{F}_q$, not all zero, under the equivalence relation given by $(x_1, \cdots, x_v) \sim (\alpha x_1, \cdots, \alpha x_v)$, $\alpha \neq 0$, $\alpha \in \mathbb{F}_q$. The definition implies that if $(x_1, \cdots, x_v)$ is a point in $\mathrm{PG}(v-1, q)$, its scalar multiple (by any non-zero scalar $\alpha \in \mathbb{F}_q$) $(\alpha x_1, \cdots, \alpha x_v)$ is the same point in $\mathrm{PG}(v-1, q)$.

A *multiset*, unlike a set, allows for multiple instances for each of its elements. A multiset $\mathcal{S}$ on a base set $\mathcal{X}$ is defined with its characteristic function, denoted as $\chi_{\mathcal{S}} : \mathcal{X} \to \mathbb{N}$, mapping $x \in \mathcal{X}$ to the multiplicity of x in $\mathcal{S}$. The cardinality of the multiset $\mathcal{S}$ is computed as $\#\mathcal{S} = \sum_{x \in \mathcal{X}} \chi_{\mathcal{S}}(x)$. The multiset $\mathcal{S}$ is also called *$\#\mathcal{S}$-multiset*. As a simple example, consider the multiset $\mathcal{S} = \{a, a, b, b, b, c\}$ on the base set $\mathcal{X} = \{a, b, c\}$ that is identified with $\chi_{\mathcal{S}}(a) = 2$, $\chi_{\mathcal{S}}(b) = 3$ and $\chi_{\mathcal{S}}(c) = 1$.

Let $\mathbf{G}$ be the generator matrix of an $[n, k]_q$ code $\mathcal{C}$ that is a k-dimensional subspace of

the n-dimensional vector space $\mathbb{F}_q^n$. Let $\mathbf{g}_i$, $i \in [n]$ be the ith column of $\mathbf{G}$. Then, each $\mathbf{g}_i$ is a point in the projective space $\mathrm{PG}(k-1,q)$, and $\mathcal{G} := \{g_1, g_2, \ldots, g_n\}$ is an n-multiset of points in $\mathrm{PG}(k-1,q)$ where each point is counted with the appropriate multiplicity. In general, $\mathcal{G}$ is called the n-multiset induced by $\mathcal{C}$.

Proposition 7. *There exists a one-to-one correspondence between the equivalence classes of full-length q-ary linear codes and the projective equivalence classes of multisets in finite projective spaces.*

An $[n,k]_q$ code can be described by a generator matrix $\mathbf{G}$ or as discussed by an n-multiset $\mathcal{G}$ of points in $\mathrm{PG}(k-1,q)$. In what follows, for the ease of notation, we restrict ourselves to the binary field. We associate the points of $\mathrm{PG}(k-1,2)$ with the non-zero vectors in $\mathbb{F}_2^k \setminus \{\mathbf{0}_k\}$, then we interpret each such vector as the binary expansion of the corresponding integer $i \in [\ell]$ where $\ell := 2^k - 1$. We denote by $\mathbf{v}_i$ the vector corresponding to the integer $i \in [\ell]$. As two examples, in $\mathbb{F}_2^3 \setminus \{\mathbf{0}_3\}$, the vectors $\mathbf{v}_3 = (0,1,1)$ and $\mathbf{v}_4 = (1,0,0)$ are corresponding to the integers 3 and 4, respectively. In order to uniquely characterize a multiset of points $\mathcal{G}$ in $\mathrm{PG}(k-1,2)$, we use multiplicities $n_i \in \mathbb{Z}_{\geq 0}$, $i \in [\ell]$, counting the number of occurrences of the vector $\mathbf{v}_i \in \mathbb{F}_2^k \setminus \{\mathbf{0}_k\}$, $i \in [\ell]$, in the generator matrix $\mathbf{G}$. Thus, we have $\sum_{i \in [\ell]} n_i = n$. Also, due to the correspondence between a generator matrix $\mathbf{G}$ and a multiset of points $\mathcal{G}$ (based on the Proposition 7), we can write $\mathcal{S}(\mathcal{G})$ instead of $\mathcal{S}(\mathbf{G})$ for the service rate region and we will directly define $\mathcal{S}(\mathcal{G})$ shortly.

7.2.5 Geometric Interpretation of the Service Rate Region

A recovery set for file f_i, $i \in [k]$, is a subset $Y \subseteq [\ell]$ such that the span of the set $\{\mathbf{v}_j \mid j \in Y\}$ contains the unit vector $\mathbf{e}_i$. A recovery set Y for f_i is called reduced if there does not exist a proper subset $Y' \subsetneq Y$ with $\mathbf{e}_i \in \langle \{\mathbf{v}_j \mid j \in Y'\} \rangle$. For $q = 2$ and a reduced recovery set Y, there is no need to specify the index i of the file that is recovered since $\sum_{j \in Y} \mathbf{v}_j = \mathbf{e}_i$. However, this is not necessarily true for $q > 2$. As an example, in $\mathbb{F}_3$ the set

$\{\mathbf{e}_1 + \mathbf{e}_2, \mathbf{e}_1 + 2\mathbf{e}_2\}$ spans a 2-dimensional subspace containing both $\mathbf{e}_1$ and $\mathbf{e}_2$, while none of these two unit vectors are contained in the span of a proper subset. Since we assume $q = 2$, we will mostly speak just of a recovery set without specifying the index i of the file that it recovers. By $\mathcal{Y}_i$ we denote the set of all reduced recovery sets for file f_i, where $i \in [k]$. For example, for $k = 3$ we have $\mathcal{Y}_2 = \{\{2\}, \{4, 6\}, \{1, 3\}, \{5, 7\}, \{1, 4, 7\}\}$. Note that the maximum cardinality of a reduced recovery set is k, which can indeed be attained.

Let $\alpha_{i,Y}$ be the portion of request rates for file f_i assigned to the recovery set $Y \in \mathcal{Y}_i$. Given a multiset of points $\mathcal{G}$ in $\mathrm{PG}(k-1, 2)$, described by the multiplicities n_j, $j \in [\ell]$, the service rate region $\mathcal{S}(\mathcal{G})$ is the set of all vectors $\boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0}$ for which there exist $\alpha_{i,Y}$'s, satisfying the following constraints:

$$\sum_{Y \in \mathcal{Y}_i} \alpha_{i,Y} = \lambda_i, \qquad \text{for all} \quad i \in [k], \tag{7.2a}$$

$$\sum_{i=1}^{k} \sum_{\substack{Y \in \mathcal{Y}_i \\ j \in Y}} \alpha_{i,Y} \leq n_j, \qquad \text{for all} \quad j \in [\ell], \tag{7.2b}$$

$$\alpha_{i,Y} \in \mathbb{R}_{\geq 0}, \qquad \text{for all} \quad i \in [k],\ Y \in \mathcal{Y}_i. \tag{7.2c}$$

As noted, for $q = 2$, each reduced recovery set uniquely characterizes the file it recovers, that is, $\mathcal{Y}_i$'s where $i \in [k]$ are pairwise disjoint and form a partition of $\mathcal{Y} := \cup_{i \in [k]} \mathcal{Y}_i$. With this we can simplify the above characterization, i.e., the service rate region $\mathcal{S}(\mathcal{G})$ is the set of all vectors $\boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0}$ for which there exists α_Y, satisfying the following:

$$\sum_{Y \in \mathcal{Y}_i} \alpha_Y \geq \lambda_i, \qquad \text{for all} \quad i \in [k], \tag{7.3a}$$

$$\sum_{\substack{Y \in \mathcal{Y} \\ j \in Y}} \alpha_Y \leq n_j, \qquad \text{for all} \quad j \in [\ell], \tag{7.3b}$$

$$\alpha_Y \in \mathbb{R}_{\geq 0}, \qquad \text{for all} \quad i \in [k],\ Y \in \mathcal{Y}_i. \tag{7.3c}$$

7.2.6 Problem Statement

After these preparations, we can state the problems that we explore to address in this chapter. Consider a practical scenario where we are asked to store k files redundantly across some number of nodes in a coded distributed storage system. Also, we are given a bounded subset $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$ as a desired service rate region for this distributed storage system. Two natural questions arising in the design of this storage system are the following: 1) What is the minimum number $n(\mathcal{R})$ of storage nodes (or servers) required for serving all demand vectors $\boldsymbol{\lambda}$ inside the desired service rate region $\mathcal{R}$? 2) What is the most storage-efficient redundancy scheme with the service rate region covering the set $\mathcal{R}$ (i.e., how should the files be stored redundantly in $n(\mathcal{R})$ storage nodes)?

In other words, for each desired service rate region $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$, the goal is to characterize the minimum number of nodes $n(\mathcal{R})$ (or derive a lower bound on $n(\mathcal{R})$) such that there exists a generator matrix $\mathbf{G}$ with $\mathcal{R} \subseteq \mathcal{S}(\mathbf{G})$ (or alternatively, a multiset of points $\mathcal{G}$ in $\mathrm{PG}(k-1, q)$ with $\mathcal{R} \subseteq \mathcal{S}(\mathcal{G})$). Thus, deriving lower bounds and constructive upper bounds for $n(\mathcal{R})$ is of great significance in the context of designing distributed storage systems, which we aim to address in this chapter.

Recall that $\mathcal{S}(\mathbf{G}) \neq \mathcal{S}(\mathbf{G}')$ (or $\mathcal{S}(\mathcal{G}) \neq \mathcal{S}(\mathcal{G}')$) even if $\mathbf{G}, \mathbf{G}'$ (or $\mathcal{G}, \mathcal{G}'$) generate the same linear code. Thus, to be more precise, instead of designing an efficient code, we have to speak of the construction of storage-efficient generator matrices or multisets of points.

7.3 Main Results

In this section, first we investigate a few structural properties and formulate the problem of determining $n(\mathcal{R})$. Then, using a geometric approach, we derive multiple lower bounds on $n(\mathcal{R})$ and finally we show that for $k = 2$ the derived lower bounds are tight by proposing an storage-efficient redundancy scheme.

7.3.1 Structural Properties of Service Rate Region

Here, before we present integer linear programming (ILP) formulations for the determination of $n(\mathcal{R})$ we first study a few structural properties.

Lemma 20. *For $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$, we have $n(\mathcal{R}) = n(\mathrm{conv}(\mathcal{R}))$.*

Proof. It suffices to observe that the service rate region $\mathcal{S}(\mathbf{G})$ of every generator matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ is convex. For more details, see [76]. □

Definition 17. *For a set $S \subseteq \mathbb{R}^k_{\geq 0}$, the set $S{\downarrow} := \left\{\mathbf{x} \in \mathbb{R}^k_{\geq 0} \mid \exists \mathbf{y} \in S : \mathbf{x} \leq \mathbf{y}\right\}$ defines the lower set of S.*

Consider the bounded subset $S = \mathrm{conv}(\{(0,0),(1,2),(2,1)\}) \subset \mathbb{R}^2_{\geq 0}$ which is a triangle with area $\frac{3}{2}$. The lower set of S is $S{\downarrow} = \mathrm{conv}(\{(0,0),(0,2),(2,0),(1,2),(2,1)\})$ which is a pentagon with area $\frac{7}{2}$.

Lemma 21. *For a subset $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$, we have $n(\mathcal{R}) = n(\mathcal{R}{\downarrow})$.*

Proof. It suffices to observe that the service rate region $\mathcal{S}(\mathbf{G})$ of every generator matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ is its own lower set, i.e., $\mathcal{S}(\mathbf{G}) = \mathcal{S}(\mathbf{G}){\downarrow}$. □

Taken the above two observations into account, we can parameterize a large class of reasonable subsets $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$ through a function $T\colon 2^{[k]} \to \mathbb{N}$ that maps the subsets of $[k]$ to integers in $\mathbb{N}$, where $T(\emptyset) = 0$.

Definition 18. *Let $T\colon 2^{[k]} \to \mathbb{N}$ with $T(\emptyset) = 0$. We define*

$$\mathcal{R}(T) := \left\{ \boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0} \mid \sum_{i\in S} \lambda_i \leq T(S),\ \ \forall \emptyset \neq S \subseteq [k] \right\}$$

By construction $\mathcal{R}(T)$ is a convex polytope and $\mathcal{R}(T){\downarrow} = \mathcal{R}(T)$, i.e., $\mathcal{R}(T)$ is its own lower set. (For more details, see e.g., [76].) It should be noted that in some cases, the values of the function $T\colon 2^{[k]} \to \mathbb{N}$ can be modified without changing $\mathcal{R}(T)$.

Lemma 22. *For each function $T\colon 2^{[k]} \to \mathbb{N}$ with $T(\emptyset) = 0$, there exists a monotone and subadditive function $T'\colon 2^{[k]} \to \mathbb{N}$ with $T'(\emptyset) = 0$, such that $R(T) = R(T')$.*[*]

Proof. Obtain T' by running the following algorithm on $T\colon 2^{[k]} \to \mathbb{N}$, $T(\emptyset) = 0$.

for each $S \subseteq \{1, \ldots, k\}$ **do**
  $T'(S) \leftarrow T(S)$
end for
$changed \leftarrow$ `true`
while $changed = true$ **do**
  $changed \leftarrow$ `false`
  for each $S \subseteq \{1, \ldots, k\}$ **do**
    for each $\emptyset \neq U \subsetneq S$ **do**
      if $T'(S) > T'(U) + T'(S \backslash U)$ **then**
        $T'(S) \leftarrow T'(U) + T'(S \backslash U)$
        $changed \leftarrow$ `true`
      end if
    end for
    for each $S \subsetneq V \subseteq \{1, \ldots, k\}$ **do**
      if $T'(S) > T'(V)$ **then**
        $T'(S) \leftarrow T'(V)$
        $changed \leftarrow$ `true`
      end if
    end for
  end for
end while

[*] $T\colon 2^{[k]} \to \mathbb{N}$ is monotone iff $T(U) \leq T(V)$ for all $\emptyset \subseteq U \subseteq V \subseteq [k]$, and is subadditive iff $T(U \cup V) \leq T(U) + T(V)$.

We remark that based on the above algorithm, the function T' is subadditive, i.e., we have $T'(U \cup V) \leq T'(U) + T'(V)$, and monotone, that is, we have $T'(U) \leq T'(V)$ for all $\emptyset \subseteq U \subseteq V \subseteq [k]$. Now, we need to prove that $\mathcal{R}(T) = \mathcal{R}(T')$.

After the first initializing loop we obviously have $\mathcal{R}(T) = \mathcal{R}(T')$. Now, let us consider a single step in which $T'(S)$ is replaced by either $T'(U) + T'(S \backslash U)$ or $T'(V)$. Inductively we know that each $\boldsymbol{\lambda} \in \mathcal{R}(T')$ satisfies $\sum_{i \in S'} \lambda_i \leq T'(S')$ for all $S' \subseteq [k]$. Since this especially holds for $S' = U$, $S' = S \backslash U$, and $S' = V$ we also have

$$\sum_{i \in S} \lambda_i \leq T'(U) + T'(S \backslash U)$$

and

$$\sum_{i \in S} \lambda_i \overset{\boldsymbol{\lambda} \geq 0}{\leq} \sum_{i \in V} \lambda_i \leq T'(V).$$

So, after each replacement we still have $\mathcal{R}(T) = \mathcal{R}(T')$. In order to show that the algorithm terminates let

$$\varepsilon = \min\{T(U) - T(V) \mid \emptyset \subseteq U, V \subseteq [k], T(U) - T(V)\}.$$

By induction over the number of replacements we can easily show that at each time after the initialization loop , we have

$$\varepsilon \leq \min\{T'(U) - T'(V) \mid \emptyset \subseteq U, V \subseteq [k], T'(U) - T'(V)\}$$

Thus, every replacement reduces the value of $\sum_{S \subseteq [k]} T'(S)$ by at least ε, so that the algorithm terminates after at least $(\sum_{S \subseteq [k]} T(S))/\varepsilon + 1$ iterations of the while loop.

Note that since in the last iteration of the while loop non of the if-conditions were true, if we apply the algorithm again on T' and obtain T'', then $T' = T''$. $\square$

Definition 19. *For a subset $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$ with property $\mathcal{R}\downarrow = \mathcal{R}$, a finite set $S \subset \mathbb{R}^k_{\geq 0}$ is a generating set of $\mathcal{R}$ if $\mathrm{conv}(S)\downarrow = \mathcal{R}$. Moreover, we call S minimal if no proper subset of S is a generating set of $\mathcal{R}$.*

In what follows, without explicitly mentioning, we only consider the minimal generating sets for each $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$. As an example, consider the function $T\colon 2^{[2]} \to \mathbb{N}$ given by $T(\emptyset) = 0$, $T(\{1\}) = T(\{2\}) = 2$, and $T(\{1,2\}) = 3$. Here, a generating set of $\mathcal{R}(T)$ is given by the set $\{(1,2),(2,1)\}$.

We remark that the generating set of $\mathcal{R}(T)$ is always unique, since $\mathcal{R}(T)$ is a polytope that can be written as $\mathcal{R}(T) = \mathrm{conv}(V)$, where V is the set of vertices of the polytope, which is a unique minimal set. The unique generating set of $\mathcal{R}(T)$ is obtained from V by removing all vectors $\mathbf{v} \in V$ such that there exists a vector $\mathbf{v}' \in V$ with $\mathbf{v} \leq \mathbf{v}'$.

Next, we present an ILP formulation for the determination of $n(\mathcal{R})$.

Proposition 8. *For a desired service rate region $\mathcal{R} \subset \mathbb{R}^k_{\geq 0}$, assume that $\mathcal{R}\downarrow = \mathcal{R}$. Let $\left\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(m)}\right\}$ be the generating set of $\mathcal{R}$. Then, $n(\mathcal{R})$ coincides with the optimal target value of the following ILP*

$$
\begin{aligned}
\min \quad & \sum_{j \in [\ell]} n_j && && (7.4)\\
\textit{s.t.} \quad & \sum_{Y \in \mathcal{Y}^j} \alpha^i_Y \geq \lambda^{(i)}_j && \forall i \in [m],\ j \in [k] \\
& \sum_{\substack{Y \in \mathcal{Y} \\ j \in Y}} \alpha^i_Y \leq n_j, && \forall j \in [\ell], \forall i \in [m] \\
& \alpha^i_Y \in \mathbb{R}_{\geq 0}, && \forall i \in [m], \forall Y \in \mathcal{Y} \\
& n_j \in \mathbb{N}, && \forall j \in [\ell]
\end{aligned}
$$

where $\lambda^{(i)}_j$ is the jth element of the $\boldsymbol{\lambda}^{(i)}$ and α^i_Y is the portion of requests coming from $\boldsymbol{\lambda}^{(i)}$ assigned to the recovery set Y.

Proof. The multiset of points $\mathcal{G}$ is uniquely characterized by the integer multiplicities n_j, $j \in [\ell]$. Thus, the stated ILP formulation minimizes the code length $n = \sum_{j\in[\ell]} n_j$ and ensures that $\boldsymbol{\lambda}^{(i)} \in \mathcal{S}(\mathcal{G})$ by using the characterization (7.3a)–(7.3c) for each $i \in [m]$. $\square$

The ILP formulation (7.4) in the Proposition 8, underlies a massive combinatorial explosion. To be more precise, when the number of files k increases, the number of recovery sets $\#\mathcal{Y}$ grows doubly exponential, that is, $\#\mathcal{Y}$ gets quite large even for moderate values of k. In order to obtain a lower bound on $n(\mathcal{R})$, one simple way is to consider the ceiling of the optimal target value for the LP relaxation of the ILP (7.4). However, this approach again suffers from the same drawback and runs into a similar problem since to list all the constraints of the LP relaxation of the ILP (7.4), one needs to explicitly know all possible recovery sets which becomes increasingly complex when the number of files k increases. Thus, introducing a technique which is not depending on the enumeration of recovery sets is of great significance. Towards this goal, we introduce a novel geometric approach.

7.3.2 Using Geometric Approach to Derive Bounds on $n(\mathcal{R})$

In this section, we present three general lower bounds for $n(\mathcal{R}(T))$ that are obtained using a geometric technique. The following Lemma is the key component of the proofs.

Lemma 23. *Given a service rate region $\mathcal{R}$ with the generating set $\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(m)}\}$, if an n-multiset of points $\mathcal{G}$ in $\mathrm{PG}(k-1,2)$ described by point multiplicities n_j for $j \in [\ell]$, covers the service rate region $\mathcal{R}$ (i.e., $\mathcal{R} \subseteq \mathcal{S}(\mathcal{G})$), then for every hyperplane $\mathcal{H}$ in $\mathrm{PG}(k-1,2)$ the following holds*

$$\sum_{j\,:\,\mathbf{v}_j\in\mathcal{G}\setminus\mathcal{H}} n_j \geq \max_{i\in[m]} \sum_{s\in\mathcal{E}(\mathcal{H})} \lambda_s^{(i)}, \tag{7.5}$$

where $\mathcal{E}(\mathcal{H}) = \{h \in [k] \mid \mathbf{e}_h \notin \mathcal{H}\}$ is the set of all indices h such that the hyperplane $\mathcal{H}$ does not contain the unit vector $\mathbf{e}_h$, i.e., $\mathbf{e}_h$ lies in $\mathrm{PG}(k-1,2) \setminus \mathcal{H}$.

Proof. Let $i \in [m]$ be an arbitrary index. From the ILP of Proposition 8, we conclude that

$$\sum_{Y \in \mathcal{Y}_s} \alpha_Y^i \geq \lambda_s^{(i)} \tag{7.6}$$

Since $\alpha_Y^i \geq 0$, for each $s \in \mathcal{E}(\mathcal{H})$ we have

$$n_j \geq \sum_{Y \in \mathcal{Y} \,:\, j \in Y} \alpha_Y^i \geq \sum_{s \in \mathcal{E}(\mathcal{H})} \sum_{Y \in \mathcal{Y}_s \,:\, j \in Y} \alpha_Y^i \tag{7.7}$$

Let $J = \{j \in [\ell] \mid \mathbf{v}_j \in \mathrm{PG}(k-1,2) \setminus \mathcal{H}\}$. Thus, we have

$$\sum_{j \in J} n_j \geq \sum_{j \in J} \sum_{s \in \mathcal{E}(\mathcal{H})} \sum_{Y \in \mathcal{Y}_s \,:\, j \in Y} \alpha_Y^i \sum_{s \in \mathcal{E}(\mathcal{H})} \sum_{j \in J} \sum_{Y \in \mathcal{Y}_s \,:\, j \in Y} \alpha_Y^i.$$

The unit vectors $\mathbf{e}_s$ with index $s \in \mathcal{E}(\mathcal{H})$ are not contained in the hyperplane $\mathcal{H}$. Thus, for each $Y \in \mathcal{Y}_s$ with $s \in \mathcal{E}(\mathcal{H})$ there exists certainly an index $j \in [\ell]$ with $j \in Y$ such that $\mathbf{v}_j \in \mathrm{PG}(k-1,2) \setminus \mathcal{H}$. Thus, from Inequality (7.6) we conclude that

$$\sum_{j \in [\ell] \,:\, \mathbf{v}_j \in \mathrm{PG}(k-1,2) \setminus \mathcal{H}} n_j \geq \sum_{s \in \mathcal{E}(\mathcal{H})} \sum_{Y \in \mathcal{Y}_s} \alpha_Y^i \geq \sum_{s \in \mathcal{E}(\mathcal{H})} \lambda_s^{(i)}$$

□

Corollary 9. *If $\left\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(m)}\right\}$ is the generating set of $\mathcal{R}$, then $n(\mathcal{R})$ is lower bounded by the optimal target value of the following ILP formulation:*

$$\begin{aligned} \min \quad & \sum_{j \in [\ell]} n_j \\ \textit{s.t.} \quad & (5) \ \textit{holds} \ \forall \ \textit{hyperplane} \ \mathcal{H} \ \textit{of} \ \mathrm{PG}(k-1,2) \\ & n_j \in \mathbb{N} \qquad \forall j \in [\ell]. \end{aligned} \tag{7.8}$$

Note that the ILP of Corollary 9 contains $2^k - 1$ constraints and (integer) variables. Thus, with respect to the LP relaxation of the ILP formulation (7.4) in Proposition 8, we have obtained a smaller formulation for the determination of a lower bound on $n(\mathcal{R})$.

Definition 20. *Let $P = \left\{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\right\}$ be a polytope in $\mathbb{R}^k$ with description $(\mathbf{A}, \mathbf{b})$. We say that constraint $\mathbf{a}^{(i)}\mathbf{x} \leq b_i$ is redundant, where $\mathbf{a}^{(i)}$ is the ith row of $\mathbf{A}$, if the polytope $P = \left\{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{A}'\mathbf{x} \leq \mathbf{b}', \mathbf{x} \geq 0\right\}$ where $\mathbf{A}'$ and $\mathbf{b}'$ obtained from $\mathbf{A}$ and $\mathbf{b}$ by removing the ith row, respectively. We say that a constraint $\mathbf{a}^{(i)}\mathbf{x} \leq b_i$ is strictly redundant if there does not exist $\bar{\mathbf{x}} \in P$ with $\mathbf{a}^{(i)}\bar{\mathbf{x}} = b_i$.*

For example, consider $T\colon 2^{[2]} \to \mathbb{N}$ defined as $T(\{1\}) = T(\{2\}) = T(\{1,2\}) = 1$ and $T(\emptyset) = 0$, and the polytope $P = \left\{\boldsymbol{\lambda} \in \mathbb{R}^2 \mid \sum_{i\in U} \lambda_i \leq T(U), \emptyset \neq U \subseteq \{1,2\}, \boldsymbol{\lambda} \geq 0\right\}$. The inequalities $\lambda_1 \leq T(\{1\})$, $\lambda_2 \leq T(\{2\})$ are redundant, while the inequality $\lambda_1 + \lambda_2 \leq T(\{1,2\})$ is not redundant since e.g. the vector $(1,1)$ is not contained in the polytope. Here, none of the inequalities are strictly redundant since the vectors $(1,0)$, $(0,1)$ are contained in the polytope.

Theorem 18. *Given the function $T\colon 2^{[k]} \to \mathbb{N}$ for some $k \in \mathbb{N}$, if none of the constraints $\sum_{i\in U} \lambda_i \leq T(U)$ are strictly redundant in $\mathbb{R}^k_{\geq 0}$, then we have*

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k]} T(U)}{2^{k-1}} \right\rceil.$$

Proof. We observe that each hyperplane $\mathcal{H}$ in $\mathrm{PG}(k-1,2)$ can be uniquely characterized by a subset $\emptyset \neq S(\mathcal{H}) \subseteq [k]$ such that $S(\mathcal{H}) = \{i \in [k] \mid \mathbf{e}_i \notin \mathcal{H}\}$. Let w.l.o.g. $\left\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(m)}\right\}$ be the generating set of $\mathcal{R}(T)$. Due to the fact that none of the constraints $\sum_{i\in U} \lambda_i \leq T(U)$ is strictly redundant, we have

$$\max\left\{ \sum_{s\in S(\mathcal{H})} \lambda_s^{(i)} \mid i \in [m] \right\} = T(S(\mathcal{H}))$$

Thus, for each hyperplane $\mathcal{H}$ of $\mathrm{PG}(k-1,2)$, by applying Lemma 23 and replacing $T(S(\mathcal{H}))$ in the right hand side of inequality (7.5), we get

$$\sum_{j\in[\ell]\,:\,\mathbf{v}_j\in\mathrm{PG}(k-1,2)\setminus\mathcal{H}} n_j \geq T(S(\mathcal{H})),$$

where $S(\mathcal{H}) = \{i \in [k] \,:\, \mathbf{e}_i \notin \mathcal{H}\}$. Since there are $2^k - 1$ hyperplanes in $\mathrm{PG}(k-1,2)$, we have $2^k - 1$ such inequalities, each of which can be uniquely characterized by subset $S(\mathcal{H})$. For each $j \in [\ell]$, one can easily verify that $\mathbf{v}_j \notin \mathcal{H}$ for exactly 2^{k-1} hyperplanes $\mathcal{H}$. This means that each n_j for $j \in [\ell]$ appears in the left side of 2^{k-1} inequalities. Thus, summing all of the $2^k - 1$ inequalities, dividing by 2^{k-1} and replacing variable $S(\mathcal{H})$ with variable U, yields

$$n = \sum_{j\in[\ell]} n_j \geq \frac{\sum_{\emptyset\neq U\subseteq[k]} T(U)}{2^{k-1}}.$$

Finally, since n has to be an integer, we have

$$n \geq \lceil \frac{\sum_{\emptyset\neq U\subseteq[k]} T(U)}{2^{k-1}} \rceil.$$

□

As we will show shortly the lower bound of Theorem 18 is indeed tight if $k = 2$ and $T\colon 2^{[k]} \to \mathbb{N}$ is monotone and subadditive. However, this bound is not tight in general for $K \geq 3$. The following example shows that for $K = 3$ this bound is not tight, while none of the constraints are strictly redundant.

Example 13. For $k = 3$, consider the desired service rate region $\mathcal{R} = \mathcal{R}(T)$ for $T\colon 2^{[3]} \to \mathbb{N}$ defined as $T(\emptyset) = 0$ and $T(S) = \#S + 1$ for $\emptyset \neq S \subseteq [3]$, that is, $\mathcal{R}$ is as follows

$$\mathcal{R} = \left\{\boldsymbol{\lambda} \in \mathbb{R}^3_{\geq 0} \,:\, \lambda_1, \lambda_2, \lambda_3 \leq 2, \lambda_1 + \lambda_2 \leq 3, \lambda_1 + \lambda_3 \leq 3, \lambda_2 + \lambda_3 \leq 3, \lambda_1 + \lambda_2 + \lambda_3 \leq 4\right\}.$$

A generating set $\{\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \boldsymbol{\lambda}^{(3)}\}$ of $\mathcal{R}$ of cardinality $m = 3$ is given by $\boldsymbol{\lambda}^{(1)} = (2,1,1)$, $\boldsymbol{\lambda}^{(2)} = (1,2,1)$, and $\boldsymbol{\lambda}^{(3)} = (1,1,2)$. The possible columns of a generator matrix $\mathbf{G}$, i.e., the non-zero vectors in $\mathbb{F}_2^3$ are $\mathbf{v}_1 = (0,0,1), \mathbf{v}_2 = (0,1,0), \mathbf{v}_3 = (0,1,1), \mathbf{v}_4 = (1,0,0), \mathbf{v}_5 = (1,0,1), \mathbf{v}_6 = (1,1,0), \mathbf{v}_7 = (1,1,1)$. In order to write down the inequalities from Lemma 23 we describe a hyperplane $\mathcal{H}$ as a set of vectors $(x_1, x_2, x_3) \in \mathbb{F}_2^3 \setminus \{\mathbf{0}\}$ satisfying a certain constraint $\sum_{i=1}^{3} c_i x_i$, where $(c_1, c_2, c_3) \in \mathbb{F}_2^3 \setminus \{\mathbf{0}\}$:

$$\mathcal{H}_1 : x_1 = 0 \Rightarrow \mathbf{e}_1 \notin \mathcal{H}_1 \;\Rightarrow\; n_4 + n_5 + n_6 + n_7 \geq 2 = \max(\lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}) \tag{7.9}$$

$$\mathcal{H}_2 : x_2 = 0 \Rightarrow \mathbf{e}_2 \notin \mathcal{H}_2 \;\Rightarrow\; n_2 + n_3 + n_6 + n_7 \geq 2 = \max(\lambda_2^{(1)}, \lambda_2^{(2)}, \lambda_2^{(3)}) \tag{7.10}$$

$$\mathcal{H}_3 : x_3 = 0 \Rightarrow \mathbf{e}_3 \notin \mathcal{H}_3 \;\Rightarrow\; n_1 + n_3 + n_5 + n_7 \geq 2 = \max(\lambda_3^{(1)}, \lambda_3^{(2)}, \lambda_3^{(3)}) \tag{7.11}$$

$$\mathcal{H}_4 : x_1 + x_2 = 0 \Rightarrow \mathbf{e}_1, \mathbf{e}_2 \notin \mathcal{H}_4 \;\Rightarrow\; n_2 + n_3 + n_4 + n_5 \geq 3 = \max_{i \in [3]} \big(\sum_{j \in \{1,2\}} \lambda_j^{(i)} \big) \tag{7.12}$$

$$\mathcal{H}_5 : x_1 + x_3 = 0 \Rightarrow \mathbf{e}_1, \mathbf{e}_3 \notin \mathcal{H}_5 \;\Rightarrow\; n_1 + n_3 + n_4 + n_6 \geq 3 = \max_{i \in [3]} \big(\sum_{j \in \{1,3\}} \lambda_j^{(i)} \big) \tag{7.13}$$

$$\mathcal{H}_6 : x_2 + x_3 = 0 \Rightarrow \mathbf{e}_2, \mathbf{e}_3 \notin \mathcal{H}_6 \;\Rightarrow\; n_1 + n_2 + n_5 + n_6 \geq 3 = \max_{i \in [3]} \big(\sum_{j \in \{2,3\}} \lambda_j^{(i)} \big) \tag{7.14}$$

$$\mathcal{H}_7 : x_1 + x_2 + x_3 = 0 \Rightarrow \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \notin \mathcal{H}_7 \;\Rightarrow\; n_1 + n_2 + n_4 + n_7 \geq 4 = \max_{i \in [3]} \big(\sum_{j \in [3]} \lambda_j^{(i)} \big) \tag{7.15}$$

Summing up inequalities (7.9)-(7.15) and dividing by four gives $n \geq \lceil \frac{19}{4} \rceil = 5$. Indeed, the LP relaxation of the ILP (7.8) from Corollary 9 has an optimal solution $n_1 = n_2 = n_4 = \frac{5}{4}$, $n_3 = n_5 = n_6 = n_7 = \frac{1}{4}$ with target value $\frac{19}{4}$. Next, we show that $n \geq 6$ for the optimal target value of ILP (7.8). Assume that there exists an integral solution with $n = 5$. Summing the inequalities over all hyperplanes $\mathcal{H}_i$ containing $\mathbf{v}_1 = \mathbf{e}_3$, i.e., (7.9), (7.10), and (7.12), and dividing by two gives $\sum_{j \in [\ell] \setminus \{1\}} n_j \geq 3.5$, so that $n_1 \leq 1$. By symmetry, we also conclude $n_2, n_4 \leq 1$. Summing the inequalities over all hyperplanes $\mathcal{H}_i$ not containing $\mathbf{v}_1 = \mathbf{e}_3$, i.e., (7.11), (7.13), (7.14), and (7.15), and dividing by two gives $2n_1 + \sum_{j \in [\ell] \setminus \{1\}} n_j \geq 6$, so that $n_1 \geq 1$. Thus, $n_1 = 1$ and, by symmetry, also $n_2 = n_4 = 1$. Summing inequalities (7.12)-(7.14), plugging in the known values, and

dividing by two gives $n_3 + n_5 + n_6 \geq 1.5$, so that $n_7 \leq 0.5$, i.e., $n_7 = 0$. However, this contradicts Inequality (7.15).

An integral solution for $n = 6$ can be attained by setting $n_1 = n_2 = n_4 = 2$, $n_3 = n_5 = n_6 = n_7 = 0$. It can be easily checked that the corresponding generator matrix $\mathbf{G}$ as given below satisfies $\mathcal{S}(\mathbf{G}) \supseteq \mathcal{R}$.

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Corollary 10. *For some $k \in \mathbb{N}$ and $X \in \mathbb{N}$, given the function $T\colon 2^{[k]} \to \mathbb{N}$ defined as $T(\emptyset) = 0$, $T(U) = X$ for all subsets $\emptyset \neq U \subseteq [k]$, we have*

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{X \cdot \left(2^k - 1\right)}{2^{k-1}} \right\rceil.$$

Moreover, if $X = t \cdot 2^{k-1}$ for some integer t, then the lower bound is tight.

Proof. One can easily check that none of the constraints $\sum_{i \in U} \lambda_i \leq T(U)$ are strictly redundant in $\mathbb{R}^k_{\geq 0}$. Thus, Theorem 18 can be applied. Thus,

$$n \geq \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k]} T(U)}{2^{k-1}} \right\rceil = \left\lceil \frac{X \cdot \left(2^k - 1\right)}{2^{k-1}} \right\rceil.$$

The generating set of $\mathcal{R}(T)$ is given by $\{X \cdot \mathbf{e}_i \mid i \in [k]\}$. Thus, for $X = t \cdot 2^{k-1}$, a t-fold k-dimensional binary simplex code achieves the desired service rate region. For more details, see [76]. □

Next, two more general lower bounds for $n(\mathcal{R}(T))$, similar to that of Theorem 18, are provided that are obtained in the search of finding a tighter lower bound for $k \geq 3$.

Theorem 19. *For some integer $k \geq 2$, let $T\colon 2^{[k]} \to \mathbb{N}$ be a function such that none of the constraints $\sum_{i\in U} \lambda_i \leq T(U)$ are strictly redundant in $\mathbb{R}^k_{\geq 0}$. For each $i \in [k]$, $n(\mathcal{R}(T)) \geq \left\lceil \frac{\alpha_i+\beta_i}{2} \right\rceil$ holds, where*

$$\alpha_i = \left\lceil \frac{\sum_{\emptyset\neq U\subseteq[k]\setminus\{i\}} T(U)}{2^{k-2}} \right\rceil,\ \beta_i = \left\lceil \frac{\sum_{\{i\}\subseteq U\subseteq[k]} T(U)}{2^{k-2}} \right\rceil.$$

Proof. Based on the same reasoning used in the proof of Theorem 18, it can be shown that for each hyperplane $\mathcal{H}$ of $\mathrm{PG}(k-1,2)$, we have

$$\sum_{j\in[\ell]\,:\,\mathbf{v}_j\in\mathrm{PG}(k-1,2)\setminus\mathcal{H}} n_j \geq T(S(\mathcal{H})), \tag{7.16}$$

where $S(\mathcal{H}) = \{i \in [k] \,:\, \mathbf{e}_i \notin \mathcal{H}\}$. Let $i \in [k]$ be arbitrary but fix and $\bar{i} = 2^{k-i}$ such that $\mathbf{v}_{\bar{i}} = \mathbf{e}_i$. Then, we proceed by summing Inequality (7.16) for all subsets $\emptyset \neq S(\mathcal{H}) \subseteq [k] \setminus \{i\}$ and replacing $S(\mathcal{H})$ everywhere with U, that gives

$$2^{k-2} \cdot \sum_{j\in[\ell]\setminus\{\bar{i}\}} n_j \geq \sum_{\emptyset\neq U\subseteq[k]\setminus\{i\}} T(U). \tag{7.17}$$

Now, summing Inequality (7.16) for all $\{i\} \subseteq S(\mathcal{H}) \subseteq [k]$ and replacing $S(\mathcal{H})$ everywhere with U gives

$$2^{k-1} \cdot n_{\bar{i}} + 2^{k-2} \cdot \sum_{j\in[\ell]\setminus\{\bar{i}\}} n_j \geq \sum_{\{i\}\subseteq U\subseteq[k]} T(U). \tag{7.18}$$

Since the n_js are integers, from equations (7.17) and (7.18), we get $\sum_{j\in[\ell]\setminus\{\bar{i}\}} n_j \geq \alpha_i$ and $2n_{\bar{i}} + \sum_{j\in[\ell]\setminus\{\bar{i}\}} n_j \geq \beta_i$, respectively, where

$$\alpha_i = \left\lceil \frac{\sum_{\emptyset\neq U\subseteq[k]\setminus\{i\}} T(U)}{2^{k-2}} \right\rceil,\ \ \beta_i = \left\lceil \frac{\sum_{\{i\}\subseteq U\subseteq[k]} T(U)}{2^{k-2}} \right\rceil.$$

Dividing the sum of these two inequalities by 2 gives

$$n = n_{\bar{i}} + \sum_{j \in [\ell] \setminus \{\bar{i}\}} n_j \geq \frac{\alpha_i + \beta_i}{2},$$

Since n has to be an integer, we have $n \geq \left\lceil \frac{\alpha_i + \beta_i}{2} \right\rceil$. □

Theorem 20. *For some integer $k \geq 2$, let $T \colon 2^{[k]} \to \mathbb{N}$ be a function such that none of the constraints $\sum_{i \in U} \lambda_i \leq T(U)$ are strictly redundant in $\mathbb{R}^k_{\geq 0}$. Then, for each $j \in [\ell]$ we have the following*

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k] \colon \#(U \cap J) \equiv 0 \pmod 2} T(U)}{2^{k-2}} \right\rceil,$$

where $J \subseteq [k]$ such that $\mathbf{v}_j = \sum_{h \in J} \mathbf{e}_h$.

Proof. Similar to the proof of Theorems 18 and 19, it can be shown that for each hyperplane $\mathcal{H}$ in $\mathrm{PG}(k-1, 2)$, we have

$$\sum_{i \in [\ell] \colon \mathbf{v}_i \in \mathrm{PG}(k-1,2) \setminus \mathcal{H}} n_i \geq T(S(\mathcal{H})), \tag{7.19}$$

where $S(\mathcal{H}) = \{i \in [k] \, : \, \mathbf{e}_i \notin \mathcal{H}\}$. For each index $i \in [\ell]$, it can be easily confirmed that $\mathbf{v}_i \notin \mathcal{H}$ for 2^{k-1} hyperplanes $\mathcal{H}$ and $\mathbf{v}_i \in \mathcal{H}$ for $2^{k-1} - 1$ hyperplanes $\mathcal{H}$ of $\mathrm{PG}(k-1, 2)$. Let $j \in [\ell]$ be arbitrary but fix. Our aim is to sum Inequality (7.19) over all $2^{k-1} - 1$ hyperplanes $\mathcal{H}$ that contain $\mathbf{v}_j$.

We proceed by proving a claim that $\mathbf{v}_j = \sum_{h \in J} \mathbf{e}_h \in \mathcal{H}$ iff $\#(U \cap J) \equiv 0 \pmod 2$, where $U = S(\mathcal{H})$. We consider two cases: (i) If $\#U = 1$, then w.l.o.g. assume $U = \{x\}$ for some $x \in [k]$. A basis of $\mathcal{H}$ is given by $\{\mathbf{e}_y \mid y \in [k] \setminus \{x\}\}$. Thus, $\mathbf{v}_j \in \mathcal{H}$ iff $x \notin J$, i.e., $\#(U \cap J) \equiv 0 \pmod 2$. (ii) If $\#U \geq 2$, then for some arbitrary element $x \in U$, a basis of $\mathcal{H}$ is given by the set $\{\mathbf{e}_y \mid y \in [k] \setminus U\} \cup \{\mathbf{e}_x + \mathbf{e}_z \mid z \in U \setminus \{x\}\}$.

In this case, it is easy to see that $\mathbf{v}_j \in \mathcal{H}$ iff $\#(U \cap J) = 2p$ for some $p \in \mathbb{Z}_{\geq 0}$, i.e., $\#(U \cap J) \equiv 0 \pmod 2$. Thus, the claim is proved. Now, by summing Inequality (7.19) over all hyperplanes $\mathcal{H}$ that contain $\mathbf{v}_j$, we obtain

$$\begin{aligned} 2^{k-2} \cdot \sum_{i \in [\ell] \setminus \{j\}} n_i &= \sum_{\text{hyperplane } \mathcal{H} \,:\, \mathbf{v}_j \in \mathcal{H}} \; \sum_{i \in [\ell] \,:\, \mathbf{v}_i \notin \mathcal{H}} n_i \\ &\geq \sum_{\emptyset \neq U \subseteq [k] \,:\, \#(U \cap J) \equiv 0 \pmod 2} T(U). \end{aligned}$$

Since $n \geq \sum_{i \in [\ell] \setminus \{j\}} n_i$ and n is an integer, we have

$$n \geq \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k] \,:\, \#(U \cap J) \equiv 0 \pmod 2} T(U)}{2^{k-2}} \right\rceil .$$

□

Example 14. For some $x \in \mathbb{N}$, let $T \colon 2^{[3]} \to \mathbb{N}$ be defined via $T(\{1\}) = T(\{2\}) = T(\{3\}) = T(\{1,2\}) = T(\{1,3\}) = x$ and $T(\{2,3\}) = T(\{1,2,3\}) = 2x$. Based on Theorem 18, we have $n(\mathcal{R}(T)) \geq \lceil \frac{9x}{4} \rceil$, and according to Theorem 20, considering $j = 3$ we have $n(\mathcal{R}(T)) \geq \lceil \frac{5x}{2} \rceil$. Thus, for $x \geq 3$, the lower bound obtained from Theorem 20 is tighter than the one obtained from Theorem 18.

7.3.3 Storage-Efficient Schemes for $k = 2$

Let w.l.o.g. (based on Lemma 22) the function $T \colon 2^{[2]} \to \mathbb{N}$ be monotone, subadditive, and satisfy $T(\emptyset) = 0$. Note that for $k = 1$ each $T \colon 2^{\{1\}} \to \mathbb{N}$ is monotone and subadditive, while for $k = 2$ the conditions can be summarized to

$$\max\{T(\{1\}), T(\{2\})\} \leq T(\{1,2\}) \leq T(\{1\}) + T(\{2\}).$$

The following Lemma describes the generating set of $\mathcal{R}(T)$ for $T \colon 2^{[2]} \to \mathbb{N}$.

Lemma 24. *If* $T\colon 2^{[2]} \to \mathbb{N}$ *is monotone, subadditive, and satisfies* $T(\emptyset) = 0$*, the generating set of* $\mathcal{R}(T)$ *is given by*

$$S = \Big\{ \Big(T(\{1\}), T(\{1,2\}) - T(\{1\}) \Big), \Big(T(\{1,2\}) - T(\{2\}), T(\{2\}) \Big) \Big\}.$$

Proof. According to the Definition 18, $\mathcal{R}(T)$ is the set of all vectors $\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0}$ that satisfy $\lambda_1 \leq T(\{1\}), \lambda_2 \leq T(\{2\})$ and $\lambda_1 + \lambda_2 \leq T(\{1,2\})$. Based on Definition 19, $S \subset \mathbb{R}^2_{\geq 0}$ is a generating set of $\mathcal{R}(T)$ if $\mathrm{conv}(S)\downarrow = \mathcal{R}(T)$.

The proof consists of two parts. First, we need to show that $\mathrm{conv}(S)\downarrow \subseteq \mathcal{R}(T)$. For this purpose, we check that each $\boldsymbol{\lambda} \in S$ satisfies the constraints $\lambda_1 \leq T(\{1\})$, $\lambda_2 \leq T(\{2\})$, and $\lambda_1 + \lambda_2 \leq T(\{1,2\})$, i.e., $\boldsymbol{\lambda} \in \mathcal{R}(T)$. Thus, $S \subseteq \mathcal{R}(T)$. Due to convex property of $\mathcal{R}(T)$ and since $\mathcal{R}(T)\downarrow = \mathcal{R}(T)$, it can be easily concluded that $\mathrm{conv}(S)\downarrow \subseteq \mathcal{R}(T)$.

Now, for the other direction, we need to show that $\mathcal{R}(T) \subseteq \mathrm{conv}(S)\downarrow$. Let $\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0}$ satisfy the constraints $\lambda_1 \leq T(\{1\})$, $\lambda_2 \leq T(\{2\})$, and $\lambda_1 + \lambda_2 \leq T(\{1,2\})$. W.l.o.g. we assume that at least one of these three inequalities is satisfied with equality, since we could increase $\boldsymbol{\lambda}$ otherwise. If $\lambda_1 + \lambda_2 = T(\{1,2\})$, then it can be readily concluded that $\boldsymbol{\lambda} \in \mathrm{conv}(S)$ since $\lambda_1 \leq T(\{1\})$ and $\lambda_2 \leq T(\{2\})$. Thus, let us now consider the case where $\lambda_1 = T(\{1\})$. If $\lambda_2 < T(\{2\})$ and $\lambda_1 + \lambda_2 < T(\{1,2\})$ then we could increase $\boldsymbol{\lambda}$, so that we can assume $\lambda_2 < T(\{2\})$ and conclude $\lambda_1 + \lambda_2 = T(\{1,2\})$ due to the subadditivity of T. The case $\lambda_2 = T(\{2\})$ can be treated analogously. Thus, $\mathcal{R}(T) \subseteq \mathrm{conv}(S) \subseteq \mathrm{conv}(S)\downarrow$. □

We remark that the cardinality of generating set of $\mathcal{R}(T)$ in Lemma 24 is 2 or 1, where the latter happens iff $T(\{1,2\}) = T(\{1\}) + T(\{2\})$.

Lemma 25. *Let* $\{\boldsymbol{\lambda}\}$ *be the generating set of* $\mathcal{R}$ *and* $\mathbf{n} = (n_1, \cdots, n_\ell)$ *be an integral solution of the ILP of Corollary* 9*. If* $\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0}$ *and* $\mathcal{G}$ *is the multiset corresponding to the* $\mathbf{n}$*, then* $\boldsymbol{\lambda} \in \mathcal{S}(\mathcal{G})$*, i.e., there exists a feasible choice of* α_Y *satisfying (*7.3a*)-(*7.3c*).*

Proof. The constraints of ILP in Corollary 9 are

$$\begin{aligned} n_2 + n_3 &\geq \lambda_1, \\ n_1 + n_3 &\geq \lambda_2, \\ n_1 + n_2 &\geq \lambda_1 + \lambda_2 \end{aligned}$$

and the recovery sets are given by

$$\begin{aligned} \mathcal{Y}_1 &= \Big\{\{2\}, \{1,3\}\Big\}, \\ \mathcal{Y}_2 &= \Big\{\{1\}, \{2,3\}\Big\}. \end{aligned}$$

Now, set the parameters as follows

$$\begin{aligned} \alpha_{\{2\}} &= \min\{n_2, \lambda_1\}, \\ \alpha_{\{1\}} &= \min\{n_1, \lambda_2\}, \\ \alpha_{\{1,3\}} &= \max\{0, \lambda_1 - n_2\}, \\ \alpha_{\{2,3\}} &= \max\{0, \lambda_2 - n_1\} \end{aligned}$$

It should be noted that since $n_1 + n_2 \geq \lambda_1 + \lambda_2$, we cannot have $n_2 < \lambda_1$ and $n_1 < \lambda_2$. Thus, it is easy to verify that

$$\begin{aligned} \alpha_{\{2\}} + \alpha_{\{1,3\}} &= \lambda_1, \\ \alpha_{\{1\}} + \alpha_{\{2,3\}} &= \lambda_2, \\ \alpha_{\{1\}} + \alpha_{\{1,3\}} &\leq n_1, \\ \alpha_{\{2\}} + \alpha_{\{2,3\}} &\leq n_2, \text{ and} \\ \alpha_{\{1,3\}} + \alpha_{\{2,3\}} &\leq n_3. \end{aligned}$$

Only the latter inequality needs a case analysis. Let us assume $n_2 \geq \lambda_1$ and $n_1 \geq \lambda_2$, then $\alpha_{\{1,3\}} + \alpha_{\{2,3\}} = 0 \leq n_3$. If $n_2 < \lambda_1$ and $n_1 \geq \lambda_2$, then $\alpha_{\{2,3\}} = 0$, $\alpha_{\{1,3\}} = \lambda_1 - n_2$, and $\alpha_{\{1,3\}} + \alpha_{\{2,3\}} = \lambda_1 - n_2$ which is at most n_3 due to $n_2 + n_3 \geq \lambda_1$. The other case, that is, $n_2 \geq \lambda_1$ and $n_1 < \lambda_2$ follows analogously. □

Definition 21. *For a set $\emptyset \neq S \subset \mathbb{N}$, we denote by $\mathrm{Simpl}(S)$ the set of all non-zero vectors in $\langle \{\mathbf{e}_i \mid i \in S\} \rangle$ over $\mathbb{F}_2$.*

Proposition 9. *([76, Theorem1]) For each $\emptyset \neq S \subseteq [k]$, $\#\,\mathrm{Simpl}(S) = 2^s - 1$ and $\mathcal{S}(\mathrm{Simpl}(S)) = \mathcal{R}(T)$, where $s = \#S$ and $T\colon 2^{[k]} \to \mathbb{N}$ is given by $T(U) = 2^{s-1}$ for all $U \subseteq [k]$ satisfying $U \cap S \neq \emptyset$ and $T(U) = 0$ otherwise (for all $U \subseteq [k]$ with $U \cap S = \emptyset$).*

Theorem 21. *For the desired service rate region $\mathcal{R}$ given by*

$$\mathcal{R} = \Big\{ \boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0} \,:\, \lambda_1 \leq X, \lambda_2 \leq Y, \lambda_1 + \lambda_2 \leq \Sigma \Big\},$$

where X, Y, Σ are non-negative integers with $\max\{X, Y\} \leq \Sigma \leq X+Y$, we have $n(\mathcal{R}) = \left\lceil \frac{X+Y+\Sigma}{2} \right\rceil$.

Proof of Theorem 21. The proof consists of a converse (lower bound) and an achievability (upper bound).

Converse: The desired service rate region $\mathcal{R}$ is given by:

$$\mathcal{R} = \left\{ (\lambda_1, \lambda_2) \in \mathbb{R}^2_{\geq 0} \,:\, \lambda_1 \leq X, \lambda_2 \leq Y, \lambda_1 + \lambda_2 \leq \Sigma \right\}.$$

It is easy to see that $\mathcal{R} = \mathcal{R}(T)$ for $T\colon 2^{[2]} \to \mathbb{N}$ defined as $T(\emptyset) = 0$, $T(\{1\}) = X$, $T(\{2\}) = Y$, and $T(\{1,2\}) = \Sigma$. Since $\max\{X, Y\} \leq \Sigma \leq X+Y$ holds, so the condition $\max\{T(\{1\}), T(\{2\})\} \leq T(\{1,2\}) \leq T(\{1\}) + T(\{2\})$ is satisfied which means that the function $T\colon 2^{[2]} \to \mathbb{N}$ is monotone and subadditive. Thus, we can apply Lemma 24 to

obtain the generating set of $\mathcal{R}(T)$ which is given by

$$S = \left\{ \lambda^{(1)} = (X, \Sigma - X), \lambda^{(2)} = (\Sigma - Y, Y) \right\}$$

The inequalities (7.5) from Lemma 23 read

$$\begin{aligned} n_1 + n_3 &\geq X = \max\{X, \Sigma - Y\}, \\ n_2 + n_3 &\geq Y = \max\{Y, \Sigma - X\}, \\ n_1 + n_2 &\geq \Sigma = \max\{\Sigma, \Sigma\}, \end{aligned}$$

so that summing up and dividing by two gives

$$n = n_1 + n_2 + n_3 \geq \frac{X + Y + \Sigma}{2}.$$

Since n is an integer, we obtain $n(\mathcal{R}) \geq \left\lceil \frac{X+Y+\Sigma}{2} \right\rceil$.

We could also use Theorem 18 for proving the converse. Since $\max\{X, Y\} \leq \Sigma \leq X + Y$, it can be simply confirmed that none of the constraints $\lambda_1 \leq X$, $\lambda_2 \leq Y$, $\lambda_1 + \lambda_2 \leq \Sigma$ are strictly redundant in $\mathbb{R}^2_{\geq 0}$. Thus, by applying Theorem 18, we directly get the stated lower bound.

Achievability: First, for the ease of notation, let us define $\frac{1}{2} \cdot \mathrm{Simpl}(\{i, j\}) \triangleq \{\mathbf{e}_i, \mathbf{e}_j\}$ for two different positive integers i and j. Note that the cardinality of $\frac{L}{2} \cdot \mathrm{Simpl}(\{i, j\})$ for some $L \in \mathbb{Z}_{\geq 0}$, is computed as $\left\lceil \frac{L}{2} \cdot \# \mathrm{Simpl}(\{i, j\}) \right\rceil = \left\lceil \frac{3L}{2} \right\rceil$ and the service rate region of $\{\mathbf{e}_i, \mathbf{e}_j\}$ contains the service rate region of the $\mathrm{Simpl}(\{i, j\})$ scaled by a factor of $\frac{1}{2}$, i.e.,

$$\begin{aligned} \mathcal{S}(\{\mathbf{e}_i, \mathbf{e}_j\}) &= \left\{ \boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0} \mid \lambda_i \leq 1, \lambda_j \leq 1 \right\} \\ &\supseteq \left\{ \boldsymbol{\lambda} \in \mathbb{R}^k_{\geq 0} \mid \lambda_i \leq 1, \lambda_j \leq 1, \lambda_i + \lambda_j \leq 1 \right\}. \end{aligned}$$

For the upper bound on $n(\mathcal{R})$, i.e., the constructive part, we need to select the multiplicities of $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_1 + \mathbf{e}_2$ in $\mathcal{G}$, a multiset of points in $\mathrm{PG}(2-1,2)$, such that $\mathcal{S}(\mathcal{G}) \supseteq \mathcal{R}(T)$. Let $\mathcal{G} = \cup_{i\in[3]}\mathcal{G}^{(i)}$ where

- $\mathcal{G}^{(1)}$ consists of $\Sigma - Y$ copies of $\mathrm{Simpl}(\{1\})$
- $\mathcal{G}^{(2)}$ consists of $\Sigma - X$ copies of $\mathrm{Simpl}(\{2\})$
- $\mathcal{G}^{(3)}$ consists of $\frac{L}{2}$ copies of $\mathrm{Simpl}(\{1,2\})$

where $L = X + Y - \Sigma$. Thus, the cardinality of the multiset $\mathcal{G}$ is given by

$$(\Sigma - Y) + (\Sigma - X) + \left\lceil \frac{3(X+Y-\Sigma)}{2} \right\rceil = \left\lceil \frac{X+Y+\Sigma}{2} \right\rceil.$$

By construction, $\mathcal{R}(\mathcal{G}) \supseteq \mathcal{R}(T)$ for $T(\emptyset) = 0, T(\{1\}) = X, T(\{2\}) = Y, T(\{1,2\}) = \Sigma$. The reason is that

$$\begin{aligned}
\mathcal{S}(\mathcal{G}^{(1)}) &\supseteq \left\{\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0} \mid \lambda_1 \leq \Sigma - Y, \lambda_2 = 0\right\}, \\
\mathcal{S}(\mathcal{G}^{(2)}) &\supseteq \left\{\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0} \mid \lambda_1 = 0, \lambda_2 \leq \Sigma - X\right\}, \\
\mathcal{S}(\mathcal{G}^{(3)}) &\supseteq \left\{\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0} \mid \lambda_1 \leq L, \lambda_2 \leq L, \lambda_1 + \lambda_2 \leq L\right\}.
\end{aligned}$$

Thus, it can be easily confirmed that

$$\mathcal{S}(\mathcal{G}) \supseteq \left\{\boldsymbol{\lambda} \in \mathbb{R}^2_{\geq 0} \mid \lambda_1 \leq X, \lambda_2 \leq Y, \lambda_1 + \lambda_2 \leq \Sigma\right\} = \mathcal{R}.$$

That is, the proposed storage scheme $\mathcal{G}$ obviously can satisfy the demands in $\mathcal{R}$. □

7.4 Example of Storage-Efficient Schemes that Cover Given Rate Regions

Consider a scenario where $k = 2$ data objects, movies "a" and "b", are stored redundantly across multiple nodes in a coded storage system. At each time, each node can serve at most one request and each user can request to download at most one of the two movies a and b. It is known that the number of users who are interested in downloading the movie a and b is less than or equal to α (i.e., $\lambda_a \leq \alpha$) and β (i.e., $\lambda_b \leq \beta$), respectively. Also, it is known that the total number of users in the area is at most γ (i.e., $\lambda_a + \lambda_b \leq \gamma$). This means that the desired service rate region of this storage system is a bounded set $\mathcal{R}$ defined as follows:

$$\mathcal{R} = \{\lambda_a, \lambda_b \geq 0, \lambda_a \leq \alpha, \lambda_b \leq \beta, \lambda_a + \lambda_b \leq \gamma\}. \tag{7.20}$$

Two natural questions that arise in the design of this distributed storage system are the following: 1) What is the minimum number of storage nodes $n(\mathcal{R})$ required to serve all request vectors (λ_a, λ_b) in the set $\mathcal{R}$? 2) How should the files a and b be stored redundantly in $n(\mathcal{R})$ storage nodes (i.e., what is the most storage-efficient redundancy scheme)?

Using the example shown in Figure 7.1, we illustrate how the storage-minimizing scheme varies with the shape of the service rate region that we wish to cover. Let $\beta = 4$

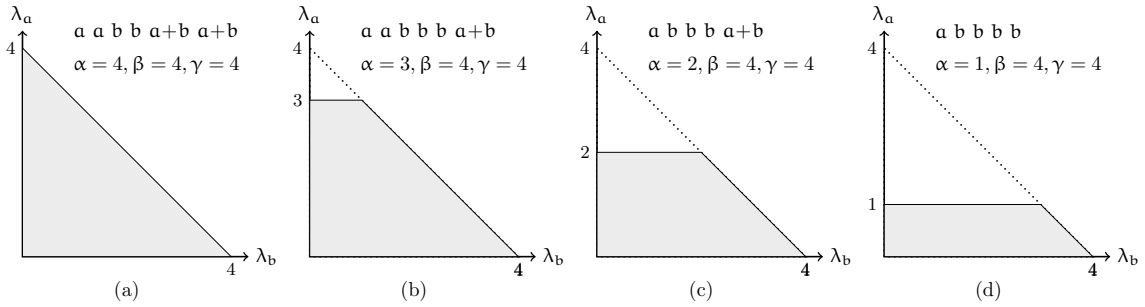

Figure 7.1: Four service rate regions defined by the constraints $\lambda_a, \lambda_b \geq 0, \lambda_a \leq \alpha, \lambda_b \leq \beta, \lambda_a + \lambda_b \leq \gamma$, and their corresponding storage schemes that cover them with a minimum number of nodes.

and $\gamma = 4$, and $\alpha \in \{1, 2, 3, 4\}$. The corresponding four storage-minimizing redundancy schemes (one for each α) together with their service rate regions are shown in Figure 7.1. In Figure 7.1(a), the rate region is dominated by points (λ_a, λ_b) for which the demands for a and b are complementary to each other, that is, if λ_a is high then λ_b is low, and vice-versa. In this case, adding two coded nodes $a + b$ is the most storage-efficient way for achieving the service rate region. On the other hand, in Figure 7.1(d), where the demand for movie b dominates the total request rate $\lambda_a + \lambda_b$, the best storage scheme does not have any coded nodes; it simply replicates object b four times, and keeps just one uncoded copy of a.

8. CONCLUSIONS AND FUTURE DIRECTIONS

The emergence of flexible and affordable cloud storage and computing systems has resulted in an exponential growth in the amount of data stored and processed in the cloud data centers. To accommodate this ever growing demand for storage and computing, cloud services are implemented over a large-scale distributed data storage system. In addition to providing low-cost and reliable content access, cloud services are expected to 1) provide private data access for the users and 2) handle a large number of requests simultaneously. Achieving these goals opens up many challenges. In this thesis, we addressed some of these challenges by developing novel algorithms and analyzing the fundamental limits of their performance metrics. In what follows, we outline the contributions of this thesis briefly and provide a number of related open problems and potential future directions.

8.1 Privacy in Distributed Systems

In part I of this thesis, we studied the problems appear in providing private data access in distributed systems. In particular, we addressed some of the challenges that arise in the Private Information Retrieval (PIR) and Private Linear Transformation (PLT) problems.

In Chapter 2 and Chapter 3, we studied the fundamental limits of single-server and multi-server settings of single-message (information-theoretic) PIR in the presence of a coded side information, respectively. Considering two different types of privacy, namely (W, S)-privacy and W-privacy, we characterized the capacity of the problem under two different models depending on whether the support set of the user's coded side information includes the requested message or not. Moreover, for each of the considered settings, we developed a novel algorithm that achieves the capacity. Our capacity results uncovered the surprising insight that by having only *one* random linear combination of M messages as side information, the privacy requirements of the user can be satisfied as efficient as (in

terms of download cost) the setting in which the user has M random messages separately as side information.

One natural question that remains open is that how much the capacity will increase if we relax the assumption that the servers know the considered model, i.e., whether the side information is a function of the demand or not. Characterizing the capacity for the settings in which the distributions of the demand and side information support index sets as well as the coding coefficients are nonuniform remains as another open problem. Such settings can find application in PIR scenarios where the data items can have different popularities, such as hot data. Another practically motivated, yet not studied, scenario of PIR with side information is when the support size of side information is a random variable whose realization is initially unknown at the server(s). The problem of PIR with side information has been mainly studied for the two types of uncoded and linearly coded side information in the literature. However, it remains open whether different types of side information, such as nonlinear-coded side information, can be leveraged in order to further reduce the download cost. Another potential direction for future work is to characterize the capacity of the single-server PIR when the user has multiple coded side information and/or wants multiple messages from the server. Our initial attempts at studying these settings suggest that there is a close relation between these problems and the problem of single-server private computation with coded side information.

In Chapter 4, we studied the fundamental limits of the multi-server setting of PLT problem under the joint privacy guarantee, where the identity of the entire set of messages in the support set of the demanded linear combinations must be kept private. We focused on the setting in which the coefficient matrix of the required linear combinations is a MDS matrix. For this setting, we established an upper bound on the capacity of the PLT for the whole range of problem parameters, and we showed the tightness of the proposed upper bound for some special cases of the problem. Several related problems remained opened.

An immediate future direction is to characterize the capacity of the multi-server PLT problem in general for all problem parameters. It is also interesting to explore the capacity of single-server or multi-server PLT problem in the presence of a prior side information. Characterizing the capacity of multi-server PLT under the individual privacy guarantee remains as another open problem. The multi-server PLT has been studied only for the replicated databases where the database is replicated over all servers. Another interesting future direction is to investigate the fundamental limits of the PLT problem for the settings in which the databases are coded. In the single-server and multi-server settings of the PLT problem, it has been assume that the user wishes to compute multiple linear combinations of a subset of the files in the database. However, the capacity of the practically motivated scenario where the user wants to compute non-linear functions of files remains open.

8.2 Service Rate of Distributed Systems

In part II of this thesis, we studied the problem of handling a large number of concurrent data access requests in distributed storage and computing systems. In particular, we addressed some of the challenges arise in the context of using the service rate region as a metric to design erasure-coded distributed systems.

In Chapter 5, we introduced a graph representation of codes, referred to as recovery graph, to capture recovery sets of a linear code, and showed that the problem of service rate allocation for a given linear code is equivalent to the fractional matching problem on the recovery graph associated with the code. This enabled us to characterize the service rate region for binary Simplex codes. We also introduced the notion of integral service rate region, where allocations are constrained to be integers. We proved that the problem of characterizing an integral service rate region can be viewed as a generalization of the problem of designing primitive multiset batch codes. A natural future direction is to analyze the service rate region for non-binary Simplex codes and other coding schemes that

are implemented in practice using the graph-based techniques. Exploring connections between the general batch codes and the problem of (integral and general) service rate region is another interesting future direction.

In Chapter 6, we proposed a novel geometric technique for addressing the problem of characterizing the service rate region of a given linear storage scheme without explicitly knowing the list of all possible recovery sets. The proposed geometric technique provides a set of half-spaces whose intersection (forms a polytope) surrounds the service rate region of a given linear storage scheme. In particular, by leveraging the introduced geometric technique, we derived upper bounds on the service rate regions of the binary first order Reed-Muller codes and binary simplex codes. Moreover, we showed how the derived upper bounds can be achieved by developing an efficient request allocation schemes for the vertices of the corresponding polytope. Utilizing the geometric technique to investigate the service rate regions of other common coding schemes such as MDS codes, second order Reed-Muller codes, non-binary Reed-Muller codes, and non-binary simplex codes are amongst the most natural future directions. It is also interesting to explore the service rate region of non-linear storage schemes.

In Chapter 7, we studied the problem of designing the underlying linear storage scheme for a coded distributed storage system storing k files where a desired service rate region $\mathcal{R}$ of the system is given and the goal is 1) to determine the minimum number of storage nodes $n(\mathcal{R})$ (or a lower bound on $n(\mathcal{R})$) for serving all data access vectors inside the set $\mathcal{R}$ and 2) to design the most storage-efficient redundancy scheme with the service rate region that covers $\mathcal{R}$. Toward this goal, we proposed three general lower bounds for $n(\mathcal{R})$. Also, for $k = 2$, we characterized $n(\mathcal{R})$, i.e., we showed that the proposed lower bounds are tight via designing a novel storage-efficient redundancy scheme with $n(\mathcal{R})$ storage nodes and the service rate region covering $\mathcal{R}$. A natural future direction is to characterize the minimum number of storage nodes $n(\mathcal{R})$ for any arbitrary number of files k. Depending

on the application, one may be interested in using a particular code with some desired properties. Then, one interesting open problem is to find the best generator matrix of a code (the service rate regions of two generator matrices of the same linear code might not be the same) with respect to the service rate region, that is the generator matrix that maximizes the volume of the service rate region with a given number of nodes/servers or covers a given service rate region with the minimum number of storage nodes.

REFERENCES

[1] M. Aktaş, G. Joshi, S. Kadhe, F. Kazemi, and E. Soljanin, "Service rate region: A new aspect of coded distributed system design," *IEEE Transactions on Information Theory*, vol. 67, no. 12, pp. 7940–7963, 2021.

[2] B. Chor and N. Gilboa, "Computationally private information retrieval," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 304–313, 1997.

[3] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," in *38th Annual Symposium on Foundations of Computer Science*, pp. 364–373, Oct 1997.

[4] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *IEEE Symposium on Foundations of Computer Science*, pp. 41–50, 1995.

[5] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond, "Breaking the O$(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval," in *43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 261–270, 2002.

[6] W. Gasarch, "A survey on private information retrieval," *The Bulletin of the EATCS*, vol. 82, no. 72-107, p. 113, 2004.

[7] S. Yekhanin, "Private information retrieval," *Communications of the ACM*, vol. 53, no. 4, pp. 68–73, 2010.

[8] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transactions on Information Theory*, vol. 63, pp. 4075–4088, July 2017.

[9] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *IEEE Transactions on Information Theory*, vol. 64, pp. 2361–2370, April 2018.

[10] K. Banawan and S. Ulukus, "Multi-message private information retrieval," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 1898–1902, June 2017.

[11] K. Banawan and S. Ulukus, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Transactions on Information Theory*, vol. 64, pp. 6842–6862, Oct 2018.

[12] N. B. Shah, K. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *2014 IEEE International Symposium on Information Theory*, pp. 856–860, IEEE, 2014.

[13] T. H. Chan, S. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2842–2846, June 2015.

[14] R. Tajeddine and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 1411–1415, 2016.

[15] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Transactions on Information Theory*, vol. 64, pp. 1945–1956, March 2018.

[16] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2852–2856, June 2015.

[17] S. R. Blackburn and T. Etzion, "Pir array codes with optimal pir rates," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2658–2662, June 2017.

[18] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 647–664, 2017.

[19] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from mds coded data in distributed storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, 2018.

[20] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.

[21] S. Li and M. Gastpar, "Single-server multi-message private information retrieval with side information," in *2018 56th Annual Allerton Conf. on Commun., Control, and Computing*, Oct 2018.

[22] S. Kadhe, B. Garcia, A. Heidarzadeh, S. E. Rouayheb, and A. Sprintson, "Private information retrieval with side information: The single server case," in *2017 55th Annual Allerton Conf. on Commun., Control, and Computing*, pp. 1099–1106, Oct 2017.

[23] A. Heidarzadeh, S. Kadhe, B. Garcia, S. E. Rouayheb, and A. Sprintson, "On the capacity of single-server multi-message private information retrieval with side information," in *2018 56th Annual Allerton Conf. on Commun., Control, and Computing*, pp. 180–187, Oct 2018.

[24] Z. Chen, Z. Wang, and S. A. Jafar, "The capacity of t-private information retrieval with private side information," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4761–4773, 2020.

[25] S. Li and M. Gastpar, "Converse for multi-server single-message pir with side information," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, IEEE, 2020.

[26] R. Tandon, "The capacity of cache aided private information retrieval," in *55th Annual Allerton Conf. on Commun., Control, and Computing*, pp. 1078–1082, Oct 2017.

[27] Y. Wei, K. Banawan, and S. Ulukus, "Cache-aided private information retrieval with partially known uncoded prefetching: Fundamental limits," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 1126–1139, June 2018.

[28] Y. Wei, K. Banawan, and S. Ulukus, "Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching," *IEEE Transactions on Information Theory*, pp. 1–1, 2018.

[29] S. P. Shariatpanahi, M. J. Siavoshani, and M. A. Maddah-Ali, "Multi-message private information retrieval with private side information," in *2018 IEEE Information Theory Workshop (ITW)*, pp. 335–339, IEEE, May 2018.

[30] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "Capacity of single-server single-message private information retrieval with coded side information," in *2018 IEEE Information Theory Workshop (ITW)*, pp. 1–5, Nov 2018.

[31] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Private information retrieval with private coded side information: The multi-server case," in *2019 57th*

Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1098–1104, IEEE, 2019.

[32] A. Heidarzadeh, S. Kadhe, S. E. Rouayheb, and A. Sprintson, "Single-server multi-message individually-private information retrieval with side information," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 1042–1046, July 2019.

[33] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Single-server single-message online private information retrieval with side information," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 350–354, July 2019.

[34] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "Capacity of single-server single-message private information retrieval with private coded side information," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 1662–1666, July 2019.

[35] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Multi-server private information retrieval with coded side information," in *2019 IEEE 16th Canadian Workshop on Information Theory (CWIT)*, pp. 1–6, 2019.

[36] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "The role of coded side information in single-server private information retrieval," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 25–44, 2021.

[37] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, "Private information retrieval with side information," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2032–2043, 2019.

[38] F. Kazemi and A. Sprintson, "Multi-server private linear transformation with joint privacy," in *2021 XVII International Symposium" Problems of Redundancy in Information and Control Systems"(REDUNDANCY)*, pp. 182–187, IEEE, 2021.

[39] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy.," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[40] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *21st International conference on data engineering (ICDE'05)*, pp. 217–228, IEEE, 2005.

[41] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 93–106, 2008.

[42] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Matching anonymized and obfuscated time series to users' profiles," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 724–741, 2018.

[43] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125, IEEE, 2008.

[44] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Privacy of dependent users against statistical matching," *IEEE Transactions on Information Theory*, vol. 66, no. 9, pp. 5842–5865, 2020.

[45] Y. H. Hwang, "Iot security & privacy: threats and challenges," in *Proceedings of the 1st ACM workshop on IoT privacy, trust, and security*, pp. 1–1, 2015.

[46] M. Abomhara and G. M. Køien, "Security and privacy in the internet of things: Current status and open issues," in *2014 international conference on privacy and security in mobile systems (PRISMS)*, pp. 1–8, IEEE, 2014.

[47] N. Takbiri, V. Shejwalkar, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Leveraging prior knowledge asymmetries in the design of location privacy-

preserving mechanisms," *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 2005–2009, 2020.

[48] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pp. 618–623, IEEE, 2017.

[49] A. Heidarzadeh, N. Esmati, and A. Sprintson, "Single-server private linear transformation: The individual privacy case," *arXiv preprint arXiv:2106.05222*, 2021.

[50] A. Heidarzadeh, N. Esmati, and A. Sprintson, "Single-server private linear transformation: The joint privacy case," *arXiv preprint arXiv:2106.05220*, 2021.

[51] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[52] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476–489, 2011.

[53] K. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *2009 47th Annual Allerton Conf. on Commun., Control, and Comput.*, 2009.

[54] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[55] M. Rabinovich and O. Spatscheck, *Web caching and replication*, vol. 67. Addison-Wesley Boston, USA, 2002.

[56] M. A. Maddah-Ali and U. Niesen, "Coding for caching: fundamental limits and practical challenges," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 23–29, 2016.

[57] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[58] T. X. Tran, F. Kazemi, E. Karimi, and D. Pompili, "Mobee: Mobility-aware energy-efficient coded caching in cloud radio access networks," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 461–465, IEEE, 2017.

[59] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4481–4493, 2016.

[60] I. Tamo and A. Barg, "Bounds on locally recoverable codes with multiple recovering sets," in *2014 IEEE International Symposium on Information Theory*, pp. 691–695, IEEE, 2014.

[61] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5843–5855, 2014.

[62] G. Joshi, Y. Liu, and E. Soljanin, "Coding for fast content download," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 326–333, IEEE, 2012.

[63] G. Liang and U. C. Kozat, "Fast cloud: Pushing the envelope on delay performance of cloud storage with coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 6, pp. 2012–2025, 2014.

[64] G. Joshi, E. Soljanin, and G. W. Wornell, "Efficient replication of queued tasks for latency reduction in cloud systems," in *53rd Annual Allerton Conference on Communication, Control, and Computing*, pp. 107–114, 2015.

[65] G. Joshi, E. Soljanin, and G. W. Wornell, "Efficient redundancy techniques for latency reduction in cloud systems," *TOMPECS*, vol. 2, no. 2, pp. 12:1–12:30, 2017.

[66] N. B. Shah, K. Lee, and K. Ramchandran, "The mds queue: Analysing the latency performance of erasure codes," in *2014 IEEE International Symposium on Information Theory*, pp. 861–865, IEEE, 2014.

[67] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, and E. Hyytia, "Reducing latency via redundant requests: Exact analysis," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 1, pp. 347–360, 2015.

[68] N. B. Shah, K. Lee, and K. Ramchandran, "When do redundant requests reduce latency?," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 715–722, 2015.

[69] S. Kadhe, E. Soljanin, and A. Sprintson, "Analyzing the download time of availability codes," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1467–1471, IEEE, 2015.

[70] M. F. Aktaş, S. Kadhe, E. Soljanin, and A. Sprintson, "Download time analysis for distributed storage codes with locality and availability," *arXiv:1912.09765*, Dec 2019.

[71] M. Noori, E. Soljanin, and M. Ardakani, "On storage allocation for maximum service rate in distributed storage systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 240–244, IEEE, 2016.

[72] P. Peng and E. Soljanin, "On distributed storage allocations of large files for maximum service rate," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 784–791, IEEE, 2018.

[73] M. Aktaş, S. E. Anderson, A. Johnston, G. Joshi, S. Kadhe, G. L. Matthews, C. Mayer, and E. Soljanin, "On the service capacity region of accessing erasure coded content," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 17–24, IEEE, 2017.

[74] S. E. Anderson, A. Johnston, G. Joshi, G. L. Matthews, C. Mayer, and E. Soljanin, "Service rate region of content access from erasure coded storage," in *2018 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2018.

[75] F. Kazemi, E. Karimi, E. Soljanin, and A. Sprintson, "A combinatorial view of the service rates of codes problem, its equivalence to fractional matching and its connection with batch codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 646–651, June 2020.

[76] F. Kazemi, S. Kurz, and E. Soljanin, "A geometric view of the service rates of codes problem and its application to the service rate of the first order Reed-Muller codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 66–71, June 2020.

[77] F. Kazemi, S. Kurz, E. Soljanin, and A. Sprintson, "Efficient storage schemes for desired service rate regions," in *2020 IEEE Information Theory Workshop (ITW)*, pp. 1–5, April 2021.

[78] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proceedings of the thirty-sixth annual ACM Symp. on Theory of computing*, pp. 262–271, ACM, 2004.

[79] S. Kadhe, A. Heidarzadeh, A. Sprintson, and O. O. Koyluoglu, "On an equivalence between single-server pir with side information and locally recoverable codes," in *2019 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2019.

[80] Y.-P. Wei and S. Ulukus, "The capacity of private information retrieval with private side information under storage constraints," *arXiv:1806.01253*, June 2018.

[81] S. Patel, G. Persiano, and K. Yeo, "Private stateful information retrieval," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, (New York, NY, USA), pp. 1002–1019, ACM, 2018.

[82] A. Heidarzadeh and A. Sprintson, "Private computation with side information: The single-server case," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1657–1661, July 2019.

[83] C. Tian, H. Sun, and J. Chen, "Capacity-achieving private information retrieval codes with optimal message size and upload cost," *IEEE Transactions on Information Theory*, vol. 65, no. 11, pp. 7613–7627, 2019.

[84] R. Roth, *Introduction to Coding Theory*. New York, NY, USA: Cambridge University Press, 2006.

[85] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, pp. 3880–3897, Jun 2019.

[86] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *IEEE Symposium on Foundations of Computer Science*, 1995.

[87] Z. Chen, Z. Wang, and S. A. Jafar, "The capacity of t-private information retrieval with private side information," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4761–4773, 2020.

[88] M. J. Siavoshani, S. P. Shariatpanahi, and M. A. Maddah-Ali, "Private information retrieval for a multi-message scenario with private side information," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3235–3244, 2021.

[89] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *2018 Iran Workshop on Communication and Information Theory (IWCIT)*, pp. 1–6, IEEE, 2018.

[90] A. Heidarzadeh and A. Sprintson, "Private computation with individual and joint privacy," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 1112–1117, IEEE, 2020.

[91] S. A. Obead and J. Kliewer, "Achievable rate of private function retrieval from mds coded databases," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2117–2121, IEEE, 2018.

[92] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Capacity of private linear computation for coded databases," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 813–820, IEEE, 2018.

[93] B. Tahmasebi and M. A. Maddah-Ali, "Private sequential function computation," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1667–1671, IEEE, 2019.

[94] M. Aliasgari, O. Simeone, and J. Kliewer, "Private and secure distributed matrix multiplication with flexible communication load," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2722–2734, 2020.

[95] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private function computation for noncolluding coded databases," *arXiv:2003.10007*, 2020.

[96] M. Aliasgari, O. Simeone, and J. Kliewer, "Distributed and private coded matrix computation with flexible communication load," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1092–1096, IEEE, 2019.

[97] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proc. of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 245–250, 2001.

[98] M. F. Aktaş, A. Behrouzi-Far, E. Soljanin, and P. Whiting, "Load balancing performance in distributed storage with regular balanced redundancy," *arXiv:1910.05791*, 2019.

[99] M. Sardari, R. Restrepo, F. Fekri, and E. Soljanin, "Memory allocation in distributed storage networks," in *2010 IEEE International Symposium on Information Theory*, pp. 1958–1962, IEEE, June 2010.

[100] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," *ACM Transactions on Storage (TOS)*, vol. 9, no. 1, p. 3, 2013.

[101] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Transactions on Information Theory*, vol. 58, no. 11, pp. 6925–6934, 2012.

[102] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 569–574, IEEE, 2014.

[103] G. Joshi, Y. Liu, and E. Soljanin, "On the delay-storage trade-off in content download from coded distributed storage systems," *IEEE Journal on Selected Areas in*

Communications, vol. 32, no. 5, pp. 989–997, 2014.

[104] S. Kadhe, E. Soljanin, and A. Sprintson, "When do the availability codes make the stored data more available?," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 956–963, IEEE, 2015.

[105] M. F. Aktaş and E. Soljanin, "Heuristics for analyzing download time in MDS coded storage systems," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 1929–1933, IEEE, 2018.

[106] M. F. Aktaş, E. Najm, and E. Soljanin, "Simplex queues for hot-data download," in *Proceedings of the SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pp. 35–36, ACM, 2017.

[107] 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*, pp. 262–271, ACM, 2004.

[108] V. Skachek, "Batch and pir codes and their connections to locally repairable codes," in *Network Coding and Subspace Designs*, pp. 427–442, Springer, 2018.

[109] H. Lipmaa and V. Skachek, "Linear batch codes," in *Coding Theory and Applications*, pp. 245–253, Springer, 2015.

[110] Z. Wang, H. M. Kiah, Y. Cassuto, and J. Bruck, "Switch codes: Codes for fully parallel reconstruction," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2061–2075, 2017.

[111] D. Leong, A. G. Dimakis, and T. Ho, "Distributed storage allocations," *IEEE Trans. Information Theory*, vol. 58, no. 7, pp. 4733–4752, 2012.

[112] N. Alon, P. Frankl, H. Huang, V. Rödl, A. Rucinski, and B. Sudakov, "Large matchings in uniform hypergraphs and the conjectures of erdős and samuels," *J. Comb. Theory, Ser. A*, vol. 119, pp. 1200–1215, 2012.

[113] Y.-H. Kao, A. G. Dimakis, D. Leong, and T. Ho, "Distributed storage allocations and a hypergraph conjecture of erdős," in *2013 IEEE International Symposium on Information Theory*, pp. 902–906, 2013.

[114] P. Erdős, "A problem on independent r-tuples," in *ARTICLE IN PRESS B. Bollobás et al./Journal of Combinatorial Theory, Series A*, Citeseer, 1965.

[115] E. R. Scheinerman and D. H. Ullman, *Fractional graph theory: a rational approach to the theory of graphs*. Courier Corporation, 2011.

[116] D. West, *Introduction to Graph Theory*. Featured Titles for Graph Theory Series, Prentice Hall, 2001.

[117] V. Voloshin, *Introduction to Graph and Hypergraph Theory*. Nova Science Publishers, 2009.

[118] R. T. Rockafellar, *Convex analysis*, vol. 28. Princeton University Press, 1970.

[119] C. Jones, E. C. Kerrigan, and J. Maciejowski, "Equality set projection: A new algorithm for the projection of polytopes in halfspace representation," tech. rep., Cambridge University Engineering Dept, 2004.

[120] J. Matousek and B. Gärtner, *Understanding and using linear programming*. Springer Science & Business Media, 2007.

[121] M. A. Tsfasman and S. G. Vladut, "Geometric approach to higher weights," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1564–1588, 1995.

[122] S. Dodunekov and J. Simonis, "Codes and projective multisets," *The Electronic Journal of Combinatorics*, vol. 5, no. 1, p. 37, 1998.

[123] A. Beutelspacher, B. Albrecht, and U. Rosenbaum, *Projective geometry: from foundations to applications*. Cambridge University Press, 1998.

[124] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *Transactions of the IRE professional group on electronic computers*, no. 3, pp. 6–12, 1954.

[125] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," tech. rep., Massachusetts Inst. of Tech. Lexington Lincoln Lab., 1953.

[126] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.