Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

PRIVATE INFORMATION RETRIEVAL AND FUNCTION COMPUTATION FOR NONCOLLUDING CODED DATABASES

by
Sarah A. Obead

The rapid development of information and communication technologies has motivated many data-centric paradigms such as big data and cloud computing. The resulting paradigmatic shift to cloud/network-centric applications and the accessibility of information over public networking platforms has brought information privacy to the focal point of current research challenges. Motivated by the emerging privacy concerns, the problem of private information retrieval (PIR), a standard problem of information privacy that originated in theoretical computer science, has recently attracted much attention in the information theory and coding communities. The goal of PIR is to allow a user to download a message from a dataset stored on multiple (public) databases without revealing the identity of the message to the databases and with the minimum communication cost. Thus, the primary performance metric for a PIR scheme is the PIR rate, which is defined as the ratio between the size of the desired message and the total amount of downloaded information.

The first part of this dissertation focuses on a generalization of the PIR problem known as *private computation (PC)* from distributed storage system (DSS). In PC, a user wishes to compute a function of f variables (or messages) stored in n noncolluding coded databases, i.e., databases storing data encoded with an $[n, k]$ linear storage code, while revealing no information about the desired function to the databases. Here, *colluding* databases refers to databases that communicate with each other in order to deduce the identity of the computed function. First, the problem of private linear computation (PLC) for linearly encoded DSS is considered. In PLC, a user wishes to privately compute a linear combination over the f messages. For the PLC

problem, the PLC capacity, i.e., the maximum achievable PLC rate, is characterized. Next, the problem of private polynomial computation (PPC) for linearly encoded DSS is considered. In PPC, a user wishes to privately compute a multivariate polynomial of degree at most g over f messages. For the PPC problem an outer bound on the PPC rate is derived, and two novel PPC schemes are constructed. The first scheme considers Reed-Solomon coded databases with Lagrange encoding and leverages ideas from recently proposed star-product PIR and Lagrange coded computation. The second scheme considers databases coded with systematic Lagrange encoding. Both schemes yield improved rates compared to known PPC schemes. Finally, the general problem of PC for arbitrary nonlinear functions from a replicated DSS is considered. For this problem, upper and lower bounds on the achievable PC rate are derived and compared.

In the second part of this dissertation, a new variant of the PIR problem, denoted as *pliable private information retrieval (PPIR)* is formulated. In PPIR, the user is pliable, i.e., interested in *any* message from a desired subset of the available dataset. In the considered setup, f messages are replicated in n noncolluding databases and classified into Γ classes. The user wishes to retrieve *any* one or more messages from *multiple* desired classes, while revealing no information about the identity of the desired classes to the databases. This problem is termed as multi-message PPIR (M-PPIR), and the single-message PPIR (PPIR) problem is introduced as an elementary special case of M-PPIR. In PPIR, the user wishes to retrieve *any one* message from *one* desired class. For the two considered scenarios, outer bounds on the M-PPIR rate are derived for arbitrary number of databases. Next, achievable schemes are designed for n replicated databases and arbitrary n. Interestingly, the capacity of PPIR, i.e., the maximum achievable PPIR rate, is shown to match the capacity of PIR from n replicated databases storing Γ messages. A similar insight is shown to hold for the general case of M-PPIR.

PRIVATE INFORMATION RETRIEVAL AND FUNCTION COMPUTATION FOR NONCOLLUDING CODED DATABASES

by
Sarah A. Obead

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

May 2022

APPROVAL PAGE

PRIVATE INFORMATION RETRIEVAL AND FUNCTION COMPUTATION FOR NONCOLLUDING CODED DATABASES

Sarah A. Obead

Dr. Jörg Kliewer, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Nirwan Ansari, Committee Member — Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Alexander Haimovich, Committee Member — Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Mary Wootters, Committee Member — Date
Assistant Professor of Computer Science and of Electrical Engineering,
Stanford University, Stanford, CA

BIOGRAPHICAL SKETCH

Author: Sarah A. Obead

Degree: Doctor of Philosophy

Date: May 2022

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2022
- Master of Science in Telecommunications,
  New Jersey Institute of Technology, Newark, NJ, 2017
- Bachelor of Science in Information Technology,
  University of Benghazi, Benghazi, Libya, 2010

Major: Electrical Engineering

Presentations and Publications:

S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private polynomial computation for noncolluding coded databases," *IEEE Trans. Inf. Forens. Secur.*, to be published. doi:10.1109/TIFS.2022.3166667.

S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private linear computation for noncolluding coded databases," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 847–861, March 2022.

S. A. Obead, B. N. Vellambi, and J. Kliewer, "Strong coordination over noisy channels," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2716–2738, May 2021.

S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "On the capacity of private nonlinear computation for replicated databases," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 25–28, 2019, pp. 1–5.

S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private polynomial computation for noncolluding coded databases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 1677–1681.

S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Capacity of private linear computation for coded databases," in *Proc. 56th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2–5, 2018, pp. 813–820.

S. A. Obead and J. Kliewer, "Achievable rate of private function retrieval from MDS coded databases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2117–2121.

S. A. Obead, J. Kliewer, and B. N. Vellambi, "Joint coordination-channel coding for strong coordination over noisy channels based on polar codes," in *Proc. 55th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 3–6, 2017, pp. 580–587.

S. A. Obead, B. N. Vellambi, and J. Kliewer, "Strong coordination over noisy channels: Is separation sufficient?" in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 25–30, 2017, pp. 2840–2844.

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

﴿وَ قُلِ اعْمَلُوا فَسَيَرَى ٱللَّهُ عَمَلَكُمْ وَ رَسُولُهُ وَ ٱلْمُؤْمِنُونَ﴾

صَدَقَ ٱللَّهُ ٱلْعَظِيمُ

والديّ الكِرام، أُمي الغالية و أبِي الحبيب

السّند و العضد و السّاعد، إخوتي و أخواتي

أساتذتي الأفاضل و كل من علمني حرفاً

كُلّ مَن ساندني بالكلمة، بالإلهام، بالتّحفيز، بالدّعاء، وبالأُمنيات الطيّبة

بِكم و مِنكم و مَعكم وَصلت

بِراً و وفاءً و إحساناً أُهدي لكُم ثَمرة جُهدي

"هذهِ بِضاعتُكم رُدّت إِليكم"

With my genuine gratefulness and warmest regard, I dedicate this work to my beloved parents, my dear siblings, my esteemed teachers, and to those who have embraced me with their prayers, inspiration, motivation, and kindness.

ACKNOWLEDGMENT

I would like to first thank my dissertation advisor Dr. Jörg Kliewer for his support, patience, and willingness to give his time so generously. His guidance helped me throughout the course of my studies at NJIT. His insightful feedback and useful critiques pushed me to sharpen my thinking and brought my work to a higher level.

Besides my advisor, I would also like to thank Dr. Nirwan Ansari, Dr. Alexander Haimovich, Dr. Ali Abdi, and Dr. Mary Wootters who graciously agreed to serve on my dissertation committee.

I would also like to extend my deepest gratitude to my collaborators Dr. Badri Vellambi, Dr. Hsuan-Yin Lin, and Dr. Eirik Rosnes for their kind advise, valuable feedback, and constructive discussions.

Many thanks to the friends I have made at NJIT and the members of the Center of Wireless Information Processing (CWiP) whom have shared this journey with me and provided a welcoming environment.

Finally, I must express my very profound gratitude to my parents, Zakia Shoiab and Dr. Ali Obead, for providing me with unfailing support, for relentlessly believing in me, and for keeping me in their prayers every step of the way. Also, to my siblings for providing me with support to pursue my aspirations not only through the process of researching and writing this dissertation but throughout all my years of study. This accomplishment would not have been possible without every one of them. Thank you!

TABLE OF CONTENTS

Chapter **Page**

1 INTRODUCTION 1

1.1 Private Information Retrieval 2

1.2 Private Computation 5

1.2.1 Main Contributions 7

1.3 Pliable Private Information Retrieval 8

1.3.1 Main Contributions 11

1.4 Organization of the Dissertation 12

2 PRELIMINARIES 14

2.1 Notation 14

2.2 Private Computation Problem Statement and System Model 16

2.3 MDS-PIR Capacity-Achieving Codes 19

3 PRIVATE LINEAR COMPUTATION FOR NONCOLLUDING CODED DATABASES 22

3.1 Introduction 22

3.2 Converse Bound 23

3.3 Private Linear Computation From Coded DSSs 28

3.3.1 Query Generation for PLC 30

3.3.2 Recovery of Desired Function Evaluation 41

3.3.3 Sign Assignment and Redundancy Elimination 42

3.3.4 Privacy 46

3.3.5 Achievable PLC Rate 50

3.4 Conclusion 50

4 PRIVATE POLYNOMIAL FUNCTION COMPUTATION FOR NONCOLLUDING CODED DATABASES 51

4.1 Introduction 51

4.1.1 Background 52

TABLE OF CONTENTS
(Continued)

Chapter **Page**

4.2 Converse Bound 53

4.3 General PPC Scheme for RS-Coded DSSs 55

4.3.1 Lagrange Coded Computation 56

4.3.2 PPC Achievable Rate Matrix 57

4.3.3 Generic Query Generation 58

4.3.4 Sign Assignment and Redundancy Elimination 60

4.3.5 Recovery and Privacy 61

4.3.6 Achievable PPC Rate 62

4.4 PPC Scheme for Systematic RS-Encoded DSSs 65

4.4.1 PPC Systematic Achievable Rate Matrix 65

4.4.2 Sign Assignment and Redundancy Elimination 68

4.4.3 Recovery and Privacy 69

4.4.4 Achievable PPC Rate 69

4.4.5 Special Case: PMC Scheme 72

4.5 Numerical Results 76

4.6 Conclusion 78

5 GENERAL PRIVATE COMPUTATION OF NONLINEAR FUNCTIONS FROM REPLICATED DATABASES 79

5.1 Converse Bound 80

5.2 Achievability 81

5.2.1 Achievable Scheme for Theorem 8 82

5.3 Discussion of the Outer Bound of Theorem 7 86

5.4 Special Case: Private Monomial Computation 86

5.5 Conclusion 87

6 MULTI-MESSAGE PLIABLE PRIVATE INFORMATION RETRIEVAL . 89

6.1 Preliminaries 90

TABLE OF CONTENTS
(Continued)

Chapter **Page**

6.1.1 System Model 90
6.1.2 Problem Statement 91
6.1.3 Special Cases 96
6.2 Pliable Private Information Retrieval 97
6.2.1 Single Server PPIR 97
6.2.2 PPIR over Replicated DSS 99
6.3 Multi-Message Pliable Private Information Retrieval 106
6.3.1 Single Server M-PPIR 107
6.3.2 M-PPIR over Replicated DSS 108
6.4 Conclusion 114
7 SUMMARY 116
APPENDIX A PROOF OF LEMMA 1 118
APPENDIX B PROOF OF LEMMA 3 120
APPENDIX C PROOF OF LEMMA 4 121
APPENDIX D PROOF OF THEOREM 4 124
APPENDIX E PROOF OF LEMMA 5 126
APPENDIX F PROOF OF LEMMA 9 128
APPENDIX G PROOF OF LEMMA 10 131
APPENDIX H PROOF OF LEMMA 11 134
REFERENCES 136

LIST OF TABLES

Table		Page
3.1	Auxiliary Query Sets for Example 4	35
3.2	PLC Query Sets for $v = 1$ after Sign Assignment	45
3.3	PLC Query Sets for $v = 3$ after Sign Assignment	48
4.1	PMC Query Sets for $v = 1$	73
4.2	Decoded and Computed Symbols from the PMC Query Sets for $v = 1$ from Table 4.1	75
5.1	Query Sets for a DSS with n Noncolluding Replicated Databases Storing f Messages	84
6.1	Query Sets for PPIR from Replication-based DSS	105

LIST OF FIGURES

Figure **Page**

1.1 Simple private information retrieval scheme. 3

1.2 Simple overview of some of the PIR problem extensions and variations. . 6

2.1 System model for PC from an $[n, k]$ coded DSS storing f messages. . . . 17

4.1 PPC rates as a function of the storage code rate $\alpha = k/n$ for $f = 2$, $k = 2$, $g = 2$, and $\mu = \mathsf{M}_2^{\mathrm{c}}(2) = \mathsf{M}_2(2) = 5$. For simplicity, we assume $\mathsf{H}_{\min} = 1$. 77

4.2 PPC rates as a function of the storage code rate $\alpha = k/n$ for $f = 10$, $k = 20$, $g = 2$, and $\mu = \mathsf{M}_2^{\mathrm{c}}(10) = \mathsf{M}_2(10) = 65$. For simplicity, we assume $\mathsf{H}_{\min} = 1$. 77

5.1 PMC rate R versus the number of messages f for the retrieval of nonparallel monomials over the field $\mathbb{F}_3$. 87

6.1 Index-mapping of f messages classified into Γ classes using class and sub-class indices, i.e., $\theta_{\gamma,\beta_\gamma} \in \mathcal{M}_\gamma \subset [f]$, $\forall \gamma \in [\Gamma]$. 91

6.2 System model for M-PPIR from an n replicated noncolluding databases storing f messages classified into Γ classes. The user intends to download λ messages each out of η desired classes. 92

6.3 Index mapping for M-PPIR problem of Example 8. The user selects $\Omega = \{1, 3\}$, i.e., $\gamma_1 = 1$ and $\gamma_2 = 3$ and wants to retrieve any two messages from each class. Highlighted in red, are two arbitrary sub-class indices from each desired class. 93

CHAPTER 1

INTRODUCTION

In today's age of information, the rapid development of information and communication technologies (ICT) has brought about voluminous amounts of data. Big data is a term that refers to vast, and continuously growing, amounts of data generated by different sources including, for example, media platforms, public databases, web logs, and internet of things (IoT) sensors. Thus, this massive volume of data introduce a new set of challenges from data storage, retrieval, search, and transfer. As a result, cloud computing paradigms for data storage and computation, with on-demand remote access to powerful computing and data storage services over the internet, has emerged as an indispensable resource for both enterprises and individuals. With this widespread use of cloud computing and with information accessibility over public networking platforms, many major concerns have risen regarding information privacy, specifically with respect to data and computation privacy.

Motivated by emerging privacy concerns, the problem of private information retrieval (PIR), a standard problem of information privacy established originally in theoretical computer science and cryptography, has recently attracted much attention in the information theory and coding communities. As a result, many interesting variations of PIR problem have surfaced. The goal of PIR is to allow a user to *efficiently* retrieve a *specific* message from a dataset stored on a database without revealing any information about the desired message to the database. The efficiency of a PIR scheme is primarily measured by the PIR rate, which is defined as the ratio between the size of desired message to the total amount of downloaded information and the maximum of this rate is known as the capacity of PIR.

In this dissertation, we focus on two directions centered around PIR. The first direction is a generalization of the PIR problem, when the data is numerical, known as *private computation (PC)*. This generalization, not only considers the private retrieval of the numerical data but also arbitrary functions evaluated on the said data. The second direction is a relaxation of the PIR problem where the users are flexible with their demands and with their privacy requirements. We denote this variation as *pliable private information retrieval (PPIR)*. We start with a brief background about PIR, then we present an overview of selected works related to private computation and pliable information retrieval.

1.1 Private Information Retrieval

The problem of PIR from public databases, introduced by Chor *et al.* [1], has been the focus of attention for several decades in the computer science community (see, e.g., [2]–[4]). The goal of PIR is to allow a user to privately access an arbitrary message stored in a set of databases, i.e., without revealing any information of the identity of the requested message to each database. If the users do not have any side information on the data stored in the databases, they can trivially request the content of the whole database to hid the identity of the desired message. This trivial solution achieves perfect privacy, however, with high storage and communication costs. Alternatively, Chor *et al.* [1] proposed a strategy to store the messages in at least two databases which provides the user with multiple point of views of the stored data while each database maintain a single view of what the user query and thus privacy can be ensured. To illustrate this concept, consider the following example

Example 1 *(Private Information Retrieval) Suppose that we have two databases each storing a dataset* $\mathbf{W}$ *consisting of* f *equal-length messages denoted by* $\mathbf{W} = \{\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(f)}\}$. *Consider a user that is interested in retrieving* $\boldsymbol{W}^{(\theta)}$ *for some* $\theta \in \{1, 2, \ldots, f\}$ *while keeping the message index* θ *hidden from each database.*

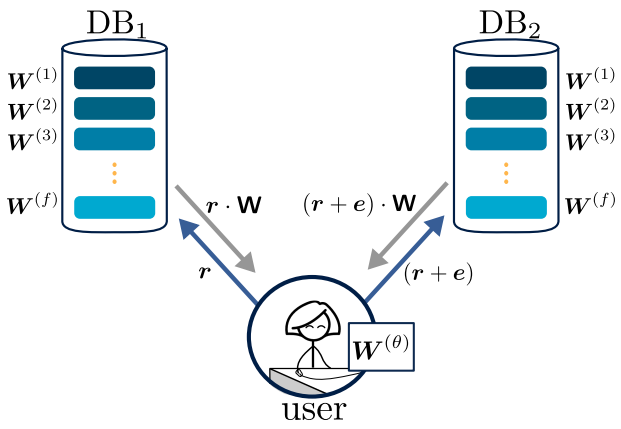

Figure 1.1 Simple private information retrieval scheme.

The PIR solution is illustrated in Figure 1.1. The user sends a uniformly random binary vector of length f, $\boldsymbol{r} \in \{0,1\}^f$ *to the first database. Then sends* $\boldsymbol{r}+\boldsymbol{e}$ *to the second database, where* $\boldsymbol{e} \in \{0,1\}^f$ *is an identity vector with* 1 *in the* θ*-th position. The first database answers with* $\boldsymbol{r}\cdot\mathbf{W}$ *and the second database answers with* $(\boldsymbol{r}+\boldsymbol{e})\cdot\mathbf{W}$. *The user substitute the answers of the databases as* $(\boldsymbol{r}+\boldsymbol{e})\cdot\mathbf{W}-\boldsymbol{r}\cdot\mathbf{W}=\boldsymbol{e}\cdot\mathbf{W}$ *and obtains the desired message* $\boldsymbol{W}^{(\theta)}$. *Here, the privacy follows from the fact that both binary vectors* $(\boldsymbol{r}+\boldsymbol{e})$ *and* $\boldsymbol{r}$ *appear uniformly distributed from the perspective of each database. Thus, no information about the desired message can be deduced, unless, the two databases share the user queries, i.e., collude. Finally, since the user obtains one message by downloading two linear combinations, the PIR rate is* $\frac{1}{2}$ *compared to* $\frac{1}{f}$ *with the trivial solution.*

Hence, the design of PIR protocols has focused on the case when multiple databases store the messages. This connects to the active and renowned research area of distributed storage systems (DSSs) usually referred to as *coded DSSs*. In coded DSSs, the data is encoded by an $[n,k]$ linear code, i.e., a storage code that generates n codewords by linear combination of k information words, then distributed and stored across n storage nodes [5]. Using coding techniques, coded DSSs possess

many practical features and benefits such as high reliability, efficient repairability, robustness, and security [6].

Recently, the aspect of minimizing the communication cost, e.g., the required rate or bandwidth of privately querying the databases with the desired requests and downloading the corresponding information from the databases has attracted a great deal of attention in the information theory and coding communities. Thus, the renewed interest in PIR primarily focused on the study and design of efficient PIR protocols for coded DSSs, starting with the fundamental limit of PIR from DSSs encoded with simple codes as for example, repetition codes in [7]. This was followed by an immense amount of work characterizing the capacity, i.e., the maximum achievable rate, for many variants of the original PIR problem (see e.g., [8]–[29]). Such variations include additional interesting privacy, storage, or security constraints. For example, in [9], multi-message PIR (M-PIR) has been proposed where the user can request more than one messages from replicated databases, i.e., databases encoded with simple repetition codes. Recently, the special case of private retrieval of multiple messages with side information, i.e., the user already knows a subset of the messages stored in the database, was studied in [25]–[28]. In [10], [30] a bounded number of databases might be colluding, adversarial (byzantine), or non-responsive. Achievable schemes for PIR from storage encoded with maximum distance separable (MDS) erasure codes have been presented in [31]–[33] and the capacity of MDS-coded storage has been established in [8].

Finally, the broad interest in PIR problem from computer science, coding theory, and information theory communities is due to its close connection to a variety of interdisciplinary problems such as oblivious transfer [34], multi-party computation [35], [36], secret sharing [37], [38], locally recoverable and decodable codes [39] and [40], respectively, and index coding [41], [42].

1.2 Private Computation

Motivated by privacy concerns in distributed computing a recently proposed generalization of the PIR problem [43]–[46] addresses private computation (PC) for functions of the messages stored in the database, also denoted as private function retrieval [47]. In PC a user has access to a number of databases and intends to compute a function of messages stored in these databases. This function is kept private from the databases, as they may be under the control of an adversary. The PC rate, defined as the ratio of the desired amount of information and the total amount of downloaded information is the main performance metric in this line of research. Accordingly, the PC capacity is defined as the maximum of all achievable PC rates over all possible PC protocols. In [43], [47], the capacity and achievable rates for the case of privately computing a given *linear* function, referred to as private linear computation (PLC), were derived as a function of the number of messages and the number of databases, respectively, for the scenario of noncolluding replicated databases. Interestingly, the obtained PLC capacity is equal to the PIR capacity of [7].

The extension to the coded case is addressed in [45], [46]. In [45], private polynomial computation (PPC) over t colluding and systematically coded databases was considered by generalizing the PIR scheme of [33]. In [45], the functions to be computed are polynomials of degree at most g, and a PC rate equal to the best asymptotic PIR rate of MDS-coded storage (when the number of messages tends to infinity) is achieved for $g = t = 1$ (the case of linear function retrieval and noncolluding databases). An alternative PPC approach was recently proposed in [46] for polynomials with higher degree, i.e., $g > 1$, by employing Reed-Solomon coded databases with Lagrange encoding. For low code rates, the scheme improves on the PC rate of [45]. The special case of private monomial computation (PMC) was addressed in [48], where the PMC capacity for an asymptotically large field size and under a mild technical condition on the size of the base field was derived.

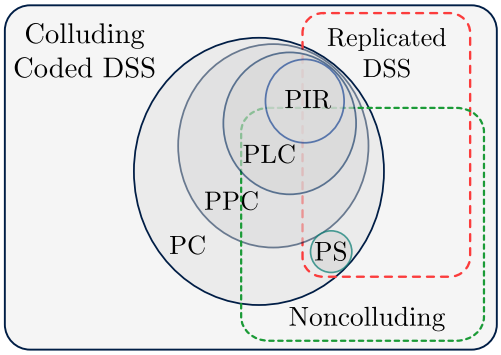

Figure 1.2 Simple overview of some of the PIR problem extensions and variations.

The technical condition on the size of the base field can be shown to be satisfied for a sufficiently large base field. Recently, PC was also extended to the single server scenario, where all messages are stored uncoded on a *single* server, with side information in [49], [50]. Here, the authors derived the capacity for both coded and uncoded side information under two different privacy conditions on the messages of the desired linear combination.

Finally, a separate but relevant form of PC, the private search (PS) problem [44] considers mapping records replicated over n noncolluding databases to binary search patterns. Each pattern represents the search result of one value out of a set of candidate alphabets. The asymptotic capacity, i.e., the information retrieval rate for PS with a large alphabet size, of privately retrieving one search pattern is found to match the asymptotic capacity of PIR for the special case of *balanced* PS. In a balanced PS scenario, the nonlinearly dependent search patterns are assumed to contain equal amount of information. An overview of how these extensions align together can be seen in Figure 1.2.

In another line of research for the case of noncolluding databases, in [15] the authors proposed two PIR protocols for a DSS where data is stored using a non-MDS linear code. For a large class of linear codes, the proposed protocols are shown to achieve, respectively, the nonasymptotic and asymptotic MDS-coded PIR capacity,

i.e., the capacity of PIR over noncolluding MDS-coded DSSs, established in [8], referred to as the MDS-PIR capacity in the sequel. The first family of non-MDS codes for which the PIR capacity is known was found in [51], [52]. Further, PIR on linearly-coded databases for the case of colluding databases was also proposed in [15], [32], [33], [53].

1.2.1 Main Contributions

For the PC with noncolluding databases, the capacity results for arbitrary linearly-coded DSSs have not been addressed so far in the open literature to the best of our knowledge. In the first part of dissertation, we intend to fill this void by proposing four PC schemes from linearly-coded DSSs and deriving outer bounds on the PC rate over all possible PC protocols. Our contributions are outlined as follows:

- We fully characterize the capacity of PLC from noncolluding coded DSSs encoded with family of non-MDS storage codes known as MDS-PIR capacity-achieving codes [15]. Towards that end, we prove a converse bound for the coded PLC capacity and construct a novel PLC scheme that achieves a rate equal to the converse bound, i.e., a capacity-achieving scheme. The proposed PLC scheme strictly generalize the replication-based PC schemes of [43], [47] and the optimal PIR scheme of [15].
- In [46], the authors were mainly concerned with constructing PPC schemes with a focus on preserving privacy against colluding DSSs. We, in contrast, aim our attention at establishing the capacity of the PPC setup. Towards that aim, we first extend our converse proof of the coded PLC to the coded PPC problem and derive an outer bound on the PPC rate from a DSS encoded with MDS-PIR capacity-achieving codes [15].Then we provide PPC solutions that minimize the download cost. Specifically, we propose two novel PPC schemes from RS-coded DSSs (one for systematic encoding) by generalizing our capacity-achieving PLC scheme and leveraging ideas from star-product PIR [33] and Lagrange coded computation [54]. Our schemes improve on the rates of the PPC schemes presented in [45], [46].
- Finally, we consider general private *nonlinear* computation for replication-based DSSs. We provide a general converse result and construct a novel PC scheme. When the message size is large and the candidate functions are the independent messages and one arbitrary nonlinear function of these, we show that the proposed PC scheme achieves a rate equal to the PC capacity. Moreover, when the number of messages grows, the PC rate approaches the outer bound on the PC capacity derived from [55, Thm. 1] and thus becomes the capacity itself. Finally, we discuss

how a PC scheme should be designed to achieve the PC capacity for general non-linear function computation.

1.3 Pliable Private Information Retrieval

Today, a growing amount of traffic over the internet is generated by content-based applications. Content-based applications are applications that provide access to information (e.g., search engines, video libraries, and digital galleries) generated by individuals or businesses. Examples of well-known content-based applications include News-feed applications, social media, and content delivery networks.

This prominent presence of content-type versus traditional message-type traffic in communication networks has recently caught the attention of the network information theory community. The main distinction is that content-type traffic is able to deliver a message within a prescribed content type instead of specific messages. For example, [56] explored the benefits of designing network and channel codes tailored to content-type requests. This work was shortly followed by the introduction of *pliable index coding (PICOD)* [57]. Index coding (IC) [41] is a well-known network information theory problem that aims to minimize the broadcast rate for communicating of messages noiselessly to n receivers, where each receiver has a different subset of messages as side-information. PICOD is a variant of the IC problem where the receivers, having a set of messages as side information, are interested in *any* other message they do not have. This is in contrast to classical IC, where the receivers are interested in *specific* messages.

Following the introduction of PICOD, converse bounds on the PICOD broadcast rate were derived in [58]. Moreover, variations of the PICOD problem are considered in [59]–[61]. Specifically, in private PICOD [60], the privacy is defined by the inability of each user to decode more than one message. In decentralized PICOD [59], the system model does not include a central transmitter with knowledge of all f messages. Alternatively, the n users share among themselves messages that can only depend on

their local set of side information messages. This work has been recently extended to secure decentralized PICOD in [61] where security is defined such that users are not allowed to gain any information about any message outside their side information set except for one message. Finally, a number of constructions for PICOD are proposed in [62]–[67].

In this dissertation, motivated by emerging content-based applications and inspired by content-type coding [56], and PICOD [57], we introduce the pliable private information retrieval (PPIR) problem as a new variant of the classical PIR problem. The PIR problem and its available variations traditionally aim to retrieve a *specific* information message from a database without revealing the identity of the desired message to the database and with the minimum communication cost. This broad aim encompasses most of the work in the PIR literature discussed in Section 1.1 and Section 1.2. However, in (single-message) PPIR, we consider that a set of f messages are replicated in n noncolluding databases and the user is flexible with her demand. She wishes to retrieve *any* message from a desired subgroup of the dataset, i.e., *class*, without revealing the identity of the desired class to each database.

One motivating example for PPIR is given by retrieving a news article of a desired topic without revealing the topic to the database. Another example would be to privately retrieve a movie from a desired genre without revealing the genre, i.e., the classification of the movie, to the content database in order to avoid targeted recommendation or undesired profiling.

To illustrate the difference between PIR and PPIR, consider the following example.

Example 2 *(Pliable Private Information Retrieval) Suppose that we have a single database consisting of $f = 5$ equal-length messages denoted by $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(f)}$ and classified into $\Gamma = 2$ classes. Suppose that the messages with indices $\mathcal{M}_1 = \{1, 3\}$ are members of the first class $\gamma = 1$ and the remaining messages, i.e., messages with*

indices $\mathcal{M}_2 = \{2, 4, 5\}$ *are members of the second class* $\gamma = 2$*. Consider a user that is interested in retrieving* any *message from class* $\gamma = 1$ *while keeping the class index hidden from the database. If the user has access to the message membership in each class, i.e., the user knows* $\mathcal{M}_1 = \{1, 3\}$ *and* $\mathcal{M}_2 = \{2, 4, 5\}$*, there are two intuitive solutions.*

- *One solution is to select one of the members of the desired class uniformly at random and attempt to privately retrieve that message using a PIR solution. For achieving information-theoretic privacy in the single-server case, it is well-known that the user has to download the entire database to hide the identity of the desired message [2]. As a result, the information retrieval rate is* $\mathsf{R} = \frac{1}{f} = \frac{1}{5}$*.*
- *Alternatively, in PPIR, the user selects* Γ *messages, one from each class, uniformly at random. Let the selected messages indices be denoted by* θ_1 *and* θ_2*, respectively, for each class. The user then queries the database for the two messages* $\boldsymbol{W}^{(\theta_1)}$ *and* $\boldsymbol{W}^{(\theta_2)}$ *resulting in* $\mathbb{P}(\gamma = 1|\theta_1, \theta_2) = \mathbb{P}(\gamma = 2|\theta_1, \theta_2) = \frac{1}{\Gamma}$*. In other words, perfect information-theoretic privacy is achieved as the desired message can be from any of the two classes. As a result, the information retrieval rate is* $\mathsf{R} = \frac{1}{\Gamma} = \frac{1}{2}$*. This matches the PIR rate for the case where we have only* $f = 2$ *messages stored in the database, indicating an apparent trade-off between a reduction of privacy with respect to five messages versus two classes and the download rate.*

It can be easily seen from Example 2 that the PPIR rate reduces to the PIR rate if $\Gamma = f$, i.e., there is only one message in each class. Accordingly, the PPIR problem is also a strict generalization of the PIR problem. Moreover, we are able to achieve a significant gain in the information retrieval rate with the PPIR solution if $f >> \Gamma$. Finally, in the PPIR problem, we make no assumptions about the user accessibility to the messages membership in each class. As a result, the PIR solution for Example 2 is not valid if the user doesn't know the identity of the messages that belong to the desired class.

To the best of our knowledge, the problem of pliable private information retrieval has not been examined before in the open literature. However, there has been some related work on other variations of PIR that explore opportunities to trade off perfect privacy with privacy leakage to increase the communication rate. The following are

some representative examples: [68] initiated the study of *leaky* private information retrieval and derived upper and lower bounds on the download rate for some bounded $\epsilon > 0$ information leakage about the message identity, an arbitrary number of messages f, and $n = 2$ replicated databases. Another related variant of PIR is known as weakly-private information retrieval (WPIR) [69]–[71]. In WPIR the perfect privacy requirement on the identity of the desired message is relaxed by allowing bounded average leakage between the queries and the corresponding requested message index. The leakage is measured by different information leakage measures including mutual information and maximal leakage (MaxL) metrics [72]–[74]. In particular, [69], [70] studied the trade-offs between different parameters of PIR, such as download rate, upload cost, and access complexity while relaxing the privacy requirement.

1.3.1 Main Contributions

In the second part of this dissertation, we introduce the multi-message PPIR (M-PPIR) problem. We consider the setup where we have a dataset of f messages replicated in n noncolluding databases and classified into Γ classes. Our contributions are outlined as follows:

- As a tutorial introduction to for the PPIR problem, we first consider single-server PPIR. For this problem, we establish the capacity and provide a simple capacity-achieving scheme.
- Then, we fully characterize the PPIR capacity from replicated noncolluding databases. Towards that end, we prove a novel converse bound on the PPIR rate for an arbitrary number of messages f, classes Γ, and databases n and we construct a capacity-achieving PPIR scheme. The significant of our PPIR converse is that it indicates an independence between the maximum achievable rate and the total number of messages f. Moreover, the derived converse bound matches the capacity of PIR when the user wishes to privately retrieve one of Γ messages only.
- Similar to the PPIR problem, we provide a tutorial introduction to the general M-PPIR problem by first considering the single-server scenario. We prove a converse bound and construct a simple capacity-achieving scheme, thus settling the single-server M-PPIR capacity.

- Next, we consider the M-PPIR problem from replicated noncolluding databases and derive two converse bounds on the achievable M-PPIR rate. The first bound is valid when the number of desired classes is at least half the total number of classes $\eta \geq \frac{\Gamma}{2}$ and the second one holds with $\eta \leq \frac{\Gamma}{2}$. Interestingly, when there is only one message in each class, i.e., $\Gamma = f$, the M-PPIR problem reduces to the M-PIR problem and our converse bounds match the M-PIR bounds.
- Finally, leveraging our construction for the single-server M-PPIR scheme and the M-PIR schemes of [9], we present two achievable M-PPIR schemes for replicated noncolluding databases. The first scheme is for the case when $\eta \geq \frac{\Gamma}{2}$ and the second when $\eta \leq \frac{\Gamma}{2}$. The achievable rates of the proposed schemes match the converse bounds when $\eta \geq \frac{\Gamma}{2}$ and when $\frac{\Gamma}{\eta}$ is an integer number. Thus, we settle the capacity of M-PPIR from replicated databases for the two former cases.

1.4 Organization of the Dissertation

The dissertation proceeds as follows:

In Chapter 2, the notation used through out the dissertation and some basic definitions are outlined. Then, we provide the general system model and the problem statement of private computation from linearly-coded DDSs. This system model is considered for the majority of the work presented in this dissertation. Finally, we introduce a particular family of linear storage codes, i.e., the MDS-PIR Capacity achieving codes.

In Chapter 3, we consider the problem of private linear computation (PLC) for coded databases. For a DSS setup where data is stored using MDS-PIR Capacity achieving codes, we derive an outer bound on the PLC rate. Further, we present a PLC scheme with rate equal to the outer bound and hence settle the PLC capacity for the considered class of linear storage codes.

In Chapter 4, we consider the problem of private polynomial computation (PPC) from a distributed storage system (DSS). For a DSS setup where data is stored using MDS-PIR Capacity achieving codes, we derive an outer bound on the PPC rate and construct two novel PPC schemes. In the first scheme, we consider Reed-Solomon coded databases with Lagrange encoding, which leverages ideas from

recently proposed star-product PIR and Lagrange coded computation. The second scheme considers the special case of coded databases with systematic Lagrange encoding.

In Chapter 5, we consider the general problem of private computation (PC) in a replicated DSS, i.e., the messages are replicated across n noncolluding databases. We provide an information-theoretically accurate achievable PC rate for the scenario of nonlinear computation. For a large message size the rate equals the PC capacity when the candidate functions are the f independent messages and one arbitrary nonlinear function of these. When the number of messages grows, the PC rate approaches an outer bound on the PC capacity. As a special case, we consider private monomial computation (PMC) and numerically compare the achievable PMC rate to the outer bound for a finite number of messages.

In Chapter 6, we formulate the pliable private information retrieval (PPIR) problem. We first provide the general system model and the problem statement of multi-message pliable private information retrieval (M-PPIR) from replicated databases and introduce the single-message PPIR (PPIR) problem as an elementary special case of M-PPIR. For the two considered scenarios we first focus on the case of the single server, i.e., $n = 1$ and derive outer bounds on the M-PPIR rate. Next, we design achievable schemes for the single server case and then extend our results to the case of replicated databases. Interestingly, we show that for PPIR capacity, i.e., the maximum achievable PPIR rate, matches the capacity of PIR with n databases storing Γ messages. A similar insight is shown to hold for the general case of M-PPIR.

Finally, in Chapter 7, we conclude the dissertation with a summary of the results, and provide directions for future research.

CHAPTER 2

PRELIMINARIES

In this chapter, we first present the notation used throughout the dissertation and provide some basic definitions. Then, we provide the general system model and the problem statement of private computation from linearly-coded DDSs. This system model is considered for the majority of the work presented in this dissertation. Namely, for the private computation protocols proposed in Chapters 3, 4, and 5. Finally, we introduce the family of linear storage codes that is considered for the private linear and private polynomial computations, i.e., the MDS-PIR Capacity achieving codes.

2.1 Notation

We denote by $\mathbb{N}$ the set of all positive integers and let $\mathbb{N}_0 \triangleq \{0\} \cup \mathbb{N}$, $[a] \triangleq \{1, 2, \ldots, a\}$, and $[a : b] \triangleq \{a, a+1, \ldots, b\}$ for $a, b \in \mathbb{N}$, $a \leq b$. Random and deterministic quantities are carefully distinguished as follows. A random variable is denoted by a capital Roman letter, e.g., X, while its realization is denoted by the corresponding small Roman letter, e.g., x. Vectors are boldfaced, e.g., $\boldsymbol{X}$ denotes a random vector and $\boldsymbol{x}$ denotes a deterministic vector, respectively. The notation $\boldsymbol{X} \sim \boldsymbol{Y}$ is used to indicate that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are identically distributed. Random matrices are represented by bold sans serif letters, e.g., $\mathbf{X}$, where X represents its realization. In addition, sets are denoted by calligraphic uppercase letters, e.g., $\mathcal{X}$, and $\mathcal{X}^{\mathrm{c}}$ denotes the complement of a set $\mathcal{X}$ in a universe set. We denote a submatrix of X that is restricted in columns by the set $\mathcal{I}$ by $\mathsf{X}|_{\mathcal{I}}$. For a given index set $\mathcal{S}$, we also write $\mathbf{X}^{\mathcal{S}}$ and $Y_{\mathcal{S}}$ to represent $\left\{\mathbf{X}^{(v)} \colon v \in \mathcal{S}\right\}$ and $\left\{Y_j \colon j \in \mathcal{S}\right\}$, respectively. Furthermore, some constants and functions are also depicted by Greek letters or a special font, e.g., X.

The function $\mathsf{H}(X)$ represents the entropy of X, and $\mathrm{I}(X;Y)$ the mutual information between X and Y. The binomial coefficient of a over b, $a, b \in \mathbb{N}_0$, is denoted by $\binom{a}{b}$ where $\binom{a}{b} = 0$ if $a < b$. The notation $\lfloor \cdot \rfloor$ denotes the floor function and $\mathbb{1}(\cdot)$ represents the indicator function, i.e., $\mathbb{1}(\text{statement})$ equals to 1 if the statement holds, and 0 otherwise. $\mathbb{P}[A]$ is the probability that the event A occurs.

We use the customary code parameters $[n, k]$ to denote a code $\mathscr{C}$ over the finite field $\mathbb{F}_q$ of blocklength n and dimension k. A generator matrix of $\mathscr{C}$ is denoted by $\mathsf{G}^{\mathscr{C}}$. A set of coordinates of $\mathscr{C}$, $\mathcal{I} \subseteq [n]$, of size k is said to be an *information set* if and only if $\mathsf{G}^{\mathscr{C}}|_{\mathcal{I}}$ is invertible. $(\cdot)^{\intercal}$ denotes the transpose operator, while $\mathrm{rank}(\mathsf{V})$ denotes the rank of a matrix V. The function $\chi(\boldsymbol{x})$ denotes the support of a vector $\boldsymbol{x}$, and the linear span of a set of vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_a\}$, $a \in \mathbb{N}$, is denoted by $\mathrm{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_a\}$. Finally, $\mathbb{F}_p[z]$ denotes the set of all univariate polynomials over $\mathbb{F}_p$ in the variable z, and we denote by $\deg(\phi(z))$ the degree of a polynomial $\phi(z) \in \mathbb{F}_p[z]$.

A monomial $\boldsymbol{z}^{\boldsymbol{i}}$ in m variables $z_1, \ldots, z_m$ with degree g is written as $\boldsymbol{z}^{\boldsymbol{i}} = z_1^{i_1} z_2^{i_2} \cdots z_m^{i_m}$, where $\boldsymbol{i} \triangleq (i_1, \ldots, i_m) \in \mathbb{N}_0^m$ is the exponent vector with $\mathsf{wt}(\boldsymbol{i}) \triangleq \sum_{j=1}^{m} i_j = g$. The set $\{\boldsymbol{z}^{\boldsymbol{i}} : \boldsymbol{i} \in \mathbb{N}_0^m,\ 1 \leq \mathsf{wt}(\boldsymbol{i}) \leq g\}$ of all monomials in m variables of degree at most g has size

$$\mathsf{M}_g(m) \triangleq \sum_{h=1}^{g} \binom{h+m-1}{h} = \binom{g+m}{g} - 1. \tag{2.1}$$

Moreover, we define a parallel monomial as a monomial resulting from raising another monomial to a positive integer power, i.e., to $\{\boldsymbol{W}^{\boldsymbol{i}} : \boldsymbol{i} \in \mathbb{N}_0^f,\ 1 \leq \mathsf{wt}(\boldsymbol{i}) \leq g,\ \boldsymbol{i} \mid \mathsf{p},\ \mathsf{p} \in \mathcal{P}_g\}$. Here, $\mathcal{P}_g$ denotes the set of prime numbers less or equal to g and $\boldsymbol{i} = (i_1, \ldots, i_m) \mid \mathsf{p}$ means that all nonzero i_j, $j \in [m]$, are divisors of p. For example, for a bivariate monomial over the variables x and y of degree at most $g = 2$ the set of possible monomials is $\{x, y, xy, x^2, y^2\}$. Note that x^2 is a parallel monomial as it can be obtained by raising the monomial x to the power of 2. Thus, x^2 and y^2 are parallel monomials. Denote by $\mathcal{P} = \{\mathsf{p}_1, \ldots, \mathsf{p}_{|\mathcal{P}|}\}$ an arbitrary nonempty subset of

$\mathcal{P}_g$. By applying the Legendre formula for counting the prime numbers less or equal to g, we obtain the number of nonparallel monomials as

$$\widetilde{\mathsf{M}}_g(m) \triangleq \mathsf{M}_g(m) + \sum_{\substack{\forall \mathcal{P} \subseteq \mathcal{P}_g : \mathcal{P} \neq \emptyset, \\ \mathsf{p}_1 \cdots \mathsf{p}_{|\mathcal{P}|} \leq g}} (-1)^{|\mathcal{P}|} \left[\binom{\left\lfloor \frac{g}{\mathsf{p}_1 \cdots \mathsf{p}_{|\mathcal{P}|}} \right\rfloor + m}{\left\lfloor \frac{g}{\mathsf{p}_1 \cdots \mathsf{p}_{|\mathcal{P}|}} \right\rfloor} - 1 \right]. \tag{2.2}$$

Finally, a polynomial $\phi(\boldsymbol{z})$ of degree at most g is represented as $\phi(\boldsymbol{z}) = \sum_{\boldsymbol{i}:\mathsf{wt}(\boldsymbol{i}) \leq g} a_{\boldsymbol{i}} \boldsymbol{z}^{\boldsymbol{i}}$, $a_{\boldsymbol{i}} \in \mathbb{F}_p$. The total number of polynomials in m variables of degree at most g generated with all possible distinct (up to scalar multiplication) $\mathsf{M}_g(m)$-dimensional coefficients vectors defined over $\mathbb{F}_p$ is equal to $\mu_g(m) \triangleq \left(p^{\mathsf{M}_g(m)} - 1\right)/(p-1)$.

We now proceed with a general description for the problem statement of private function computation from linearly-coded DSSs.

2.2 Private Computation Problem Statement and System Model

The PC problem for coded DSSs is described as follows. We consider a DSS that stores in total f independent messages $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$, where each message $\mathbf{W}^{(m)}, m \in [f]$, consists of L symbols $W_1^{(m)}, \ldots, W_{\mathsf{L}}^{(m)}$ chosen independently and uniformly at random from $\mathbb{F}_p$. Thus,

$$\mathsf{H}(\mathbf{W}^{(m)}) = \mathsf{L}, \, \forall \, m \in [f],$$
$$\mathsf{H}(\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}) = f\mathsf{L} \quad \text{(in } p\text{-ary units)}.$$

Let $\mathsf{L} \triangleq \beta k$, for some $\beta, k \in \mathbb{N}$. The DSS stores the f messages encoded using an $[n, k]$ code as follows. Shown in Figure 2.1, first, the symbols of each message $\mathbf{W}^{(m)}$, $m \in [f]$, are presented as a $\beta \times k$ matrix, i.e., $\mathbf{W}^{(m)} = \left(W_{i,j}^{(m)}\right), i \in [\beta], j \in [k]$. Let $\boldsymbol{W}_i^{(m)} = \left(W_{i,1}^{(m)}, \ldots, W_{i,k}^{(m)}\right)$, $i \in [\beta]$, denote a message vector corresponding to the i-th row of $\mathbf{W}^{(m)}$. Second, each $\boldsymbol{W}_i^{(m)}$ is encoded by an $[n, k]$ code $\mathscr{C}$ over $\mathbb{F}_p$ into a length-n codeword $\boldsymbol{C}_i^{(m)} = \left(C_{i,1}^{(m)}, \ldots, C_{i,n}^{(m)}\right)$. The βf generated codewords $\boldsymbol{C}_i^{(m)}$ are then arranged in the array $\mathbf{C} = \left((\mathbf{C}^{(1)})^{\mathsf{T}} | \ldots | (\mathbf{C}^{(f)})^{\mathsf{T}}\right)^{\mathsf{T}}$ of dimensions $\beta f \times n$, where

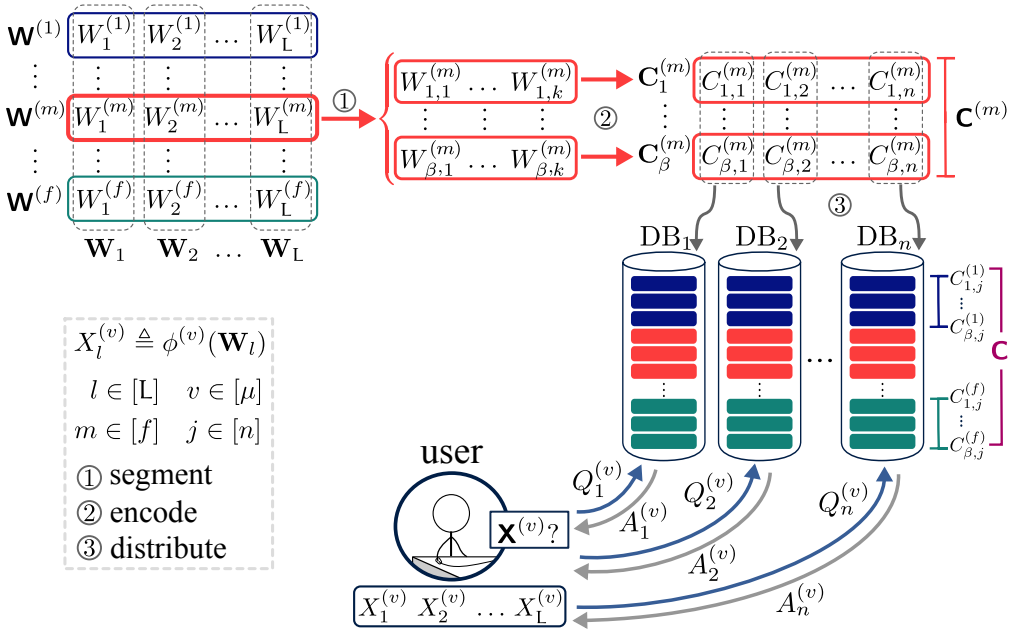

Figure 2.1 System model for PC from an $[n, k]$ coded DSS storing f messages.

$\mathbf{C}^{(m)} = \big((\boldsymbol{C}_1^{(m)})^{\mathsf{T}} | \ldots | (\boldsymbol{C}_\beta^{(m)})^{\mathsf{T}}\big)^{\mathsf{T}}$. Finally, the code symbols $C_{1,j}^{(m)}, \ldots, C_{\beta,j}^{(m)}$, $m \in [f]$, for all f messages are stored on the j-th database, $j \in [n]$.

We consider the case of n noncolluding databases. In private function computation, a user wishes to privately compute exactly one function image $X_l^{(v)} \triangleq \phi^{(v)}(\boldsymbol{W}_l)$, where $\boldsymbol{W}_l = (W_l^{(1)}, \ldots, W_l^{(f)})$, $\forall\, l \in [\mathrm{L}]$, out of μ arbitrary *candidate* functions $\phi^{(1)}, \ldots, \phi^{(\mu)} \colon \mathbb{F}_p^f \to \mathbb{F}_p$ from the coded DSS. Let $\mathbf{X}^{(v)} = \big(X_1^{(v)}, \ldots, X_{\mathrm{L}}^{(v)}\big)$, where $X_1^{(v)}, \ldots, X_{\mathrm{L}}^{(v)}$ are independent and identically distributed according to a prototype random variable $X^{(v)}$ with probability mass function $P_{X^{(v)}}$. Thus,

$$\mathsf{H}(\mathbf{X}^{(v)}) = \mathrm{L}\,\mathsf{H}(X^{(v)}),\ \forall\, v \in [\mu],$$
$$\mathsf{H}(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}) = \mathrm{L}\,\mathsf{H}\big(X^{(1)}, \ldots, X^{(\mu)}\big) \quad \text{(in } p\text{-ary units)},$$

and we let $\mathsf{H}_{\min} \triangleq \min_{v \in [\mu]} \mathsf{H}\big(X^{(v)}\big)$ and $\mathsf{H}_{\max} \triangleq \max_{v \in [\mu]} \mathsf{H}\big(X^{(v)}\big)$.

The user privately selects an index $v \in [\mu]$ and wishes to compute the v-th function while keeping the requested function index v private from each database. In order to retrieve the desired function evaluation $\mathbf{X}^{(v)}$, $v \in [\mu]$, from the coded

DSS, the user sends a query $Q_j^{(v)}$ to the j-th database for all $j \in [n]$ as illustrated in Figure 2.1. The queries are generated by the user without any prior knowledge of the realizations of the candidate functions, consequently, they are independent of the candidate functions evaluations. In other words, we have

$$\mathrm{I}\Big(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}; Q_1^{(v)}, \ldots, Q_n^{(v)}\Big) = 0, \, \forall \, v \in [\mu]. \tag{2.3}$$

In response to the received query, database j generates the answer $A_j^{(v)}$ as a deterministic function of $Q_j^{(v)}$ and the data stored in the database, and then sends it back to the user. Let $\boldsymbol{C}_j \triangleq \big(C_{1,j}^{(1)}, \ldots, C_{\beta,j}^{(1)}, C_{1,j}^{(2)}, \ldots, C_{\beta,j}^{(f)}\big)^{\mathsf{T}}$ denote the f coded chunks that are stored in the j-th database. Thus, $\forall \, v \in [\mu]$,

$$\mathrm{H}\Big(A_j^{(v)} \,\Big|\, Q_j^{(v)}, \boldsymbol{C}_j\Big) = 0, \, \forall \, j \in [n]. \tag{2.4}$$

To guarantee user privacy, in an information-theoretic sense, the query-answer function must be identically distributed for each possible desired function index $v \in [\mu]$ from the perspective of each database $j \in [n]$. In other words, the scheme's queries and answer strings must be independent from the desired function index, therefore, revealing no information about the identity of the desired function evaluation. Moreover, the user must be able to reliably decode the desired function evaluation $\mathbf{X}^{(v)}$. Accordingly, we define a PC protocol for an $[n, k]$ coded DSSs as follows.

Consider a DSS with n noncolluding databases storing f messages using an $[n, k]$ code. The user wishes to retrieve the v-th function evaluation $\mathbf{X}^{(v)}$, $v \in [\mu]$, from the available information $Q_j^{(v)}$ and $A_j^{(v)}$, $j \in [n]$. For a PC protocol, the following conditions must be satisfied $\forall \, v, v' \in [\mu]$, $v \neq v'$, and $\forall \, j \in [n]$,

$$\text{[Privacy]} \qquad (Q_j^{(v)}, A_j^{(v)}, \mathbf{X}^{[\mu]}) \sim (Q_j^{(v')}, A_j^{(v')}, \mathbf{X}^{[\mu]}), \tag{2.5}$$

$$\text{[Recovery]} \qquad \mathrm{H}\big(\mathbf{X}^{(v)} \,\big|\, A_{[n]}^{(v)}, Q_{[n]}^{(v)}\big) = o(\mathrm{L}), \tag{2.6}$$

where any function of L, say $\lambda(\mathsf{L})$, is said to be $o(\mathsf{L})$ if $\lim_{\mathsf{L}\to\infty} \lambda(\mathsf{L})/\mathsf{L} = 0$.

From an information-theoretic perspective, the efficiency of a PC protocol is measured by the *PC rate*, which is defined as follows.

Definition 1 (PC rate and capacity for linearly-coded DSSs) *The exact information-theoretic rate of a PC scheme, denoted by* R*, is defined as the ratio of the minimum desired function size* $\mathsf{L}\,\mathsf{H}_{\min}$ *over the total required download cost, i.e.,*

$$\mathsf{R} \triangleq \frac{\mathsf{L}\,\mathsf{H}_{\min}}{\mathsf{D}},$$

where D *is the total required download cost. The PC* capacity C_{PC} *is the maximum of all achievable PC rates over all possible PC protocols for a given* $[n, k]$ *storage code.*

2.3 MDS-PIR Capacity-Achieving Codes

A PIR protocol for any linearly-coded DSS that uses an $[n, k]$ code to store f messages, named Protocol 1, was proposed in [15]. The PIR rate of Protocol 1 can be derived by finding a *PIR achievable rate matrix* of the underlying storage code $\mathscr{C}$, which is defined as follows.

Definition 2 ([15, Def. 10]) *Let* $\mathscr{C}$ *be an arbitrary* $[n, k]$ *code. A* $\nu \times n$ *binary matrix* $\Lambda^{\mathrm{PIR}}_{\kappa,\nu}(\mathscr{C})$ *is said to be a PIR achievable rate matrix for* $\mathscr{C}$ *if the following conditions are satisfied.*

1. *The Hamming weight of each column of* $\Lambda^{\mathrm{PIR}}_{\kappa,\nu}$ *is* κ*, and*
2. *for each matrix row* $\boldsymbol{\lambda}_i$*,* $i \in [\nu]$*,* $\chi(\boldsymbol{\lambda}_i)$ *always contains an information set of* $\mathscr{C}$*, where* $\chi(\boldsymbol{\lambda}_i)$ *denotes the support of the vector* $\boldsymbol{\lambda}_i$*.*

In other words, each coordinate j *of* $\mathscr{C}$*,* $j \in [n]$*, appears exactly* κ *times in* $\{\chi(\boldsymbol{\lambda}_i)\}_{i\in[\nu]}$*, and every set* $\chi(\boldsymbol{\lambda}_i)$ *contains an information set of* $\mathscr{C}$*.*

Example 3 *Consider a* $[4, 2]$ *code* $\mathscr{C}$ *with generator matrix*

$$\mathsf{G}^{\mathscr{C}} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

One can verify that

$$\mathsf{\Lambda}_{1,2}^{\text{PIR}} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

is a valid PIR achievable rate matrix for $\mathscr{C}$ *with* $(\kappa, \nu) = (1, 2)$*. This is true given that, column-wise, the Hamming weight of each column in* $\mathsf{\Lambda}_{1,2}^{\text{PIR}}$ *is* $\kappa = 1$*. On the other hand, row-wise,* $\chi(\boldsymbol{\lambda}_1) = \{1, 3\}$ *and* $\chi(\boldsymbol{\lambda}_2) = \{2, 4\}$ *are two information sets of* $\mathscr{C}$*.* $\bigtriangledown$

It is shown in [15] that the MDS-PIR capacity [8] can be achieved using Protocol 1 for a special class of $[n, k]$ codes. In particular, to achieve the MDS-PIR capacity using Protocol 1, the $[n, k]$ storage code should possess a specific underlying structure as given by the following theorem.

Theorem 1 ([15, Cor. 1]) *Consider a DSS that uses an* $[n, k]$ *code* $\mathscr{C}$ *to store* f *messages. If a PIR achievable rate matrix* $\mathsf{\Lambda}_{\kappa,\nu}^{\text{PIR}}(\mathscr{C})$ *with* $\frac{\kappa}{\nu} = \frac{k}{n}$ *exists, then the MDS-PIR capacity*

$$\mathsf{C}_{\text{MDS-PIR}} \triangleq \big(1 - \frac{k}{n}\big)\left[1 - \big(\frac{k}{n}\big)^f\right]^{-1}$$

is achievable.

This gives rise to the following definition.

Definition 3 ([15, Def. 13]) *Given an* $[n, k]$ *code* $\mathscr{C}$*, if a PIR achievable rate matrix* $\mathsf{\Lambda}_{\kappa,\nu}^{\text{PIR}}(\mathscr{C})$ *with* $\frac{\kappa}{\nu} = \frac{k}{n}$ *exists, then the code* $\mathscr{C}$ *is referred to as an* MDS-PIR capacity-achieving *code, and the matrix* $\mathsf{\Lambda}_{\kappa,\nu}^{\text{PIR}}(\mathscr{C})$ *is called an* MDS-PIR capacity-achieving *matrix.*

Accordingly, one can easily see that the $[4, 2]$ code $\mathscr{C}$ given in Example 3 is an MDS-PIR capacity-achieving code. Note that the class of MDS-PIR capacity-achieving codes includes MDS codes, cyclic codes, Reed-Muller codes, and certain classes of distance-optimal local reconstruction codes [15]. In the following chapter, we present a PLC protocol and a general achievable rate using the PIR achievable rate matrix $\Lambda^{\mathrm{PIR}}_{\kappa,\nu}$ of an $[n, k]$ code.

CHAPTER 3

PRIVATE LINEAR COMPUTATION FOR NONCOLLUDING CODED DATABASES

3.1 Introduction

In this chapter[1], we consider the problem of private linear computation (PLC) for coded databases. In PLC, a user wishes to compute a linear combination over the f messages while keeping the coefficients of the desired linear combination hidden from the databases. For a DSS setup where data is stored using a code from a particular family of linear storage codes, we derive an outer bound on the PLC rate, which is defined as the ratio of the desired amount of information and the total amount of downloaded information. In particular, the proposed converse is valid for any number of messages and linear combinations, and depends on the rank of the coefficient matrix obtained from all linear combinations. Further, we present a PLC scheme with rate equal to the outer bound and hence settle the PLC capacity for the considered class of linear storage codes. Interestingly, the PLC capacity matches the maximum distance separable coded capacity of PIR for the considered class of linear storage codes.

The PLC problem from n noncolluding coded DSSs is described in Section 2.2. However, in PLC a user wishes to privately compute exactly one *linear* function evaluation $\mathbf{X}^{(v)} = \left(X_1^{(v)}, \ldots, X_{\mathsf{L}}^{(v)}\right)$, out of μ *candidate* linear combinations $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}$ from the coded DSS. As a result, the μ-tuple $\left(X_l^{(1)}, \ldots, X_l^{(\mu)}\right)^{\intercal}$, $\forall\, l \in [\mathsf{L}]$, is mapped by a deterministic matrix V of size $\mu \times f$ over $\mathbb{F}_p$ by

$$\begin{pmatrix} X_l^{(1)} \\ \vdots \\ X_l^{(\mu)} \end{pmatrix} = \mathsf{V}_{\mu \times f} \begin{pmatrix} W_l^{(1)} \\ \vdots \\ W_l^{(f)} \end{pmatrix}. \tag{3.1}$$

[1]The material presented in this chapter is published in [75].

The user privately selects an index $v \in [\mu]$ and wishes to compute the v-th function while keeping the requested function index v private from each database. Here, we also assume that the rank of V is equal to $\text{rank}(\mathsf{V}) = r \leq \min\{\mu, f\}$ and the indices corresponding to a basis for the row space of V are denoted by the set $\mathcal{L} \triangleq \{\ell_1, \ldots, \ell_r\} \subseteq [\mu]$. Finally, we assume error-free recovery; hence, the recoverability constraint of the PC protocol given in equation (2.6) becomes

$$\text{[Recovery]} \qquad \mathsf{H}\big(\mathbf{X}^{(v)} \,\big|\, A_{[n]}^{(v)}, Q_{[n]}^{(v)}\big) = 0. \tag{3.2}$$

The remainder of this chapter is organized as follows. We derive the converse bound for an arbitrary number of messages and linear combinations in Section 3.2. A capacity-achieving PLC scheme for linearly-coded storage with an MDS-PIR capacity-achieving code is presented in Section 3.3. Some conclusions are drawn in Section 3.4.

3.2 Converse Bound

In [51], [52], the PIR capacity for a coded DSS using an MDS-PIR capacity-achieving code is shown to be equal to the MDS-PIR capacity. In this section, we derive a converse bound for the PLC rate (Theorem 2 below) by adapting the converse proof of [52, Thm. 4] to the linearly-coded PLC problem, where the storage code is an MDS-PIR capacity-achieving code. Then, we show that the PLC capacity matches the MDS-PIR capacity (i.e., the PIR capacity for a DSS where data is encoded and stored using an MDS code). The converse is valid for any number of messages f and candidate linear functions μ. The following theorem states an upper bound on the PLC capacity for a coded DSS where data is stored using an MDS-PIR capacity-achieving code.

Theorem 2 *Consider a DSS with n noncolluding databases that uses an $[n, k]$ MDS-PIR capacity-achieving code $\mathscr{C}$ to store f messages. Then, the rate R of any PLC*

protocol is upper bounded by

$$\mathsf{R} \leq \mathsf{C}_{\mathrm{PLC}} \triangleq \left[1 + \sum_{v=1}^{r-1} \left(\frac{k}{n}\right)^v\right]^{-1} = \left(1 - \frac{k}{n}\right)\left[1 - \left(\frac{k}{n}\right)^r\right]^{-1},$$

where r is the rank of the linear mapping from equation (3.1)*.*

Note that by simply assuming that the candidate functions are linearly independent linear combinations, i.e., $\mu = r$, the PLC problem reduces to PIR from $[n, k]$ linearly-encoded DSSs. If these linear combinations are also uniformly distributed, the proof of Theorem 2 follows directly from the PIR capacity of [52, Thm. 4]. However, by providing a formal proof for Theorem 2, we confirm that with added computation, i.e., $\mu > r$, we can not achieve a better rate. In the following, we present a general converse proof for *dependent* messages and detail the conditions that lead to this conclusion. Before we proceed with the converse proof, we provide some general results.

- From the condition of privacy,

$$\mathsf{H}\big(A_j^{(v)} \,\big|\, \mathbf{X}^{(v)}, \mathcal{Q}\big) = \mathsf{H}\big(A_j^{(v')} \,\big|\, \mathbf{X}^{(v)}, \mathcal{Q}\big), \tag{3.3}$$

  where $v \neq v'$, $v, v' \in [\mu]$, and $\mathcal{Q} \triangleq \big\{Q_j^{(v)} \colon v \in [\mu], j \in [n]\big\}$ denotes the set of all queries. Although this seems to be intuitively true, a proof of this property is still required and can be found in [8].

- Consider a PLC protocol for a coded DSS that uses an $[n, k]$ code $\mathscr{C}$ to store f messages.

  Lemma 1 *(Independence of answers from k databases forming an information set): For any* information set $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| = k$*, of the $[n, k]$ linear storage code $\mathscr{C}$, and for any $v \in [\mu]$,*

$$\mathsf{H}\big(A_{\mathcal{I}}^{(v)} \,\big|\, \mathcal{Q}\big) = \sum_{j \in \mathcal{I}} \mathsf{H}\big(A_j^{(v)} \,\big|\, \mathcal{Q}\big). \tag{3.4}$$

Moreover, equation (3.4) *is true conditioned on any subset of linear combinations* $\mathsf{X}^{\mathcal{V}}$, $\mathcal{V} \subseteq [\mu]$, *i.e.,*

$$\mathsf{H}\big(A_{\mathcal{I}}^{(v)} \,\big|\, \mathsf{X}^{\mathcal{V}}, \mathcal{Q}\big) = \sum_{j \in \mathcal{I}} \mathsf{H}\big(A_{j}^{(v)} \,\big|\, \mathsf{X}^{\mathcal{V}}, \mathcal{Q}\big). \tag{3.5}$$

The proof of Lemma 1 is a simple extension of [8, Lem. 1] based on [16, Lem. 1] and is presented in Appendix A.

Next, we state Shearer's Lemma, which represents a very useful entropy method for combinatorial problems.

Lemma 2 (Shearer's Lemma [76]) *Let* $\mathscr{S}$ *be a collection of subsets of* $[n]$, *with each* $j \in [n]$ *included in at least* κ *members of* $\mathscr{S}$. *For random variables* $Z_1, \ldots, Z_n$, *we have* $\sum_{\mathcal{S} \in \mathscr{S}} \mathsf{H}(Z_{\mathcal{S}}) \geq \kappa\, \mathsf{H}(Z_1, \ldots, Z_n)$.

For our converse proof for the coded PLC problem, we also need the following lemma, whose proof is presented in Appendix B.

Lemma 3 *Consider the linear mapping* $\mathsf{V} = (v_{i,j})$ *defined in equation* (3.1) *with* $\mathrm{rank}(\mathsf{V}) = r$ *where* $v_{i_1,j_1}, \ldots, v_{i_r,j_r}$ *are the entries corresponding to the pivot elements of* V. *It follows that* $\big(\mathsf{X}^{(i_1)}, \ldots, \mathsf{X}^{(i_h)}\big)$ *and* $\big(\mathsf{W}^{(j_1)}, \ldots, \mathsf{W}^{(j_h)}\big)$ *are identically distributed, for some* $h \in [r]$. *In other words,* $\mathsf{H}\big(\mathsf{X}^{(i_1)}, \ldots, \mathsf{X}^{(i_h)}\big) = h\mathsf{L}$, $h \in [r]$.

Now, we are ready for the converse proof. By [15, Lem. 2], since the code $\mathscr{C}$ is MDS-PIR capacity-achieving, there exist ν information sets $\mathcal{I}_1, \ldots, \mathcal{I}_\nu$ such that each coordinate $j \in [n]$ is included in exactly κ members of $\mathscr{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_\nu\}$ with $\frac{\kappa}{\nu} = \frac{k}{n}$.

Applying the chain rule of entropy we have $\mathsf{H}\big(A_{[n]}^{(v)} \,\big|\, \mathsf{X}^{\mathcal{V}}, \mathcal{Q}\big) \geq \mathsf{H}\big(A_{\mathcal{I}_i}^{(v)} \,\big|\, \mathsf{X}^{\mathcal{V}}, \mathcal{Q}\big)$, $\forall\, i \in [\nu]$, where $\mathcal{V} \subseteq [\mu]$ is arbitrary.

Let $v \in \mathcal{V}$ and $v' \in \mathcal{V}^{\mathrm{c}} \triangleq [\mu] \setminus \mathcal{V}$. Following similar steps as in the proof given in [8], [77], we get

$$\begin{aligned}
\nu\, \mathsf{H}\big(A_{[n]}^{(v)} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) &\geq \sum_{i=1}^{\nu} \mathsf{H}\big(A_{\mathcal{I}_i}^{(v)} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \\
&\overset{(a)}{=} \sum_{i=1}^{\nu} \left(\sum_{j \in \mathcal{I}_i} \mathsf{H}\big(A_j^{(v)} \big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \right) \overset{(b)}{=} \sum_{i=1}^{\nu} \left(\sum_{j \in \mathcal{I}_i} \mathsf{H}\big(A_j^{(v')} \big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \right) \\
&\overset{(a)}{=} \sum_{i=1}^{\nu} \mathsf{H}\big(A_{\mathcal{I}_i}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \overset{(c)}{\geq} \kappa\, \mathsf{H}\big(A_{[n]}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \\
&= \kappa \Big[\mathsf{H}\big(A_{[n]}^{(v')}, \mathbf{X}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) - \mathsf{H}\big(\mathbf{X}^{(v')} \,\big|\, A_{[n]}^{(v')}, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \Big] \\
&\overset{(d)}{=} \kappa \Big[\mathsf{H}\big(\mathbf{X}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) + \mathsf{H}\big(A_{[n]}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathbf{X}^{(v')}, \mathcal{Q}\big) - 0 \Big] \\
&\overset{(e)}{=} \kappa \Big[\mathsf{H}\big(\mathbf{X}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}\big) + \mathsf{H}\big(A_{[n]}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathbf{X}^{(v')}, \mathcal{Q}\big) \Big],
\end{aligned}$$

where (a) follows from equation (3.5); (b) is because of equation (3.3); (c) is due to Shearer's Lemma; (d) is from the fact that the v'-th linear combination $\mathbf{X}^{(v')}$ is determined by the answers $A_{[n]}^{(v')}$ and all possible queries $\mathcal{Q}$; and finally, (e) follows from the independence between all possible queries and the messages. Therefore, we can conclude that

$$\begin{aligned}
\mathsf{H}\big(A_{[n]}^{(v)} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) &\geq \frac{\kappa}{\nu}\, \mathsf{H}\big(\mathbf{X}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}\big) + \frac{\kappa}{\nu}\, \mathsf{H}\big(A_{[n]}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathbf{X}^{(v')}, \mathcal{Q}\big) \\
&= \frac{k}{n}\, \mathsf{H}\big(\mathbf{X}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}\big) + \frac{k}{n}\, \mathsf{H}\big(A_{[n]}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathbf{X}^{(v')}, \mathcal{Q}\big), \qquad (3.6)
\end{aligned}$$

where we have used Definition 3 to obtain equation (3.6).

Since there are in total μ linear combinations and $\mathcal{L} \triangleq \{\ell_1, \ldots, \ell_r\} \subseteq [\mu]$ is the set of row indices corresponding to the selected basis for the row space of V, we can recursively use equation (3.6) $r-1$ times to obtain

$$\begin{aligned}
&\mathsf{H}\big(A_{[n]}^{(\ell_1)} \,\big|\, \mathbf{X}^{(\ell_1)}, \mathcal{Q}\big) \\
&\quad \geq \sum_{v=1}^{r-1} \Big(\frac{k}{n}\Big)^{v} \mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})} \,\big|\, \mathbf{X}^{\{\ell_1, \ldots, \ell_v\}}\big) + \Big(\frac{k}{n}\Big)^{r-1} \mathsf{H}\big(A_{[n]}^{(\ell_r)} \,\big|\, \mathbf{X}^{\{\ell_1, \ldots, \ell_r\}}, \mathcal{Q}\big)
\end{aligned}$$

$$\overset{(a)}{\geq} \sum_{v=1}^{r-1} \Big(\frac{k}{n}\Big)^v \mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})} \,\big|\, \mathbf{X}^{\{\ell_1,\ldots,\ell_v\}}\big) \overset{(b)}{=} \sum_{v=1}^{r-1} \Big(\frac{k}{n}\Big)^v \mathsf{L}, \tag{3.7}$$

where (a) follows from the nonnegativity of entropy, and (b) holds since $\mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})} \,\big|\, \mathbf{X}^{\{\ell_1,\ldots,\ell_v\}}\big) = \mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})}\big) = \mathsf{L}$ (see Lemma 3). Here, we also remark that the recursive steps follow the same principle of the general converse for DPIR from [44, Thm. 1]. In [44], the authors claim that the general converse for the DPIR problem strongly depends on the chosen permutation of the indices of the candidate functions. However, for the PLC problem, the index permutation of the candidate linear functions intuitively follows from finding a basis for V. Now,

$$\begin{aligned}
\mathsf{L} = \mathsf{H}\big(\mathbf{X}^{(\ell_1)}\big) &\overset{(a)}{=} \mathsf{H}\big(\mathbf{X}^{(\ell_1)} \,\big|\, \mathcal{Q}\big) - \underbrace{\mathsf{H}\big(\mathbf{X}^{(\ell_1)} \,\big|\, A_{[n]}^{(\ell_1)}, \mathcal{Q}\big)}_{=0} \\
&= \mathsf{H}\Big(A_{[n]}^{(\ell_1)} \,\Big|\, \mathcal{Q}\Big) - \mathsf{H}\Big(A_{[n]}^{(\ell_1)} \,\Big|\, \mathbf{X}^{(\ell_1)}, \mathcal{Q}\Big) \\
&\overset{(b)}{\leq} \mathsf{H}\Big(A_{[n]}^{(\ell_1)} \,\Big|\, \mathcal{Q}\Big) - \sum_{v=1}^{r-1} \Big(\frac{k}{n}\Big)^v \mathsf{L},
\end{aligned} \tag{3.8}$$

where (a) follows since any message is independent of the queries $\mathcal{Q}$, and by knowing the answers $A_{[n]}^{(\ell_1)}$ and the queries $\mathcal{Q}$, one can determine $\mathbf{X}^{(\ell_1)}$, and (b) holds because of equation (3.7).

Finally, the converse proof is completed by showing that

$$\begin{aligned}
\mathsf{R} = \frac{\mathsf{L}}{\sum_{j=1}^{n} \mathsf{H}\big(A_j^{(\ell_1)}\big)} &\leq \frac{\mathsf{L}}{\mathsf{H}\big(A_{[n]}^{(\ell_1)}\big)} \overset{(a)}{\leq} \frac{\mathsf{L}}{\mathsf{H}\big(A_{[n]}^{(\ell_1)} \,\big|\, \mathcal{Q}\big)} \\
&\overset{(b)}{\leq} \frac{1}{1 + \sum_{v=1}^{r-1} \big(\frac{k}{n}\big)^v} = \mathsf{C}_{\text{PLC}},
\end{aligned}$$

where (a) is due to the fact that conditioning reduces entropy, and we apply equation (3.8) to obtain (b).

It can be easily seen that the converse bound of Theorem 2 matches the MDS-PIR capacity $\mathsf{C}_{\text{MDS-PIR}}$ for $f = r$ files given in Theorem 1. The capacity-achieving PLC scheme is provided in the following section.

3.3 Private Linear Computation From Coded DSSs

One of the main results of this chapter is the derivation of the PLC capacity for a coded DSS where data is encoded and stored using a linear code from the class of MDS-PIR capacity-achieving codes [15]. Based on the PLC converse bound of Theorem 2, in this section we construct a capacity-achieving PLC scheme. Our capacity-achieving PLC scheme is also a generalization of the replication-based PLC scheme in [43]. Although the two schemes are build upon a different PIR construction, both schemes adapt the underlying PIR construction for *dependent* virtual messages through an *index assignment* structure. Moreover, in order to optimize the download rate, both schemes deploy a *sign assignment* structure to induce redundancy within the queries of the modified underlying PIR construction. In this section, we first present our modified underlying PIR construction with Algorithm 1 in Section 3.3.1. Then, we elaborate on the sign assignment procedure in Section 3.3.3. In Theorem 3 we settle the PLC capacity for a DSS where data is stored using an MDS-PIR capacity-achieving code.

Theorem 3 *Consider a DSS with n noncolluding databases that uses an $[n, k]$ MDS-PIR capacity-achieving code $\mathscr{C}$ to store f messages. Then, the PLC capacity is equal to C_{PLC}, where r is the rank of the linear mapping from equation* (3.1).

We remark that since all MDS codes are MDS-PIR capacity-achieving codes, it follows that if $\text{rank}(\mathsf{V}) = f$, then the PLC capacity for an MDS-coded DSS [78] is equal to the MDS-PIR capacity $\mathsf{C}_{\text{MDS-PIR}}$.

We now start by constructing a query generation algorithm for a coded PIR-like scheme, where its *dependent virtual* messages represent the evaluations of the μ candidate linear combinations. A PIR-like scheme achieves a private retrieval of the desired *virtual* message by following three important design principles: 1) Enforcing symmetry across databases. Each database is queried for an equal number of symbols and the query structure does not depend on the individual database, i.e., the scheme

structure is fixed for all databases. 2) Enforcing symmetry across virtual messages. 3) Exploiting side information represented by undesired information downloaded to maintain message symmetry.

Given that the messages are stored using an $[n, k]$ MDS-PIR capacity-achieving code $\mathscr{C}$, we can construct a $\nu \times n$ MDS-PIR capacity-achieving matrix $\Lambda_{\kappa,\nu}^{\mathrm{PIR}}$ of Definition 2, and obtain the PIR interference matrices $\mathsf{A}_{\kappa\times n}$ and $\mathsf{B}_{(\nu-\kappa)\times n}$ as given by the following definition.

Definition 4 ([15]) *For a given $\nu \times n$ PIR achievable rate matrix $\Lambda_{\kappa,\nu}^{\mathrm{PIR}}(\mathscr{C}) = (\lambda_{u,j})$, we define the PIR interference matrices $\mathsf{A}_{\kappa\times n} = (a_{i,j})$ and $\mathsf{B}_{(\nu-\kappa)\times n} = (b_{i,j})$ for the code $\mathscr{C}$ as*

$$a_{i,j} \triangleq u \text{ if } \lambda_{u,j} = 1,\ \forall j \in [n], i \in [\kappa], u \in [\nu],$$
$$b_{i,j} \triangleq u \text{ if } \lambda_{u,j} = 0,\ \forall j \in [n], i \in [\nu - \kappa], u \in [\nu].$$

Note that in Definition 4, for each $j \in [n]$, distinct values of $u \in [\nu]$ should be assigned for all i. Thus, the assignment is not unique in the sense that the order of the entries of each column of A and B can be permuted. Moreover, for $j \in [n]$, let $\mathcal{A}_j \triangleq \{a_{i,j} \colon i \in [\kappa]\}$ and $\mathcal{B}_j \triangleq \{b_{i,j} \colon i \in [\nu - \kappa]\}$. Note that the j-th column of $\mathsf{A}_{\kappa\times n}$ contains the row indices of $\Lambda_{\kappa,\nu}^{\mathrm{PIR}}$ whose entries in the j-th column are equal to 1, while $\mathsf{B}_{(\nu-\kappa)\times n}$ contains the remaining row indices of $\Lambda_{\kappa,\nu}^{\mathrm{PIR}}$. Hence, it can be observed that $\mathcal{B}_j = [\nu] \setminus \mathcal{A}_j,\ \forall\, j \in [n]$.

Next, for the sake of illustrating our query generation algorithm, we make use of the following definition.

Definition 5 *By $\mathcal{S}(u|\mathsf{A}_{\kappa\times n})$ we denote the set of column coordinates of matrix $\mathsf{A}_{\kappa\times n} = (a_{i,j})$ in which at least one of its entries is equal to u, i.e., $\mathcal{S}(u|\mathsf{A}_{\kappa\times n}) \triangleq \{j \in [n] \colon \exists\, a_{i,j} = u, i \in [\kappa]\}$.*

As a result, we require the size of the message to be $\mathsf{L} = \nu^{\mu} \cdot k$ (i.e., $\beta = \nu^{\mu}$).

3.3.1 Query Generation for PLC

Before running the main algorithm to generate the query sets, the following index preparation for the coded symbols stored in each database is performed.

***1) Index Preparation*:** The goal is to make the symbols queried from each database to appear to be chosen randomly and independently from the desired linear function index. Note that the function is computed separately for the t-th row of all messages, $t \in [\beta]$. Therefore, similar to the PLC scheme in [43] and the MDS-coded PLC scheme in [78], we apply a permutation that is fixed across all coded symbols for the t-th row to maintain the dependency across the associated message elements. Let $\pi(\cdot)$ be a random permutation function over $[\beta]$, and let

$$U_{t,j}^{(v')} \triangleq \boldsymbol{v}_{v'} \boldsymbol{C}_{\pi(t),j},\ t \in [\beta],\ j \in [n],\ v' \in [\mu], \tag{3.9}$$

denote the t-th permuted symbol associated with the v'-th virtual message $\mathbf{X}^{(v')}$ stored in the j-th database, where $\boldsymbol{C}_{t,j} \triangleq \left(C_{t,j}^{(1)}, \ldots, C_{t,j}^{(f)}\right)^{\mathsf{T}}$ and $\boldsymbol{v}_{v'}$ represents the v'-th row vector of the matrix $\mathsf{V}_{\mu \times f} = (v_{i,j})$. The permutation $\pi(\cdot)$ is randomly selected privately and uniformly by the user.

***2) Preliminaries*:** The query generation procedure is subdivided into μ rounds, where in each round τ we generate the queries based on the concept of *τ-sums* as defined in the following.

Definition 6 (τ-sum) *For $\tau \in [\mu]$, a sum $U_{i_1,j}^{(v_1)} + U_{i_2,j}^{(v_2)} + \cdots + U_{i_\tau,j}^{(v_\tau)}$, $j \in [n]$, of τ distinct symbols is called a τ-sum for any $(i_1, \ldots, i_\tau) \in [\beta]^\tau$, and $\{v_1, \ldots, v_\tau\} \subseteq [\mu]$ determines the* type *of the τ-sum.*

Since we have $\binom{\mu}{\tau}$ different selections of τ distinct elements out of μ elements, a τ-sum can have $\binom{\mu}{\tau}$ different *types*. For a requested linear function evaluation indexed by $v \in [\mu]$, a query set $Q_j^{(v)}$, $j \in [n]$, is composed of μ disjoint subsets of queries, each subset of queries is generated by the operations of each round $\tau \in [\mu]$. In

a round we generate the queries for all possible $\binom{\mu}{\tau}$ types of τ-sums. For each round $\tau \in [\mu]$ the corresponding query subset is further subdivided into two subsets $Q_j^{(v)}(\mathcal{D};\tau)$ and $Q_j^{(v)}(\mathcal{U};\tau)$. The first subset $Q_j^{(v)}(\mathcal{D};\tau)$ corresponds to τ-sums with a single symbol from the *desired* function evaluation and $\tau - 1$ symbols from the evaluations of *undesired* functions, while the second subset $Q_j^{(v)}(\mathcal{U};\tau)$ corresponds to τ-sums with symbols only from the evaluations of undesired functions. Here, $\mathcal{D}$ is an indicator for "desired function evaluations", while $\mathcal{U}$ an indicator for "undesired functions evaluations". Note that we require $\kappa^{\mu-(\tau-1)}(\nu-\kappa)^{\tau-1}$ distinct instances of each τ-sum type for every query set $Q_j^{(v)}$. To this end, the algorithm will generate κn auxiliary query sets $Q_j^{(v)}(a_{i,j},\mathcal{D};\tau)$, $i \in [\kappa]$, where each query requests a distinct symbol from the desired function evaluation and $\tau - 1$ symbols from undesired functions evaluations, and $(\nu-\kappa)n$ auxiliary query sets $Q_j^{(v)}(b_{i,j},\mathcal{U};\tau)$, $i \in [\nu-\kappa]$, to represent the query sets of symbols from the undesired functions evaluations for each database $j \in [n]$. We utilize these sets to generate the query sets of each round according to the PIR interference matrices $\mathsf{A}_{\kappa\times n}$ and $\mathsf{B}_{(\nu-\kappa)\times n}$.

To illustrate the key concepts of the coded PLC scheme, we use the following example, i.e., Example 4, as a running example for this section.

Example 4 *Consider $f = 4$ messages $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, $\mathbf{W}^{(3)}$, and $\mathbf{W}^{(4)}$ that are stored in a DSS using the $[4, 2]$ MDS-PIR capacity-achieving code $\mathscr{C}$ given in Example 3 for which*

$$\Lambda_{1,2}^{\text{PIR}} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \mathsf{A}_{1\times 4} = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \text{ and } \mathsf{B}_{1\times 4} = \begin{pmatrix} 2 & 1 & 2 & 1 \end{pmatrix},$$

are a PIR achievable rate matrix with $(\kappa,\nu) = (1, 2)$ and the corresponding PIR interference matrices $\mathsf{A}_{1\times 4}$ and $\mathsf{B}_{1\times 4}$, respectively, according to Definition 4. Suppose that the user wishes to obtain a linear function evaluation $\mathbf{X}^{(v)}$ from a set of $\mu = 4$

candidate linear functions, whose $\mathsf{V}_{\mu\times f}$ *from equation* (3.1) *is given by*

$$\mathsf{V}_{4\times 4} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 \\ 4 & 1 & 0 & 3 \end{pmatrix}.$$

We simplify notation by letting $x_{t,j} = U_{t,j}^{(1)}$, $y_{t,j} = U_{t,j}^{(2)}$, $z_{t,j} = U_{t,j}^{(3)}$, *and* $w_{t,j} = U_{t,j}^{(4)}$ *for all* $t \in [\beta]$, $j \in [n]$, *where* $\beta = \nu^{\mu} = 16$. *First, let the desired linear function index be* $v = 1$. $\triangledown$

The query sets for all databases are generated by Algorithm 1 through the following procedures.[2]

3) Initialization (Round $\tau = 1$***):*** In the initialization step, the algorithm generates the auxiliary queries for the first round. This round is described in lines 5 to 11 of Algorithm 1, where we have $\tau = 1$ for the τ-sum. At this point, Algorithm 1 invokes the subroutine `Initial-Round` given in Algorithm 2 to generate $Q_j^{(v)}(a_{i,j}, \mathcal{D}; 1)$, $i \in [\kappa]$, such that each of these query sets contains $\alpha_1 = \kappa^{\mu-1}$ distinct symbols. Furthermore, to maintain function symmetry, the algorithm asks each database for the same number of distinct symbols of all other linear functions evaluations in $Q_j^{(v)}(a_{i,j}, \mathcal{U}; 1)$, $i \in [\kappa]$, resulting in a total number of $\binom{\mu-1}{1}\kappa^{\mu-1}$ symbols. As a result, the queried symbols in the auxiliary query sets for each database are symmetric with respect to all function evaluation vectors indexed by $v' \in [\mu]$.

In the following steps, we will associate the symbols of undesired functions evaluations in κ groups, each placed in the undesired query sets $Q_j^{(v)}(a_{i,j}, \mathcal{U}; 1)$, $i \in [\kappa]$.

[2]Note that a query $Q_j^{(v)}$ sent to the j-th database usually indicates the row indices of the symbols that the user requests, while the answer $A_j^{(v)}$ to the query $Q_j^{(v)}$ refers to the particular symbols requested through the query. In Algorithm 1, with some abuse of notation for the sake of simplicity, the generated queries are sets containing their answers.

Algorithm 1: `Q-Gen`

Input : v, μ, κ, ν, n, $\mathsf{A}_{\kappa\times n}$, and $\mathsf{B}_{(\nu-\kappa)\times n}$

Output: $Q_1^{(v)},\ldots,Q_n^{(v)}$

for $\tau\in[\mu]$ **do**
  $Q_j^{(v)}(\mathcal{D};\tau)\leftarrow\emptyset$, $Q_j^{(v)}(\mathcal{U};\tau)\leftarrow\emptyset$, $j\in[n]$
  $\alpha_\tau\leftarrow\kappa^{\mu-1}+\sum_{h=1}^{\tau-1}\binom{\mu-1}{h}\kappa^{\mu-(h+1)}(\nu-\kappa)^h$
  ▷ Generate query sets for the initial round
  if $\tau=1$ **then**
    for $u\in[\nu]$ **do**
      for $j\in\mathcal{S}(u|\mathsf{A}_{\kappa\times n})$ **do**
        $Q_j^{(v)}(u,\mathcal{D};\tau),Q_j^{(v)}(u,\mathcal{U};\tau)\leftarrow$ `Initial-Round`(u,α_τ,j,v,τ)
      end
    end
  end
  ▷ Generate query sets for the following rounds $\tau>1$
  else
    for $u\in[\nu]$ **do**
      ▷ Generate desired symbols for the following rounds $\tau>1$
      for $j\in\mathcal{S}(u|\mathsf{A}_{\kappa\times n})$ **do**
        $Q_j^{(v)}(u,\mathcal{D};\tau)\leftarrow$ `Desired-Q`(u,α_τ,j,v,τ)
      end
      ▷ Generate side information for the following rounds $\tau>1$
      for $j\in\mathcal{S}(u|\mathsf{B}_{(\nu-\kappa)\times n})$ **do**
        $Q_j^{(v)}(u,\mathcal{U};\tau-1)\leftarrow$
        `Exploit-SI`$(u,Q_1^{(v)}(u,\mathcal{U},\tau-1),\ldots,Q_n^{(v)}(u,\mathcal{U},\tau-1),j,v,\tau)$
      end
    end
    ▷ Generate the final desired query sets for the following rounds $\tau>1$
    for $j\in[n]$ **do**
      $\tilde{Q}_j^{(v)}(\mathcal{U};\tau-1)\leftarrow\bigcup_{i=1}^{\nu-\kappa}Q_j^{(v)}(b_{i,j},\mathcal{U};\tau-1)$
      $\tilde{Q}_j^{(v)}(1,\mathcal{U};\tau-1),\ldots,\tilde{Q}_j^{(v)}(\kappa,\mathcal{U};\tau-1)\leftarrow$ `Partition`$\big(\tilde{Q}_j^{(v)}(\mathcal{U};\tau-1)\big)$
      for $i\in[\kappa]$ **do**
        $Q_j^{(v)}(a_{i,j},\mathcal{D};\tau)\leftarrow$ `SetAddition`$\big(Q_j^{(v)}(a_{i,j},\mathcal{D};\tau),\tilde{Q}_j^{(v)}(i,\mathcal{U};\tau-1)\big)$
      end
    end
    ▷ Generate the query sets of undesired symbols by forcing message symmetry for the following rounds $\tau>1$
    for $u\in[\nu]$ **do**
      for $j\in\mathcal{S}(u|\mathsf{A}_{\kappa\times n})$ **do**
        $Q_j^{(v)}(u,\mathcal{U};\tau)\leftarrow$ `M-Sym`$\big(Q_j^{(v)}(u,\mathcal{D};\tau),j,v,\tau\big)$
      end
    end
  end
  for $u\in[\nu]$ **do**
    for $j\in\mathcal{S}(u|\mathsf{A}_{\kappa\times n})$ **do**
      $Q_j^{(v)}(\mathcal{D};\tau)\leftarrow Q_j^{(v)}(\mathcal{D};\tau)\cup Q_j^{(v)}(u,\mathcal{D};\tau)$
      $Q_j^{(v)}(\mathcal{U};\tau)\leftarrow Q_j^{(v)}(\mathcal{U};\tau)\cup Q_j^{(v)}(u,\mathcal{U};\tau)$
    end
  end
end
for $j\in[n]$ **do**
  $Q_j^{(v)}\leftarrow\bigcup_{\tau=1}^{\mu}\Big(Q_j^{(v)}(\mathcal{D};\tau)\cup Q_j^{(v)}(\mathcal{U};\tau)\Big)$
end

Since this procedure produces κ undesired query sets for each database, database symmetry is maintained.

Example 4 (continued) *The initialization step is described in the following. Algorithm 1 starts with* $\tau = 1$ *to generate auxiliary query sets* $Q_j^{(v)}(a_{i,j}, \mathcal{D}; 1)$, $Q_j^{(v)}(a_{i,j}, \mathcal{U}; 1)$, $i \in [\kappa]$, *for each database* $j \in [n]$. *Starting at line 6 of Algorithm 1, since* $\nu = 2$, *we have the row indicator* $u \in [2]$. *This indicator is first used to identify the code coordinates pertaining to different entries* $u = a_{i,j}$, *as specified by the interference matrix* $\mathsf{A}_{1\times 4}$. *For example, when* $u = 1$, *following Definition 5, we have* $\mathcal{S}(1|\mathsf{A}_{\kappa\times n}) = \{1, 3\}$. *In line 7 of Algorithm 1, for* $j \in \{1, 3\}$, *algorithm* `Initial-Round` *is invoked to generate the desired and undesired query subsets* $Q_j^{(1)}(1, \mathcal{D}; 1)$ *and* $Q_j^{(1)}(1, \mathcal{U}; 1)$. *The set* $Q_j^{(1)}(1, \mathcal{D}; 1)$ *queries* $\alpha_1 = \kappa^{\mu-1} = 1$ *distinct instances of the desired function evaluation* $x_{t,j}$ *and the set* $Q_j^{(1)}(1, \mathcal{U}; 1)$ $\alpha_1 = 1$ *distinct instances of the remaining linear functions evaluations* $y_{t,j}$, $z_{t,j}$, *and* $w_{t,j}$. *To this end, the row indicator* u *is passed to the subroutine* `Initial-Round`, *i.e., Algorithm 2, where it is used to determine the indices of the queried symbols. For example, the first auxiliary query set for* $u = 1$ *generated by Algorithm 2 is given by* $Q_j^{(1)}(1, \mathcal{D}; 1) = \{U^{(1)}_{(1-1)\cdot 1+1,j}\} = \{x_{1,j}\}$, $j \in \{1, 3\}$. *A similar process is followed for* $Q_j^{(1)}(1, \mathcal{U}; 1)$. *The same process is then repeated for* $u = 2$. *By the end of this step, we have queried* $\nu\alpha_1 = 2$ *distinct instances of the desired function evaluation* $x_{t,j}$ *and by message symmetry,* $\nu\alpha_1 = 2$ *distinct instances of the remaining functions evaluations* $y_{t,j}$, $z_{t,j}$, *and* $w_{t,j}$. *In total, the first round of queries comprises* $n\kappa\alpha_1\mu = 16$ *symbols, which can be written in the form* $n\binom{\mu}{1}\kappa^{\mu-1+1}(\nu-\kappa)^{1-1}$. *The resulting auxiliary query sets for the first round of queries are shown in Table 3.1(a), where we highlight in red the row indicator* $u \in [\nu]$ *as specified by the interference matrix* $\mathsf{A}_{1\times 4}$, *i.e.,* $u = a_{1,j}$. $\triangledown$

4) Desired Function Symbols for Rounds $\tau > 1$**:** For the following rounds a similar process is repeated in terms of generating auxiliary query sets containing distinct code symbols from the desired linear function evaluation $\mathbf{U}^{(v)} = (U_{t,j}^{(v)})$. This is accomplished in lines 16 to 18 by calling the subroutine `Desired-Q`, given in Algorithm 3, to generate $Q_j^{(v)}(a_{i,j}, \mathcal{D}; \tau)$, $i \in [\kappa]$, such that each of these query sets

Table 3.1 Auxiliary Query Sets for Example 4

j	1	2	3	4
$Q_j^{(1)}(a_{1,j}, \mathcal{D}; 1)$	$x_{(1-1)\cdot 1+1,1}$	$x_{(2-1)\cdot 1+1,2}$	$x_{(1-1)\cdot 1+1,3}$	$x_{(2-1)\cdot 1+1,4}$
$Q_j^{(1)}(a_{1,j}, \mathcal{U}; 1)$	$y_{(1-1)\cdot 1+1,1}$ $z_{(1-1)\cdot 1+1,1}$ $w_{(1-1)\cdot 1+1,1}$	$y_{(2-1)\cdot 1+1,2}$ $z_{(2-1)\cdot 1+1,2}$ $w_{(2-1)\cdot 1+1,2}$	$y_{(1-1)\cdot 1+1,3}$ $z_{(1-1)\cdot 1+1,3}$ $w_{(1-1)\cdot 1+1,3}$	$y_{(2-1)\cdot 1+1,4}$ $z_{(2-1)\cdot 1+1,4}$ $w_{(2-1)\cdot 1+1,4}$

(a)

j	1	2	3	4
$Q_j^{(1)}(a_{1,j}, \mathcal{D}; 2)$	$x_{1\cdot 2+1,1} + y_{2,1}$ $x_{2\cdot 2+1,1} + z_{2,1}$ $x_{3\cdot 2+1,1} + w_{2,1}$	$x_{1\cdot 2+2,2} + y_{1,2}$ $x_{2\cdot 2+2,2} + z_{1,2}$ $x_{3\cdot 2+2,2} + w_{1,2}$	$x_{1\cdot 2+1,3} + y_{2,3}$ $x_{2\cdot 2+1,3} + z_{2,3}$ $x_{3\cdot 2+1,3} + w_{2,3}$	$x_{1\cdot 2+2,4} + y_{1,4}$ $x_{2\cdot 2+2,4} + z_{1,4}$ $x_{3\cdot 2+2,4} + w_{1,4}$
$Q_j^{(1)}(a_{1,j}, \mathcal{U}; 2)$	$y_{4+1,1} + z_{2+1,1}$ $y_{6+1,1} + w_{2+1,1}$ $z_{6+1,1} + w_{4+1,1}$	$y_{4+2,2} + z_{2+2,2}$ $y_{6+2,2} + w_{2+2,2}$ $z_{6+2,2} + w_{4+2,2}$	$y_{4+1,3} + z_{2+1,3}$ $y_{6+1,3} + w_{2+1,3}$ $z_{6+1,3} + w_{4+1,3}$	$y_{4+2,4} + z_{2+2,4}$ $y_{6+2,4} + w_{2+2,4}$ $z_{6+2,4} + w_{4+2,4}$

(b)

j	1	2	3	4
$Q_j^{(1)}(a_{1,j}, \mathcal{D}; 3)$	$x_{4\cdot 2+1,1} + y_{6,1} + z_{4,1}$ $x_{5\cdot 2+1,1} + y_{8,1} + w_{4,1}$ $x_{6\cdot 2+1,1} + z_{8,1} + w_{6,1}$	$x_{4\cdot 2+2,2} + y_{5,2} + z_{3,2}$ $x_{5\cdot 2+2,2} + y_{7,2} + w_{3,2}$ $x_{6\cdot 2+2,2} + z_{7,2} + w_{5,2}$	$x_{4\cdot 2+1,3} + y_{6,3} + z_{4,3}$ $x_{5\cdot 2+1,3} + y_{8,3} + w_{4,3}$ $x_{6\cdot 2+1,3} + z_{8,3} + w_{6,3}$	$x_{4\cdot 2+2,4} + y_{5,4} + z_{3,4}$ $x_{5\cdot 2+2,4} + y_{7,4} + w_{3,4}$ $x_{6\cdot 2+2,4} + z_{7,4} + w_{5,4}$
$Q_j^{(1)}(a_{1,j}, \mathcal{U}; 3)$	$y_{12+1,1} + z_{10+1,1} + w_{8+1,1}$	$y_{12+2,2} + z_{10+2,2} + w_{8+2,2}$	$y_{12+1,3} + z_{10+1,3} + w_{8+1,3}$	$y_{12+2,4} + z_{10+2,4} + w_{8+2,4}$

(c)

j	1	2	3	4
$Q_j^{(1)}(a_{1,j}, \mathcal{D}; 4)$	$x_{7\cdot 2+1,1} + y_{14,1} + z_{12,1} + w_{10,1}$	$x_{7\cdot 2+2,2} + y_{13,2} + z_{11,2} + w_{9,2}$	$x_{7\cdot 2+1,3} + y_{14,3} + z_{12,3} + w_{10,3}$	$x_{7\cdot 2+2,4} + y_{13,4} + z_{11,4} + w_{9,4}$

(d)

Note: auxiliary query sets for each round $\tau \in [4]$. Highlighted in red is the row indicator $u \in [\nu]$ used in determining the indices of the queried symbols. The magenta dashed arrows and the cyan arrows indicate that the `Exploit-SI` algorithm and the `M-Sym` algorithm are used, respectively.

contains $(\alpha_\tau - 1) - \alpha_{\tau-1} + 1 = \binom{\mu-1}{\tau-1}\kappa^{\mu-(\tau-1+1)}(\nu-\kappa)^{\tau-1}$ distinct symbols from the desired linear function evaluation $\mathbf{U}^{(v)}$.

Example 4 (continued) *After successfully generating the queries for $\nu\alpha_1 = 2$ distinct symbols from the desired linear function evaluation in the initiation step, for round $\tau = 2$ we generate the queries for the following $\nu(\alpha_2 - \alpha_1) = 6$ symbols. To this end, subroutine* `Desired-Q`*, given in Algorithm 3, generates auxiliary query sets $Q_j^{(1)}(a_{i,j}, \mathcal{D}; 2)$ containing distinct symbols from the desired linear function evaluation, following a process similar to Algorithm 2, however with a different method for determining the queried indices. The output of lines 16 to 18 after calling the subroutine* `Desired-Q` *for $u \in [2]$ is as follows.*

j	1	2	3	4
$Q_j^{(1)}(1, \mathcal{D}; 2)$	$x_{1\cdot 2+1,1}$ $x_{5,1}, x_{7,1}$		$x_{1\cdot 2+1,3}$ $x_{5,3}, x_{7,3}$	
$Q_j^{(1)}(2, \mathcal{D}; 2)$		$x_{1\cdot 2+2,2}$ $x_{6,2}, x_{8,2}$		$x_{1\cdot 2+2,4}$ $x_{6,4}, z_{8,4}$

$\triangledown$

***5) Side Information Exploitation*:** In lines 20 to 22, we generate the *side information* query sets $Q_j^{(v)}(b_{i',j}, \mathcal{U}; \tau - 1)$, $i' \in [\nu - \kappa]$, from the auxiliary query sets $Q_1^{(v)}(a_{i,1}, \mathcal{U}; \tau - 1), \ldots, Q_n^{(v)}(a_{i,n}, \mathcal{U}; \tau - 1)$, $i \in [\kappa]$, of the previous round $\tau - 1$, $\tau \in [2 : \mu]$, by applying the subroutine `Exploit-SI`, given by Algorithm 4. This subroutine is extended from [43] based on our coded storage scenario. These side information query sets will be exploited by the user to ensure the recovery and privacy of the PLC scheme. Note that in Algorithm 4 the function `Reproduce`$(j, Q_{j'}^{(v)}(u, \mathcal{U}; \tau - 1))$, $j' \in [n] \setminus \{j\}$, simply reproduces all the queries in the auxiliary query set $Q_{j'}^{(v)}(u, \mathcal{U}; \tau - 1)$ with a different coordinate j.

Next, we update the desired query sets $Q_j^{(v)}(a_{i,j}, \mathcal{D}; \tau)$ in lines 25 to 31. First, the function `Partition`$\big(\tilde{Q}_j^{(v)}(\mathcal{U}; \tau - 1)\big)$ denotes a procedure that divides a set into κ disjoint equally-sized subsets. This is viable since based on the subroutine

`Initial-Round` and the following subroutine `M-Sym`, one can show that $\left|\tilde{Q}_j^{(v)}(\mathcal{U};\tau-1)\right| = \binom{\mu-1}{\tau-1}\kappa^{\mu-(\tau-1)}(\nu-\kappa)^{(\tau-1)-1}\cdot(\nu-\kappa)$ for each round $\tau \in [2:\mu]$, which is always divisible by κ. Secondly, we assign the new query set of desired symbols $Q_j^{(v)}(a_{i,j},\mathcal{D};\tau)$ for the current round by using an element-wise set addition $\texttt{SetAddition}(Q_1,Q_2)$. The element-wise set addition is defined as $\left\{q_{i_l}+q_{i'_l}\colon q_{i_l}\in Q_1, q_{i'_l}\in Q_2, l\in[\rho]\right\}$ with $|Q_1|=|Q_2|=\rho$, where ρ is an appropriate integer.

Algorithm 2: `Initial-Round`

Input : u, α_τ, j, v, and τ

Output: $\varphi^{(v)}(u,\mathcal{D};\tau), \varphi^{(v)}(u,\mathcal{U};\tau)$

$\varphi^{(v)}(u,\mathcal{D};\tau) \leftarrow \emptyset$, $\varphi^{(v)}(u,\mathcal{U};\tau) \leftarrow \emptyset$

for $l \in [\alpha_\tau]$ **do**

$\quad \varphi^{(v)}(u,\mathcal{D};\tau) \leftarrow \varphi^{(v)}(u,\mathcal{D};\tau) \cup \left\{U^{(v)}_{(u-1)\cdot\alpha_\tau+l,j}\right\}$

$\quad \varphi^{(v)}(u,\mathcal{U};\tau) \leftarrow \varphi^{(v)}(u,\mathcal{U};\tau) \cup \left(\bigcup_{v'=1}^{\mu}\left\{U^{(v')}_{(u-1)\cdot\alpha_\tau+l,j}\right\} \setminus \left\{U^{(v)}_{(u-1)\cdot\alpha_\tau+l,j}\right\}\right)$

end

Algorithm 3: `Desired-Q`

Input : u, α_τ, j, v, and τ

Output: $\varphi^{(v)}(u,\mathcal{D};\tau)$

$\varphi^{(v)}(u,\mathcal{D};\tau) \leftarrow \emptyset$

for $l \in [\alpha_{\tau-1}:\alpha_\tau-1]$ **do**

$\quad \varphi^{(v)}(u,\mathcal{D};\tau) \leftarrow \varphi^{(v)}(u,\mathcal{D};\tau) \cup \left\{U^{(v)}_{l\cdot\nu+u,j}\right\}$

end

6) Message and Index Symmetry in Rounds $\tau > 1$**:** In lines 33 to 37, the subroutine `M-Sym`, given in Algorithm 5, is invoked to generate the undesired query sets $Q_j^{(v)}(a_{i,j},\mathcal{U};\tau)$ by utilizing message symmetry. This subroutine selects symbols of undesired functions evaluations to generate τ-sums that enforce symmetry in the round queries. The procedure resembles the subroutine `M-Sym` proposed in [43]. In Algorithm 5, Π_τ denotes the set of all possible selections of τ distinct indices in

Algorithm 4: `Exploit-SI`

Input : u, $Q_1^{(v)}(u,\mathcal{U};\tau-1),\ldots,Q_n^{(v)}(u,\mathcal{U};\tau-1)$, j, v, and τ

Output: $\varphi^{(v)}(u,\mathcal{U};\tau-1)$

$\varphi^{(v)}(u,\mathcal{U};\tau-1) \leftarrow \emptyset$

for $i \in [\kappa]$ **do**

  for $j' \in [n] \setminus \{j\}$ **do**

    if $u = a_{i,j'}$ **then**

      $\varphi^{(v)}(u,\mathcal{U};\tau-1) \leftarrow \texttt{Reproduce}(j, Q_{j'}^{(v)}(u,\mathcal{U};\tau-1))$

      break

    end

  end

end

Algorithm 5: `M-Sym`

Input : $Q_j^{(v)}(u,\mathcal{D};\tau)$, j, v, and τ

Output: $\varphi^{(v)}(u,\mathcal{U};\tau)$

$\varphi^{(v)}(u,\mathcal{U};\tau) \leftarrow \emptyset$

for $(v_1,\ldots,v_\tau) \in \texttt{Lexico}(\Pi_\tau)$, $v \notin \{v_1,\ldots,v_\tau\}$ **do**

  $\varphi^{(v)}(u,\mathcal{U};\tau) \leftarrow \varphi^{(v)}(u,\mathcal{U};\tau) \cup \left\{U_{i_1,j}^{(v_1)} + \ldots + U_{i_\tau,j}^{(v_\tau)}\right\}$ such that $\forall\, z \in [\tau]$, $\exists\, U_{i_z,j}^{(v)} + \sum_{\substack{x\in[\tau]\\ x\neq z}} U_{*,j}^{(v_x)} \in Q_j^{(v)}(u,\mathcal{D};\tau)$

end

$[\mu]$ and $\texttt{Lexico}(\Pi_\tau)$ denotes the corresponding set of ordered selections (the indices $(v_1, \ldots, v_\tau)$ of a selection of Π_τ are ordered in natural lexicographical order). Further, the notation $U_{*,j}^{(v_x)}$ implies that the row index of the symbol can be arbitrary. This is the case since only the function indices $(v_1, \ldots, v_\tau)$ are necessary to determine i_z, $\forall\, z \in [\tau]$. As a result, symmetry over the linear functions is maintained. Moreover, for $Q_j^{(v)}(a_{i,j}, \mathcal{U}; \tau)$, $i \in [\kappa]$, we obtain for each $\tau \in [2:\mu]$ the remaining τ-sum types, such that each of these query sets contains $\binom{\mu-1}{\tau}\kappa^{\mu-(\tau-1+1)}(\nu-\kappa)^{\tau-1}$ symbols.

Example 4 (continued) *After determining the indices of the desired function evaluations to be queried by each database in round $\tau = 2$, we now deploy side information to preserve the privacy for the desired function evaluation. This is accomplished by generating τ-sums of each possible* type *and enforcing index symmetry. To this end, we first identify the side information available from the previous round, queried from the neighboring databases, to be exploited according to the interference matrix $\mathsf{B}_{1\times 4}$. This process is performed by invoking Algorithm 4, which generates* complement *sets for the undesired query sets of the previous round, i.e., $Q_j^{(1)}(a_{i,j}, \mathcal{U}; 1)$. The output of Algorithm 4 for $u \in [2]$ is as follows.*

j	1	2	3	4
$Q_j^{(1)}(1, \mathcal{U}; 1)$		$y_{1,2}, z_{1,2}, w_{1,2}$		$y_{1,4}, z_{1,4}, w_{1,4}$
$Q_j^{(1)}(2, \mathcal{U}; 1)$	$y_{2,1}, z_{2,1}, w_{2,1}$		$y_{2,3}, z_{2,3}, w_{2,3}$	

Next, these side information query sets are then partitioned into κ groups to be exploited in different $Q_j^{(1)}(a_{i,j}, \mathcal{D}; 2)$ for $i \in [\kappa]$. The partitioning guarantees that the two sets used in generating the τ-sums in lines 28 to 30 of Algorithm 1 have an equal number of elements. Finally, message and index symmetry is guaranteed by passing the generated auxiliary query sets $Q_j^{(1)}(a_{i,j}, \mathcal{D}; 2)$ to the subroutine `M-Sym`*, i.e., Algorithm 5, that generates τ-sums of the remaining* types. *Table 3.1(b) illustrates the final query sets for round $\tau = 2$.*

Next, Steps 4) to 6) are repeated for the following rounds, i.e., for $\tau = 3$ and $\tau = 4$. As a result, the queries for $\nu(\alpha_3 - \alpha_2) = 6$ and the remaining $\nu(\alpha_4 - \alpha_3) = 2$ distinct symbols of the desired linear function evaluation are generated by rounds $\tau = 3$ and $\tau = 4$, respectively. Tables 3.1(c)-(d) illustrate the final query sets for the

final rounds. Similar to Table 3.1(a), in Tables 3.1(b)–(d), we highlight with red the row indicator $u = a_{1,j} \in [\nu]$ and with magenta dashed arrows the side information exploitation following the algorithm `Exploit-SI`*, i.e., Algorithm 4. In addition, we indicate with cyan arrows the message symmetry enforcement procedure following the algorithm* `M-Sym`*, i.e., Algorithm 5, and with red the resulting index symmetry in $Q_j^{(1)}(a_{1,j}, \mathcal{U}; \tau)$ based on the desired linear function indices.* $\triangledown$

***7) Query Set Assembly*:** Finally, in lines 39 to 48, we assemble each query set from disjoint query subsets obtained in all τ rounds. It can be shown that $Q_j^{(v)}(\mathcal{D}; \tau) \cup Q_j^{(v)}(\mathcal{U}; \tau)$ contains $\kappa^{\mu-(\tau-1)}(\nu-\kappa)^{\tau-1}$ τ-sums for every τ-sum type as follows. For the initialization round, $\tau = 1$, from Step 3) above, the total number of queried symbols is given by

$$\left|Q_j^{(v)}(\mathcal{D}; 1) \cup Q_j^{(v)}(\mathcal{U}; 1)\right| = \kappa\left[\kappa^{\mu-1} + \binom{\mu-1}{1}\kappa^{\mu-1}\right] = \binom{\mu}{1}\kappa^{\mu-1+1}(\nu-\kappa)^{1-1}.$$

For the following rounds, $\tau \in [2:\mu]$, from Steps 4), 5), and 6) above, we have

$$\begin{aligned}\left|Q_j^{(v)}(\mathcal{D}; \tau) \cup Q_j^{(v)}(\mathcal{U}; \tau)\right| &= \kappa\left[\binom{\mu-1}{\tau-1}\kappa^{\mu-\tau}(\nu-\kappa)^{\tau-1} + \binom{\mu-1}{\tau}\kappa^{\mu-\tau}(\nu-\kappa)^{\tau-1}\right] \\ &= \binom{\mu}{\tau}\kappa^{\mu-\tau+1}(\nu-\kappa)^{\tau-1}.\end{aligned}$$

In summary, the total number of queries generated by Algorithm 1 is

$$\sum_{j=1}^{n}\left|Q_j^{(v)}\right| = n\sum_{\tau=1}^{\mu}\binom{\mu}{\tau}\kappa^{\mu-\tau+1}(\nu-\kappa)^{\tau-1}. \tag{3.10}$$

Remark 1 *The practicality of implementing an algorithm is measured by the algorithm's computational complexity, i.e., the number of operations an algorithm performs to complete its task. The computational complexity of Algorithm 1 can be shown to be $\mathcal{O}(n\kappa\mu\nu^{\mu-1})$. To the best of our knowledge, our scheme shares this exponential time complexity with the PIR and PLC schemes of [7], [8], [15], [43].*

Example 4 (continued) *In the final step, i.e., Step 7), the auxiliary query subsets are aggregated according to the row indicator $u = a_{i,j}$, $i \in [\kappa]$, to form the final*

query set for each database. Note that, by utilizing the code coordinates forming an information set in the code array, it can be shown that the side information based on $\mathsf{B}_{(\nu-\kappa)\times n}$ *can be decoded. For example, in round* 3*, since* $\{2, 4\}$ *is an information set of the storage code* $\mathscr{C}$*, the code symbols* $y_{6,1} + z_{4,1}$ *and* $y_{6,3} + z_{4,3}$ *can be obtained by knowing* $y_{6,2} + z_{4,2}$ *and* $y_{6,4} + z_{4,4}$*, from which the corresponding symbols* $x_{6,1}$ *and* $x_{6,3}$ *can be obtained by canceling the side information. Hence, the symbols from the desired linear function evaluation can be obtained.*

3.3.2 Recovery of Desired Function Evaluation

The construction of the capacity-achieving PLC scheme is, so far, a PIR-like scheme that privately retrieve a virtual message from a linearly-coded DSS. This virtual message represents the evaluation of the desired function over coded symbols, however, the user wishes to privately retrieve the evaluation of the desired function over the original information symbols. As a result, due to the fact that we are performing computation over coded storage, the coded PLC scheme includes two extra steps over other uncoded PC schemes. Namely, decoding the desired function evaluation symbols and decoding and canceling the side information. Thus, the correct decoding of the desired function evaluation relies on the correct decoding of the queried symbols from all virtual messages. To this end, in the following, we show that we can reliably recover the desired function evaluation from the queried symbols.

The main argument behind the reliable recovery of the desired function evaluation is the fact that the candidate linear functions and linear coding commute, i.e., evaluating a function over coded symbols is equal to encoding the symbols of the function evaluation. To see that, let $\hat{t} = \pi(t)$ where $t, \hat{t} \in [\beta]$ be the private permutation selected by the user and let $\boldsymbol{g}_j = \big(g_{1,j}, g_{2,j}, \ldots, g_{k,j}\big)^{\mathsf{T}}$ be the j-th column of the generator matrix $\mathsf{G}^{\mathscr{C}}$ for the $[n, k]$ linear storage code. One can verify, from

equation (3.9), that for all $v' \in [\mu]$, we have

$$U_{t,j}^{(v')} = \boldsymbol{v}_{v'} \boldsymbol{C}_{\hat{t},j} = \sum_{i=1}^{f} v_{v',i} C_{\hat{t},j}^{(i)} = \sum_{i=1}^{f} v_{v',i} \sum_{h=1}^{k} W_{\hat{t},h}^{(i)} g_{h,j}$$
$$= \sum_{h=1}^{k} g_{h,j} \sum_{i=1}^{f} v_{v',i} W_{\hat{t},h}^{(i)} = \sum_{h=1}^{k} X_{\hat{t},h}^{(v')} g_{h,j}, \tag{3.11}$$

where $(X_{1,1}^{(v')}, \ldots, X_{1,k}^{(v')}, X_{2,1}^{(v')}, \ldots, X_{\beta,k}^{(v')}) = (X_1^{(v')}, \ldots, X_k^{(v')}, X_{k+1}^{(v')}, \ldots, X_{\mathrm{L}}^{(v')}) = \mathbf{X}^{(v')}$. Note that equation (3.11) resembles the process of encoding the segment $\left(X_{\hat{t},1}^{(v')}, \ldots, X_{\hat{t},k}^{(v')}\right)$ of the candidate linear function evaluation $\mathbf{X}^{(v')}$ using the $[n, k]$ storage code. Thus, one can consider the construction of our PLC scheme, so far, as a coded PIR scheme over a virtual coded DSS storing the evaluations of the candidate functions. As a result, using the same $[n, k]$ linear code for decoding the symbols obtained from the answer sets guarantees the reliable retrieval of the desired function evaluation.

3.3.3 Sign Assignment and Redundancy Elimination

In contrast to simple PIR solutions, in PLC we have the opportunity to exploit the dependencies induced by performing computations over the same set of messages, i.e., the f independent messages $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$, while keeping the requested index v private from each database. As shown in the recent PC literature (e.g., [43], [47], [78]), one is able to exploit this dependency to optimize the download cost by trading communication overhead with offline computation performed at the user side. To this end, our proposed PLC scheme is further constructed with two additional procedures: *Sign assignment* and *redundancy elimination*.

After running Algorithm 1, the user will know which row indices of the stored code symbols he/she is going to request. To reduce the total number of downloaded symbols, the linear dependency among the candidate linear functions evaluations is exploited. To this end, an initial sign $\sigma_t^{(v)}$ is first privately generated by the user with

a uniform distribution over $\{-1,+1\}$ for all $t \in [\beta]$, i.e., the same selected sign is identically applied to all symbols from different function evaluations with the same index.

Next, depending on the desired linear function index $v \in [\mu]$, we apply a deterministic sign assignment procedure that carefully scales each pre-signed symbol in the query sets, i.e., $\sigma_t^{(v)} U_{t,j}^{(v')}$, $v' \in [\mu]$, by $\{+1,-1\}$. The intuition behind the sign assignment is to introduce a uniquely solvable equation system from the different τ-sum types given the side information available from all other databases. By obtaining such a system of equations in each round, the user can determine some of the queries offline to decode the desired linear function evaluations and/or interference, thus reducing the download rate. On the other hand, the privately selected initial sign for $\sigma_t^{(v)}$, $t \in [\beta]$, acts as a one-time pad that randomizes over the deterministic sign assignment procedure. Here, we adopt a similar sign assignment process over each symbol in the query sets, as introduced in [43, Sec. IV-B]. The sign assigned to each symbol relies on two factors; the position of that symbol within a lexicographically ordered τ-sum query and whether that query contains a symbol from the desired function evaluation. Specifically, let a lexicographically ordered τ-sum query be q, i.e., $q \triangleq \sum_{\ell=1}^{\tau} U^{(v_\ell)}$, $v_1 < \cdots < v_\tau$.[3] Let $\Delta^{(v)}(q)$ denote the position of the symbol associated with the desired function evaluation $\mathbf{X}^{(v)}$ within q, where $\Delta^{(v)}(q) = 0$ indicates that the query does not contain a symbol from the desired function evaluation. The queries generated by Algorithm 1 are sorted by round $\tau \in [\mu]$, then the queries for each round are divided into subgroups indexed by $S(\Delta^{(v)}(q)) \in \{1,2,\ldots\}$ based on the value of $\Delta^{(v)}(q)$ for each query. Finally, a '+' or '$-$' sign is assigned as a function of the subgroup index $S(\cdot)$ and the position of each symbol relative to the desired function evaluation symbol in each query. The details of the sign selection follow [43, Sec. IV-B] and are omitted for brevity. Moreover, we

[3]Segment and database indices are suppressed here for clarity.

remark that after sign assignment, the recovery condition of the scheme is inherently maintained since it can be seen as a coded PIR scheme as Protocol 1 in [15]. The key idea of redundancy elimination is illustrated with Example 4 below.

Example 4 (continued) *First, without loss of generality, we assume the initial sign assignment $\sigma_t^{(v)} = +1$ is privately selected by the user for all $t \in [\beta]$. Next, we apply the sign assignment process to the query sets for $v = 1$. The resulting queries after sign assignment are shown in Table 3.2. In the following, we show that we can remove some redundant queries from each database and the desired linear function evaluation $\mathbf{X}^{(1)}$ can still be recovered. For example, in the first round ($\tau = 1$), it can be easily seen from $\mathsf{V}_{\mu\times f}$ that the queried symbols of $z_{t,j}$ and $w_{t,j}$ can be generated offline by the user as functions of $x_{t,j}$ and $y_{t,j}$, i.e., $z_{t,j} = x_{t,j} + y_{t,j}$ and $w_{t,j} = 3x_{t,j} + y_{t,j}$ for all $t \in [\beta]$ and $j \in [n]$. Moreover, the coefficient vectors associated with $x_{t,j}$ and $y_{t,j}$ are the two row basis vectors of the coefficient matrix $\mathsf{V}_{\mu\times f}$ ($r = \mathrm{rank}(\mathsf{V}) = 2$). Thus, we can represent the candidate functions evaluations in terms of this basis with a deterministic linear mapping $\hat{\mathsf{V}}_{\mu\times r} = (\hat{v}_{i,l})$ of size $\mu \times r$ as follows:*

$$(x_{t,j}, y_{t,j}, z_{t,j}, w_{t,j})^{\mathsf{T}} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 3 & 1 \end{pmatrix}}_{\hat{\mathsf{V}}_{\mu\times r}} (x_{t,j}, y_{t,j})^{\mathsf{T}}. \tag{3.12}$$

That is true due to the commutativity of the performed linear functions, i.e., the storage code and the candidate functions, and given that the coefficient matrix $\mathsf{V}_{\mu\times f}$ of the candidate functions is available to the user. Thus, the queries for these symbols, i.e., $z_{t,j}$ and $w_{t,j}$, are redundant and can be removed from the query sets regardless of which function evaluation is desired by the user. Next, in round $\tau = 2$ and for the 1st database, from the deterministic linear mapping $\hat{\mathsf{V}}_{\mu\times r} = (\hat{v}_{i,l})$ of equation (3.12), one can verify that

$$\begin{aligned}
&\hat{v}_{3,2}(y_{7,1} - w_{3,1}) - \hat{v}_{4,2}(y_{5,1} - z_{3,1}) - (\hat{v}_{3,1} \cdot \hat{v}_{4,2} - \hat{v}_{4,1} \cdot \hat{v}_{3,2})x_{3,1} - \hat{v}_{4,1}x_{5,1} + \hat{v}_{3,1}x_{7,1} \\
&\quad = 1(y_{7,1} - w_{3,1}) - 1(y_{5,1} - z_{3,1}) - (1 \cdot 1 - 3 \cdot 1)x_{3,1} - 3x_{5,1} + 1x_{7,1} \\
&\quad = 1(y_{7,1} - 3x_{3,1} - 1y_{3,1}) - (y_{5,1} - x_{3,1} - y_{3,1}) + 2x_{3,1} - 3x_{5,1} + x_{7,1}
\end{aligned}$$

$$= (x_{7,1} + y_{7,1}) - (3x_{5,1} + y_{5,1}) = z_{7,1} - w_{5,1}, \tag{3.13}$$

and hence we do not need to download the 2-sum $z_{7,1} - w_{5,1}$. *Similarly, we can do the same exercise for the other databases. The redundant queries are marked in blue in Table 3.2, shown at the top of the following page, and the indices* $t \in [\beta]$ *of the desired linear function evaluations are marked in red. This completes the recovery part. The resulting PLC rate becomes* $\frac{\nu^{\mu} \cdot k}{\mathrm{D}} = \frac{16 \cdot 2}{12 \cdot 4} = \frac{2}{3}$, *which is equal to the PLC capacity in Theorem 3 with* $r = \mathrm{rank}(\mathsf{V}) = 2$. *This demonstrates the optimality of the PLC scheme.* ▽

Table 3.2 PLC Query Sets for $v = 1$ after Sign Assignment

j	1	2	3	4
$Q_j^{(v)}(\mathcal{D}; 1)$	$x_{1,1}$	$x_{2,2}$	$x_{1,3}$	$x_{2,4}$
$Q_j^{(v)}(\mathcal{U}; 1)$	$y_{1,1}, z_{1,1}, w_{1,1}$	$y_{2,2}, z_{2,2}, w_{2,2}$	$y_{1,3}, z_{1,3}, w_{1,3}$	$y_{2,4}, z_{2,4}, w_{2,4}$
$Q_j^{(v)}(\mathcal{D}; 2)$	$x_{3,1} - y_{2,1}$ $x_{5,1} - z_{2,1}$ $x_{7,1} - w_{2,1}$	$x_{4,2} - y_{1,2}$ $x_{6,2} - z_{1,2}$ $x_{8,2} - w_{1,2}$	$x_{3,3} - y_{2,3}$ $x_{5,3} - z_{2,3}$ $x_{7,3} - w_{2,3}$	$x_{4,4} - y_{1,4}$ $x_{6,4} - z_{1,4}$ $x_{8,4} - w_{1,4}$
$Q_j^{(v)}(\mathcal{U}; 2)$	$y_{5,1} - z_{3,1}$ $y_{7,1} - w_{3,1}$ $z_{7,1} - w_{5,1}$	$y_{6,2} - z_{4,2}$ $y_{8,2} - w_{4,2}$ $z_{8,2} - w_{6,2}$	$y_{5,3} - z_{3,3}$ $y_{7,3} - w_{3,3}$ $z_{7,3} - w_{5,3}$	$y_{6,4} - z_{4,4}$ $y_{8,4} - w_{4,4}$ $z_{8,4} - w_{6,4}$
$Q_j^{(v)}(\mathcal{D}; 3)$	$x_{9,1} - y_{6,1} + z_{4,1}$ $x_{11,1} - y_{8,1} + w_{4,1}$ $x_{13,1} - z_{8,1} + w_{6,1}$	$x_{10,2} - y_{5,2} + z_{3,2}$ $x_{12,2} - y_{7,2} + w_{3,2}$ $x_{14,2} - z_{7,2} + w_{5,2}$	$x_{9,3} - y_{6,3} + z_{4,3}$ $x_{11,3} - y_{8,3} + w_{4,3}$ $x_{13,3} - z_{8,3} + w_{6,3}$	$x_{10,4} - y_{5,4} + z_{3,4}$ $x_{12,4} - y_{7,4} + w_{3,4}$ $x_{14,4} - z_{7,4} + w_{5,4}$
$Q_j^{(v)}(\mathcal{U}; 3)$	$y_{13,1} - z_{11,1} + w_{9,1}$	$y_{14,2} - z_{12,2} + w_{10,2}$	$y_{13,3} - z_{11,3} + w_{9,3}$	$y_{14,4} - z_{12,4} + w_{10,4}$
$Q_j^{(v)}(\mathcal{D}; 4)$	$x_{15,1} - y_{14,1} + z_{12,1} - w_{10,1}$	$x_{16,2} - y_{13,2} + z_{11,2} - w_{9,2}$	$x_{15,3} - y_{14,3} + z_{12,3} - w_{10,3}$	$x_{16,4} - y_{13,4} + z_{11,4} - w_{9,4}$

Note: PLC query sets for $v = 1$ for rounds one to four for the $[4, 2]$ code of Example 4, $f = 4$ messages, and $\mu = 4$ candidate linear functions. Red subscripts indicate the indices of the desired linear function evaluations. The redundant queries are marked in blue.

From the above example we note the following.

- There is a deterministic linear mapping, i.e., $\hat{\mathsf{V}}_{\mu \times r}$, that captures the dependencies among the candidate linear functions evaluations.
- We maintain the same characteristics of the query construction that facilitate the exploitation of the linear dependencies among the candidate functions evaluations as for the uncoded PLC scheme in [43]. These characteristics include index assignment, sign assignment, and lexicographic ordering of the elements of

τ-sums. As a result, some of the queries become redundant and can be removed from the query sets while maintaining the decodability of the desired function evaluation.

- The candidate functions are computed over the coded symbols stored in each database individually. Consequently, from the perspective of the queries of each database, the linear dependency among the symbols of the candidate functions evaluations is present, i.e., the fact that the computation is performed over coded storage is transparent to the redundancy elimination process. This can be seen from equation (3.13).
- The number of redundant queries depends on the rank of the coefficient matrix $\mathsf{V}_{\mu\times f}$, i.e., $r = \text{rank}(\mathsf{V})$. This can be clearly observed for the 1-sum symbols where out of the μ symbols, $\mu - r$ can be computed offline given that the symbols of the functions evaluations associated with the r row basis vectors of $\mathsf{V}_{\mu\times f}$ are available.

Based on this insight we can state the following lemma for redundancy elimination.

Lemma 4 *For all $v \in [\mu]$, each database $j \in [n]$, and based on the side information available from the databases, any $\binom{\mu-r}{\tau}$ τ-sum types out of all possible $\binom{\mu}{\tau}$ types in each round $\tau \in [\mu - r]$ of the query sets are redundant.*

The proof of Lemma 4 is presented in Appendix C. The proof is based on the insight that the redundancy resulting from the linear dependencies between virtual messages is also present with MDS-PIR capacity-achieving codes. Since both repetition and MDS codes are MDS-PIR capacity-achieving codes, Lemma 4 generalizes both [43, Lem. 1] and [78, Lem. 1]. We now make the final modification to our PLC query sets by first directly applying the sign assignment over $\sigma_t^{(v)} U_{t,j}^{(v')}$, $v' \in [\mu]$, and then remove the τ-sums corresponding to the redundant τ-sum types from every round $\tau \in [\mu - r]$. Note that the amount of redundancy is dependent on the rank of the functions matrix, $\text{rank}(\mathsf{V}) = r \leq \min\{\mu, f\}$, thus generalizing the MDS-coded PLC case. Finally, we generate the queries $Q_{[n]}^{(v)}$.

3.3.4 Privacy

It is worth mentioning that the queries generated by Algorithm 1 inherently satisfy the privacy condition of equation (2.5), which is guaranteed by satisfying the index,

message, and database symmetry principles as for all the PIR schemes in [7], [8], [15]. That is, given the fixed and symmetric construction of the queries, there always exists a one-to-one mapping between the queries, $Q_j^{(v)} \leftrightarrow Q_j^{(v')}$, $\forall\, j \in [n]$, in terms of the queried symbols indices $t \in [\beta]$, where $v, v' \in [\mu]$ and $v \neq v'$. Given this one-to-one mapping along with a permutation $\pi(t)$ over these indices privately selected uniformly at random by the user, the queries are indistinguishable and equally likely.

Moreover, after the sign assignment process a one-to-one mapping between the assigned signs is found following a simple sign flipping rule for $\sigma_t^{(v')}$. The rule states that, to map the queries of $Q_j^{(v)}$ to $Q_j^{(v')}$, one should only consider the desired queries, i.e., queries that contain symbols associated with $\mathbf{X}^{(v')}$. For such queries in each round τ, we replace $\sigma_*^{(v')}$ with $-\sigma_*^{(v')}$ for each element to the right of the desired function evaluation symbol $U_*^{(v')}$ in the lexicographically ordered query if the query is sorted in a subgroup indexed with an odd S (see Section 3.3.3). Next, we flip the sign of elements to the left of the desired function evaluation symbol $U_*^{(v')}$ if the query is sorted in a subgroup indexed with an even S. The proof of the correctness of this rule and thus the privacy after sign assignment follows directly from [43, Sec. VI-B]. For completeness, we also show with Example 4 that the user's privacy is still maintained after the sign assignment process and the removal of redundant queries.

Example 4 (continued) *Here, to show that the queries are identically distributed regardless of the desired function evaluation index $v \in [4]$ we show that there exists a one-to-one mapping from the queries for $v = 1$ to the queries for $v = 3$ for all databases. Without loss of generality, we again assume the initial sign assignment $\sigma_t^{(3)} = +1$ to be privately selected by the user for all $t \in [\beta]$. In Table 3.3, shown at the top of the following page, the queries for $v = 3$ are presented following Algorithm 1 and the sign assignment process. From Tables 3.2 and 3.3 one can verify that the index and sign mapping*

Table 3.3 PLC Query Sets for $v = 3$ after Sign Assignment

j	1	2	3	4
$Q_j^{(v)}(\mathcal{D};1)$	$z_{1,1}$	$z_{2,2}$	$z_{1,3}$	$z_{2,4}$
$Q_j^{(v)}(\mathcal{U};1)$	$x_{1,1}, y_{1,1}, w_{1,1}$	$x_{2,2}, y_{2,2}, w_{2,2}$	$x_{1,3}, y_{1,3}, w_{1,3}$	$x_{2,4}, y_{2,4}, w_{2,4}$
$Q_j^{(v)}(\mathcal{D};2)$	$x_{2,1} - z_{3,1}$ $y_{2,1} - z_{5,1}$ $z_{7,1} - w_{2,1}$	$x_{1,2} - z_{4,2}$ $y_{1,2} - z_{6,2}$ $z_{8,2} - w_{1,2}$	$x_{2,3} - z_{3,3}$ $y_{2,3} - z_{5,3}$ $z_{7,3} - w_{2,3}$	$x_{1,4} - z_{4,4}$ $y_{1,4} - z_{6,4}$ $z_{8,4} - w_{1,4}$
$Q_j^{(v)}(\mathcal{U};2)$	$x_{5,1} - y_{3,1}$ $x_{7,1} - w_{3,1}$ $y_{7,1} - w_{5,1}$	$x_{6,2} - y_{4,2}$ $x_{8,2} - w_{4,2}$ $y_{8,2} - w_{6,2}$	$x_{5,3} - y_{3,3}$ $x_{7,3} - w_{3,3}$ $y_{7,3} - w_{5,3}$	$x_{6,4} - y_{4,4}$ $x_{8,4} - w_{4,4}$ $y_{8,4} - w_{6,4}$
$Q_j^{(v)}(\mathcal{D};3)$	$x_{6,1} - y_{4,1} + z_{9,1}$ $-x_{8,1} - z_{11,1} + w_{4,1}$ $-y_{8,1} - z_{13,1} + w_{6,1}$	$x_{5,2} - y_{3,2} + z_{10,2}$ $-x_{7,2} - z_{12,2} + w_{3,2}$ $-y_{7,2} - z_{14,2} + w_{5,2}$	$x_{6,3} - y_{4,3} + z_{9,3}$ $-x_{8,3} - z_{11,3} + w_{4,3}$ $-y_{8,3} - z_{13,3} + w_{6,3}$	$x_{5,4} - y_{3,4} + z_{10,4}$ $-x_{7,4} - z_{12,4} + w_{3,4}$ $-y_{7,4} - z_{14,4} + w_{5,4}$
$Q_j^{(v)}(\mathcal{U};3)$	$x_{13,1} - y_{11,1} + w_{9,1}$	$x_{14,2} - y_{12,2} + w_{10,2}$	$x_{13,3} - y_{11,3} + w_{9,3}$	$x_{14,4} - y_{12,4} + w_{10,4}$
$Q_j^{(v)}(\mathcal{D};4)$	$x_{14,1} - y_{12,1} + z_{15,1} + w_{10,1}$	$x_{13,2} - y_{11,2} + z_{16,2} + w_{9,2}$	$x_{14,3} - y_{12,3} + z_{15,3} + w_{10,3}$	$x_{13,4} - y_{11,4} + z_{16,4} + w_{9,4}$

Note: PLC query sets for rounds one to four for the $[4, 2]$ code of Example 4, $f = 4$ messages, and $\mu = 4$ candidate linear functions. Red subscripts indicate the indices of the desired linear function evaluations. The redundant queries are marked in blue.

Databases 1 and 3:

$$\begin{aligned}&(3, 2, 5, 9, 6, 4, 11, 8, 13, \sigma_{13}^{(1)}, 15, 14, 12, \sigma_{10}^{(1)})\\ \xrightarrow{v=3}\ &(5, 3, 2, 6, 4, 9, 13, 11, 8, -\sigma_{8}^{(3)}, 14, 12, 15, -\sigma_{10}^{(3)})\end{aligned} \tag{3.14}$$

Databases 2 and 4:

$$\begin{aligned}&(4, 1, 6, 10, 5, 3, 12, 7, 14, \sigma_{14}^{(1)}, 16, 13, 11, \sigma_{9}^{(1)})\\ \xrightarrow{v=3}\ &(6, 4, 1, 5, 3, 10, 14, 12, 7, -\sigma_{7}^{(3)}, 13, 11, 16, -\sigma_{9}^{(3)})\end{aligned} \tag{3.15}$$

converts the queries for $v = 1$ to the queries for $v = 3$. To see this mapping, compare the τ-sums $x_{t_1,1} - y_{t_2,1}$ and $x_{t'_1,1} - y_{t'_2,1}$ from the queries of the first database of Tables 3.2 and 3.3, respectively. It can be seen that the indices $t_1 = 3$ and $t_2 = 2$ of the queries for $v = 1$ convert into the indices $t'_1 = 5$ and $t'_2 = 3$ of the queries for $v = 3$, respectively. Thus, we have the mapping $((t_1, t_2) \to (t'_1, t'_2)) = ((3, 2) \to (5, 3))$ and due to the index symmetry of the query construction this mapping is fixed for all

symbols with the corresponding indices. A similar comparison between the remaining τ-sums results in the index and sign mapping of equations (3.14) *and* (3.15).

One can similarly verify that there exists a mapping from the queries for $v = 1$ to the queries for $v = 2$ or those for $v = 4$, i.e., $Q_{[n]}^{(1)} \leftrightarrow Q_{[n]}^{(2)}$ and $Q_{[n]}^{(1)} \leftrightarrow Q_{[n]}^{(4)}$. Since a permutation over these indices, i.e., $\pi(t)$ and an initial sign $\sigma_t^{(v)}$ are uniformly and privately selected by the user independently of the desired function evaluation index v, these queries are equally likely and indistinguishable.

Next, to verify the correctness of the sign flipping rule stated above, consider the desired queries of the third round ($\tau = 3$) for the query sets for $v = 3$ in Table 3.3. For database 1, *one can verify that the query $x_{6,1} - y_{4,1} + z_{9,1}$ is sorted in the subgroup indexed by $S = 1$. As S is odd and no element is placed to the right of $z_{9,1}$ the signs are left unchanged. However, for the query $-x_{8,1} - z_{11,1} + w_{4,1}$ which falls in the subgroup indexed by $S = 2$, the sign of the element to the left of $z_{11,1}$, i.e., $x_{8,1}$, is flipped. That is, we change $\sigma_8^{(3)}$ to $-\sigma_8^{(3)}$ and that matches the sign mapping in equation* (3.14) *for this query. Moreover, due to index symmetry, this mapping also matches the sign assignment for $\sigma_8^{(3)}$ for the query $-y_{8,1} - z_{13,1} + w_{6,1}$.*

Finally, for redundancy elimination, we only need to show that for any desired index $v \in [4]$, the removed redundant τ-sums can be chosen to be of the same type. For instance, let us consider the 1*st database. In the* 2*nd round, see Table 3.3, it can be shown that the queries for desired index $v = 3$ satisfy the equation*

$$\begin{aligned}
&(1 \cdot 1 - 3 \cdot 1)(x_{5,1} - y_{3,1}) - 1(x_{7,1} - w_{3,1}) - 3z_{3,1} - 1z_{5,1} + 1z_{7,1} \\
&= -2(x_{5,1} - y_{3,1}) - (x_{7,1} - 3x_{3,1} - y_{3,1}) - 3(x_{3,1} + y_{3,1}) \\
&\qquad -(x_{5,1} + y_{5,1}) + (x_{7,1} + y_{7,1}) \\
&= 1(y_{7,1} - (3x_{5,1} + y_{5,1})) \\
&= y_{7,1} - w_{5,1}, \qquad (3.16)
\end{aligned}$$

which implies that the 2*-sum $z_{7,1} - w_{2,1}$ can be removed from the download, since $z_{7,1}$ can be obtained from downloading $x_{5,1} - y_{3,1}$, $x_{7,1} - w_{3,1}$, $x_{2,1} - z_{3,1}$, $y_{2,1} - z_{5,1}$, and $y_{7,1} - w_{5,1}$. Hence, the redundant τ-sum type for $v = 3$ can be chosen to be equal to the redundant τ-sum type for $v = 1$ (see equation* (3.13)*). A similar argument can be made for $v = 2$ and $v = 4$, which ensures that the privacy of the scheme is not affected by redundancy elimination.* $\triangledown$

3.3.5 Achievable PLC Rate

The resulting achievable PLC rate of Algorithm 1 after removing redundant τ-sums according to Lemma 4 becomes

$$\begin{aligned}
\mathsf{R} &\overset{(a)}{=} \frac{k\nu^{\mu}}{n\sum_{\tau=1}^{\mu}\left(\binom{\mu}{\tau}-\binom{\mu-r}{\tau}\right)\kappa^{\mu-(\tau-1)}(\nu-\kappa)^{\tau-1}} \\
&\overset{(b)}{=} \frac{\kappa\nu^{\mu}}{\nu\sum_{\tau=1}^{\mu}\left(\binom{\mu}{\tau}-\binom{\mu-r}{\tau}\right)\kappa^{\mu-(\tau-1)}(\nu-\kappa)^{\tau-1}} \\
&= \frac{\nu^{\mu}\left(\frac{\nu-\kappa}{\nu}\right)}{\sum_{\tau=1}^{\mu}\left(\binom{\mu}{\tau}-\binom{\mu-r}{\tau}\right)\kappa^{\mu-\tau}(\nu-\kappa)^{\tau}} \\
&\quad\vdots \\
&\overset{(c)}{=} \frac{\nu^{\mu}\left(1-\frac{\kappa}{\nu}\right)}{\nu^{\mu}-\kappa^{r}\nu^{\mu-r}} = \left(1-\frac{\kappa}{\nu}\right)\left[1-\left(\frac{\kappa}{\nu}\right)^{r}\right]^{-1},
\end{aligned} \tag{3.17}$$

where we recall that $\binom{m}{n} = 0$ if $m < n$; (a) follows from the PLC rate in Definition 1, equation (3.10), and Lemma 4; (b) follows from Definition 3; and (c) follows by adapting similar steps as in the proof given in [78]. Note that the rate in equation (3.17) matches the converse in Theorem 2, which proves Theorem 3.

3.4 Conclusion

In this chapter, we have provided the capacity of PLC from coded DSSs, where data is encoded and stored using an arbitrary linear code from a large class of linear storage codes. Interestingly, for the considered family of linear storage codes, the capacity of PLC is equal to the corresponding PIR capacity. Thus, privately retrieving arbitrary linear combinations of the stored messages does not incur any overhead in rate compared to retrieving a single message from the databases and provides a significant advantage over individually downloading each message via a PIR scheme and combining them offline.

CHAPTER 4

PRIVATE POLYNOMIAL FUNCTION COMPUTATION FOR NONCOLLUDING CODED DATABASES

4.1 Introduction

In this chapter[4], we consider the problem of private polynomial computation (PPC) from a distributed storage system (DSS). In such setting a user wishes to compute a multivariate polynomial of degree at most g over f variables (or messages) stored in n noncolluding coded databases, i.e., databases storing data encoded with an $[n, k]$ linear storage code, while revealing no information about the desired polynomial evaluation to the databases.

For a DSS setup where data is stored using linear storage codes, we derive an outer bound on the PPC rate, which is defined as the ratio of the (minimum) desired amount of information and the total amount of downloaded information, and construct two novel PPC schemes. In the first scheme, we consider Reed-Solomon coded databases with Lagrange encoding, which leverages ideas from recently proposed star-product PIR and Lagrange coded computation. The second scheme considers the special case of coded databases with systematic Lagrange encoding. Both schemes yield improved rates, while asymptotically, as $f \to \infty$, the systematic scheme gives a significantly better computation retrieval rate compared to all known schemes up to some storage code rate that depends on the maximum degree of the candidate polynomials.

The PPC problem for coded DSSs is described as in Section 2.2. Here, without loss of generality, we also assume that the polynomial candidate set contains its monomial basis, i.e., all monomials required to represent the polynomials in the candidate set as linear combinations of monomials, are included in the candidate

[4]The material presented in this chapter is published in [79].

set. Moreover, similar to the PLC problem for linearly coded DSSs of Chapter 3, we assume error-free recovery; hence, the recoverability constraint of the PC protocol given in equation (2.6) becomes

$$\text{[Recovery]} \qquad \mathsf{H}\big(\mathbf{X}^{(v)} \,\big|\, A_{[n]}^{(v)}, Q_{[n]}^{(v)}\big) = 0. \tag{4.1}$$

In the following, we outlines some useful definitions. Then, the remainder of the chapter is organized as follows. We derive the converse bound for an arbitrary number of messages and polynomial functions in Section 4.2. In Sections 4.3 and 4.4, we propose two PPC schemes for RS-coded storage with examples . Then, in Section 4.5, numerical results for the proposed PPC schemes and the converse bound from Section 4.2 are presented, establishing the achievability of larger retrieval rates compared with PPC schemes from the literature. Some conclusions are drawn in Section 4.6.

4.1.1 Background

Definition 7 (Star-product) *Let $\mathscr{C}$ and $\mathscr{D}$ be two linear codes of length n over $\mathbb{F}_q$. The star-product (Hadamard product) of $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathscr{C}$ and $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathscr{D}$ is defined as $\boldsymbol{v} \star \boldsymbol{u} = (v_1 u_1, \ldots, v_n u_n) \in \mathbb{F}_q^n$. Further, the star-product of $\mathscr{C}$ and $\mathscr{D}$, denoted by $\mathscr{C} \star \mathscr{D}$, is defined by* $\operatorname{span}\{\boldsymbol{v} \star \boldsymbol{u} : \boldsymbol{v} \in \mathscr{C}, \boldsymbol{u} \in \mathscr{D}\}$ *and the g-fold star-product of $\mathscr{C}$ with itself is given by* $\mathscr{C}^{\star g} = \operatorname{span}\{\boldsymbol{v}_1 \star \cdots \star \boldsymbol{v}_g : \boldsymbol{v}_i \in \mathscr{C}, i \in [g]\}$.

Definition 8 (RS code) *Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ be a vector of n distinct elements of $\mathbb{F}_q$. For $n \in \mathbb{N}$, $k \in [n]$, and $q \geq n$, the $[n, k]$ RS code (over $\mathbb{F}_q$) is defined as*

$$\mathcal{RS}_k(\boldsymbol{\alpha}) \triangleq \{(\phi(\alpha_1), \ldots, \phi(\alpha_n)) \colon \phi \in \mathbb{F}_q[z], \deg(\phi) < k\}. \tag{4.2}$$

It is well-known that RS codes are MDS codes that behave well under the star-product. We state the following proposition that was introduced in [33].

Proposition 1 *Let $\mathcal{RS}_k(\boldsymbol{\alpha})$ be a length-n RS code. Then, for $g \in \mathbb{N}$, the g-fold star-product of $\mathcal{RS}_k(\boldsymbol{\alpha})$ with itself is the RS code given by $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha}) = \mathcal{RS}_{\min\{g(k-1)+1, n\}}(\boldsymbol{\alpha})$.*

Let $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k)$ be a vector of k distinct elements of $\mathbb{F}_q$. For a message vector $\boldsymbol{W} = (W_1, \ldots, W_k)$, let $\ell(z) \in \mathbb{F}_q[z]$ be a polynomial of degree at most $k-1$ such that $\ell(\gamma_i) = W_i$ for all $i \in [k]$. Using the Lagrange interpolation formula we present this polynomial as $\ell(z) = \sum_{i\in[k]} W_i \iota_i(z)$, where $\iota_i(z)$ is the Lagrange basis polynomial

$$\iota_i(z) = \prod_{t\in[k]\setminus\{i\}} \frac{z-\gamma_t}{\gamma_i - \gamma_t}.$$

It was shown in [46] that Lagrange encoding is equivalent to the choice of a specific basis for an RS code. Therefore, for encoding we choose the set of Lagrange basis polynomials as the code generating polynomials of equation (4.2) [54]. Thus, a generator matrix of $\mathcal{RS}_k(\boldsymbol{\alpha})$ is $\mathsf{G}_{\mathcal{RS}_k}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = (\iota_i(\alpha_j))$, $i \in [k]$, $j \in [n]$. Note that if we choose $\gamma_i = \alpha_i$ for $i \in [k]$, then the generator matrix $\mathsf{G}_{\mathcal{RS}_k}(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ becomes systematic.

4.2 Converse Bound

In Section 3.2, the PLC capacity for a coded DSS using an MDS-PIR capacity-achieving code is shown to be equal to the MDS-PIR capacity. In this section, we derive an outer bound on the PPC rate (Theorem 4 below) by adapting the converse proof of Theorem 2 in Section 3.2 to the scenario of the linearly-coded PPC problem, where the storage code is MDS-PIR capacity-achieving. The converse is valid for any number of messages f and candidate functions μ. We first define an effective rank for the PPC problem as follows.

Definition 9 *Let* $\mathbf{X}^{[\mu]} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}\}$ *denote the set of candidate polynomials evaluations where* $\mathbf{X}^{(\ell)} = \big(X_1^{(\ell)}, \ldots, X_{\mathrm{L}}^{(\ell)}\big)$, $\ell \in [\mu]$. *The* effective rank $r\big(\mathbf{X}^{[\mu]}\big)$ *is defined as*

$$r\big(\mathbf{X}^{[\mu]}\big) \triangleq \min\Big\{s\colon \mathsf{H}\big(X_l^{(\ell_1)}, \ldots, X_l^{(\ell_s)}\big) = \mathsf{H}\big(X_l^{[\mu]}\big), \\ \{\ell_1, \ldots, \ell_s\} \subseteq [\mu],\, s \in [\mu],\, \forall\, l \in [\mathrm{L}]\Big\}, \tag{4.3}$$

and we define the set $\mathcal{L} \triangleq \{\ell_1, \ldots, \ell_r\} \subseteq [\mu]$ *to be a minimum set that satisfies equation* (4.3).[5]

Accordingly, an upper bound on the capacity of PPC for a coded DSS where data is encoded and stored using an MDS-PIR capacity-achieving code introduced in Definition 3, is stated as follows.

Theorem 4 *Consider a DSS with* n *noncolluding databases that uses an* $[n, k]$ *MDS-PIR capacity-achieving code* $\mathscr{C}$ *to store* f *messages. Then, the maximum achievable PPC rate over all possible PPC protocols, i.e., the PPC capacity* $\mathsf{C}_{\mathrm{PPC}}$*, is upper bounded by*

$$\mathsf{C}_{\mathrm{PPC}} \leq \frac{\mathsf{H}_{\min}}{\mathsf{H}^{(\mathrm{B})}_{\min} + \sum_{v=1}^{r-1} \left(\frac{k}{n}\right)^v \mathsf{H}\left(X^{(\ell_{v+1})} \middle| X^{(\ell_1)}, \ldots, X^{(\ell_v)}\right)},$$

for any effective rank $r\left(\mathbf{X}^{[\mu]}\right) = r$*, where* $\mathsf{H}^{(\mathrm{B})}_{\min} \triangleq \min_{\ell \in \mathcal{L}} \mathsf{H}\left(X^{(\ell)}\right)$.

Here, we remark that Theorem 4 generalizes [44, Thm. 1], which is a converse bound on the capacity of dependent PIR (DPIR) for noncolluding replicated databases.

Remark 2 *Restricting the candidate set to degree* $g = 1$ *polynomials reduces the PPC problem to a PLC problem where there is a deterministic linear mapping* $\mathsf{V}_{\mu \times f}$ *between the* μ *functions evaluations and the* f *information messages. Thus, the effective rank given in Definition 9 becomes the rank of said mapping, i.e.,* $r = \mathrm{rank}(\mathsf{V}_{\mu \times f})$*. Moreover, the candidate functions evaluations with indices from the set* $\mathcal{L} = \{\ell_1, \ldots, \ell_r\}$ *that satisfies equation* (4.3) *are independent and identically distributed according to a uniform distribution (see Lemma 3 in Section 3.2). As a result, for* $v \in [r-1]$ *we have* $\mathsf{H}\left(X^{(\ell_{v+1})} \middle| X^{\{\ell_1, \ldots, \ell_v\}}\right) = \mathsf{H}\left(X^{(\ell_{v+1})}\right) = 1$*,* $\mathsf{H}^{(\mathrm{B})}_{\min} = \mathsf{H}_{\min} = 1$*, and the capacity of PLC (see Theorem 2 in Section 3.2) i.e.,* $\mathsf{C}_{\mathrm{PLC}} = \left(1 - k/n\right)\left[1 - \left(k/n\right)^r\right]^{-1}$*, follows.*

[5] *There always exists a subset* $\{\ell_1, \ldots, \ell_s\} \subseteq [\mu]$ *that satisfies the joint entropy condition of equation* (4.3)*. For the case where the candidate functions result in independent functions evaluations, this set is the set of all function evaluations, i.e.,* $r = \mu$*. Moreover, we naturally assume that* $r > 1$*, as* $\mu > 1$ *and* $f > 1$*. Otherwise, the problem becomes trivial in the sense that there is only one candidate message/computation to retrieve.*

Accordingly, the proof of Theorem 4 is an extension to our converse proof of PLC in Section 3.2 and is presented in Appendix D.

Remark 3 *Note that the converse bound of Theorem 4 is generally difficult to compute for a large number of candidate polynomials. However, it is worth mentioning that there are two cases where the computation of the converse bound is straightforward. Namely, the case of the candidate functions being from the* linear polynomials *class, following Remark 2, and the case where the set of μ candidate polynomials evaluations includes the f* independent *files, i.e., $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}\} \subset \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}\}$. For this case, the rank of the candidate functions set is simply $r = f$ as all the remaining candidate polynomials evaluations are a function of these f files and no other smaller subset captures the value of the joint entropy $\mathsf{H}\big(X_l^{[\mu]}\big)$ of equation* (4.3)*. Since these f files are independent and uniformly distributed, computing the capacity bound reduces to computing only the minimum entropy.*

4.3 General PPC Scheme for RS-Coded DSSs

In the following, we build PPC schemes based on Lagrange encoding and our PLC scheme in Section 3.3. Note that a polynomial can be written as a linear combination of monomials, and therefore PMC is a special case of PPC. Thus, a PPC scheme can be obtained from a PLC scheme by replacing independent messages with a monomial basis. We first discuss the PPC case in general in the following scheme. In RS-coded DSSs, each message vector $\boldsymbol{W}_i^{(m)}$ is encoded by an RS code $\mathcal{RS}_k(\boldsymbol{\alpha})$ with evaluation vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ over $\mathbb{F}_q$ into a length-n codeword $\boldsymbol{C}_i^{(m)}$ where $\boldsymbol{C}_i^{(m)} = \boldsymbol{W}_i^{(m)} \mathsf{G}_{\mathcal{RS}_k}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = \big(C_{i,1}^{(m)}, \ldots, C_{i,n}^{(m)}\big)$ and $C_{i,j}^{(m)} = \ell(\alpha_j)$, $j \in [n]$. Consider an RS-coded DSS with n noncolluding databases storing f messages. The user wishes to retrieve the v-th polynomial evaluation $\mathbf{X}^{(v)}$, $v \in [\mu]$, from the available information from queries $Q_j^{(v)}$ and answer strings $A_j^{(v)}$, $j \in [n]$, satisfying the conditions of equations (2.5) and (4.1).

4.3.1 Lagrange Coded Computation

Lagrange coded computation [54] is a framework that can be applied to any function computation when the function of interest is a multivariate polynomial of the messages. We extend the application of this framework to PMC and PPC by utilizing the following argument.

Let $\ell_t^{(m)}(z)$ be the Lagrange interpolation polynomial associated with the length-k message segment $\boldsymbol{W}_t^{(m)}$ for some $t \in [\beta]$ and $m \in [f]$. Recall that $\ell_t^{(m)}(z)$ evaluated at γ_j results in an information symbol $W_{t,j}^{(m)}$ and when evaluated at α_j we obtain a code symbol $C_{t,j}^{(m)}$. Let $\boldsymbol{\ell}_t(z) = (\ell_t^{(1)}(z), \ldots, \ell_t^{(f)}(z))$ be a vector of f Lagrange interpolation polynomials associated with the messages $\boldsymbol{W}_t^{(1)}, \ldots, \boldsymbol{W}_t^{(f)}$. Now, given a multivariate polynomial $\phi(\boldsymbol{W}_{t,j})$ of degree at most g, where $\boldsymbol{W}_{t,j} \triangleq \left(W_{t,j}^{(1)}, \ldots, W_{t,j}^{(f)}\right)^{\mathsf{T}}$, we introduce the composition function $\psi_t(z) = \phi(\boldsymbol{\ell}_t(z))$. Accordingly, evaluating $\psi_t(z)$ at any γ_j, $j \in [k]$, is equal to evaluating the polynomial over the uncoded information symbols, i.e., $\phi(\boldsymbol{W}_{t,j})$ and similarly, evaluating $\psi_t(z)$ at α_j, $j \in [n]$, will result in the evaluation of the polynomial over the coded symbols, i.e., $\phi(\boldsymbol{C}_{t,j})$, where $\boldsymbol{C}_{t,j} \triangleq \left(C_{t,j}^{(1)}, \ldots, C_{t,j}^{(f)}\right)^{\mathsf{T}}$. Since each Lagrange interpolation polynomial of $\boldsymbol{\ell}_t(z)$ is a polynomial of degree at most $k-1$, it follows that $\deg(\psi_t(z)) \leq g(k-1)$ and we require up to $g(k-1)+1$ coefficients to interpolate and determine the polynomial $\psi_t(z)$.

Note that $\psi_t(z)$ is a linear combination of monomials $z^i \in \mathbb{F}_q[z]$, $i \leq g(k-1)$, and the underlying code $\tilde{\mathscr{C}}$ for $(\psi_t(\alpha_1), \ldots, \psi_t(\alpha_n))$, referred to as the *polynomial decoding code*, is given by the g-fold star-product $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha})$ of the storage code $\mathcal{RS}_k(\boldsymbol{\alpha})$ according to [46, Lem. 6]. This is due to the fact that the span of $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha})$ is given by linear combinations of codewords in $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha})$ where each code symbol represents a monomial. In other words, to construct coded PPC schemes that retrieve polynomials of degree at most g, we require $g(k-1)+1 \leq n$ and $d_{\min}^{\tilde{\mathscr{C}}} \geq n - (g(k-1)+1) + 1$, where $d_{\min}^{\tilde{\mathscr{C}}}$ denotes the minimum distance of $\tilde{\mathscr{C}}$, to be able to decode the computation

correctly. It follows from Proposition 1 that $\tilde{\mathscr{C}} = \mathcal{RS}_{\tilde{k}}(\boldsymbol{\alpha})$ with dimension $\tilde{k} = \min\{g(k-1)+1, n\} = g(k-1)+1$ and $d_{\min}^{\tilde{\mathscr{C}}} = n - \tilde{k} + 1 = n - (g(k-1)+1) + 1$.

4.3.2 PPC Achievable Rate Matrix

We now extend the notion of a PIR achievable rate matrix for the coded PIR problem in Definition 2 to the coded PPC problem.

Definition 10 *Let $\mathscr{C}$ be an arbitrary $[n,k]$ code and denote by $\tilde{\mathscr{C}} = \mathscr{C}^{\star g}$ the $\tilde{k}$-dimensional code generated by the g-fold star-product of $\mathscr{C}$ with itself. A $\nu \times n$ binary matrix $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{PPC}}$ is called a* PPC achievable rate matrix *for $(\mathscr{C}, \tilde{\mathscr{C}})$, if*

1. *$\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{PPC}}$ is a κ-column regular matrix, i.e., its column sums are equal to κ, with $\kappa/\nu = \tilde{k}/n$, and*
2. *for each matrix row $\boldsymbol{\lambda}_i$, $\chi(\boldsymbol{\lambda}_i)$ is always an information set for $\tilde{\mathscr{C}}$, $i \in [\nu]$.*

In [15, Def. 11], two PIR interference matrices are defined from a PIR achievable rate matrix. Similar to the notion of PIR interference matrices, given a PPC achievable rate matrix $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{PPC}}$, the PPC *interference matrices* $\mathsf{A}_{\kappa\times n}$ and $\mathsf{B}_{(\nu-\kappa)\times n}$, are defined as follows.

Definition 11 *For a given $\nu \times n$ PPC achievable rate matrix $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{PPC}}(\mathscr{C}, \tilde{\mathscr{C}}) = (\lambda_{u,j})$, we define the interference matrices $\mathsf{A}_{\kappa\times n} = (a_{i,j})$ and $\mathsf{B}_{(\nu-\kappa)\times n} = (b_{i,j})$ for the code pair $(\mathscr{C}, \tilde{\mathscr{C}})$ as*

$$a_{i,j} \triangleq u \text{ if } \lambda_{u,j} = 1, \forall\, j \in [n], i \in [\kappa], u \in [\nu],$$
$$b_{i,j} \triangleq u \text{ if } \lambda_{u,j} = 0, \forall\, j \in [n], i \in [\nu-\kappa], u \in [\nu].$$

For $j \in [n]$, let $\mathcal{A}_j \triangleq \{a_{i,j} \colon i \in [\kappa]\}$ and $\mathcal{B}_j \triangleq \{b_{i,j} \colon i \in [\nu-\kappa]\}$. Then, the j-th column of $\mathsf{A}_{\kappa\times n}$ contains the row indices of $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{PPC}}$ whose entries in the j-th column are equal to 1, while $\mathsf{B}_{(\nu-\kappa)\times n}$ contains the remaining row indices of $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{PPC}}$. Hence, $\mathcal{B}_j = [\nu] \setminus \mathcal{A}_j$, $\forall\, j \in [n]$.

Note that in Definition 11, for each $j \in [n]$, distinct values of $u \in [\nu]$ should be assigned for all i. Thus, the assignment is not unique in the sense that the order of the entries of each column of A and B can be permuted.

Example 5 *Consider a DSS storing messages using a* $[4, 2]$ *RS code* $\mathscr{C}$ *over* $\mathbb{F}_5$ *with*

$$\mathsf{G}^{\mathscr{C}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

and candidate polynomials of degree at most $g = 2$*. We have* $\tilde{\mathscr{C}} = \mathscr{C}^{\star 2}$*,* $\tilde{k} = g(k-1)+1 = 3$*, and the generator matrix for* $\tilde{\mathscr{C}}$ *is given by*

$$\mathsf{G}^{\tilde{\mathscr{C}}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 4 & 4 \end{pmatrix}.$$

One can verify that

$$\mathsf{\Lambda}_{3,4}^{\mathrm{PPC}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

is a valid PPC achievable rate matrix for $(\mathscr{C}, \tilde{\mathscr{C}})$*, with* $(\kappa, \nu) = (3, 4)$*, generated using the four information sets of* $\tilde{\mathscr{C}}$ *and the corresponding interference matrices are given by*

$$\mathsf{A}_{3\times 4} = \begin{pmatrix} 1 & 1 & 1 & 2 \\ 2 & 2 & 3 & 3 \\ 3 & 4 & 4 & 4 \end{pmatrix} \text{ and } \mathsf{B}_{1\times 4} = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}.$$

$\bigtriangledown$

4.3.3 Generic Query Generation

In this subsection, we utilize the query generation algorithm `Q-Gen` that is introduced for PLC from MDS-PIR capacity-achieving coded DSSs in Section 3.3 as the basis for our PPC scheme. More specifically, the query generation algorithm `Q-Gen` generates the query of a PIR-like scheme from a linearly-coded DSS with dependent virtual messages representing the evaluations of the μ candidate functions. Accordingly, the

PPC scheme requires the length of each message to be $\mathsf{L} = \nu^{\mu} \cdot k$. Before running the main algorithm to generate the query sets, the following index preparation for the coded symbols stored in each database is performed.

***1) Index Preparation*:** Given that the query generation algorithm `Q-Gen` generates a fixed query set structure as a deterministic function of the desired polynomial index, we introduce an index permutation. The goal is to make the symbols queried from each database appear to be chosen randomly and independently from the desired polynomial index. Note that the polynomial is computed separately for the t-th row of all messages, $t \in [\beta]$. Therefore, similar to the coded PLC scheme in Section 3.3, we apply a permutation that is fixed across all coded symbols for the t-th row to maintain the dependency across the associated message elements. Let $\pi(\cdot)$ be a random permutation function over $[\beta]$, and let

$$U_{t,j}^{(v')} \triangleq \phi^{(v')}(\boldsymbol{C}_{\pi(t),j}),\ t \in [\beta],\ j \in [n],\ v' \in [\mu],$$

denote the t-th permuted symbol associated with the v'-th virtual message $\mathbf{X}^{(v')}$ stored in the j-th database, where $\boldsymbol{C}_{t,j} = \left(C_{t,j}^{(1)}, \ldots, C_{t,j}^{(f)}\right)^{\intercal}$. The permutation $\pi(\cdot)$ is randomly selected privately and uniformly by the user.

***2) Preliminaries*:** The query generation procedure is subdivided into μ rounds, where each round τ generates the queries based on the concept of τ-*sums* as defined as in Definition 6, of Section 3.3.1. Since we have $\binom{\mu}{\tau}$ different selections of τ distinct elements out of μ elements, a τ-sum can have $\binom{\mu}{\tau}$ different *types*. For a requested polynomial evaluation indexed by $v \in [\mu]$, a query set $Q_j^{(v)}$, $j \in [n]$, is composed of μ disjoint subsets of queries, each subset of queries is generated by the operations of each round $\tau \in [\mu]$. In a round we generate the queries for all possible $\binom{\mu}{\tau}$ types of τ-sums. For each round $\tau \in [\mu]$ the corresponding query subset is further subdivided into two subsets $Q_j^{(v)}(\mathcal{D}; \tau)$ and $Q_j^{(v)}(\mathcal{U}; \tau)$. The first subset $Q_j^{(v)}(\mathcal{D}; \tau)$ corresponds to τ-sums with a single symbol from the *desired* polynomial evaluation

and $\tau - 1$ symbols from the evaluations of *undesired* polynomials, while the second subset $Q_j^{(v)}(\mathcal{U};\tau)$ corresponds to τ-sums with symbols only from the evaluations of undesired polynomials. Here, $\mathcal{D}$ is an indicator for "desired function evaluation", while $\mathcal{U}$ an indicator for "undesired functions evaluations". Note that we require $\kappa^{\mu-(\tau-1)}(\nu-\kappa)^{\tau-1}$ distinct instances of each τ-sum type for every query set $Q_j^{(v)}$. We utilize these sets to generate the query sets of each round according to the interference matrices $\mathsf{A}_{\kappa\times n}$ and $\mathsf{B}_{(\nu-\kappa)\times n}$.

The queries $Q_j^{(v)}$ are generated by setting $(\kappa,\nu)=(\tilde{k},n)$ and invoking the query generation algorithm `Q-Gen` of Section 3.3.1 with the PPC problem parameters as follows:

$$\{Q_1^{(v)},\ldots,Q_n^{(v)}\} \leftarrow \texttt{Q-Gen}(v,\mu,\kappa,\nu,n,\mathsf{A}_{\kappa\times n},\mathsf{B}_{(\nu-\kappa)\times n}).$$

The total number of queries generated by the algorithm is given by

$$\sum_{j=1}^{n}\left|Q_j^{(v)}\right| = n\sum_{\tau=1}^{\mu}\binom{\mu}{\tau}\kappa^{\mu-\tau+1}(\nu-\kappa)^{\tau-1}. \tag{4.4}$$

4.3.4 Sign Assignment and Redundancy Elimination

Here, we generalize the coded PLC scheme of Chapter 3 in terms of exploiting the dependency between the virtual messages. Let $\mathsf{M}_g^{\mathrm{c}}(f)$ denote the size of the monomial basis of the polynomial candidate set. Then, since any polynomial in the candidate set is a linear function of its monomial basis of size $\mathsf{M}_g^{\mathrm{c}}(f)$, a PPC scheme can be seen as a PLC scheme performed over a set of $\mathsf{M}_g^{\mathrm{c}}(f)$ messages. Hence, the redundancy resulting from the linear dependencies between the virtual messages is also present for PPC and we can extend Lemma 4 in Section 3.3.3 and [43, Lem. 1] to this scheme. To exploit the dependency between the virtual messages we adopt a similar sign assignment process to each queried symbol of the virtual monomial messages as detailed in [43, Sec. IV-B]. Using Lagrange interpolation, we will show that it results in a uniquely solvable equation system from the different τ-sum types

given the side information available from all other databases. By obtaining such a system of equations in each round $\tau \in [\mu]$ of the protocol, the user can determine some of the answers offline.

Now, consider τ-sum types for $\tau = 1$, where we download individual segments of each virtual message including f independent messages. For this type, the user can determine any polynomial from the f obtained message segments. Based on this insight we can state the following lemma.

Lemma 5 *Let $\mu \in [f : \mathsf{M}_g(f)]$ be the number of candidate polynomials evaluations, including the f independent messages. For each query set, for all $v \in [\mu]$, each database $j \in [n]$, and based on the queried segments from the f independent messages, any $\binom{\mu-f}{1}$ 1-sum types out of all possible $\binom{\mu}{1}$ types are redundant. On the other hand, for $\tau \in [2 : \mu]$, any $\binom{\mu-\mathsf{M}_g^{\mathrm{c}}(f)}{\tau}$ τ-sum types out of $\binom{\mu}{\tau}$ types are redundant. Thus, the number of nonredundant τ-sum types with $\tau > 1$ is given by $\rho(\mu, \tau) \triangleq \binom{\mu}{\tau} - \binom{\mu-\mathsf{M}_g^{\mathrm{c}}(f)}{\tau}$.*

The proof of Lemma 5 is presented in Appendix E. In the following subsection, we show that the recovery and privacy conditions of our proposed PPC scheme are satisfied.

4.3.5 Recovery and Privacy

The scheme works as the PLC scheme in Chapter 3 by using the code $\tilde{\mathscr{C}}$ instead of the storage code $\mathscr{C}$. This is the case since for *any* polynomial evaluation code $\mathscr{D}$, $\mathscr{D}^{*i} \subseteq \mathscr{D}^{*j}$ for all $i \in [j]$, $j \in \mathbb{N}$, since the all-ones codeword is in $\mathscr{D}$ (see also [46, Lem. 6]). Moreover, since the definition of the PPC achievable rate matrix in Definition 10 is analogous to the corresponding definition of a PIR achievable rate matrix in Definition 2 (by using $\tilde{\mathscr{C}}$ instead of $\mathscr{C}$), it can directly be seen that the arguments in the proof of [15, Thm. 1] (see [15, App. B]) can be applied. Hence, it follows that $\tilde{k}$ distinct evaluations of $\psi_t(z) = \phi(\boldsymbol{\ell}_t(z))$ for each segment t can be recovered. Since $\deg(\psi_t(z)) \leq \tilde{k} - 1$, it follows that the polynomial $\psi_t(z)$ can be reconstructed via polynomial interpolation and then the desired polynomial

evaluations can be recovered by evaluating $\psi_t(z)$ at γ_j, $j \in [k]$. This is equal to evaluating the desired polynomial $\phi(\cdot)$ over the uncoded information symbols, i.e., $\phi(\boldsymbol{W}_{t,j})$ due to Lagrange encoding.

As for the privacy of the PPC scheme, using an argumentation similar to the PLC scheme privacy argument in Section 3.3.4, it can be seen that for any desired index $v \in [\mu]$, the redundant τ-sum types according to Lemma 5 can be fixed, i.e., the same τ-sum types are redundant for all $v \in [\mu]$, and hence the queries satisfy the privacy condition.

4.3.6 Achievable PPC Rate

Since $\tilde{\mathscr{C}}$ is an $[n, \tilde{k}]$ MDS code ($\mathscr{C}$ is an RS code), there always exists a PPC achievable rate matrix $\Lambda^{\text{PPC}}_{\kappa,\nu}$ with $\kappa/\nu = \tilde{k}/n$. Hence, using Lemma 5 we can prove the following theorem.

Theorem 5 *Consider a DSS that uses an $[n, k]$ RS code $\mathscr{C}$ to store f messages over n noncolluding databases using Lagrange encoding. Let $\mu \in [f : \mu_g(f)]$ be the number of candidate polynomials evaluations of degree at most g, including the f independent messages. Then, the PPC rate*

$$\mathsf{R}_{\text{PPC}} = \begin{cases} \frac{1}{f}\,\mathsf{H}_{\min} & \text{if } n \le g(k-1)+1, \\ \dfrac{\frac{k}{\tilde{k}}\left(1-\frac{\tilde{k}}{n}\right)\mathsf{H}_{\min}}{1-\left(\frac{\tilde{k}}{n}\right)^{\mathsf{M}^{\mathrm{c}}_g(f)}-(\mathsf{M}^{\mathrm{c}}_g(f)-f)\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1}} & \text{otherwise} \end{cases} \tag{4.5}$$

is achievable.

Proof: From equation (4.4) and Lemma 5, the achievable PPC rate after removing redundant τ-sums becomes

$$\begin{aligned} \mathsf{R} &\overset{(a)}{=} \frac{k\nu^{\mu}\,\mathsf{H}_{\min}}{n\left(\binom{\mu}{1}-\binom{\mu-f}{1}\right)\kappa^{\mu}+n\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau+1}(\nu-\kappa)^{\tau-1}} \\ &= \frac{k\nu^{\mu}\,\mathsf{H}_{\min}}{n\left[f\kappa^{\mu}+\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau+1}(\nu-\kappa)^{\tau-1}\right]}, \end{aligned} \tag{4.6}$$

where (a) follows from the PPC rate in Definition 1, equation (4.4), and Lemma 5. Now, if $\nu = \kappa$, or equivalently (from Definition 10) $n = \tilde{k} \overset{(b)}{=} \min\{g(k-1)+1, n\}$, i.e., $n = g(k-1)+1$ (since n cannot be strictly smaller than $g(k-1)+1$ by assumption and (b) is from Proposition 1), then it follows directly from equation (4.6) that $\mathsf{R} = {}^{k\,\mathsf{H}_{\min}}\!/_{nf}$. Moreover, it can be seen in this case that the proposed scheme reduces to the trivial scheme where the f independent files are downloaded and then the desired polynomial evaluation is performed offline. However, the proposed scheme requires an unnecessarily high redundancy to decode the f files, i.e., $\tilde{k} = n$ instead of $\tilde{k} = k$. As a result, for the case of $n \leq g(k-1)+1$, we opt out of any other achievable scheme and achieve the PPC rate ${}^{\mathsf{H}_{\min}}\!/_{f}$ by simply downloading all f files and performing the desired polynomial evaluation offline. Otherwise, i.e., $\nu > \kappa$, or equivalently (from Definition 10), $n > \tilde{k} = \min\{g(k-1)+1, n\}$, i.e., $n > g(k-1)+1$, then from equation (4.6) we have

$$
\begin{aligned}
\mathsf{R} &\overset{(c)}{=} \frac{\frac{k(\nu-\kappa)}{n\kappa}\,\mathsf{H}_{\min}}{\left[\frac{f(\nu-\kappa)}{\nu}\left(\frac{\kappa}{\nu}\right)^{\mu-1} + \frac{1}{\nu^{\mu}}\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau}(\nu-\kappa)^{\tau}\right]} \\
&\overset{(d)}{=} \frac{\frac{k(n-\tilde{k})}{n\tilde{k}}\,\mathsf{H}_{\min}}{\left[f\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1} + \frac{1}{n^{\mu}}\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\tilde{k}^{\mu-\tau}(n-\tilde{k})^{\tau}\right]} \\
&\overset{(e)}{=} \frac{k}{\tilde{k}}\left(1-\frac{\tilde{k}}{n}\right)\mathsf{H}_{\min}\Bigg[f\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1} \\
&\quad + \frac{1}{n^{\mu}}\left(\sum_{\tau=0}^{\mu}\binom{\mu}{\tau}\tilde{k}^{\mu-\tau}(n-\tilde{k})^{\tau} - \mu\tilde{k}^{\mu-1}(n-\tilde{k}) - \tilde{k}^{\mu}\right) \\
&\quad - \frac{1}{n^{\mu}}\sum_{\tau=2}^{\mu}\binom{\mu-\mathsf{M}_{g}^{\mathrm{c}}(f)}{\tau}\tilde{k}^{\mu-\tau}(n-\tilde{k})^{\tau}\Bigg]^{-1} \\
&\overset{(f)}{=} \frac{k}{\tilde{k}}\left(1-\frac{\tilde{k}}{n}\right)\mathsf{H}_{\min}\Bigg[f\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1} + \frac{1}{n^{\mu}}\left(n^{\mu} - \mu\tilde{k}^{\mu-1}(n-\tilde{k}) - \tilde{k}^{\mu}\right) \\
&\quad - \frac{1}{n^{\mu}}\left(\sum_{\tau=0}^{\eta}\binom{\eta}{\tau}\tilde{k}^{\mu-\tau}(n-\tilde{k})^{\tau} - \eta\tilde{k}^{\mu-1}(n-\tilde{k}) - \tilde{k}^{\mu}\right)\Bigg]^{-1} \\
&= \frac{k}{\tilde{k}}\left(1-\frac{\tilde{k}}{n}\right)\mathsf{H}_{\min}\Bigg[f\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1} - \mu\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1} \\
&\quad + 1 - \left(\frac{\tilde{k}}{n}\right)^{\mu} - \frac{1}{n^{\mu}}\left(\tilde{k}^{\mu-\eta}\sum_{\tau=0}^{\eta}\binom{\eta}{\tau}\tilde{k}^{\eta-\tau}(n-\tilde{k})^{\tau}\right) \\
&\quad + \eta\left(1-\frac{\tilde{k}}{n}\right)\left(\frac{\tilde{k}}{n}\right)^{\mu-1} + \left(\frac{\tilde{k}}{n}\right)^{\mu}\Bigg]^{-1}
\end{aligned}
$$

$$
\begin{aligned}
&= \tfrac{k}{\tilde{k}}\Big(1-\tfrac{\tilde{k}}{n}\Big)\,\mathsf{H}_{\min}\Bigg[1+(f-\mu+\eta)\Big(1-\tfrac{\tilde{k}}{n}\Big)\Big(\tfrac{\tilde{k}}{n}\Big)^{\mu-1}-\tfrac{1}{n^{\mu}}\Big(\tilde{k}^{\mu-\eta}n^{\eta}\Big)\Bigg]^{-1}\\
&= \tfrac{k}{\tilde{k}}\Big(1-\tfrac{\tilde{k}}{n}\Big)\,\mathsf{H}_{\min}\Bigg[1-(\mu-\eta-f)\Big(1-\tfrac{\tilde{k}}{n}\Big)\Big(\tfrac{\tilde{k}}{n}\Big)^{\mu-1}-\Big(\tfrac{\tilde{k}}{n}\Big)^{\mu-\eta}\Bigg]^{-1}\\
&= \frac{\tfrac{k}{\tilde{k}}\big(1-\tfrac{\tilde{k}}{n}\big)\,\mathsf{H}_{\min}}{1-\big(\tfrac{\tilde{k}}{n}\big)^{\mathsf{M}_g^{\mathrm{c}}(f)}-(\mathsf{M}_g^{\mathrm{c}}(f)-f)\big(1-\tfrac{\tilde{k}}{n}\big)\big(\tfrac{\tilde{k}}{n}\big)^{\mu-1}},
\end{aligned}
$$

where (c) follows since $\nu > \kappa$; (d) holds since we have $\kappa/\nu = \tilde{k}/n$ from Definition 10; (e) follows from expanding the summation over the terms of $\rho(\mu,\tau)$; and (f) follows by defining $\eta \triangleq \mu - \mathsf{M}_g^{\mathrm{c}}(f)$ and the fact that $\binom{m}{n} = 0$ if $m < n$. ■

Corollary 1 *Consider a DSS that uses an $[n,k]$ RS code $\mathscr{C}$ to store f messages over n noncolluding databases using Lagrange encoding. Let $\mu \in [f : \mu_g(f)]$ be the number of candidate polynomials evaluations of degree at most g, including the f independent messages. Then, the PPC rate*

$$
\mathsf{R}_{\mathrm{PPC},\infty} = \frac{k}{n}\left(\frac{\max\{n-g(k-1)-1,0\}}{g(k-1)+1}\right)\mathsf{H}_{\min} \tag{4.7}
$$

is achievable as $f \to \infty$.

Proof: If $n \leq g(k-1)+1$, then it follows from equation (4.5) that the PPC rate approaches zero as $f \to \infty$, which is in accordance with equation (4.7). Otherwise, if $n > g(k-1)+1$, the result follows directly from equation (4.5) by taking the limit $f \to \infty$ and using the fact that $\tilde{k} \overset{(a)}{=} \min\{g(k-1)+1, n\} = g(k-1)+1 < n$, where (a) follows from Proposition 1. ■

Note that the asymptotic PPC rate in equation (4.7) is equal to the rate of the general scheme from [46] when $\mathsf{H}_{\min} = 1$. This difference is due to the simplified rate definition used in [46]. Moreover, our proposed scheme cannot be obtained using the concept of refinement and lifting of so-called one-shot schemes as introduced for PIR in [80], since this concept cannot readily be applied to the function computation case.

Remark 4 *Note that in Lemma 5 and Theorem 5 we assume that the set of μ candidate functions includes its monomial basis which at least consists of the f*

independent files, i.e., $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}\} \subseteq \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}\}$ *and* $\mu \geq f$*. However, for the PPC problem where this is not the case, one can see that the PPC rate*

$$\mathsf{R}_{\text{PPC}} = \begin{cases} \frac{1}{f}\,\mathsf{H}_{\min} & \text{if } n \leq g(k-1)+1, \\ \frac{\frac{k}{\tilde{k}}\left(1-\frac{\tilde{k}}{n}\right)\mathsf{H}_{\min}}{1-\left(\frac{\tilde{k}}{n}\right)^{\mu}} & \text{otherwise} \end{cases}$$

is achievable with our general PPC scheme for RS-coded DSSs based on equation (4.4)*. Moreover, Corollary 1 holds when* $\mu \to \infty$*.*

4.4 PPC Scheme for Systematic RS-Encoded DSSs

In this section, we consider the case of RS-coded DSSs with systematic Lagrange encoding and first adapt the concept of the PPC achievable rate matrix from Definition 10.

4.4.1 PPC Systematic Achievable Rate Matrix

In contrast to the PPC scheme in Section 4.3, the basic idea is to utilize the systematic part of the RS code to recover the requested polynomial evaluation directly, i.e., we do not need to interpolate the systematic downloaded symbols to determine the requested polynomial evaluation. Thus, we can further enhance the download rate. However, due to the generic PC query design principles, namely, message symmetry and side information exploitation, we are restricted in how to exploit side information obtained from the systematic nodes. Specifically, for decodability (side information cancellation) to be possible, the side information obtained from the systematic nodes must be utilized in an isolated manner within an information set of the *polynomial decoding code* (see Section 4.3.1), such that we can reverse the order of the decoding procedure (i.e., unlike our RS-coded PPC scheme, we interpolate first and then cancel the side information). This restriction is further illustrated by a careful construction of a PPC systematic achievable rate matrix (Definition 12 below) and the corresponding interference matrices. Moreover, we modify the general

PPC scheme to utilize only the necessary number of nodes, denoted by $\hat{n}$, that guarantee the isolated use of systematic side information. Accordingly, we introduce an achievable rate matrix for the systematic PPC scheme as follows.

Definition 12 *Let $\mathscr{C}$ be an arbitrary $[n,k]$ code and denote by $\tilde{\mathscr{C}} = \mathscr{C}^{\star g}$ the $\tilde{k}$-dimensional code generated by the g-fold star-product of $\mathscr{C}$ with itself. Moreover, let*[6]

$$\hat{n} \triangleq \begin{cases} n & \text{if } \left\lfloor \frac{n}{\tilde{k}} \right\rfloor = 1 \text{ and } n - \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \tilde{k} < k, \\ k + (\left\lfloor \frac{n}{\tilde{k}} \right\rfloor - 1)\tilde{k} & \text{if } \left\lfloor \frac{n}{\tilde{k}} \right\rfloor > 1 \text{ and } n - \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \tilde{k} < k, \\ k + \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \tilde{k} & \text{if } \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \geq 1 \text{ and } n - \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \tilde{k} \geq k. \end{cases} \tag{4.8}$$

Then, a $\nu \times \hat{n}$ binary matrix $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{S,PPC}}$ is called a PPC systematic achievable rate matrix *for $(\mathscr{C}, \tilde{\mathscr{C}})$ if the following conditions are satisfied.*

1. *$\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{S,PPC}}$ is a κ-column regular matrix, and*
2. *there are exactly $\varrho \triangleq \left\lfloor \hat{n}/\tilde{k} \right\rfloor \kappa$ rows $\{\boldsymbol{\lambda}_i\}_{i\in[\varrho]}$ and $\nu - \varrho$ rows $\{\boldsymbol{\lambda}_{i+\varrho}\}_{i\in[\nu-\varrho]}$ of $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{S,PPC}}$ such that $\forall\, i \in [\varrho]$, $\chi(\boldsymbol{\lambda}_i)$ contains an information set for $\tilde{\mathscr{C}}$ and $\forall\, i \in [\nu - \varrho]$, $\chi(\boldsymbol{\lambda}_{i+\varrho}) = [k]$.*

The following lemma shows how to construct a PPC systematic achievable rate matrix with $(\kappa, \nu) = \big(k, \hat{n} - \left\lfloor \hat{n}/\tilde{k} \right\rfloor (\tilde{k} - k)\big)$.

Lemma 6 *Let $\mathscr{C}$ be an arbitrary $[n,k]$ code and $\tilde{\mathscr{C}} = \mathscr{C}^{\star g}$. Then, there exists a PPC systematic achievable rate matrix $\mathsf{\Lambda}_{\kappa,\nu}^{\mathrm{S,PPC}}$ for $(\mathscr{C}, \tilde{\mathscr{C}})$ with $(\kappa, \nu) = \big(k, \hat{n} - \left\lfloor \hat{n}/\tilde{k} \right\rfloor (\tilde{k} - k)\big)$, where $\tilde{k}$ is the dimension of $\tilde{\mathscr{C}}$.*

Proof: Let $\hat{\delta} \triangleq \left\lfloor \hat{n}/\tilde{k} \right\rfloor$ and $\Gamma \triangleq \hat{n} - \hat{\delta}\tilde{k}$. From our choices of $\hat{n}$ in equation (4.8), one can verify that $\Gamma \leq k$ and Γ is well-defined. Accordingly, construct a matrix $\mathsf{A}_{k\times\hat{n}}$ as in Definition 11 with

$$a_{i,j} = \hat{\delta}k + i, \text{ if } j \in [k],\, i \in [\Gamma]. \tag{4.9}$$

[6] *Note that the first requirement of the final case of equation* (4.8) *is unnecessary as $\left\lfloor n/\tilde{k} \right\rfloor \geq 1$ always. However, it is included for symmetry reasons.*

In this way, $k\Gamma$ entries of $\mathsf{A}_{k\times\hat{n}}$ are filled. Next, let $\{a_{i_1^{(j)},j},\ldots,a_{i_{u(j)}^{(j)},j}\}$, $j\in[\hat{n}]$, denote the remaining empty entries in column j of $\mathsf{A}_{k\times\hat{n}}$, where $u(j)\leq k$ is the number of empty entries in column j. Hence, the $k\hat{n}-k\Gamma=k(\hat{n}-\Gamma)$ entries

$$\left\{a_{i_1^{(1)},1},\ldots,a_{i_{u(1)}^{(1)},1},\ldots,a_{i_1^{(\hat{n})},\hat{n}},\ldots,a_{i_{u(\hat{n})}^{(\hat{n})},\hat{n}}\right\} \tag{4.10}$$

are empty. Now, observe that $(\hat{n}-\Gamma)\hat{\delta}^{-1}=\big(\hat{n}-(\hat{n}-\hat{\delta}\tilde{k})\big)\hat{\delta}^{-1}=\tilde{k}\in\mathbb{N}$. By consecutively assigning $\{1,\ldots,\hat{\delta}k\}$ to the entries of $\mathsf{A}_{k\times\hat{n}}$ in equation (4.10) and repeating this process $\tilde{k}$ times, the remaining $\hat{\delta}k\cdot{}^{(\hat{n}-\Gamma)}\!/_{\hat{\delta}}=k(\hat{n}-\Gamma)$ empty entries of $\mathsf{A}_{k\times\hat{n}}$ are filled. Note that since values of $[\hat{\delta}k]$ are consecutively assigned, the largest number of empty entries of each column of $\mathsf{A}_{k\times\hat{n}}$ is k, and $\hat{\delta}=\big\lfloor{}^{\hat{n}}\!/_{\tilde{k}}\big\rfloor\geq 1$, there are no repeated values of $[\hat{\delta}k]$ in any column of $\mathsf{A}_{k\times\hat{n}}$, which implies that condition 1) in Definition 12 is satisfied. From equations (4.9) and (4.10), it can be seen that each $a\in[\hat{\delta}k]=[\varrho]$ occurs in $\tilde{k}$ columns of $\mathsf{A}_{k\times\hat{n}}$ and each $a\in[\hat{\delta}k+1:\hat{\delta}k+\Gamma]$ occurs in k columns of $\mathsf{A}_{k\times\hat{n}}$. This implies that condition 2) in Definition 12 is satisfied with $\kappa=k$, $\varrho=\hat{\delta}k$, and $\nu=\Gamma+\hat{\delta}k$, which completes the proof. ■

Lemma 7 *For the PPC systematic achievable rate matrix from Lemma 6, it holds that*

$$\nu=\begin{cases}n-\tilde{k}+k & \text{if } \big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor=1 \text{ and } n-\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor\tilde{k}<k,\\ \big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor k & \text{if } \big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor>1 \text{ and } n-\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor\tilde{k}<k,\\ \big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor k+k & \text{if } \big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor\geq 1 \text{ and } n-\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor\tilde{k}\geq k.\end{cases} \tag{4.11}$$

Proof: To prove the results, we use Definition 12 and the fact that $\nu=\hat{n}-\big\lfloor{}^{\hat{n}}\!/_{\tilde{k}}\big\rfloor(\tilde{k}-k)$. Now, if $\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor=1$ and $n-\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor\tilde{k}<k$ (the first case from Definition 12), then it follows directly that $\nu=\hat{n}-\big\lfloor{}^{\hat{n}}\!/_{\tilde{k}}\big\rfloor(\tilde{k}-k)=n-\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor(\tilde{k}-k)=n-\tilde{k}+k$. On the other hand, if $\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor>1$ and $n-\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor\tilde{k}<k$ (the second case from Definition 12), then after inserting $\hat{n}=k+\big(\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor-1\big)\tilde{k}$ into the expression for ν, $\nu=k\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor-\big\lfloor{}^{k}\!/_{\tilde{k}}\big\rfloor(\tilde{k}-k)=k\big\lfloor{}^{n}\!/_{\tilde{k}}\big\rfloor$, since $\big\lfloor{}^{k}\!/_{\tilde{k}}\big\rfloor(\tilde{k}-k)=0$. In a similar manner, the remaining case in equation (4.11) can be shown. ■

In the following lemma, we show a lower bound to the fraction κ/ν.

Lemma 8 *If a matrix* $\Lambda_{\kappa,\nu}^{\mathrm{S,PPC}}(\mathscr{C},\tilde{\mathscr{C}})$ *exists for an* $[n,k]$ *code* $\mathscr{C}$ *and the* $[n,\tilde{k}]$ *code* $\tilde{\mathscr{C}}$*, then we have* $\kappa/\nu \geq k/\left(\hat{n}-\lfloor\hat{n}/\tilde{k}\rfloor(\tilde{k}-k)\right)$.

Proof: Since by definition each row $\boldsymbol{\lambda}_i$ of $\Lambda_{\kappa,\nu}^{\mathrm{S,PPC}}$ contains an information set for $\tilde{\mathscr{C}}$, $i \in [\varrho]$, $\varrho = \left\lfloor \hat{n}/\tilde{k} \right\rfloor \kappa$, and each row $\boldsymbol{\lambda}_{i+\varrho} = [k]$, $i \in [\nu-\varrho]$, we have $w_{\mathsf{H}}(\boldsymbol{\lambda}_i) \geq \tilde{k}$, $i \in [\varrho]$, and $w_{\mathsf{H}}(\boldsymbol{\lambda}_{i+\varrho}) = k$, $i \in [\nu-\varrho]$. Let $\boldsymbol{v}_j$, $j \in [\hat{n}]$, be the j-th column of $\Lambda_{\kappa,\nu}^{\mathrm{S,PPC}}$. If we look at $\Lambda_{\kappa,\nu}^{\mathrm{S,PPC}}$ from both a row-wise and a column-wise point of view, we obtain

$$\begin{aligned}\varrho\tilde{k} + (\nu-\varrho)k &\leq \sum_{i=1}^{\varrho} w_{\mathsf{H}}(\boldsymbol{\lambda}_i) + \sum_{i=1}^{\nu-\varrho} w_{\mathsf{H}}(\boldsymbol{\lambda}_{i+\varrho}) \\ &= \sum_{j=1}^{\hat{n}} w_{\mathsf{H}}(\boldsymbol{v}_j) = \kappa\hat{n}.\end{aligned}$$

Thus, we have

$$\varrho\tilde{k} - \varrho k + \nu k = \varrho(\tilde{k}-k) + \nu k \leq \kappa\hat{n},$$

from which the result follows. ■

The systematic PPC scheme requires the length of each message to be $\mathsf{L} = \nu^{\mu}\cdot k$. The queries $Q_j^{(v)}$ are generated by setting $(\kappa,\nu) = (k, \hat{n} - \left\lfloor \hat{n}/\tilde{k} \right\rfloor(\tilde{k}-k))$ and invoking the query generation algorithm `Q-Gen` of Section 3.3.1 with the systematic PPC problem parameters as follows:

$$\{Q_1^{(v)},\ldots,Q_{\hat{n}}^{(v)}\} \leftarrow \texttt{Q-Gen}(v,\mu,\kappa,\nu,\hat{n},\mathsf{A}_{\kappa\times\hat{n}},\mathsf{B}_{(\nu-\kappa)\times\hat{n}}).$$

Note that we utilize $\hat{n} \leq n$ databases, including the systematic nodes, in constructing the scheme, while the remaining $n-\hat{n}$ databases are not queried.

4.4.2 Sign Assignment and Redundancy Elimination

Since this scheme is a modified version of the general PPC scheme where we utilize the systematic part of the RS code to recover the requested polynomial evaluation

directly, the scheme inherently extends the same redundancy and sign assignment arguments stated in Section 4.3.4. The only difference between the general PPC scheme and the systematic PPC scheme lies within the recovery argument.

4.4.3 Recovery and Privacy

The scheme works as the PPC scheme in Section 4.3, however by mixing between the code $\tilde{\mathscr{C}}$ and the storage code $\mathscr{C}$. Due to this mixture, we require a more complicated decoding process. The key idea of the recovery process of the scheme is illustrated with Example 6 in Section 4.4.5.

4.4.4 Achievable PPC Rate

Using Lemmas 5 and 6, the following theorem follows.

Theorem 6 *Consider a DSS that uses an $[n, k]$ RS code $\mathscr{C}$ to store f messages over n noncolluding databases using systematic Lagrange encoding. Let $\mu \in [f : \mu_g(f)]$ be the number of candidate polynomials evaluations of degree at most g, including the f independent messages. Then, the PPC rate*

$$\mathsf{R}^{\mathrm{S}}_{\mathrm{PPC}} = \begin{cases} \frac{1}{f}\,\mathsf{H}_{\min} & \text{if } n \leq g(k-1)+1, \\ \frac{\frac{k}{\hat{n}}\left(\frac{\nu-\kappa}{\kappa}\right)\mathsf{H}_{\min}}{1-\left(\frac{\kappa}{\nu}\right)^{\mathsf{M}^{\mathrm{c}}_g(f)}-(\mathsf{M}^{\mathrm{c}}_g(f)-f)\left(1-\frac{\kappa}{\nu}\right)\left(\frac{\kappa}{\nu}\right)^{\mu-1}} & \text{otherwise}, \end{cases} \tag{4.12}$$

with $(\kappa, \nu) = (k, \hat{n} - \lfloor \hat{n}/\tilde{k} \rfloor(\tilde{k} - k))$ and $\hat{n}$ as defined in equation (4.8), is achievable.

Proof: From equation (4.4) and by removing redundant τ-sums from the query sets according to Lemma 5, the achievable PPC rate becomes

$$\begin{aligned} \mathsf{R} &\overset{(a)}{=} \frac{k\nu^{\mu}\,\mathsf{H}_{\min}}{\hat{n}\left(\binom{\mu}{1}-\binom{\mu-f}{1}\right)\kappa^{\mu}+\hat{n}\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau+1}(\nu-\kappa)^{\tau-1}} \\ &= \frac{k\nu^{\mu}\,\mathsf{H}_{\min}}{\hat{n}\kappa\left[f\kappa^{\mu-1}+\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau}(\nu-\kappa)^{\tau-1}\right]}, \end{aligned} \tag{4.13}$$

where (a) follows from the PPC rate in Definition 1, equation (4.4), and Lemma 5.

Now, we first consider the case where $\nu = \kappa$ and show that it is equivalent to $n \leq g(k-1)+1$. Assume that $\nu = \kappa = k$. Then, for the first case of equation (4.11) it follows that $\tilde{k} = n$. For the second and third cases of equation (4.11), to obtain $\nu = k$, we must have $\lfloor n/\tilde{k} \rfloor = 1$ or $\lfloor n/\tilde{k} \rfloor = 0$, respectively, which violates the condition of the second case and is never true for the third case. Since, by Proposition 1, $\tilde{k} = \min\{g(k-1)+1, n\} = n$, it follows that $n \leq g(k-1)+1$. Conversely, if $n \leq g(k-1)+1$, then $\tilde{k} = \min\{g(k-1)+1, n\} = n$, and it follows from equation (4.11) (the first case) that $\nu = \kappa$. Hence, in summary, we have shown that $\nu = \kappa$ is equivalent to $n \leq g(k-1)+1$. As a result, for $n \leq g(k-1)+1$, it follows directly from equation (4.13) that $\mathsf{R} = {}^{k\,\mathsf{H}_{\min}}/_{\hat{n}f}$. Moreover, it can be seen in this case that the proposed systematic PPC scheme reduces to the trivial scheme for which all the f independent files are downloaded and the desired polynomial evaluation is performed offline. However, similar to the general PPC scheme, the proposed systematic PPC scheme requires an unnecessarily high redundancy to decode the f files, i.e., $\tilde{k} = \hat{n}$ instead of $\tilde{k} = k$. As a result, for the case of $n \leq g(k-1)+1$, we again opt out of any other achievable scheme and achieve the PPC rate ${}^{\mathsf{H}_{\min}}/_{f}$ by simply downloading all f files and performing the desired polynomial evaluation offline.

On the other hand, if $\nu > \kappa$, or equivalently, $n > g(k-1)+1$, then from equation (4.13) we have

$$\begin{aligned}
\mathsf{R} &\overset{(b)}{=} \frac{\frac{k}{\hat{n}\kappa}\,\mathsf{H}_{\min}}{\frac{f\kappa^{\mu-1}}{\nu^{\mu}} + \frac{1}{\nu^{\mu}(\nu-\kappa)}\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau}(\nu-\kappa)^{\tau}} \\
&= \frac{\frac{k(\nu-\kappa)}{\hat{n}\kappa}\,\mathsf{H}_{\min}}{\frac{f(\nu-\kappa)}{\nu}\left(\frac{\kappa}{\nu}\right)^{\mu-1} + \frac{1}{\nu^{\mu}}\sum_{\tau=2}^{\mu}\rho(\mu,\tau)\kappa^{\mu-\tau}(\nu-\kappa)^{\tau}} \\
&\quad\ \vdots \\
&\overset{(c)}{=} \frac{\frac{k}{\hat{n}}\left(\frac{\nu-\kappa}{\kappa}\right)\mathsf{H}_{\min}}{1-\left(\frac{\kappa}{\nu}\right)^{\mathsf{M}_g^{\mathrm{c}}(f)} - \left(\mathsf{M}_g^{\mathrm{c}}(f)-f\right)\left(1-\frac{\kappa}{\nu}\right)\left(\frac{\kappa}{\nu}\right)^{\mu-1}},
\end{aligned}$$

where (b) follows since $\nu > \kappa$ and (c) results from following similar steps as in the proof of the achievable PPC rate of Theorem 5 in Section 4.3.6. ■

Corollary 2 *Consider a DSS that uses an $[n, k]$ RS code $\mathscr{C}$ to store f messages over n noncolluding databases using systematic Lagrange encoding. Let $\mu \in [f : \mu_g(f)]$ be the number of candidate polynomials evaluations of degree at most g, including the f independent messages. Then, the PPC rate*

$$\mathsf{R}^{\mathrm{S}}_{\mathrm{PPC},\infty} = \begin{cases} \frac{1}{n}\big(\max\{n - g(k-1) - 1, 0\}\big)\,\mathsf{H}_{\min} & \text{if } \left\lfloor \frac{n}{\tilde{k}} \right\rfloor = 1 \text{ and} \\ & n - \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \tilde{k} < k, \\ \frac{1}{\hat{n}}\Big(\left\lfloor \frac{n}{g(k-1)+1} \right\rfloor k - k\Big)\,\mathsf{H}_{\min} & \text{if } \left\lfloor \frac{n}{\tilde{k}} \right\rfloor > 1 \text{ and} \\ & n - \left\lfloor \frac{n}{g(k-1)+1} \right\rfloor (g(k-1)+1) < k, \\ \frac{1}{\hat{n}}\Big(\left\lfloor \frac{n}{g(k-1)+1} \right\rfloor k\Big)\,\mathsf{H}_{\min} & \text{if } \left\lfloor \frac{n}{\tilde{k}} \right\rfloor \geq 1 \text{ and} \\ & n - \left\lfloor \frac{n}{g(k-1)+1} \right\rfloor (g(k-1)+1) \geq k, \end{cases} \tag{4.14}$$

with $\hat{n}$ as defined in equation (4.8), is asymptotically achievable for $f \to \infty$.

Proof: If $n \leq g(k-1)+1$, then it follows from equation (4.12) that the PPC rate approaches zero as $f \to \infty$, which is in accordance with equation (4.14) (first case, since $\left\lfloor n/\tilde{k} \right\rfloor = 1$ and $n - \left\lfloor n/\tilde{k} \right\rfloor \tilde{k} = 0 < k$). Otherwise, if $n > g(k-1)+1$, the result follows directly from equation (4.12) by taking the limit $f \to \infty$ and using equation (4.11) and the fact (see Proposition 1) that $\tilde{k} = \min\{g(k-1)+1, n\} = g(k-1)+1$. ■

Note that when $n - \tilde{k} \leq k$, the asymptotic PPC rate in equation (4.14) is equal to the rate of the systematic scheme from [45, Thm. 3], [46] when $\mathsf{H}_{\min} = 1$. This difference is due to the simplified rate definition used in [45], [46]. However, for the case when $n - \tilde{k} > k$, with the simplified rate definition, i.e., for $\mathsf{H}_{\min} = 1$, the asymptotic PPC rate in equation (4.14) is larger compared to the PPC rate of the systematic scheme from [45, Thm. 3], [46].

Remark 5 *Similar to Remark 4, in Theorem 6 we assume that the set of μ candidate functions includes its monomial basis which at least consists of the f independent files, i.e., $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}\} \subseteq \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}\}$ and $\mu \geq f$. However, for the PPC problem*

where this is not the case,one can see that the PPC rate

$$\mathsf{R}^{\mathrm{S}}_{\mathrm{PPC}} = \begin{cases} \frac{1}{f}\,\mathsf{H}_{\min} & \text{if } n \leq g(k-1)+1, \\ \frac{k}{\hat{n}}\left(\frac{\nu-\kappa}{\kappa}\right)\mathsf{H}_{\min}\left[1-\left(\frac{\kappa}{\nu}\right)^{\mu}\right]^{-1} & \text{otherwise,} \end{cases}$$

with $(\kappa,\nu) = (k, \hat{n} - \lfloor \hat{n}/\tilde{k} \rfloor(\tilde{k}-k))$ *and* $\hat{n}$ *as defined in (4.8), is achievable with our PPC scheme for RS-coded DSSs with systematic Lagrange encoding based on equation* (4.4)*. Moreover, Corollary 2 holds when* $\mu \to \infty$*.*

We illustrate the key concept of our proposed scheme in Theorem 6 with an example for the special case of PMC.

4.4.5 Special Case: PMC Scheme

As the rate of PMC is a decreasing function of the number of candidate monomials, we can increase the PMC rate by limiting ourselves to the set of monomials excluding *parallel* monomials as defined by equation (2.2) in Chapter 2 . Recall that, given a bivariate monomial over the variables x and y of degree at most $g = 2$, the set of possible monomials is $\{x, y, xy, x^2, y^2\}$. Moreover, x^2 is said to be a parallel monomial as it can be obtained by raising the monomial x to the power of 2. Thus, x^2 and y^2 are parallel monomials and can be excluded from the set of candidate monomials.

Example 6 *Consider two messages* $\mathbf{W}^{(1)}$ *and* $\mathbf{W}^{(2)}$ *that are stored in a noncolluding DSS using a systematic* $[4, 2]$ *RS code* $\mathscr{C}$*. Suppose that the user wishes to obtain a monomial function evaluation* $\mathbf{X}^{(v)}$ *from the set of nonparallel monomial functions of degree at most* $g = 2$*. We have* $\mu = \mathsf{M}^{\mathrm{c}}_2(2) = \widetilde{\mathsf{M}}_2(2) = 3$*,* $v \in [3]$*, and the candidate set of monomial functions evaluations is* $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(1)} \star \mathbf{W}^{(2)}\}$*, where* $\star$ *denotes element-wise multiplication. Let the desired monomial function index be* $v = 1$*, i.e., the user wishes to obtain the function evaluation* $\mathbf{X}^{(1)} = \mathbf{W}^{(1)}$*. We have* $\tilde{k} = g(k-1)+1 = 3$ *and* $\hat{n} = n = 4$*. It follows that* $\nu = \hat{n} - \left\lfloor \hat{n}/\tilde{k} \right\rfloor(\tilde{k}-k) = 3$*,*

Table 4.1 PMC Query Sets for $v = 1$

j	1	2	3	4
$Q_j^{(1)}(\mathcal{D}; 1)$	$x_{1:4,1},\ x_{9:12,1}$	$x_{5:8,2},\ x_{9:12,2}$	$x_{1:4,3},\ x_{5:8,3}$	$x_{1:4,4},\ x_{5:8,4}$
$Q_j^{(1)}(\mathcal{U}; 1)$	$y_{1:4,1},\ y_{9:12,1}$	$y_{5:8,2},\ y_{9:12,2}$	$y_{1:4,3},\ y_{5:8,3}$	$y_{1:4,4},\ y_{5:8,4}$
$Q_j^{(1)}(\mathcal{D}; 2)$	$x_{13:14,1} - y_{5:6,1}$ $x_{15:16,1} - z_{5:6,1}$ $x_{21:22,1} - y_{7:8,1}$ $x_{23:24,1} - z_{7:8,1}$	$x_{17:18,2} - y_{1:2,2}$ $x_{19:20,2} - z_{1:2,2}$ $x_{21:22,2} - y_{3:4,2}$ $x_{23:24,2} - z_{3:4,2}$	$x_{13:14,3} - y_{9:10,3}$ $x_{15:16,3} - z_{9:10,3}$ $x_{17:18,3} - y_{11:12,3}$ $x_{19:20,3} - z_{11:12,3}$	$x_{13:14,4} - y_{9:10,4}$ $x_{15:16,4} - z_{9:10,4}$ $x_{17:18,4} - y_{11:12,4}$ $x_{19:20,4} - z_{11:12,4}$
$Q_j^{(1)}(\mathcal{U}; 2)$	$y_{15:16,1} - z_{13:14,1}$ $y_{23:24,1} - z_{21:22,1}$	$y_{19:20,2} - z_{17:18,2}$ $y_{23:24,2} - z_{21:22,2}$	$y_{15:16,3} - z_{13:14,3}$ $y_{19:20,3} - z_{17:18,3}$	$y_{15:16,4} - z_{13:14,4}$ $y_{19:20,4} - z_{17:18,4}$
$Q_j^{(1)}(\mathcal{D}; 3)$	$x_{25,1} - y_{19,1} + z_{17,1}$ $x_{27,1} - y_{20,1} + z_{18,1}$	$x_{26,2} - y_{15,2} + z_{13,2}$ $x_{27,2} - y_{16,2} + z_{14,2}$	$x_{25,3} - y_{23,3} + z_{21,3}$ $x_{26,3} - y_{24,3} + z_{22,3}$	$x_{25,4} - y_{23,4} + z_{21,4}$ $x_{26,4} - y_{24,4} + z_{22,4}$

Note: query sets after sign assignment and removal of redundant queries for a $[4, 2]$ RS-coded DSS with systematic Lagrange encoding storing $f = 2$ messages, where the $\mu = 3$ candidate monomial functions evaluations are $\{\mathbf{X}^{(1)} = \mathbf{W}^{(1)}, \mathbf{X}^{(2)} = \mathbf{W}^{(2)}, \mathbf{X}^{(3)} = \mathbf{W}^{(1)} \star \mathbf{W}^{(2)}\}$. Blue and red subscripts indicate side information exploitation in rounds $\tau = 2$ and $\tau = 3$, respectively.

$\kappa = k = 2$, $\varrho = \left\lfloor \hat{n}/\tilde{k} \right\rfloor \kappa = 2$, *and*

$$\Lambda_{2,3}^{\mathrm{S,PPC}} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

is a valid PPC systematic achievable rate matrix (see Lemma 6). We further obtain the PC interference matrices

$$\mathsf{A}_{2\times 4} = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 3 & 3 & 2 & 2 \end{pmatrix} \text{ and } \mathsf{B}_{1\times 4} = \begin{pmatrix} 2 & 1 & 3 & 3 \end{pmatrix}$$

from $\Lambda_{2,3}^{\mathrm{S,PPC}}$ *using Definition 11.*

We simplify the notation by letting $x_{t,j} = C_{t,j}^{(1)}$, $y_{t,j} = C_{t,j}^{(2)}$, *and* $z_{t,j} = C_{t,j}^{(1)} \cdot C_{t,j}^{(2)}$ *for all* $t \in [\beta]$, $j \in [4]$, *where* $\beta = \nu^{\mu} = 27$. *Since the desired function evaluation is*

$\mathbf{X}^{(1)}$, *the goal is to privately obtain* $x_{t,j}$, $\forall\, t \in [27]$, *and successfully decode* $\mathbf{X}^{(1)}$. *The construction of the query sets is briefly presented in the following steps.*[7]

Initialization (Round $\tau = 1$***):*** *We start with* $\tau = 1$ *to generate query sets for each database* j *holding* $\kappa^{\mu} = 8$ *instances of* $x_{t,j}$. *By message symmetry this also applies to* $y_{t,j}$ *and* $z_{t,j}$.

Following Rounds ($\tau \in [2:3]$***):*** *Using the PC interference matrices* $\mathsf{A}_{2\times 4}$ *and* $\mathsf{B}_{1\times 4}$ *for the exploitation of side information for the* j*-th database,* $j \in [n]$, *we generate the desired query sets* $Q_j^{(1)}(\mathcal{D};\tau)$ *by querying a number of new symbols of the desired monomial jointly combined with symbols from other monomials queried in the previous round from database* $i \neq j$. *Next, the undesired query sets* $Q_j^{(1)}(\mathcal{U};\tau)$ *(if* $\tau = 2$*) are generated by enforcing message symmetry.*

In the end, we apply the sign assignment procedure to the query sets for $v = 1$ *and make the final modification to the queries by removing all the* 1*-sums corresponding to the redundant* 1*-sum types from the first round (see Lemma 5). This translates to removing the queries for* $z_{t,j}$, *since they can be generated offline by the user given* $x_{t,j}$ *and* $y_{t,j}$. *The resulting query sets are shown in Table 4.1, where* $u_{a:b,j} \triangleq \{u_{a,j}, \ldots, u_{b,j}\}$ *for* $u = x, y, z$, *and the side information is highlighted with blue and red for rounds* $\tau = 2$ *and* $\tau = 3$, *respectively. The PMC rate* ${}^{k\nu^{\mu}\,\mathsf{H}_{\min}}\!/_{\mathsf{D}} = {}^{(2\times 3^3\times \mathsf{H}_{\min})}\!/_{(2\times 4\times 15)} = 0.45\cdot \mathsf{H}_{\min}$ *is achievable, where the value of* $\mathsf{H}_{\min} = \mathsf{H}(\mathbf{X}^{(3)})$ *depends on the underlying field.*

Now we show that the $\mathsf{L} = k\beta = 54$ *symbols of the desired function evaluation can be reliably decoded. Note that here we assume that the nodes* $j \in \{1,2\}$ *are systematic.*

Initialization Round ($\tau = 1$**):** *The following steps are taken.*

1. *Obtain the desired symbols: From the answers retrieved for the query sets* $Q_j^{(1)}(\mathcal{D},1)$, *utilize the information sets* $\tilde{\mathcal{I}}_1 = \{1,3,4\}$ *and* $\tilde{\mathcal{I}}_2 = \{2,3,4\}$ *of* $\tilde{\mathscr{C}}$ *to decode the symbols of the desired function evaluation* $\mathbf{X}^{(1)}$ *for* $j \in \{1,2\}$. *In other words, from* $x_{1:4,1}$, $x_{1:4,3}$, *and* $x_{1:4,4}$ *we use Lagrange interpolation to obtain* $x_{1:4,2}$. *Similarly, from* $x_{5:8,2}$, $x_{5:8,3}$, *and* $x_{5:8,4}$ *we obtain* $x_{5:8,1}$. *Finally, from the information set* $\mathcal{I} = \{1,2\}$ *of* $\mathscr{C}$ *we readily have* $x_{9:12,1}$ *and* $x_{9:12,2}$. *By the end of this round, we obtain* $k\nu(\kappa^{\mu-1}) = 24$ *symbols from the desired function evaluation* $\mathbf{X}^{(1)}$.

[7] *With some abuse of notation for the sake of simplicity, the generated queries are sets containing their answers.*

2. *Prepare the side information: We prepare the side information symbols retrieved in this round to be used in the next round by the following steps. First, for the answers of the query sets $Q_j^{(1)}(\mathcal{U}, 1)$, repeat the previous step to decode the undesired symbols $y_{5:8,1}$ and $y_{1:4,2}$. Next, since in this round, due to redundancy elimination, we retrieve symbols of polynomials of degree one, i.e., symbols from the $f = 2$ independent files, we can use Lagrange interpolation with $k = 2$ symbols from the systematic nodes to obtain coded symbols for $j \notin \{1, 2\}$. Accordingly, from $x_{9:12,1}$ and $x_{9:12,2}$ we obtain $x_{9:12,3}$ and $x_{9:12,4}$, and similarly for $y_{9:12,3}$ and $y_{9:12,4}$. Finally, using the dependency between x, y, and z and the available symbols, compute $z_{5:8,1}$, $z_{1:4,2}$, $z_{9:12,3}$, and $z_{9:12,4}$. The obtained symbols are shown in Table 4.2(a).*

Table 4.2 Decoded and Computed Symbols from the PMC Query Sets for $v = 1$ from Table 4.1

j	1	2	3	4
$\tilde{Q}_j^{(1)}(\mathcal{D}; 1)$	$x_{5:8,1}$	$x_{1:4,2}$	$x_{9:12,3}$	$x_{9:12,4}$
$\tilde{Q}_j^{(1)}(\mathcal{U}; 1)$	$y_{5:8,1}$, $z_{5:8,1}$	$y_{1:4,2}$, $z_{1:4,2}$	$y_{9:12,3}$, $z_{9:12,3}$	$y_{9:12,4}$, $z_{9:12,4}$

(a)

j	1	2
$\tilde{Q}_j^{(1)}(\mathcal{D}; 2)$	$x_{17:18,1}$, $x_{19:20,1}$	$x_{13:14,2}$, $x_{15:16,2}$
$\tilde{Q}_j^{(1)}(\mathcal{U}; 2)$	$y_{19:20,1} - z_{17:18,1}$	$y_{15:16,2} - z_{13:14,2}$

(b)

j	1	2
$\tilde{Q}_j^{(1)}(\mathcal{D}; 3)$	$x_{25,1}$, $x_{27,1}$	$x_{26,2}$, $x_{27,2}$
$\tilde{Q}_j^{(1)}(\mathcal{U}; 3)$	$x_{25,1} + y_{23,1} - z_{21,1}$	$x_{26,2} + y_{24,2} - z_{22,2}$

(c)

Second Round ($\tau = 2$): *The decoding procedure is as follows.*

1. *Interference cancellation: Utilize the decoded symbols from the set $\tilde{Q}_j^{(1)}(\mathcal{U}, 1)$ of Table 4.2(a) to cancel the side information, marked in blue in Table 4.1, from the answers of the query sets $Q_j^{(1)}(\mathcal{D}, 2)$.*

2. *Obtain the desired symbols: Similar to the first round, utilize the information sets $\tilde{\mathcal{I}}_1 = \{1, 3, 4\}$ and $\tilde{\mathcal{I}}_2 = \{2, 3, 4\}$ of $\tilde{\mathscr{C}}$ to decode the symbols of the desired function evaluation $\mathbf{X}^{(1)}$ for $j \in \{1, 2\}$ shown in $\tilde{Q}_j^{(1)}(\mathcal{D}, 2)$ of Table 4.2(b). Together with the symbols directly obtained from $j \in \{1, 2\}$, by the end of this round, we would*

have obtained an additional $k\nu\binom{\mu-1}{\tau-1}\kappa^{\mu-\tau}(\nu-\kappa)^{\tau-1} = 24$ *symbols from the desired function evaluation.*

3. *Prepare the side information: We prepare the side information* τ*-sums retrieved in this round to be used in the next round by repeating the previous step to decode the undesired* τ*-sums* $y_{19:20,1} - z_{17:18,1}$ *and* $y_{15:16,2} - z_{13:14,2}$ *of the query sets* $\tilde{Q}_j^{(1)}(\mathcal{U}, 2)$*. Note that, unlike in the previous round, we do not have enough symbols to utilize Lagrange interpolation to re-encode the* τ*-sums* $y_{19:20,3} - z_{17:18,3}$ *and* $y_{19:20,4} - z_{17:18,4}$ *as they represent polynomials of degree strictly larger than one.*

Final Round ($\tau = 3$)**:** *The decoding procedure is as follows.*

1. *Interference cancellation: Utilize the decoded* τ*-sums from the set* $\tilde{Q}_j^{(1)}(\mathcal{U}, 2)$ *of Table 4.2(b) to cancel the side information, marked in red in Table 4.1, from the query sets* $Q_j^{(1)}(\mathcal{D}, 3)$ *for* $j \in \{1, 2\}$*. As a result we obtain the desired symbols of the set* $\tilde{Q}_j^{(1)}(\mathcal{D}, 3)$ *shown in Table 4.2(c).*

2. *Generate new symbols: This step is only required when* $\hat{n} - \left\lfloor \hat{n}/\tilde{k} \right\rfloor \tilde{k} < k$ *due to the construction of the interference matrix in the proof of Lemma 6. In particular, the condition is equivalent to* $\Gamma < k$*. Using the obtained symbols from the previous step, colored in Table 4.2 for* $\tilde{Q}_j^{(1)}(\mathcal{D}, 3)$ *with blue, along with the side information downloaded in the previous round in* $Q_j^{(1)}(\mathcal{U}, 2)$*, generate* $\left\lfloor \hat{n}/\tilde{k} \right\rfloor \tilde{k} - (n-k) = 1$ *new* τ*-sum with identical indices to the* τ*-sums retrieved from the nonsystematic nodes. These newly generated symbols are shown in* $\tilde{Q}_j^{(1)}(\mathcal{U}, 3)$*.*

3. *Obtain the desired symbols: Here, we reverse the order of operation of the previous rounds where we use Lagrange interpolation first and then cancel the side information. First, utilize the information sets* $\tilde{\mathcal{I}}_1 = \{1, 3, 4\}$ *and* $\tilde{\mathcal{I}}_2 = \{2, 3, 4\}$ *of* $\tilde{\mathcal{C}}$ *to decode the* τ*-sums containing the desired function evaluation for* $j \in \{1, 2\}$*. As a result, we obtain* $x_{26,1} + y_{24,1} - z_{22,1}$ *and* $x_{25,2} + y_{23,2} - z_{21,2}$*. Next, cancel the side information from the* τ*-sums directly obtained from* $Q_j^{(1)}(\mathcal{U}, 2)$ *for* $j \in \{1, 2\}$*. Finally, by the end of this round, we would have obtained the final* $k\nu\binom{\mu-1}{\tau-1}\kappa^{\mu-\tau}(\nu-\kappa)^{\tau-1} = 6$ *symbols from the desired function evaluation* $\mathbf{X}^{(1)}$*.*

In summary, the total number of desired function evaluation symbols obtained from this decoding process is $k\nu\sum_{\tau=1}^{\mu}\binom{\mu-1}{\tau-1}\kappa^{\mu-\tau}(\nu-\kappa)^{\tau-1} = k\nu^{\mu} = 54$*.* $\triangledown$

4.5 Numerical Results

In Figures 4.5 and 4.5, we compare the PPC rates of Theorems 5 and 6 and those of the schemes from [45], [46] as well as the converse bound from Theorem 4 for various values of the storage code rate $\alpha = k/n$, fixed k, $g = 2$, $f = 2$, $\mu = \mathsf{M}_2^{\mathrm{c}}(2) = \mathsf{M}_2(2) = 5$

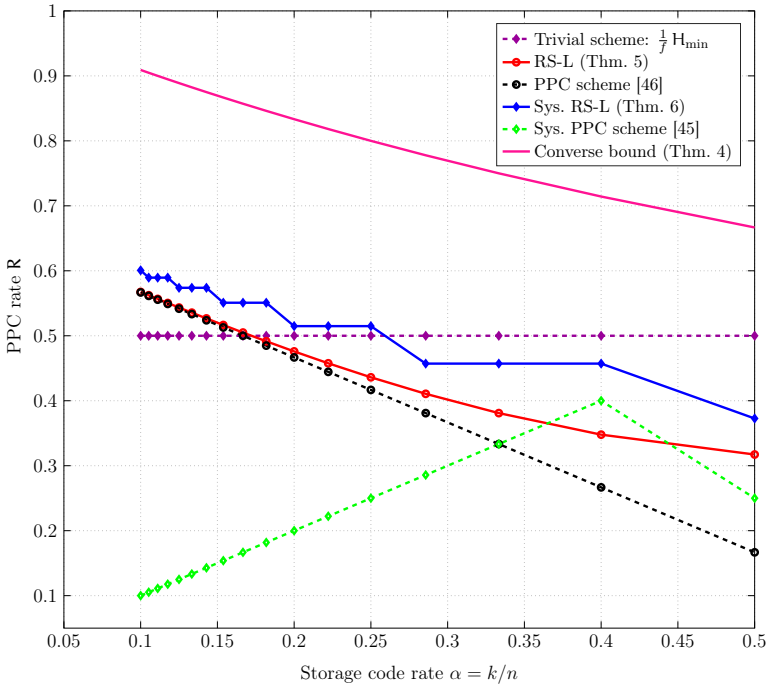

Figure 4.1 PPC rates as a function of the storage code rate $\alpha = k/n$ for $f = 2$, $k = 2$, $g = 2$, and $\mu = \mathsf{M}_2^{\mathrm{c}}(2) = \mathsf{M}_2(2) = 5$. For simplicity, we assume $\mathsf{H}_{\min} = 1$.

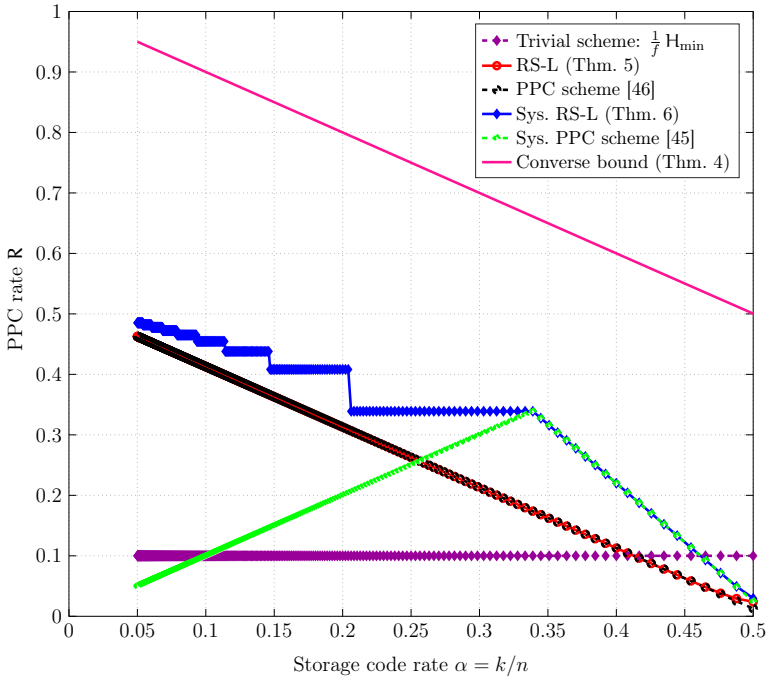

Figure 4.2 PPC rates as a function of the storage code rate $\alpha = k/n$ for $f = 10$, $k = 20$, $g = 2$, and $\mu = \mathsf{M}_2^{\mathrm{c}}(10) = \mathsf{M}_2(10) = 65$. For simplicity, we assume $\mathsf{H}_{\min} = 1$.

for Figure 4.5, and $f = 10$, $\mu = \mathsf{M}_2^{\mathrm{c}}(10) = \mathsf{M}_2(10) = 65$ for Figure 4.5. For a small number of files ($f = 2$), the proposed schemes show improved performance for all code rates, while for a relatively large number of files ($f = 10$), the systematic scheme from Theorem 6 shows improved performance up to some code rate. The converse bound from Theorem 4 shows a relatively large gap for all values of f and storage code rate $\alpha = k/n$. Observe that when neglecting the computational cost at the user, the trivial scheme which downloads all the f files and computes the desired function evaluation offline outperforms all considered PPC schemes when the code rate is above some threshold that depends on both f and g. For $f = 10$ the code rate needs to be close to $1/2$ for the trivial scheme to be the best. Note that the curve for the systematic scheme follows a staircase in which there are $\tilde{k}$ points on each horizontal line of the staircase. This follows directly from the term $\lfloor n/\tilde{k} \rfloor$ in the definition of $\hat{n}$ in equation (4.8).

4.6 Conclusion

For the PPC problem, we have presented two PPC schemes for RS-coded DSSs with Lagrange encoding showing improved computation rates compared to the best known PPC schemes from the literature when the number of messages is small. Asymptotically, as the number of messages tends to infinity, the rate of our RS-coded nonsystematic PPC scheme approaches the rate of the best known nonsystematic PPC scheme. However, for systematically RS-coded DSSs, our scheme significantly outperforms all known PPC schemes up to some specific storage code rate that depends on the maximum degree of the candidate polynomials. Finally, a general converse bound on the PPC rate was derived and compared to the achievable rates of the proposed schemes with some numerical results.

CHAPTER 5

GENERAL PRIVATE COMPUTATION OF NONLINEAR FUNCTIONS FROM REPLICATED DATABASES

In this chapter[8], we consider the general problem of private computation (PC) in a distributed storage system. In such a setting a user wishes to compute a function of f messages replicated across n noncolluding databases, while revealing no information about the desired function to the databases. We provide an information-theoretically accurate achievable PC rate, which is the ratio of the smallest desired amount of information and the total amount of downloaded information, for the scenario of nonlinear computation. For a large message size the rate equals the PC capacity, i.e., the maximum achievable PC rate, when the candidate functions are the f independent messages and one arbitrary nonlinear function of these. When the number of messages grows, the PC rate approaches an outer bound on the PC capacity. As a special case, we consider private monomial computation (PMC) and numerically compare the achievable PMC rate to the outer bound for a finite number of messages.

The PC problem for replicated DSSs differs PC problem from coded-DSSs, that is described in Section 2.2, as follows. We consider a DSS that stores in total f independent messages $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$, where each message $\mathbf{W}^{(m)} = \left(W_1^{(m)}, \ldots, W_{\beta \mathsf{L}}^{(m)}\right)$, $m \in [f]$, is a random length-$\beta\mathsf{L}$ vector with independent and identically distributed symbols that are chosen at random from the field $\mathbb{F}_p$ for some $\beta, \mathsf{L} \in \mathbb{N}$. The messages are replicated and stored on the j-th database, $j \in [n]$. Without loss of generality, we assume that the candidate functions evaluations are ordered descendingly with respect to their entropy, i.e., $\mathsf{H}\left(X^{(1)}\right) = \max_{v \in [\mu]} \mathsf{H}\left(X^{(v)}\right) \triangleq \mathsf{H}_{\max}$ and $\mathsf{H}\left(X^{(\mu)}\right) = \min_{v \in [\mu]} \mathsf{H}\left(X^{(v)}\right) \triangleq \mathsf{H}_{\min}$. Thus, in p-ary

[8]The material presented in this chapter is published in [81].

units, we have

$$\mathsf{H}(X^{(1)}) \geq \mathsf{H}(X^{(2)}) \geq \cdots \geq \mathsf{H}(X^{(\mu)}) \geq 0.$$

In the following sections, we first derive an outer bound on the PC rate of any PC protocol from [55, Thm. 1] (Theorem 7 below) and then an achievable rate for the special case of large message sizes (Theorem 8 below).

5.1 Converse Bound

Theorem 7 *Consider a DSS with n noncolluding replicated databases storing f messages, where the number of arbitrary candidate functions to be computed is $\mu \geq 1$. Then, the PC capacity C_{PC} is upperbounded as*

$$\mathsf{C}_{\mathrm{PC}} \leq \frac{n^{\mu}\,\mathsf{H}_{\min}}{\sum_{v=1}^{\mu} n^{\mu-v+1}\big[\mathsf{H}(X^{[v]}) - \mathsf{H}(X^{[v-1]})\big]}, \tag{5.1}$$

where $X^{[0]}$ is the empty set and $\mathsf{H}(\emptyset) = 0$.

Proof: From the converse proof of either [43] or [55], it is not difficult to see that the total download cost D of a PC protocol is lowerbounded as

$$\mathsf{D} \geq \mathsf{H}\big(\mathbf{X}^{(1)}\big) + \frac{\mathsf{H}\big(\mathbf{X}^{(2)} \,\big|\, \mathbf{X}^{(1)}\big)}{n} + \frac{\mathsf{H}\big(\mathbf{X}^{(3)} \,\big|\, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}\big)}{n^2} + \cdots + \frac{1}{n^{\mu-1}}\,\mathsf{H}\big(\mathbf{X}^{(\mu)} \,\big|\, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu-1)}\big),$$

from which the result follows directly from Definition 13. ■

Corollary 3 *The outer bound from equation* (5.1) *equals*

$$\mathsf{H}_{\min}\frac{1-\frac{1}{n}}{1-(\frac{1}{n})^{f}} \triangleq \mathsf{H}_{\min}\mathsf{C}_{\mathrm{PIR}} \tag{5.2}$$

when $\mu \geq f$ and the candidate functions include the f independent messages $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$, where $\mathsf{C}_{\mathrm{PIR}} = \frac{1-\frac{1}{n}}{1-(\frac{1}{n})^{f}}$ is the PIR capacity for a DSS with n noncolluding replicated databases storing f messages [7].

5.2 Achievability

Theorem 8 *Consider a DSS with n noncolluding replicated databases storing f messages of length $\beta\mathsf{L}$, where the number of arbitrary candidate functions to be computed is $\mu \geq 1$. Then, as $\mathsf{L} \to \infty$, the PC rate*

$$\mathsf{R} = \frac{\mathsf{H}_{\min}}{\sum\limits_{v=1}^{\mu-1} \frac{1}{n^{v-1}} \mathsf{H}(X^{(v)}) + \frac{1}{n^{\mu-1}} \Big[\mathsf{H}(X^{[\mu]}) - \sum\limits_{v=1}^{\mu-1} \mathsf{H}(X^{(v)}) \Big]} \tag{5.3}$$

is achievable.

Corollary 4 *The PC rate R from equation (5.3) is lowerbounded as*

$$\mathsf{R} \geq \frac{\mathsf{H}_{\min}}{\mathsf{H}_{\max}} \frac{1 - \frac{1}{n}}{1 - (\frac{1}{n})^{\mu}}.$$

Corollary 5 *Consider a DSS with n noncolluding replicated databases storing f messages of length $\beta\mathsf{L}$. Then, as $\mathsf{L} \to \infty$, the PC rate*

$$\mathsf{R} = \begin{cases} \mathsf{H}_{\min} \frac{1-\frac{1}{n}}{1-\left(\frac{1}{n}\right)^{f}} = \mathsf{H}_{\min} \mathsf{C}_{\mathrm{PIR}}, & \text{if } \mu = f+1, \\ \frac{\mathsf{H}_{\min}(1-\frac{1}{n})}{1-\left(\frac{1}{n}\right)^{f} + \left(1-\frac{1}{n}\right) \sum\limits_{v=f+1}^{\mu-1} \mathsf{H}(X^{(v)}) \left[\frac{1}{n^{v-1}} - \frac{1}{n^{\mu-1}}\right]}, & \text{if } \mu \geq f+2 \end{cases} \tag{5.4}$$

is achievable when the candidate functions include the f independent messages $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$.

Remark 6

- *For $\mu = f+1$ the PC rate from Corollary 5 equals the outer bound from Corollary 3. Thus, the proposed scheme is capacity-achieving.*
- *The PC rate from Corollary 5 and the outer bound from Corollary 3 converge to $\mathsf{H}_{\min}(1-1/n)$ as $f \to \infty$. A similar result was stated in [43, Thm. 2], however for a simplified definition of the PC rate.*
- *The rate of equation (5.3) extends the elementary capacity result for the case of two arbitrary correlated functions [43, Sec. VII], while the lower bound from Corollary 4 matches the lower bound on the capacity of DPIR [55, Sec. III-B].*
- *If all the μ functions are uniformly distributed, $\mathsf{H}_{\min} = \mathsf{H}_{\max}$ and we obtain the PC rate*

$$\mathsf{R} = \frac{1 - \frac{1}{n}}{1 - (\frac{1}{n})^{\mu}}. \tag{5.5}$$

A PMC problem is a PC problem where the candidate functions to be computed are restricted to a subset of all possible multivariate monomials in f variables (or messages) with degree at most g which includes $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$, where $f \leq \mu \leq \mathsf{M}_g(f)$, $g \in \mathbb{N}$. The goal here is to find a scheme that achieves the outer bound in equation (5.2). Towards this goal, we state the following remark.

Remark 7

- *For multivariate monomials in f variables with degree at most g, it can be seen that the PMC rate*

$$\frac{1 - \frac{1}{n}}{1 - (\frac{1}{n})^{\mu}} \tag{5.6}$$

  can be achieved via the PIR protocol from [7] by considering each candidate monomial as a virtual *message.*

- *In the case of monomials with degree at most $g = 1$, $\mu = f$ (since $\mathsf{M}_g(f) = f$) and $\mathsf{H}_{\min} = \mathsf{H}_{\max}$, and the PMC rate reduces to the PIR capacity $\mathsf{C}_{\mathrm{PIR}}$.*

- *Finally, for monomials with higher degree, i.e., $g \geq 2$, we can achieve a PMC rate R strictly larger than equation* (5.6) *by Corollary 5, using a similar approach of redundancy elimination as in the PPC schemes in Chapter 4, specifically, Section 4.3.4. Moreover, the gap between the achievable PMC rate and the outer bound from equation* (5.2) *decreases with the degree of the monomials and the number of messages (see Section 5.4).*

5.2.1 Achievable Scheme for Theorem 8

We start with a PIR query scheme for μ *virtual* messages, where the μ arbitrary candidate functions of the PC problem are considered as μ arbitrary correlated messages. Given that μ virtual messages are replicated over n noncolluding databases, we require the length of each message to be $\beta\mathsf{L} = n^{\mu}\mathsf{L}$ with a sufficiently large L. Let $\mathbf{X}^{(v)} = (\boldsymbol{X}_1^{(v)}, \ldots, \boldsymbol{X}_{\beta}^{(v)})$, where each segment $\boldsymbol{X}_i^{(v)}$, $i \in [\beta]$, contains L symbols. For $\tau \in [\mu]$, a sum $\boldsymbol{X}_{i_1}^{(v_1)} + \cdots + \boldsymbol{X}_{i_\tau}^{(v_\tau)}$ of τ distinct candidate function segments is called

a τ-sum for any $(i_1, \ldots, i_\tau) \in [\beta]^\tau$, and $\{v_1, \ldots, v_\tau\} \subseteq [\mu]$ determines the type of the τ-sum.

Here, we rely on lossless data compression of large-enough message segments to achieve the PC rate presented in Theorem 8. However, due to possible dependency across message symbols associated with the same subindex, we follow similar index assignment and message symmetry principles as for the PLC scheme in [43] and our PLC scheme in Chapter 3.

The overall protocol is composed of μ rounds. For a desired function indexed by $v \in [\mu]$, a query set $Q_j^{(v)}$, $j \in [n]$, is composed of μ disjoint subsets, one generated by each round $\tau \in [\mu]$. For each round τ the query subset is further subdivided into two subsets. The first subset $Q_j^{(v)}(\mathcal{D}; \tau)$ consists of τ-sums with a single symbol from the *desired* message and $\tau - 1$ symbols from *undesired* messages, while the second subset $Q_j^{(v)}(\mathcal{U}; \tau)$ contains τ-sums with symbols only from undesired messages.[9] We let π be a random permutation over the β message segments. For $v \in [\mu]$,

$$\boldsymbol{U}_t^{(v)} \triangleq \boldsymbol{X}_{\pi(t)}^{(v)}, \quad t \in [\beta],$$

denotes a permuted segment from the virtual message $\mathbf{X}^{(v)}$, where the permutation π is selected privately by the user and is applied as a one-time pad to all messages. Without loss of generality, let the desired virtual message be $\mathbf{X}^{(1)}$. The construction of the queries for arbitrary n and μ is done round-wise for each round $\tau \in [\mu]$ and each database as shown in Table 5.1. The answer string of each database is generated as follows.

- For the first round ($\tau = 1$), optimally compress the length-L segments $\{\boldsymbol{U}_t^{(1)}, \boldsymbol{U}_t^{(2)}, \ldots, \boldsymbol{U}_t^{(\mu)}\}$, $t \in [\beta]$, jointly, which results in $\mathsf{L}\,\mathsf{H}(X^{[\mu]}) + o(\mathsf{L})$ units.
- In the second round ($\tau = 2$), for the 2-sum $\boldsymbol{U}_t^{(v)} + \boldsymbol{U}_{t'}^{(v')}$, $\forall\, v, v' \in [\mu]$, $v < v'$, and $t, t' \in [\beta]$, compress each message segment independently based on $\max\{\mathsf{H}(X^{(v)}), \mathsf{H}(X^{(v')})\}$ and then return the sum of the two compressed

[9] With some abuse of notation, the generated queries are sets containing their answers.

segments, which results in $\mathsf{L}\max\{\mathsf{H}(X^{(v)}), \mathsf{H}(X^{(v')})\} + o(\mathsf{L})$ units. For this round, one can show that in total $(n-1)\sum_{v=1}^{\mu-1}(\mu-v)\mathsf{L}\,\mathsf{H}(X^{(v)}) + o(\mathsf{L})$ units are downloaded.

- For the following rounds ($\tau > 2$), each database compresses the segments of each queried τ-sum $\sum_{l=1}^{\tau} \boldsymbol{U}_{t_l}^{(v_l)}$, where $\{v_1, \ldots, v_\tau\} \subseteq [\mu]$ and $(t_1, \ldots, t_\tau) \in [\beta]^\tau$, separately based on $\max\{\mathsf{H}(X^{(v_1)}), \ldots, \mathsf{H}(X^{(v_\tau)})\}$. Each database then returns the sum of the compressed segments in $\mathsf{L}\max\{\mathsf{H}(X^{(v_1)}), \ldots, \mathsf{H}(X^{(v_\tau)})\} + o(\mathsf{L})$ units. By the end of each round, one can show that in total $(n-1)^{\tau-1}\sum_{v=1}^{\mu-(\tau-1)}\binom{\mu-v}{\tau-1}\mathsf{L}\,\mathsf{H}(X^{(v)}) + o(\mathsf{L})$ units are downloaded for each $\tau \in [3:\mu]$.

Table 5.1 Query Sets for a DSS with n Noncolluding Replicated Databases Storing f Messages

j	1	$\ldots$	n
$Q_j^{(1)}(\mathcal{D};1)$	$\boldsymbol{U}_1^{(1)}$	$\ldots$	$\boldsymbol{U}_n^{(1)}$
$Q_j^{(1)}(\mathcal{U};1)$	$\boldsymbol{U}_1^{(2)}, \ldots, \boldsymbol{U}_1^{(\mu)}$	$\cdots$	$\boldsymbol{U}_n^{(2)}, \ldots, \boldsymbol{U}_n^{(\mu)}$
$Q_j^{(1)}(\mathcal{D};2)$	$\boldsymbol{U}_{n+1}^{(1)} + \boldsymbol{U}_2^{(2)}$ $\vdots$ $\boldsymbol{U}_{n+\mu-1}^{(1)} + \boldsymbol{U}_2^{(\mu)}$ $\vdots$ $\boldsymbol{U}_{n+(\mu-1)(n-1)}^{(1)} + \boldsymbol{U}_n^{(\mu)}$	$\cdots$ $\vdots$ $\cdots$ $\vdots$ $\cdots$	$\boldsymbol{U}_{n+(\mu-1)(n-1)^2+1}^{(1)} + \boldsymbol{U}_1^{(2)}$ $\vdots$ $\boldsymbol{U}_{n+(\mu-1)(n-1)^2+(\mu-1)}^{(1)} + \boldsymbol{U}_1^{(\mu)}$ $\vdots$ $\boldsymbol{U}_{n+n(\mu-1)(n-1)}^{(1)} + \boldsymbol{U}_{n-1}^{(\mu)}$
$Q_j^{(1)}(\mathcal{U};2)$	$\boldsymbol{U}_{n+2}^{(2)} + \boldsymbol{U}_{n+1}^{(3)}$ $\vdots$ $\boldsymbol{U}_{n+(\mu-1)(n-1)}^{(\mu-1)} + \boldsymbol{U}_*^{(\mu)}$	$\cdots$ $\vdots$ $\cdots$	$\boldsymbol{U}_*^{(2)} + \boldsymbol{U}_{n+(\mu-1)(n-1)^2+1}^{(3)}$ $\vdots$ $\boldsymbol{U}_{n+n(\mu-1)(n-1)}^{(\mu-1)} + \boldsymbol{U}_*^{(\mu)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$Q_j^{(1)}(\mathcal{D};\mu)$	$\boldsymbol{U}_*^{(1)} + \cdots + \boldsymbol{U}_*^{(\mu)}$ $\vdots$ $\boldsymbol{U}_*^{(1)} + \cdots + \boldsymbol{U}_*^{(\mu)}$	$\cdots$ $\vdots$ $\cdots$	$\boldsymbol{U}_*^{(1)} + \cdots + \boldsymbol{U}_*^{(\mu)}$ $\vdots$ $\boldsymbol{U}_{n^\mu}^{(1)} + \cdots + \boldsymbol{U}_*^{(\mu)}$

Note: the first ($v = 1$) out of μ candidate functions is privately computed. For simplicity, $\boldsymbol{U}_*^{(v)}$ indicates that the exact requested subindex $t \in [\beta]$ is omitted.

Recovery and Privacy: The scheme inherently satisfies the recovery and privacy conditions stated in Section 2.2. Privacy is guaranteed by satisfying the index, message, and database symmetry principles as for the PLC scheme in [43] and our PLC scheme in Chapter 3. As for the recovery, one can easily see from the PIR query structure that the user is able to obtain all β segments of the desired function based on the answers received from the n databases. Then, each segment is decoded (or optimally decompressed) to obtain in total $\beta\mathsf{L}$ symbols with a probability of decoding error that is arbitrarily close to zero for a sufficiently large L.

Achievable Rate: The PC rate of the scheme, assuming $\mathsf{L} \to \infty$, is given by

$$\begin{aligned}
\mathsf{R} &\overset{(a)}{=} \frac{\beta\mathsf{L}\,\mathsf{H}_{\min}}{\mathsf{D}} \\
&= \frac{n^{\mu}\mathsf{L}\,\mathsf{H}_{\min}}{n\mathsf{L}\left[\mathsf{H}(X^{[\mu]}) + \sum_{\tau=2}^{\mu}(n-1)^{\tau-1}\sum_{v=1}^{\mu-(\tau-1)}\binom{\mu-v}{\tau-1}\mathsf{H}(X^{(v)})\right]} \\
&= \frac{n^{\mu}\,\mathsf{H}_{\min}}{n\left[\mathsf{H}(X^{[\mu]}) + \sum_{\tau=2}^{\mu}(n-1)^{\tau-1}\sum_{v=1}^{\mu-(\tau-1)}\binom{\mu-v}{\tau-1}\mathsf{H}(X^{(v)})\right]} \\
&\overset{(b)}{=} \frac{n^{\mu-1}\,\mathsf{H}_{\min}}{\mathsf{H}(X^{[\mu]}) + \sum_{v=1}^{\mu-1}\sum_{\tau=2}^{\mu-(v-1)}(n-1)^{\tau-1}\binom{\mu-v}{\tau-1}\mathsf{H}(X^{(v)})} \\
&\overset{(c)}{=} \frac{n^{\mu-1}\,\mathsf{H}_{\min}}{\mathsf{H}(X^{[\mu]}) + \sum_{v=1}^{\mu-1}\mathsf{H}(X^{(v)})\sum_{\tau'=1}^{\mu-v}\binom{\mu-v}{\tau'}(n-1)^{\tau'}} \\
&\overset{(d)}{=} \frac{n^{\mu-1}\,\mathsf{H}_{\min}}{\mathsf{H}(X^{[\mu]}) + \sum_{v=1}^{\mu-1}\mathsf{H}(X^{(v)})(n^{\mu-v}-1)} \\
&= \frac{\mathsf{H}_{\min}}{\sum_{v=1}^{\mu-1}\frac{1}{n^{v-1}}\mathsf{H}(X^{(v)}) + \frac{1}{n^{\mu-1}}\left[\mathsf{H}(X^{[\mu]}) - \sum_{v=1}^{\mu-1}\mathsf{H}(X^{(v)})\right]},
\end{aligned} \tag{5.7}$$

where (a) follows from Definition 13, (b) follows from changing the order of the two summations, (c) results by defining $\tau' = \tau - 1$ of the second summation term, and (d) follows from the binomial identity.

For the scenario of Corollary 5, by a similar approach of redundancy elimination as in the PPC schemes in Chapter 4, the PC scheme above can be modified by removing the redundant 1-sums. Using Lemma 5 from Section 4.3.4 and $\mathsf{H}(X^{(v)}) = \mathsf{H}_{\max} = 1$, $\forall\, v \in [f]$, the PC rate can be shown to be equal to equation (5.4).

5.3 Discussion of the Outer Bound of Theorem 7

By expanding the denominator of equation (5.1), denoted by D_{opt}, we get

$$\begin{aligned}\mathsf{D}_{\text{opt}} &= \sum_{v=1}^{\mu} n^{\mu-v+1}\big[\mathsf{H}(X^{[v]}) - \mathsf{H}(X^{[v-1]})\big] \\ &= n\,\mathsf{H}\big(X^{[\mu]}\big) + n(n-1)\,\mathsf{H}\big(X^{[\mu-1]}\big) + \; n(n-1)\cdot n\,\mathsf{H}\big(X^{[\mu-2]}\big) + \cdots \\ &\qquad \cdots + \; n(n-1)\cdot n^{\mu-2}\,\mathsf{H}(X^{(1)}).\end{aligned}$$

Next, consider the total download cost of the achievable scheme for Theorem 8 divided by L, i.e., the denominator of equation (5.7), and denote it by D_1. We have

$$\begin{aligned}\mathsf{D}_1 &= n\,\mathsf{H}(X^{[\mu]}) + \sum_{\tau=2}^{\mu} n(n-1)^{\tau-1} \sum_{v=1}^{\mu-(\tau-1)} \tbinom{\mu-v}{\tau-1}\,\mathsf{H}(X^{(v)}) \\ &= n\,\mathsf{H}(X^{[\mu]}) + n(n-1)\sum_{v=1}^{\mu-1}\tbinom{\mu-v}{1}\,\mathsf{H}(X^{(v)}) + n(n-1)\sum_{v=1}^{\mu-2}(n-1)\tbinom{\mu-v}{2}\,\mathsf{H}(X^{(v)}) + \cdots \\ &\qquad \cdots + \; n(n-1)\cdot(n-1)^{\mu-2}\,\mathsf{H}(X^{(1)}).\end{aligned}$$

By comparing D_{opt} with D_1, it can be seen that because joint compression of the virtual message segments is not utilized, the outer bound of Theorem 7 is not achieved. An open question is to design an optimal scheme that achieves a download cost of D_{opt}.

5.4 Special Case: Private Monomial Computation

In this section, we consider the special case of PMC. One can easily see that the assumption of Corollary 5 covers the scenario of PMC, which includes the f

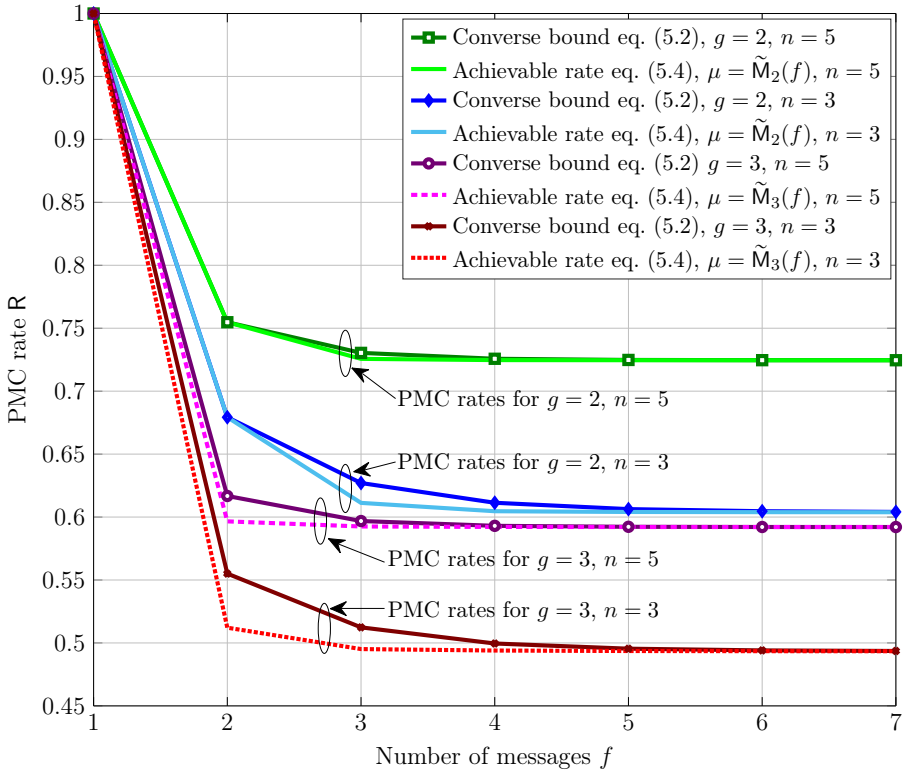

Figure 5.1 PMC rate R versus the number of messages f for the retrieval of nonparallel monomials over the field $\mathbb{F}_3$.

independent messages as candidate functions. Hence, as $\mathsf{L} \to \infty$, the rate in equation (5.4) is achievable for PMC.

In Figure 5.1, for the field $\mathbb{F}_3$ and $n = 3$ and 5, we plot the PMC rate computed from equation (5.4) and the outer bound from equation (5.2) as a function of the number of messages f for $\mu = \widetilde{\mathsf{M}}_g(f)$ with $g = 2$ and $g = 3$, where $\widetilde{\mathsf{M}}_g(f)$ denotes the number of *nonparallel* monomials as defined by equation (2.2) in Chapter 2. Note that the PMC rate is close to the outer bound even for a small number of messages. As $f \to \infty$, it follows from Remark 6 that the PMC rate approaches $\mathsf{H}_{\min}(1 - 1/n)$.

5.5 Conclusion

In this chapter, we presented a novel PC scheme for noncolluding replicated databases and the scenario of nonlinear computation and showed that the resulting PC rate equals the PC capacity as the message size grows for the case when the candidate

functions are the independent messages and one arbitrary nonlinear function of these. Moreover, the PC rate approaches an outer bound on the PC capacity and thus becomes the capacity itself when the number of messages grows. Finally, we compared the outer bound and the achievable rate for the special case of PMC.

CHAPTER 6

MULTI-MESSAGE PLIABLE PRIVATE INFORMATION RETRIEVAL

In this chapter, we formulate a new variant of the Private Information Retrieval (PIR) problem where the user is pliable, i.e., interested in *any* message from a desired subset of the available dataset, coined as Pliable Private Information Retrieval (PPIR). We consider the setup where a dataset consisting of f messages is replicated in n noncolluding databases and classified or categorized into Γ classes. For this setup, the user wishes to retrieve *any* $\lambda \geq 1$ messages from *multiple* desired classes, i.e., $\eta \geq 1$, while revealing no information about the identity of the desired classes to the databases. We term this problem multi-message PPIR (M-PPIR) and introduce the single-message PPIR (PPIR) problem as an elementary special case of M-PPIR. In PPIR, the user wishes to retrieve *any* $\lambda = 1$ message from *one* desired class, i.e., $\eta = 1$, while revealing no information about the identity of the desired class to the databases. For the two considered scenarios we first focus on the case of the single server, i.e., $n = 1$ and derive outer bounds on the M-PPIR rate, which is defined as the ratio of the desired amount of information and the total amount of downloaded information. Next, we design achievable schemes for the single server case and then extend our results to the case of replicated databases. Interestingly, we show that for PPIR from n noncolluding databases, the capacity, i.e., the maximum achievable PPIR rate, is $\mathsf{C}_{\text{PPIR}} = 1/\Gamma$ for $n = 1$ and $\mathsf{C}_{\text{PPIR}} = (1 - 1/n)(1 - 1/n^{\Gamma})^{-1}$ for $n > 1$ which matches the capacity of PIR with n databases and Γ messages. Thus, enabling flexibility, i.e., pliability, allows to trade-off privacy versus download rate compared to classical PIR. A similar insight is shown to hold for the general case of M-PPIR.

The reminder of this chapter is organized as follows. In Section 6.1, we outline the notation and formally define the M-PPIR problem. In Section 6.2, we derive

the converse bound for single-message PPIR as special case of M-PPIR and present a scheme that achieves this bound, hence settling the PPIR capacity for the n replicated DSSs. In section 6.3, we consider the general case of M-PPIR and derive upper and lower bounds on its capacity. Section 6.4 offers the conclusion.

6.1 Preliminaries

6.1.1 System Model

We consider a dataset that consists of a number of f independent messages $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(f)}$. Each message $\boldsymbol{W}^{(m)} = \big(W_1^{(m)}, \ldots, W_{\mathsf{L}}^{(m)}\big)$, $m \in [f]$, is a random length-L vector for some $\mathsf{L} \in \mathbb{N}$, with independent and identically distributed symbols that are chosen at random from the field $\mathbb{F}_p$. The messages are classified into Γ classes for $\Gamma \leq f$[10], $\Gamma \in \mathbb{N}$, and replicated in a distributed storage system (DSS) consisting of n noncolluding databases. Without loss of generality, we assume that the symbols of each message are selected uniformly over the field $\mathbb{F}_p$. Thus,

$$\mathsf{H}\big(\boldsymbol{W}^{(m)}\big) = \mathsf{L}, \, \forall \, m \in [f], \tag{6.1}$$

$$\mathsf{H}\big(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(f)}\big) = f\mathsf{L} \quad (\text{in } p\text{-ary units}). \tag{6.2}$$

Let $\mathcal{M}_\gamma$ be the set of *message indices* belonging to the class indexed with $\gamma \in [\Gamma]$ where $M_\gamma = |\mathcal{M}_\gamma|$ is the size of this set. Note that here, we assume that every message is classified into one class only i.e., $\forall \gamma', \gamma \in [\Gamma]$ and $\gamma' \neq \gamma$, $\mathcal{M}_\gamma \cap \mathcal{M}_{\gamma'} = \phi$ and $\sum_{\gamma=1}^{\Gamma} M_\gamma = f$. Moreover, we assume that there are at least two classes, i.e., $1 \leq M_\gamma \leq f - 1$. Finally, for simplicity of presentation and without loss of generality, we assume that messages are ordered in an ascending order based on their class membership with $\mathcal{M}_\gamma = [(1 + \sum_{i=1}^{\gamma-1} M_i) : (\sum_{i=1}^{\gamma} M_i)]$ for all $\gamma \in [\Gamma]$, i.e.,

$$\{\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(M_1)}\} \in \boldsymbol{W}^{\mathcal{M}_1}$$

$$\{\boldsymbol{W}^{(M_1+1)}, \ldots, \boldsymbol{W}^{(M_1+M_2)}\} \in \boldsymbol{W}^{\mathcal{M}_2}$$

[10]Note that we assume that every message is classified into one class only and no class is empty, i.e., $\Gamma \not> f$.

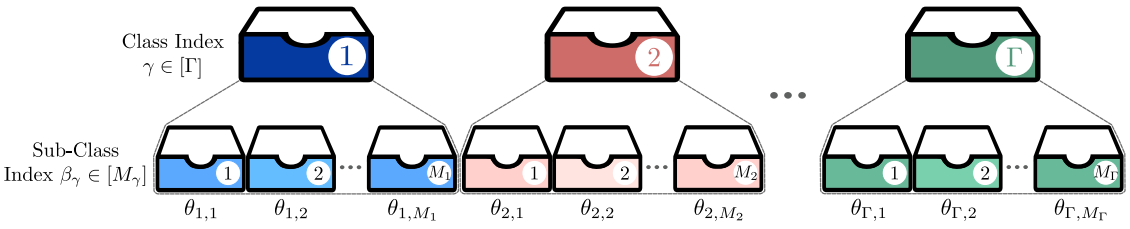

Figure 6.1 Index-mapping of f messages classified into Γ classes using class and sub-class indices, i.e., $\theta_{\gamma,\beta_\gamma} \in \mathcal{M}_\gamma \subset [f]$, $\forall \gamma \in [\Gamma]$.

$$\vdots$$

$$\{\boldsymbol{W}^{(1+\sum_{i=1}^{\Gamma-1} M_i)}, \ldots, \boldsymbol{W}^{(f)}\} \in \boldsymbol{W}^{\mathcal{M}_\Gamma}.$$

To represent the message index-mapping that results from classifying the f messages into Γ classes, let, for $\gamma \in [\Gamma]$, $\theta_{\gamma,\beta_\gamma}$ be the index of a message that belongs to class γ where $\beta_\gamma \in [M_\gamma]$ is a sub-class index and $\theta_{\gamma,\beta_\gamma} \in \mathcal{M}_\gamma$. Here, the sub-class index β_γ represents the membership of a message *within* the class γ as shown in Figure 6.1.

Hence, $\forall \gamma \in [\Gamma]$ and $\forall \beta_\gamma \in [M_\gamma]$, we have the index-mapping

$$\theta_{\gamma,\beta_\gamma} \triangleq \beta_\gamma + \sum_{l=1}^{\gamma-1} M_l. \tag{6.3}$$

Example 7 *Consider that the messages with indices* $\{9, 10, 11\} \subset [f]$ *are members of the second class, i.e.,* $\mathcal{M}_2 = \{9, 10, 11\}$ *and* $M_2 = 3$. *Then,* $\boldsymbol{W}^{(\theta_{2,1})} = \boldsymbol{W}^{(9)}$, $\boldsymbol{W}^{(\theta_{2,2})} = \boldsymbol{W}^{(10)}$, *and* $\boldsymbol{W}^{(\theta_{2,3})} = \boldsymbol{W}^{(11)}$.

6.1.2 Problem Statement

In multi-message PPIR (M-PPIR) problem, the user wishes to retrieve a total of *any* μ messages from a subset of η *desired* classes indexed by the index set $\Omega \subseteq [\Gamma]$ where $|\Omega| = \eta$. The desired number of messages μ is distributed among the desired classes as $\mu = \sum_{i=1}^{\eta} \lambda_{\gamma_i}$ where λ_{γ_i} is the number of *desired* messages

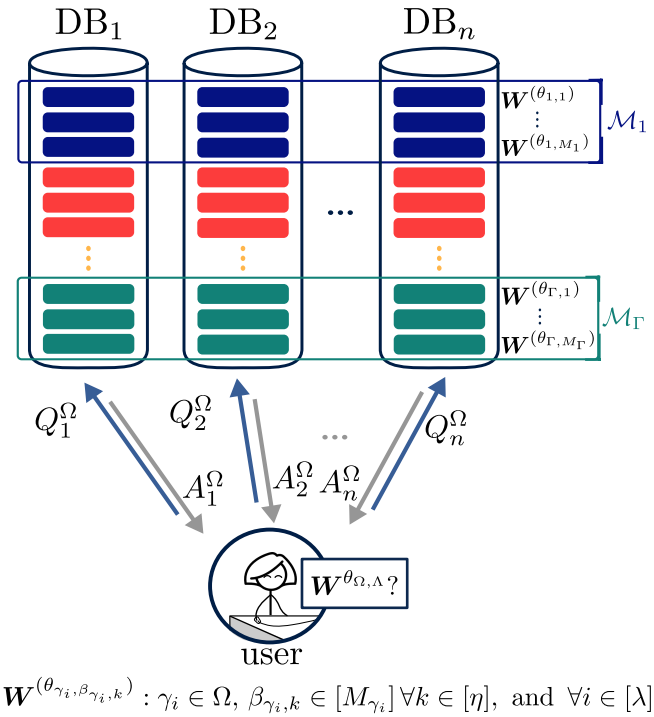

Figure 6.2 System model for M-PPIR from an n replicated noncolluding databases storing f messages classified into Γ classes. The user intends to download λ messages each out of η desired classes.

from the desired class $\gamma_i \in \Omega$. For the scope of this work and for tractability we restrict ourselves to a fixed number of requested messages from each desired class, i.e., $\lambda_{\gamma_i} = \lambda\ \forall \gamma_i \in \Omega$ and $\mu = \lambda\eta$. Moreover, we impose the mild assumption that the user only has prior knowledge of LCM, i.e., the least common multiple of the sizes of the Γ classes $\delta \triangleq \text{LCM}(M_1, \ldots, M_\Gamma)$. In other words, the user *does not* know the *size* of each class nor the total number of files stored at the database. Accordingly, the user wishes to privately retrieve *any* λ messages out of M_{γ_i} messages within a desired *class* $\gamma_i \in \Omega$, $\forall i \in [\eta]$, which are denoted by $\{\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,1}})}, \boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,2}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,\lambda}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\gamma_\eta,\lambda}})}\}$, i.e.,

$$\{\boldsymbol{W}^{(\theta_{\gamma_i,\beta_{\gamma_i,k}})} : \gamma_i \in \Omega,\ \beta_{\gamma_i,k} \in [M_{\gamma_i}] \quad \forall k \in [\lambda],\ \text{and}\ \forall i \in [\eta]\}.$$

Example 8 *Consider a dataset consisting of $f = 15$ messages classified into $\Gamma = 3$ classes with sizes $\{6, 4, 5\}$, respectively. Suppose a user that wishes to retrieve any*

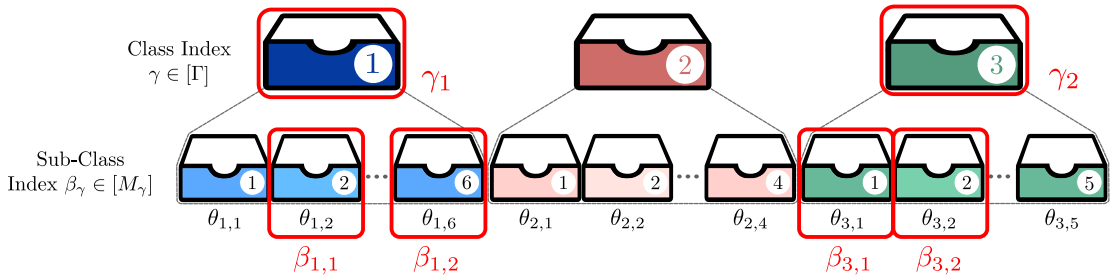

Figure 6.3 Index mapping for M-PPIR problem of Example 8. The user selects $\Omega = \{1, 3\}$, i.e., $\gamma_1 = 1$ and $\gamma_2 = 3$ and wants to retrieve any two messages from each class. Highlighted in red, are two arbitrary sub-class indices from each desired class.

$\lambda = 2$ *messages from the set of classes* $\Omega = \{1, 3\}$. *The indices of the* two *arbitrary selected messages from each class are shown in Figure 6.3. The sub-class index of the* first *message from the* first *class, i.e.,* $i = 1$, $k = 1$ *and* $\gamma_1 = 1$, *is given by* $\beta_{1,1} = 2$. *From the index-mapping of equation* (6.3), *we have* $\theta_{1,\beta_{1,1}} = \beta_{1,1} = 2$ *and similarly,* $\theta_{1,\beta_{1,2}} = \beta_{1,2} = M_1 = 6$. *Next, the sub-class index of the* first *message from the* second *class, i.e.,* $i = 2$, $k = 1$ *and* $\gamma_2 = 3$, *is given by* $\beta_{3,1} = 1$. *From the index-mapping equation* (6.3), *we have* $\theta_{3,\beta_{3,1}} = 1 + \sum_{l=1}^{2} M_l = 11$ *and similarly for* $\theta_{3,\beta_{3,2}} = 2 + \sum_{l=1}^{2} M_l = 12$.

The user privately selects a subset of η class indices $\Omega = \{\gamma_1, \gamma_2, \ldots, \gamma_\eta\} \subseteq [\Gamma]$, and wishes to retrieve *any* λ messages from each of the desired classes while keeping the identities of the requested classes in Ω private from each database. In order to retrieve the desired messages $\{\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{1,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_1,\beta_{1,\lambda}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\eta,\lambda}})}\}$, the user sends a random query Q_j^Ω to the database $j \in [n]$. The query is generated by the user without any prior knowledge of the realizations of the stored messages. In other words,

$$\mathrm{I}\big(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(f)}; Q_1^\Omega, \ldots, Q_n^\Omega\big) = 0. \tag{6.4}$$

In response to the received query, the j-th database sends the answer A_j^Ω back to the user, where A_j^Ω is a deterministic function of Q_j^Ω and the data stored in the database.

Thus,

$$\mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{[f]}\big) = 0, \ \forall\, j \in [n]. \tag{6.5}$$

Note that, here we assume that there exists *at least* λ messages in each class, i.e., $M_\gamma \geq \lambda, \forall \gamma \in [\Gamma]$. Let $\mathcal{V}$ and $\mathcal{T}$ be two arbitrary subsets of $\mathcal{M}_\gamma$ such that $\mathcal{V} \subseteq \mathcal{T} \subseteq \mathcal{M}_\gamma$ and $|\mathcal{V}| = \lambda$. It follows from the definition of the M-PPIR problem

$$\mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{V}}\big) = \mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{T}}\big). \tag{6.6}$$

This is unlike the classical PIR setup where the answer string is generate given all of the messages in the dataset. Hence, from the chain rule of entropy we have $\mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{V}}\big) \geq \mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{T}}\big)$ for the classical M-PIR. In other words, in M-PPIR, the answer from the database $j \in [n]$ is generated as a deterministic function given a sufficient amount of information, i.e., at least *any* λ messages from a class for any class $\gamma \in [\Gamma]$. Similarly, let $v' \in \mathcal{M}_\gamma^{\mathrm{c}} \triangleq [f] \setminus \mathcal{M}_\gamma$ and $\mathcal{V}' \subseteq \mathcal{M}_\gamma^{\mathrm{c}}$. Then it follows from equation (6.6) that

$$\mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{V}}\boldsymbol{W}^{(v')}\big) = \mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{T}}\boldsymbol{W}^{(v')}\big), \tag{6.7}$$

and

$$\mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{V}}\boldsymbol{W}^{\mathcal{V}'}\big) = \mathsf{H}\big(A_j^{\Omega} \,\big|\, Q_j^{\Omega}, \boldsymbol{W}^{\mathcal{T}}\boldsymbol{W}^{\mathcal{V}'}\big). \tag{6.8}$$

To satisfy the user privacy requirement, the query-answer function must be identically distributed for all possible subset of class indices $\Omega \subseteq [\Gamma]$ from the perspective of each database. In other words, the scheme's query and answer string must be independent from the desired class index set, i.e.,

$$\mathsf{I}\big(\Omega; Q_j^{\Omega}, A_j^{\Omega}\big) = 0, \ \forall j \in [n]. \tag{6.9}$$

Moreover, the user must be able to reliably decode, from the received databases answers, any λ messages from the desired classes i.e., $\{\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,1}})},\ldots,\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,\lambda}})},\ldots,\boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\gamma_\eta,\lambda}})}\}$ for $\gamma_i \in \Omega$. Accordingly, the M-PPIR protocol from replicated DSS is defined as follows.

Consider a DSS with n noncolluding replicated databases storing f messages classified into Γ classes. The user wishes to retrieve any λ messages from each class in the desired class index set $\Omega \subseteq [\Gamma]$, from the queries Q_j^Ω and answers A_j^Ω, $\forall j \in [n]$. Let $\mathfrak{S}$ be the set of all unique subsets of $[\Gamma]$ of size η, and $\mathcal{M}_{\gamma_i}$ be the index set of the messages classified into the class $\gamma_i \in \Omega$, then for an M-PPIR protocol, the following conditions must be satisfied $\forall\, \Omega, \Omega' \in \mathfrak{S}$, $\Omega \neq \Omega'$, and $j \in [n]$

$$\text{[Privacy]} \qquad (Q_j^\Omega, A_j^\Omega, \boldsymbol{W}^{[f]}) \sim (Q_j^{\Omega'}, A_j^{\Omega'}, \boldsymbol{W}^{[f]})^{11}, \tag{6.10}$$

$$\text{[Correctness]}\ \mathsf{H}\big(\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,1}})},\ldots,\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,\lambda}})},\ldots \boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\gamma_\eta,\lambda}})} \,\big|\, A_{[n]}^\Omega, Q_{[n]}^\Omega\big) = 0. \tag{6.11}$$

To measure the efficiency of an M-PPIR protocol, we consider the required number of downloaded symbols for retrieving the L symbols of the $\mu = \lambda\eta$ desired messages.

Definition 13 (M-PPIR rate and capacity for replicated DSSs) *The rate of an M-PPIR protocol, denoted by* R*, is defined as the ratio of the desired information size,* $\lambda\eta$ *messages each consisting of* L *symbols, to the total required download cost* D*, i.e.,*

$$\mathsf{R} \triangleq \frac{\eta\lambda\mathsf{L}}{\mathsf{D}} = \frac{\eta\lambda\mathsf{L}}{\sum_{j=1}^{n}\mathsf{H}\big(A_j^\Omega\big)}.$$

The M-PPIR capacity*, denoted by* $\mathsf{C}_{\text{M-PPIR}}$*, is the maximum achievable M-PPIR rate over all possible M-PPIR protocols.*

[11]The privacy constraint can be alternatively expressed as equation (6.9).

6.1.3 Special Cases

In this subsection, we introduce some special cases of the general M-PPIR problem presented in Section 6.1.1 emerging from choosing different values of λ and η. We use these special cases, namely PPIR, single-class M-PPIR, and multi-class M-PPIR, as building-blocks for the general M-PPIR problem. As this work is an introduction to the PPIR problem, we find it useful to see how these special cases relate to and extend classical PIR problems.

Single-Message PPIR (in short denoted as PPIR $(\lambda = 1, \eta = 1)$**)** Here, the user is interested in a *single* message from a *single* desired class[12]. In PPIR, the user privately selects a class index $\gamma \in [\Gamma]$ and wishes to privately retrieve *any one* message out of the M_γ *candidate* messages of the desired class, i.e., $\boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,1}})}$: $\theta_{\gamma,\beta_{\gamma,1}} \in \mathcal{M}_{\gamma_1}$, $\gamma \in [\Gamma]$, while keeping the desired class index γ private from each database $j \in [n]$. Note that when the number of classes is equal to the number of messages, i.e., there is only one message in each class and $\Gamma = f$, the PPIR problem reduces to the classical PIR problem [7].

Single-Class M-PPIR $(\lambda \geq 1, \eta = 1)$ Here, the user is interested in *multiple* messages from a *single* desired class. In single-class M-PPIR, the user privately selects a class index $\gamma \in [\Gamma]$ and wishes to privately retrieve *any* $\lambda \geq 1$ messages out of M_γ *candidate* messages within the desired class, i.e., $\{\boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,\lambda}})}$: $\theta_{\gamma,\beta_{\gamma,k}} \in \mathcal{M}_\gamma, \forall k \in [\lambda]\}$, without revealing the identity of the desired class γ to each database $j \in [n]$.

Multi-Class M-PPIR $(\lambda = 1, \eta \geq 1)$ Here, the user is interested in a *single* message from *multiple* desired classes. In this case, the user privately selects a subset of class indices $\Omega \subseteq [\Gamma]$ of size η and wishes to retrieve *any* one message from each of

[12]For notation simplicity, we drop the desired class subscript when it is understood from the context, i.e., there is only one desired class $\eta = 1$.

the η desired classes $\gamma_i \in \Omega$, i.e., $\{\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\gamma_\eta,1}})} : \theta_{\gamma_i,\beta_{\gamma_i,1}} \in \mathcal{M}_{\gamma_i}, \gamma_i \in \Omega, \forall i \in [\eta]\}$, without revealing the identity of the desired class index set Ω to each database $j \in [n]$. Note that when the number of classes is equal to the number of messages, i.e., there is only one message in each class and $\Gamma = f$, the multi-class M-PPIR problem reduces to the multi-message PIR (MPIR) problem [9].

6.2 Pliable Private Information Retrieval

In this section, we discuss the PPIR problem as a special case of the M-PPIR problem with $\lambda = 1, \eta = 1$. The significance of presenting this special case lies within the direct connection to the well known classical PIR problem in [7], thus, providing an intuitive tutorial style introduction to the general M-PPIR problem. In the following, we derive the capacity of PPIR, which indicates a significant (possible) reduction in download rate compared to the capacity of classical PIR. In the PPIR problem we assume that the user is *oblivious* to the structure of the database, i.e., has no knowledge of the messages membership in each class and construct achievable schemes accordingly. To this end, we first consider the single server case, i.e., $n = 1$, characterize the capacity of single-server PPIR (see theorem 9), and present a capacity-achieving scheme. Then, we extend our capacity result to replication-based DSSs, i.e., $n > 1$ (see theorem 10).

6.2.1 Single Server PPIR

Theorem 9 *For the PPIR problem with single server storing f messages classified into Γ classes, the maximum achievable PPIR rate over all possible PPIR protocols, i.e., the PPIR capacity C_{PPIR}, is given by*

$$\mathsf{C}_{\text{PPIR}} = \frac{1}{\Gamma}.$$

In the following we start with the converse proof of Theorem 9.

Converse proof of Theorem 9 We base this proof on an induction argument. We first prove the outer bound for $M_\gamma = 1, \forall \gamma \in [\Gamma]$, i.e., each class contains only one message and $f = \Gamma$, for arbitrary f, then proceed to the case of arbitrary M_γ, $\forall \gamma \in [\Gamma]$.

- For $M_\gamma = 1, \forall \gamma \in [\Gamma]$, we have $\Gamma = f$. Accordingly, in order to maintain the privacy of the desired class identity $\gamma \in [\Gamma]$, we must maintain the privacy of the retrieved message identity $\theta_{\gamma,1} \in [f]$. As a result, the capacity of single server PPIR matches the capacity of the single server PIR problem, i.e., $\mathsf{C}_{\text{PPIR}} = \frac{1}{f} = \frac{1}{\Gamma}$ [2].
- For $M_\gamma > 1, \forall \gamma \in [\Gamma]$, in order to maintain the privacy of the desired class identity, we must download *at least* one message from each class. Accordingly, the probability that any one of the classes is the desired class is uniformly distributed, thus achieving perfect information theoretic privacy. Since there is more than one message in each class, and the user requests *any* message from her desired class, the identity of the selected message is not relevant. Accordingly, by randomly selecting one message from each class as an answer to the user it follows that the best information retrieval rate, i.e., PPIR capacity, must be bounded by $\frac{1}{\Gamma}$, i.e., $\mathsf{C}_{\text{PPIR}} = \frac{1}{\Gamma}$.

Achievability of Theorem 9 For the achievability of theorem 9, recall that the user has prior knowledge of $\delta \triangleq \text{LCM}(M_1, \ldots, M_\Gamma)$. To privately retrieve a message from the desired class, the user selects a random number $s \in [\delta]$ and sends it to the database. Based on the selected random number, a subset of size Γ from the messages is selected by the database, one message from each class[13]. The identity of the selected message from each class is computed as $\beta_{\gamma,1} = \lceil \frac{s}{\delta} M_\gamma \rceil$ and due to the ascending order of the messages based on their class membership, from equation (6.3), the message index is $\theta_{\gamma,\beta_{\gamma,1}} = \beta_{\gamma,1} + \sum_{l=1}^{\gamma-1} M_l$. Finally, the set of candidate messages $\{\boldsymbol{W}^{(\theta_{1,\beta_{1,1}})}, \boldsymbol{W}^{(\theta_{2,\beta_{2,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\Gamma,\beta_{\Gamma,1}})}\}$ are set as the answer to the user. Note that the query is fixed, independently of the desired class. Consequently, from the perspective of the database, the query and answer string of any desired class are indistinguishable and the privacy constraint of equation (6.10) is satisfied. Moreover, since the user

[13]Given that the random number s is selected from the set of size equal to the LCM of the sizes of all classes, each message in a given class is equally likely to be a member of the candidate message set.

obtains one message $\boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,1}})}$ from the desired class $\gamma \in [\Gamma]$ the correctness constraint of equation (6.11) is satisfied. As a result, the PPIR rate $\mathsf{R} = \frac{\mathsf{L}}{\Gamma\mathsf{L}} = \frac{1}{\Gamma}$ is achieved.

6.2.2 PPIR over Replicated DSS

Next, we state our main result for PPIR over replicated DSS with theorem 10 as follows.

Theorem 10 *Consider a DSS with n noncolluding replicated databases storing f messages classified into Γ classes. The maximum achievable PPIR rate over all possible PPIR protocols, i.e., the PPIR capacity C_{PPIR}, is given by*

$$\begin{aligned}\mathsf{C}_{\text{PPIR}} &= \left(1 + \frac{1}{n} + \frac{1}{n^2} + \cdots + \frac{1}{n^{\Gamma-1}}\right)^{-1} \\ &= \left(1 - \frac{1}{n}\right)\left(1 - \frac{1}{n^{\Gamma}}\right)^{-1}.\end{aligned}$$

Before we start the converse proof, we present a number of useful lemmas and simplifying assumptions. Without loss of generality, assume that,

- From the queries and answers of each database $j \in [n]$, we can successfully decode the first λ message in each desired class $\gamma_i \in \Omega$ for any $\Omega \in \mathfrak{S}$ where $\mathfrak{S}$ is the set of all unique subsets of $[\Gamma]$ of size η. As a result, $\beta_{\gamma_i,k} = k$ for all $k \in [\lambda], i \in [\eta]$ and we can write the message index $\theta_{\gamma_i,\beta_{\gamma_i,k}}$ as $\theta_{\gamma_i,k}$ for all $k \in [\lambda]$. Let $\theta_{\gamma_i,k}$ denote the index of the k-th message in class $\gamma_i \in [\Gamma]$. Then, for example, from the answers of desired classes indexed with set $\Omega = [\eta] = \{1, 2, \ldots, \eta\}$ we can successfully decode $\{\boldsymbol{W}^{(\theta_{1,1})}, \ldots, \boldsymbol{W}^{(\theta_{1,\lambda})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,\lambda})}\}$. For simplicity, with some abuse of notation, we let $\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \triangleq \{\boldsymbol{W}^{(\theta_{1,1})}, \ldots, \boldsymbol{W}^{(\theta_{1,\lambda})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,\lambda})}\}$. As a result, from equation (6.11), we have $\mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \,\big|\, A_{[n]}^{[\eta]}, Q_{[n]}^{[\eta]}\big) = 0$.
- Let $\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}$ be the complement subset of files for the set $\boldsymbol{W}^{\theta_{[\eta],[\lambda]}}$, where

  $$\theta_{[\eta],[\lambda]} \triangleq \{\theta_{1,1}, \theta_{1,2}, \ldots, \theta_{1,\lambda}, \theta_{2,1}, \ldots, \theta_{\eta,1}, \ldots, \theta_{\eta,\lambda}\},$$

  i.e., $\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}} \triangleq \boldsymbol{W}^{[\theta_{1,\lambda+1}:\theta_{2,1}-1]} \cup \boldsymbol{W}^{[\theta_{2,\lambda+1}:\theta_{3,1}-1]} \cup \cdots \cup \boldsymbol{W}^{[\theta_{\eta-1,\lambda+1}:\theta_{\eta,1}-1]} \cup \boldsymbol{W}^{[\theta_{\eta,\lambda+1}:f]}$.

Lemma 9 $\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{[\eta]} A_{[n]}^{[\eta]} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \leq \eta\lambda\mathsf{L}(\frac{1}{\mathsf{R}} - 1)$.

The proof of lemma 9 is given in Appendix F.

Lemma 10 *Let $\Omega_1, \Omega_2 \in \mathfrak{S}$, such that $\Omega_1 \cap \Omega_2 = \phi$, without loss of generality, assume that $\Omega_1 = [\eta]$ and $\Omega_2 = [\eta + 1 : 2\eta]$. Then*

$$\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{\Omega_1} A_{[n]}^{\Omega_1} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \geq \frac{\eta\lambda \mathsf{L}}{n} + \frac{1}{n}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[2\eta],[\lambda]}}; Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[2\eta],[\lambda]}}\Big). \quad (6.12)$$

The proof of lemma 10 is given in Appendix G.

Converse proof of Theorem 10 We now proceed to the proof of the converse of Theorem 10. For $\gamma \in [\Gamma]$, let $\boldsymbol{W}^{\theta_{[\gamma],1}} \triangleq \{\boldsymbol{W}^{(\theta_{1,1})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma,1})}\}$.

Proof: From Lemma 9 we have for $\lambda = 1$ and $\eta = 1$

$$\mathrm{I}\Big(\boldsymbol{W}^{[\theta_{1,2:f}]}; Q_{[n]}^{(1)} A_{[n]}^{(1)} \,\Big|\, \boldsymbol{W}^{(\theta_{1,1})}\Big) \leq \mathsf{L}\big(\frac{1}{\mathsf{R}} - 1\big). \quad (6.13)$$

Next, from Lemma 10 we have for $\lambda = 1$, $\eta = 1$, and $\gamma \in [2 : \Gamma]$

$$\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; Q_{[n]}^{(\gamma-1)} A_{[n]}^{(\gamma-1)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \geq \frac{\mathsf{L}}{n} + \frac{1}{n}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma],1}}; Q_{[n]}^{(\gamma)} A_{[n]}^{(\gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma],1}}\Big). \quad (6.14)$$

Now, starting by $\gamma = 2$, then applying equation (6.14) repeatedly for $\gamma \in [3 : \Gamma]$, we have

$$\begin{aligned}
&\mathrm{I}\Big(\boldsymbol{W}^{[\theta_{1,2:f}]}; Q_{[n]}^{(1)} A_{[n]}^{(1)} \,\Big|\, \boldsymbol{W}^{(\theta_{1,1})}\Big) \\
&\geq \frac{\mathsf{L}}{n} + \frac{1}{n}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[2],1}}; Q_{[n]}^{(2)} A_{[n]}^{(2)} \,\Big|\, \boldsymbol{W}^{\theta_{[2],1}}\Big) \\
&\geq \frac{\mathsf{L}}{n} + \frac{1}{n}\left[\frac{\mathsf{L}}{n} + \frac{1}{n}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[3],1}}; Q_{[n]}^{(3)} A_{[n]}^{(3)} \,\Big|\, \boldsymbol{W}^{\theta_{[3],1}}\Big)\right] \\
&= \frac{\mathsf{L}}{n} + \frac{\mathsf{L}}{n^2} + \frac{1}{n^2}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[3],1}}; Q_{[n]}^{(3)} A_{[n]}^{(3)} \,\Big|\, \boldsymbol{W}^{\theta_{[3],1}}\Big) \\
&\geq \quad \vdots \\
&\geq \frac{\mathsf{L}}{n} + \cdots + \frac{\mathsf{L}}{n^{\Gamma-2}} + \frac{1}{n^{\Gamma-2}}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\Gamma-1],1}}; Q_{[n]}^{(\Gamma-1)} A_{[n]}^{(\Gamma-1)} \,\Big|\, \boldsymbol{W}^{\theta_{[\Gamma-1],1}}\Big) \\
&\geq \frac{\mathsf{L}}{n} + \cdots + \frac{\mathsf{L}}{n^{\Gamma-2}} + \frac{\mathsf{L}}{n^{\Gamma-1}} + \frac{1}{n^{\Gamma-1}}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\Gamma],1}}; Q_{[n]}^{(\Gamma)} A_{[n]}^{(\Gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\Gamma],1}}\Big)
\end{aligned}$$

$$\stackrel{(a)}{=} \frac{\mathsf{L}}{n} + \cdots + \frac{\mathsf{L}}{n^{\Gamma-2}} + \frac{\mathsf{L}}{n^{\Gamma-1}} + \frac{1}{n^{\Gamma-1}} \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\Gamma],1}}; A_{[n]}^{(\Gamma)} \,\Big|\, Q_{[n]}^{(\Gamma)}\boldsymbol{W}^{\theta_{[\Gamma],1}}\Big)}_{=0}$$

where in (a) the last term equals zero due to the independence of the messages and the queries as given by equation (6.4) and the fact that the answer strings are a deterministic function of the queries and a *sufficient* number of messages from each classes, i.e., combining equations (6.5) and (6.6) we have

$$\begin{aligned}
&\mathsf{H}\big(A_{[n]}^{(\Gamma)} \,\big|\, Q_{[n]}^{(\Gamma)}\boldsymbol{W}^{\theta_{[\Gamma],1}}\big) \\
&= \mathsf{H}\big(A_{[n]}^{(\Gamma)} \,\big|\, Q_{[n]}^{(\Gamma)}\boldsymbol{W}^{(\theta_{1,1})}\boldsymbol{W}^{\theta_{[2:\Gamma],1}}\big) \\
&= \mathsf{H}\big(A_{[n]}^{(\Gamma)} \,\big|\, Q_{[n]}^{(\Gamma)}\boldsymbol{W}^{[\theta_{1,1}:\theta_{2,1}-1]}\boldsymbol{W}^{(\theta_{2,1})}\boldsymbol{W}^{\theta_{[3:\Gamma],1}}\big) \\
&= \vdots \\
&= \mathsf{H}\big(A_{[n]}^{(\Gamma)} \,\big|\, Q_{[n]}^{(\Gamma)}\boldsymbol{W}^{[f]}\big) = 0.
\end{aligned}$$

As a result, we obtain

$$\mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; Q_{[n]}^{(1)}A_{[n]}^{(1)} \,\Big|\, \boldsymbol{W}^{(\theta_{1,1})}\Big) \geq \frac{\mathsf{L}}{n} + \cdots + \frac{\mathsf{L}}{n^{\Gamma-2}} + \frac{\mathsf{L}}{n^{\Gamma-1}}. \tag{6.15}$$

Combining equations (6.15) and (6.13) yields

$$\mathsf{L}\left(\frac{1}{\mathsf{R}} - 1\right) \geq \frac{\mathsf{L}}{n} + \cdots + \frac{\mathsf{L}}{n^{\Gamma-2}} + \frac{\mathsf{L}}{n^{\Gamma-1}}, \tag{6.16}$$

and by eliminating L from both sides, we finally obtain

$$\mathsf{R} \leq \left(1 + \frac{1}{n} + \frac{1}{n^2} + \cdots + \frac{1}{n^{\Gamma-1}}\right)^{-1} \tag{6.17}$$

$$= \left(1 - \frac{1}{n}\right)\left(1 - \frac{1}{n^{\Gamma}}\right)^{-1}. \tag{6.18}$$

■

Achievability of Theorem 10 We now present a scheme that achieve the PPIR capacity bound of theorem 10. The capacity of the PIR problem with n noncolluding replicated databases, each storing f messages, was characterized in [7] as $\left(1-\frac{1}{n}\right)\left(1-\frac{1}{n^f}\right)^{-1}$. From the capacity bound of PPIR in theorem 10, one can observe that PPIR effectively reduces the size of the database from f to Γ messages. Thus, our capacity achieving PPIR scheme extends the single server PPIR solution of section 6.2.1 to a replicated DSS setup through adaptation of the capacity achieving PIR scheme in [7]. Given Γ, n, $\gamma \in [\Gamma]$, and $\delta = \text{LCM}(M_1, \ldots, M_\Gamma)$, the high-level implementation of the PPIR scheme is outlined with the following steps.

1. The user selects a number uniformly at random from the set $[\delta]$.
2. The user constructs queries according to [7, Section IV] for n noncolluding replicated database storing Γ candidate messages $\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(\Gamma)}\}$, to privately retrieve $\boldsymbol{X}^{(\gamma)}$, $\gamma \in [\Gamma]$, i.e., $Q_1^{(\gamma)}, \ldots, Q_n^{(\gamma)}$. Each message is assumed to be of length $\mathsf{L} = n^\Gamma$ [7].
3. The user sends the selected random number from Step 1, $s \in [\delta]$, followed by the constructed queries $Q_1^{(\gamma)}, \ldots, Q_n^{(\gamma)}$, in a random order to each database $j \in [n]$.
4. Given the random number $s \in [\delta]$, each database $j \in [n]$ computes the indices of Γ messages, one from each class, to be used in constructing its answer string $A_j^{(\gamma)}$. These indices are computed as $\theta_{\gamma,\beta_{\gamma,1}} = \lceil \frac{s}{\delta} M_\gamma \rceil + \sum_{l=1}^{\gamma-1} M_l$. Herein, $\beta_{\gamma,1} = \lceil \frac{s}{\delta} M_\gamma \rceil$ is the membership of the selected message within its class, and $\theta_{\gamma,\beta_{\gamma,1}}$ follows due to the fact that the messages are ordered in an ascending order based on their class membership. Each of these Γ messages are mapped to the user's queries of Step 2 as $\boldsymbol{X}^{(\gamma)} = \boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,1}})}$ for all $\gamma \in [\Gamma]$.

Privacy: Note that the query structure of the PIR capacity achieving scheme in [7] is fixed independently of the desired *candidate* message index $\gamma \in [\Gamma]$. This fixed structure adheres to three principles to achieve this independence, namely,

- Database symmetry: symmetry across databases is enforced. This is accomplished by querying each database $j \in [n]$ for the same number of message symbols. Thus, the query structure does not depend on the individual database, i.e., the scheme structure is constructed to be fixed for all databases.
- Message symmetry: symmetry across messages is enforced. This is accomplished by querying the same number of message symbols form each message $\boldsymbol{X}^{(\gamma)}, \forall \gamma \in [\Gamma]$.

- Side-information exploitation: the symbols of *undesired* messages, i.e., $\boldsymbol{X}^{[\Gamma]\setminus\{\gamma\}}$, that are obtained as a result from enforcing database and message symmetry are exploited to obtain new symbols of the *desired* message $\boldsymbol{X}^{(\gamma)}$.

Since the achievable construction of [7] guarantees that any message $\boldsymbol{X}^{(\gamma)}$ for all $\gamma \in [\Gamma]$ is equally likely to be the user's demand, it follows that the Γ messages $\boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,1}})}$ for $\gamma \in [\Gamma]$, i.e., one message from each class, are also equally likely to be the user's demand. As a result, $\mathbb{P}(\gamma = \gamma | Q_j^{(\gamma)}, A_j^{(\gamma)}) = \frac{1}{\Gamma}$ for any $\gamma \in [\Gamma]$, $j \in [n]$, and the query and answer string of any desired class $\gamma \in [\Gamma]$ are indistinguishable from the perspective of each database. This in turns satisfies the M-PPIR privacy constraint of equation (6.10).

Correctness: Given that the scheme in [7] guarantees the retrieval of all the n^{Γ} symbols of $X^{(\gamma)}$ which is mapped by each database to $\boldsymbol{W}^{(\theta_{\gamma,\beta_{\gamma,1}})}$, the user obtains all the symbols of a message that belongs to the class γ. Thus, the M-PPIR correctness constraint of equation (6.11) is satisfied.

Calculation of achievable rate: For privately retrieving one message from a candidate set of size Γ from n replicated databases, the scheme of [7] achieves an information retrieval rate of $\left(1 + \frac{1}{n} + \frac{1}{n^2} + \cdots + \frac{1}{n^{\Gamma-1}}\right)^{-1}$, as shown in [7, Thm. 1], which matches the PPIR capacity of theorem 10.

The key concepts of the capacity-achieving PPIR scheme are illustrated with the following example.

Example 9 *Consider the case where we have a number of $f = 20$ messages classified into $\Gamma = 3$ classes where the number of messages in each class are given by $[4, 6, 10]$ respectively. The f messages are replicated in $n = 2$ databases. Suppose that the user is interested in retrieving a message from class $\gamma = 3$.*

1)- Queries to databases: *First, the user selects a number $s \in [\delta]$, where $\delta \triangleq LCM(4, 6, 10) = 60$, uniformly at randomly and send this number to the n databases. Next, the user utilizes the achievable scheme in [7] to generate the query sets for privately retrieving one message from a set of Γ candidate messages $\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \boldsymbol{X}^{(3)}\}$ where $\boldsymbol{X}^{(\gamma)} = \{X_1^{(\gamma)}, X_2^{(\gamma)} \ldots, X_{\mathrm{L}}^{(\gamma)}\}$, for $\gamma \in [3]$.*

The achievable scheme in [7] requires the size of each message to be $\mathsf{L} = n^{\Gamma} = 8$ *and its query sets are constructed as follows. First, to make the symbols downloaded from each database appear random and independent from the desired message, the indices of the* L *symbols of each message are randomly permuted prior to the query construction. Let,* $U_i^{(\gamma)} = X_{\pi_\gamma(i)}^{(\gamma)}, \forall i \in [\mathsf{L}], \gamma \in [\Gamma]$*, where* $\pi_\gamma(\cdot)$ *is a uniform random permutation privately selected by the user independently for each candidate message. We simplify the notation by letting* $U_i^{(1)} = x_i$*,* $U_i^{(2)} = y_i$ *and* $U_i^{(3)} = z_i$ *for* $i \in [\mathsf{L}]$*. To retrieve a message from the desired class* $\gamma = 3$*, i.e., the candidate message* $\boldsymbol{z} = \{z_1, z_2, \ldots, z_8\}$*, symbols are queried from the two databases in a total of* $\tau = 3$ *rounds. This is shown in table 6.1(a) where the queries of round* τ *are indicated with* $Q_j^{(\gamma)}(\tau)$*.*

Initialization Round (*$\tau = 1$****):*** *The user first queries* $(n-1)^{\tau-1} = 1$ *distinct instance of* z_i *from each database. By message and index symmetries this also applies to* x_i *and* y_i*, resulting in total* $n\binom{\Gamma}{1}(n-1)^{(1-1)} = 6$ *symbols. The symbols queried in the first round are shown in the row indicated by* $Q_j^{(3)}(1)$ *in table 6.1-(a).*

Following Rounds (*$\tau \in [2:3]$****):*** *In each round and for each database, the user further queries sums of* τ *symbols with each symbol is from a different message. The queried sums either contain a single symbol from the desired message (so-called desired* τ*-sum) or only symbols from undesired messages (so-called undesired* τ*-sum, referred to as* side information*). One can see that by utilizing the undesired* τ*-sums obtained from the previous round, the desired message can be decoded. For example, in round* $\tau = 3$*, the desired symbol* z_7 *can be obtained by canceling the side information* $x_6 + y_5$ *which is obtained from the 2nd database in round* $\tau = 2$*. Similarly, one can verify the successful recovery of all symbols of the desired message* $\boldsymbol{z}$ *from the queried desired* τ*-sums shown in table 6.1-(a). Note that after deciding which desired sums to query, the undesired sums to query can be decided by enforcing message and index symmetry and the total number of symbols queried in round* τ *is equal to* $n\binom{\Gamma}{\tau}(n-1)^{(\tau-1)}$*. Finally, the queries are sent to each database* $j \in [2]$*.*

2)- Database answers: *Assume that the randomly selected number in Step 1) is given as* $s = 13$*. Accordingly, each database selects the same subset of candidate messages as follows:* $\boldsymbol{X}^{(1)} = \boldsymbol{W}^{(\theta_{1,\beta_{1,1}})}$ *,* $\boldsymbol{X}^{(2)} = \boldsymbol{W}^{(\theta_{2,\beta_{2,1}})}$ *, and* $\boldsymbol{X}^{(3)} = \boldsymbol{W}^{(\theta_{3,\beta_{3,1}})}$ *where* $\theta_{1,\beta_{1,1}} = \lceil 0.216 \times 4 \rceil = 1$*,* $\theta_{2,\beta_{2,1}} = \lceil 0.216 \times 6 \rceil + 4 = 6$*, and* $\theta_{3,\beta_{3,1}} = \lceil 0.216 \times 10 \rceil + 10 = 13$*, respectively. Using this mapping between the identity of the candidate*

Table 6.1 Query Sets for PPIR from Replication-based DSS

j	1	2
$Q_j^{(3)}(1)$	$x_1,\ y_1,\ z_1$	$x_2,\ y_2,\ z_2$
$Q_j^{(3)}(2)$	$x_4 + y_3$ $x_2 + z_3$ $y_2 + z_4$	$x_6 + y_5$ $x_1 + z_5$ $y_1 + z_6$
$Q_j^{(3)}(3)$	$x_6 + y_5 + z_7$	$x_4 + y_3 + z_8$

(a)

j	1	2
$Q_j^{(1)}(1)$	$x_1,\ y_1,\ z_1$	$x_2,\ y_2,\ z_2$
$Q_j^{(1)}(2)$	$x_3 + y_2$ $x_4 + z_2$ $y_4 + z_3$	$x_5 + y_1$ $x_6 + z_1$ $y_6 + z_5$
$Q_j^{(1)}(3)$	$x_7 + y_6 + z_5$	$x_8 + y_4 + z_3$

(b)

Note: The $n = 2$ databases store $f = 20$ messages classified into $\Gamma = 3$ classes. (a) shows the query sets for retrieving a message from desired class $\gamma = 3$ and (b) for $\gamma = 1$.

messages and the identity of the stored messages[14]*, each database then generates its answer string according to the queries of table 6.1-(a). In other words, the query for* x_i *is answered by each database with the symbol* $W_i^{(1)}$*, the query of* y_i *is answered with the symbol* $W_i^{(6)}$*, and query of* z_i *is answered with the symbol* $W_i^{(13)}$*.*

3)- Privacy and correctness of the retrieved message: *By decoding the downloaded symbols, we obtain the corresponding symbols of the message* $\boldsymbol{W}^{(13)}$ *which is indeed a message from the desired class* $\gamma = 3$*. Moreover, since the achievable scheme in [7] follows the symmetry principles, i.e., message, index, and database symmetries within the query sets of each database, the privacy is inherently ensured. Specifically, the achievable scheme in [7] guarantees the private retrieval of the message* $\boldsymbol{W}^{(13)}$ *among the set* $\{\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(6)}, \boldsymbol{W}^{(13)}\}$ *from the perspective of each database. With each message representing a class* $\gamma \in [\Gamma]$*, the desired class is also indistinguishable. For example, Table 6.1(b) illustrates the query sets for desired class* $\gamma = 1$*. From Tables 6.1(a) and 6.1(b) one can verify that the index mapping*

$$\textit{Databases 1:} \quad (1,2,3,4,5,6,7) \xrightarrow{\gamma=1} (1,4,2,3,6,7,5) \tag{6.19}$$

$$\textit{Databases 2:} \quad (1,2,3,4,5,6,8) \xrightarrow{\gamma=1} (6,2,4,8,1,5,3) \tag{6.20}$$

[14] *Note that, if we assume the user has knowledge of the size of each class, then* δ *is not needed. An achievable scheme is generated by first randomly selecting one message from each class to construct a set of* Γ *candidate messages. The mapping between the class index and the message index is made locally by the user and the queries are generated as PIR queries with the selected messages identities directly.*

converts the queries for $\gamma = 3$ to the queries for $\gamma = 1$. To see this mapping, compare $x_{i_1} + y_{i_2}$ and $x_{\hat{i}_1} + y_{\hat{i}_2}$ from the queries of the first database of Tables 6.1(a) and 6.1(b), respectively. It can be seen that the indices $i_1 = 4$ and $i_2 = 3$ of the queries for $\gamma = 3$ convert into the indices $\hat{i}_1 = 3$ and $\hat{i}_2 = 2$ of the queries for $\gamma = 1$, respectively. Thus, we have the mapping $((i_1, i_2) \to (\hat{i}_1, \hat{i}_2)) = ((4, 3) \to (3, 2))$. A similar comparison between the remaining queries results in the index and sign mapping of equations (6.19) *and* (6.20)*. One can similarly verify that there exists a mapping from the queries for $\gamma = 3$ to the queries for $\gamma = 2$, i.e., $Q^{(3)}_{[2]} \leftrightarrow Q^{(2)}_{[2]}$. Since a preliminary permutation over these indices, i.e., $\pi_\gamma(t)$ is uniformly and privately selected by the user independently of the desired class index γ, these queries are equally likely and indistinguishable.*

4)- Achievable Rate: *By counting the number of symbols to be downloaded as answer for the queries of table 6.1-(a), we obtain the PPIR rate $\mathsf{R} = \frac{8}{14} = \frac{4}{7} = \mathsf{C}_{PPIR}$.*

6.3 Multi-Message Pliable Private Information Retrieval

In this section, we consider the general problem of M-PPIR as presented in Section 6.1.1 with $\lambda \geq 1, \eta \geq 1$ and derive upper and lower bounds on the M-PPIR rate. Recall that in the M-PPIR problem, the user is *oblivious* to the structure of the database, i.e., has no knowledge of the messages membership in each class. Thus, we cannot directly utilize multi-message PIR solutions for the M-PPIR problem. To this end, again to provide a gentle introduction, we first consider the single server case, i.e., $n = 1$, characterize the capacity of single-server M-PPIR (see theorem 11), and present a capacity-achieving scheme. Then, we extend our results to replication-based DSSs, i.e., $n > 1$ and derive upper and lower bounds on the M-PPIR rate (see theorem 12). As mentioned in Section 6.2, the single-message PPIR problem is a special case of M-PPIR, thus, the results of Theorem 9 and Theorem 10 can be obtained by setting $\lambda = 1$ and $\eta = 1$ in the bounds derived in Theorem 12 below.

6.3.1 Single Server M-PPIR

Theorem 11 *For the M-PPIR problem with single server storing f messages classified into Γ classes, the maximum achievable M-PPIR rate over all possible M-PPIR protocols, i.e., the M-PPIR capacity $\mathsf{C}_{\text{M-PPIR}}$, for arbitrary λ number of messages from $\eta \in [\Gamma]$ desired classes is given by*

$$\mathsf{C}_{\text{M-PPIR}} = \frac{\eta}{\Gamma}.$$

Converse proof of Theorem 11 The converse proof for the single server M-PPIR follows intuitively from Theorem 9. We prove the outer bound for $\eta = 1$ and $M_\gamma = \lambda$, $\forall \gamma \in [\Gamma]$, i.e., each class contains exactly λ messages and $f = \lambda\Gamma$. For arbitrary f we then proceed to the case of arbitrary $M_\gamma \geq \lambda$, $\forall \gamma \in [\Gamma]$ and arbitrary $\eta \geq 1$.

- For $M_\gamma = \lambda$, $\forall \gamma \in [\Gamma]$, and $\eta = 1$ we have $f = \lambda\Gamma$. Accordingly, we can consider the λ messages in each class as a one *super* message of length $\lambda\mathsf{L}$, for arbitrary L. Thus, in order to maintain the privacy of the desired class, we must maintain the privacy of the retrieved *super* message. To this end, it is well-known that information-theoretic privacy for a single-server can only be achieved by downloading all content of the database [2], hence $\mathsf{C}_{\text{M-PPIR}} = \frac{\lambda\mathsf{L}}{f\mathsf{L}} = \frac{\lambda\mathsf{L}}{\lambda\Gamma\mathsf{L}} = \frac{1}{\Gamma}$.
- Finally, for arbitrary $\eta \geq 1$ and $M_\gamma \geq \lambda$, $\forall \gamma \in [\Gamma]$, in order to maintain the privacy of a set of η desired classes and obtain λ messages from each desired class $\gamma_i \in \Omega$, we must download *at least* one *super* message from each possible class. Accordingly, the probability that any subset of η classes is the desired subset is equally likely, i.e., achieving perfect information theoretic privacy. Since there are more than λ messages in each class and the user requests *any* λ messages from each of the desired classes, the identities of the selected λ messages to form the Γ *super* messages are not relevant. Accordingly, by randomly selecting one *super* message from each class as an answer to the user it follows that the best information retrieval rate, i.e., the M-PPIR capacity, must be upper bounded by $\mathsf{C}_{\text{M-PPIR}} = \frac{\eta\lambda\mathsf{L}}{\lambda\Gamma\mathsf{L}} = \frac{\eta}{\Gamma}$.

Achievability of Theorem 11 For the achievability of Theorem 11 we extend our solution for the single-server PPIR from Section 6.2.1 to the case of multiple messages from multiple classes. To privately retrieve λ messages from the desired set of classes Ω, let $\delta = \text{LCM}(M_1, \ldots, M_\Gamma)$. The user selects a number $s \in [\delta]$ uniformly at random and sends it to the database. Based on the selected random number, a subset of size

$\lambda\Gamma$ from the messages is selected by the database, λ messages from each class. The identity of the selected λ messages from each class is computed as follows:

- The first message from each class is given by

$$\theta_{\gamma_i,\beta_{\gamma_i,1}} = \left\lceil \frac{s}{\delta} M_{\gamma_i} \right\rceil + \sum_{l=1}^{\gamma_i - 1} M_l, \tag{6.21}$$

  where $\beta_{\gamma_i,1} = \lceil \frac{s}{\delta} M_{\gamma_i} \rceil$ and $\theta_{\gamma_i,\beta_{\gamma_i,1}}$ follows due to the fact that the messages are ordered in an ascending order based on their class membership as outlined in Section 6.1.1.

- The following $\lambda - 1$ messages from each class are selected, without loss of generality, in a cyclic order over the members of the class starting with $\theta_{\gamma_i,\beta_{\gamma_i,2}} = \theta_{\gamma_i,\beta_{\gamma_i,1}} + 1$. That is, for any $k \in [\lambda]$,

$$\theta_{\gamma_i,\beta_{\gamma_i,k+1}} = \begin{cases} \theta_{\gamma_i,\beta_{\gamma_i,k}} - M_{\gamma_i} + 1 & \text{if } \theta_{\gamma_i,\beta_{\gamma_i,k}} = \sum_{l=1}^{\gamma_i} M_l \\ \theta_{\gamma_i,\beta_{\gamma_i,k}} + 1 & \text{otherwise.} \end{cases} \tag{6.22}$$

Finally, the set of candidate $\lambda\Gamma$ messages $\{\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,\lambda}})}, \ldots \ldots, \boldsymbol{W}^{(\theta_{\gamma_\Gamma,\beta_{\gamma_\Gamma,\lambda}})}\}$ are set as the answer to the user. Note that the query is fixed independently of the desired set of classes. Consequently, from the perspective of the database, the query and answer string of any set $\Omega \in \mathfrak{S}$ of desired classes are indistinguishable and the privacy constraint of equation (6.10) is satisfied. Moreover, since the user obtains $\lambda\eta$ messages $\boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_1,\beta_{\gamma_1,\lambda}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\gamma_\eta,1}})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma_\eta,\beta_{\gamma_\eta,\lambda}})}$ from the desired classes $\gamma_i \in \Omega \subseteq [\Gamma]$ the correctness constraint of equation (6.11) is also satisfied. As a result, the M-PPIR rate $\mathsf{R} = \frac{\lambda\eta\mathsf{L}}{\lambda\Gamma\mathsf{L}} = \frac{\eta}{\Gamma}$ is achieved.

Next, we extend the single server result to the M-PPIR over replicated DSS setup with theorem 12 as follows.

6.3.2 M-PPIR over Replicated DSS

Theorem 12 *Consider a DSS with n noncolluding replicated databases storing f messages classified into Γ classes. For the M-PPIR problem with $\lambda \geq 1$ and $\eta \geq 1$,*

the maximum achievable M-PPIR rate over all possible M-PPIR protocols, i.e., the M-PPIR capacity $\mathsf{C}_{\text{M-PPIR}}$*, is as*

$$\underline{\mathsf{R}} \leq \mathsf{C}_{\text{M-PPIR}} \leq \bar{\mathsf{R}}$$

where

$$\begin{cases} \bar{\mathsf{R}} = \underline{\mathsf{R}} = \left[1 + \dfrac{\Gamma - \eta}{n\eta}\right]^{-1} & \text{if } \eta \geq \frac{\Gamma}{2}, \qquad (6.23a) \\ \bar{\mathsf{R}} = \left[\dfrac{1 - (\frac{1}{n})^{\lfloor \frac{\Gamma}{\eta} \rfloor}}{1 - \frac{1}{n}} + \left(\frac{\Gamma}{\eta} - \lfloor \frac{\Gamma}{\eta} \rfloor\right)(\frac{1}{n})^{\lfloor \frac{\Gamma}{\eta} \rfloor}\right]^{-1} & \text{if } \eta \leq \frac{\Gamma}{2} \qquad (6.23b) \\ \underline{\mathsf{R}} = \dfrac{\sum_{i=1}^{\eta} \tau_i \kappa_i^{\Gamma-\eta}\left[\left(1 + \frac{1}{\kappa_i}\right)^{\Gamma} - \left(1 + \frac{1}{\kappa_i}\right)^{\Gamma-\eta}\right]}{\sum_{i=1}^{\eta} \tau_i \kappa_i^{\Gamma-\eta}\left[\left(1 + \frac{1}{\kappa_i}\right)^{\Gamma} - 1\right]} & \text{if } \eta \leq \frac{\Gamma}{2}, \qquad (6.23c) \end{cases}$$

for $\kappa_i \triangleq \frac{e^{j2\pi(i-1)/\eta}}{n^{(1/\eta)} - e^{j2\pi(i-1)/\eta}}$*,* τ_i*,* $i \in [\eta]$ *is the solution of the* η *linear equations*

$$\sum_{i=1}^{\eta} \tau_i \kappa_i^{-\eta} = (n-1)^{\Gamma-\eta}$$

$$\sum_{i=1}^{\eta} \tau_i \kappa_i^{-k} = 0 \quad \text{for } k \in [\eta - 1].$$

The converse bounds of Theorem 12 are derived in Section 6.3.2 and Section 6.3.2, respectively. The achievability lower bounds in Theorem 12 are shown in Section 6.3.2. The following corollary states that if $\frac{\Gamma}{\eta} \in \mathbb{N}$, i.e., the number of classes is divisible by the number of desired classes, then the achievability bound of equation (6.23c) matches the upper bound of equation (6.23b).

Corollary 6 *For the M-PPIR problem from* $n > 1$ *noncolluding replicated databases where* $\eta \leq \frac{\Gamma}{2}$*,* $\frac{\Gamma}{\eta} \in \mathbb{N}$*, the derived upper bound of equation* (6.23b) *is tight, i.e., matches the lower bound of equation* (6.23c)*, and the M-PPIR capacity is given by*

$$\mathsf{C}_{\text{M-PPIR}} = \left(1 - \frac{1}{n}\right)\left[1 - \left(\frac{1}{n}\right)^{\frac{\Gamma}{\eta}}\right]^{-1}. \qquad (6.24)$$

Remark 8 *Theorem 12 and Corollary 6 yield a simple yet powerful observation. One can observe that privately retrieving multiple messages $\lambda > 1$ from* multiple *desired classes $\eta > 1$, while keeping the identity of the desired classes indices hidden from each database, imposes no penalty on the download rate compared to privately retrieving only* one *message from each of the desired classes. That can be seen from the independence of the bounds of theorem 12 of λ. Moreover, the presented bounds match the MPIR rates for the case where the user is interested in privately retrieving η messages from a dataset consisting of Γ messages, i.e., each class contains only one message, [9, Thm. 1,Thm. 2, Cor. 3].*

In the following, we first derive an upper bound for the M-PPIR problem by adapting the classical PIR converse proofs of [7], [9] to our pliable setup. We now proceed with the proving the upper bound on the capacity of M-PPIR.

Converse proof of Theorem 12 for $\eta \geq \frac{\Gamma}{2}$ Here, since $\eta \geq \frac{\Gamma}{2}$, for any $\Omega, \Omega' \in \mathfrak{S}$, such that $\Omega \neq \Omega'$, we have $\Omega \cap \Omega' \neq \phi$. In other words, there is always some overlap between the possible sets of desired classes. As a result of this overlap we have the following lemma.

Lemma 11 *For the M-PPIR problem with $\eta \geq \frac{\Gamma}{2}$, the following bound holds*

$$\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{[\eta]} A_{[n]}^{[\eta]} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \geq \frac{\lambda \mathsf{L}}{n}(\Gamma - \eta). \tag{6.25}$$

Moreover, equation (6.25) *holds for any set $\Omega \in \mathfrak{S}$, i.e.,*

$$\mathrm{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{\Omega,[\lambda]}}; Q_{[n]}^{\Omega} A_{[n]}^{\Omega} \,\big|\, \boldsymbol{W}^{\theta_{\Omega,[\lambda]}}\big) \geq \frac{\lambda \mathsf{L}}{n}(\Gamma - \eta). \tag{6.26}$$

The proof of Lemma 11 follows similar steps as the ones for Lemma 10 and can be found in Appendix H. Now, we are ready to prove the converse for the case $\eta \geq \frac{\Gamma}{2}$.

Proof: By combining Lemma 9 and Lemma 11, we have

$$\eta\lambda \mathsf{L}\left(\frac{1}{\mathsf{R}} - 1\right) \geq \frac{\lambda \mathsf{L}}{n}(\Gamma - \eta), \tag{6.27}$$

and by eliminating $\lambda \mathsf{L}$, we obtain

$$\mathsf{R} \leq \left[1 + \frac{\Gamma - \eta}{n\eta}\right]^{-1}. \tag{6.28}$$

That proves the upper bound on the M-PPIR capacity for $\eta \geq \frac{\Gamma}{2}$ as given in equation (6.23a). ■

Converse proof of Theorem 12 for $\eta \leq \frac{\Gamma}{2}$ *Proof:* Let $\Omega_1 = [\eta]$, $\Omega_i = [\eta(i-1)+1 : \eta(i)]$ for $i \in [2:\rho]$ and $\rho = \lfloor \frac{\Gamma}{\eta} \rfloor$. Let $\Omega_{\rho'} = [\Gamma - \eta + 1 : \Gamma]$. We have $\bigcap_{i=1}^{\rho} \Omega_i = \phi$ and $\Omega_\rho \cap \Omega_{\rho'} = [\Gamma - \eta + 1 : \eta \lfloor \frac{\Gamma}{\eta} \rfloor]$. Starting by $\Omega_1 = [\eta]$, then applying Lemma 10 repeatedly we have

$$\begin{aligned}
&\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{\Omega_1} A_{[n]}^{\Omega_1} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\geq \frac{\eta\lambda\mathsf{L}}{n} + \frac{1}{n}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[2\eta],[\lambda]}}; Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[2\eta],[\lambda]}}\Big) \\
&\geq \frac{\eta\lambda\mathsf{L}}{n} + \frac{1}{n}\left[\frac{\eta\lambda\mathsf{L}}{n} + \frac{1}{n}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[3\eta],[\lambda]}}; Q_{[n]}^{\Omega_3} A_{[n]}^{\Omega_3} \,\Big|\, \boldsymbol{W}^{\theta_{[3\eta],[\lambda]}}\Big)\right] \\
&= \frac{\eta\lambda\mathsf{L}}{n} + \frac{\eta\lambda\mathsf{L}}{n^2} + \frac{1}{n^2}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[3\eta],[\lambda]}}; Q_{[n]}^{\Omega_3} A_{[n]}^{\Omega_3} \,\Big|\, \boldsymbol{W}^{\theta_{[3\eta],[\lambda]}}\Big) \\
&\geq \quad \vdots \\
&\geq \frac{\eta\lambda\mathsf{L}}{n} + \cdots + \frac{\eta\lambda\mathsf{L}}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-2}} + \frac{1}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-2}}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta\lfloor\frac{\Gamma}{\eta}\rfloor-1],[\lambda]}}; Q_{[n]}^{\Omega_{\rho-1}} A_{[n]}^{\Omega_{\rho-1}} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta\lfloor\frac{\Gamma}{\eta}\rfloor-1],[\lambda]}}\Big) \\
&\geq \frac{\eta\lambda\mathsf{L}}{n} + \cdots + \frac{\eta\lambda\mathsf{L}}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-2}} + \frac{\eta\lambda\mathsf{L}}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-1}} + \frac{1}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-1}}\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta\lfloor\frac{\Gamma}{\eta}\rfloor],[\lambda]}}; Q_{[n]}^{\Omega_\rho} A_{[n]}^{\Omega_\rho} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta\lfloor\frac{\Gamma}{\eta}\rfloor],[\lambda]}}\Big) \\
&\geq \frac{\eta\lambda\mathsf{L}}{n} + \cdots + \frac{\eta\lambda\mathsf{L}}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-2}} + \frac{\eta\lambda\mathsf{L}}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-1}} + \frac{1}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor}}\Big[\lambda\mathsf{L}\big(\Gamma - \eta\lfloor\tfrac{\Gamma}{\eta}\rfloor\big)\Big]
\end{aligned} \tag{6.29}$$

where equation (6.29) results from bounding the last mutual information term, similar to Lemma 11, as follows

$$n\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta\lfloor\frac{\Gamma}{\eta}\rfloor],[\lambda]}}; Q_{[n]}^{\Omega_\rho} A_{[n]}^{\Omega_\rho} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta\lfloor\frac{\Gamma}{\eta}\rfloor],[\lambda]}}\Big) \geq \lambda\mathsf{L}\Big(\Gamma - \eta\lfloor\tfrac{\Gamma}{\eta}\rfloor\Big).$$

Now, combining equation (6.29) and Lemma 9 yields

$$\eta\lambda\mathsf{L}\left(\frac{1}{\mathsf{R}} - 1\right) \geq \eta\lambda\mathsf{L}\left(\frac{1}{n} + \cdots + \frac{1}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-2}} + \frac{1}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor-1}} + \frac{1}{n^{\lfloor\frac{\Gamma}{\eta}\rfloor}}\Big[\tfrac{\Gamma}{\eta} - \lfloor\tfrac{\Gamma}{\eta}\rfloor\Big]\right). \tag{6.30}$$

Eliminating $\eta\lambda\mathsf{L}$ from both sides, we obtain

$$\mathsf{R} \leq \left(1 + \frac{1}{n} + \frac{1}{n^2} + \cdots + \frac{1}{n^{\lfloor \frac{\Gamma}{\eta} \rfloor - 1}} + \frac{1}{n^{\lfloor \frac{\Gamma}{\eta} \rfloor}} \left[\frac{\Gamma}{\eta} - \left\lfloor \frac{\Gamma}{\eta} \right\rfloor\right]\right)^{-1} \tag{6.31}$$

$$= \left[\frac{1 - (\frac{1}{n})^{\lfloor \frac{\Gamma}{\eta} \rfloor}}{1 - \frac{1}{n}} + \frac{\frac{\Gamma}{\eta} - \lfloor \frac{\Gamma}{\eta} \rfloor}{n^{\lfloor \frac{\Gamma}{\eta} \rfloor}}\right]^{-1}, \tag{6.32}$$

which proves the upper bound on the M-PPIR capacity for the case $\eta \leq \frac{\Gamma}{2}$ as given in equation (6.23b). ■

Achievability of Theorem 12 The M-PPIR schemes needed for Theorem 12 utilize the single-message and multi-message PIR solutions of [7], [9]. If we only consider retrieving a single-message from multiple desired classes, i.e., $\lambda = 1$ and $\eta \geq 1$, we can directly adapt the multi-message scheme of [9], similarly to the approach for PPIR in Section 6.2.2. In the following, we outline the required steps for this adaptation with the extension to multiple desired classes $\eta \geq 1$.

The achievable rate of M-PIR problem with n noncolluding replicated databases, each storing f messages, and $\lambda\eta$ desired messages to download is characterized in [9, Thm. 1, Thm. 2], as

$$\underline{\mathsf{R}} = \begin{cases} \left[1 + \dfrac{f-\mu}{n\mu}\right]^{-1} & \text{if } \mu \geq \frac{f}{2}, \qquad (6.33a) \\ \dfrac{\sum_{i=1}^{\mu} \tau_i \kappa_i^{f-\mu} \left[\left(1 + \frac{1}{\kappa_i}\right)^f - \left(1 + \frac{1}{\kappa_i}\right)^{f-\mu}\right]}{\sum_{i=1}^{\mu} \tau_i \kappa_i^{f-\mu} \left[\left(1 + \frac{1}{\kappa_i}\right)^f - 1\right]} & \text{if } \mu \leq \frac{f}{2}, \qquad (6.33b) \end{cases}$$

where $\kappa_i \triangleq \frac{e^{j2\pi(i-1)/\mu}}{n^{(1/\mu)} - e^{j2\pi(i-1)/\mu}}$, τ_i, $i \in [\mu]$, is the solution of the linear equations

$$\sum_{i=1}^{\lambda\eta} \tau_i \kappa_i^{-\lambda\eta} = (n-1)^{f-\lambda\eta}$$

$$\sum_{i=1}^{\lambda\eta} \tau_i \kappa_i^{-k} = 0 \text{ for } k \in [\lambda\eta - 1].$$

From comparing the upper bounds of M-PPIR of Theorem 12 in equations (6.23a)-(6.23b) with equations (6.33a)-(6.33b), we can observe that M-PPIR effectively reduces the size of the database from f to Γ messages and the number of desired messages from $\lambda\eta$ to simply η. Thus, our achievable M-PPIR schemes extends the single server M-PPIR solution of section 6.3.1 to a replicated DSS setup through adaptation of the M-PIR scheme achievable scheme in [9] for $\eta \geq \frac{\Gamma}{2}$ and $\eta \leq \frac{\Gamma}{2}$, respectively.

Given Γ, η, λ, n, $\Omega \in \mathfrak{S}$, and $\delta = \text{LCM}(M_1, \ldots, M_\Gamma)$, the high-level implementation of the M-PPIR scheme is outlined with the following steps.

1. The user selects a number uniformly at random from the set $[\delta]$.
2. If $\eta \geq \frac{\Gamma}{2}$, the user constructs the queries according to the achievable M-PIR scheme in [9, Sec.IV] for n noncolluding replicated databases storing Γ candidate *super* messages $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(\Gamma)}\}$ to privately retrieve η messages $\mathbf{X}^{(\gamma_i)}, \forall \gamma_i \in \Omega$, i.e., $Q_1^\Omega, \ldots, Q_n^\Omega$. Each *super* message is assumed to be of length $\hat{\mathsf{L}} = n^2$ super symbols, i.e., $\mathbf{X}^{(\gamma)} = (\boldsymbol{X}_1^{(\gamma)}, \ldots, \boldsymbol{X}_{\hat{\mathsf{L}}}^{(\gamma)})$ and the super symbol $\boldsymbol{X}_l^{(\gamma)} = (X_1^{(\gamma)}, \ldots, X_\lambda^{(\gamma)})$ corresponds to a vector of symbols from the λ messages of class $\gamma \in [\Gamma]$.
3. The user sends the selected random number from Step 1 $s \in [\delta]$, then the constructed queries $Q_1^\Omega, \ldots, Q_n^\Omega$, in a random order to each database $j \in [n]$.
4. Given the random number $s \in [\delta]$, each database $j \in [n]$ computes the indices of $\lambda\Gamma$ messages, λ messages from each class. These indices are computed as follows:
   - The first message from each class is given by equation (6.21).
   - The following $\lambda - 1$ messages from each class are selected in a cyclic order over the members of the class according to equation (6.22).
5. Super messages are assembled in each database using the selected $\lambda\Gamma$ messages of the previous step to be used in constructing its answer string A_j^Ω as follows. Each of Γ *super* messages are mapped to the user's queries of Step 2 as $\boldsymbol{X}_l^{(\gamma)} = (W_l^{(\theta_{\gamma,\beta_{\gamma,1}})} W_l^{(\theta_{\gamma,\beta_{\gamma,2}})}, \ldots, W_l^{(\theta_{\gamma,\beta_{\gamma,\lambda}})})$ for all $\gamma \in [\Gamma]$ and $l \in \hat{\mathsf{L}}$. Note that any operation involving a super symbol is performed element wise.
6. If $\eta \leq \frac{\Gamma}{2}$, repeat steps 1-5 by constructing the queries of according to the achievable M-PIR scheme in [9, Sec.V] for n noncolluding replicated database

storing Γ candidate *super* messages of length

$$\hat{\mathsf{L}} = \frac{1}{\eta}\sum_{i=1}^{\eta}\tau_i\kappa_i^{\Gamma-\eta}\left[\left(1+\frac{1}{\kappa_i}\right)^{\Gamma}-\left(1+\frac{1}{\kappa_i}\right)^{\Gamma-\eta}\right],$$

where $\kappa_i \triangleq \frac{e^{j2\pi(i-1)/\eta}}{n^{(1/\eta)}-e^{j2\pi(i-1)/\eta}}$, τ_i, $i \in [\eta]$, is the solution of the two linear equations

$$\sum_{i=1}^{\eta}\tau_i\kappa_i^{-\eta} = (n-1)^{\Gamma-\eta}$$
$$\sum_{i=1}^{\eta}\tau_i\kappa_i^{-k} = 0 \text{ for } k \in [\eta-1].$$

Privacy and Correctness: The arguments of privacy and correctness follow from the underlying guarantees of the M-PIR solutions of [9, Sec.IV] and [9, Sec.V], similarly to the capacity achieving scheme of PPIR in Section 6.2.2.

Calculation of achievable rate: The achievable rates follow directly from equations (6.33a) and (6.33b) by substituting f with Γ and $\lambda\eta$ with η, respectively.

6.4 Conclusion

In this chapter, we formulated the problem of multi-message pliable private information retrieval (M-PPIR) from noncolluding replicated database as a new variant of the classical PIR problem. In M-PPIR, f messages are replicated in n noncolluding databases and classified into Γ classes. The user wishes to retrieve *any* $\lambda \geq 1$ messages from *multiple* desired classes while revealing no information about the identity of the desired classes to the databases. From this general problem, we considered the special case of (single-message) PPIR where the user is interested in retrieving only one message from one desired class. We characterized the capacity of PPIR from replicated noncolluding databases for arbitrary number of databases $n \geq 1$ and presented capacity-achieving schemes. Interestingly, the capacity of PPIR from n noncolluding databases matches the capacity of PIR with n databases and Γ messages. Thus, enabling flexibility, i.e., pliability, in private information retrieval allowed to

trade-off privacy versus download rate compared to classical information-theoretic PIR schemes. Finally, we extended our results to the general M-PPIR problem, derived upper and lower bounds on the M-PPIR rate, and showed a similar insight, i.e., the derived M-PPIR bounds match the multi-message PIR bounds.

CHAPTER 7

SUMMARY

In this dissertation, we first considered the problem of private computation (PC) as generalization of the classical private information retrieval (PIR) problem. For PC, three variations were studied. Namely, private linear computation (PLC) from linearly-coded DSS, private polynomial computations (PPC) from Reed-Solomon coded DSS with Langrange encoding, and PC of nonlinear functions from replicated DSS. In the second part of this dissertation, we considered a relaxation of PIR problem denoted as pliable private information (PPIR).

We have provided the capacity of PLC from coded DSSs, where data is encoded and stored using an arbitrary linear code from the class of MDS-PIR capacity-achieving codes. Interestingly, the capacity of PLC is equal to the corresponding MDS-PIR capacity. Thus, privately retrieving arbitrary linear combinations of the stored messages does not incur any overhead in rate compared to retrieving a single message from the databases.

For the PPC problem, we have presented two PPC schemes for RS-coded DSSs with Lagrange encoding showing improved computation rates compared to the best known PPC schemes from the literature when the number of messages is small. Asymptotically, as the number of messages tends to infinity, the rate of our RS-coded nonsystematic PPC scheme approaches the rate of the best known nonsystematic PPC scheme. However, for systematically RS-coded DSSs, our scheme significantly outperforms all known PPC schemes up to some specific storage code rate that depends on the maximum degree of the candidate polynomials. Finally, a general converse bound on the PPC rate was derived and compared to the achievable rates of the proposed schemes with some numerical results. The numerical results depicted a gap between the derived converse bound and the achievable rates of the proposed

schemes and the best known PPC schemes from literature. Naturally, this gap raises two promising open problems. One is to prove that the converse of Theorem 4 is tight, and the other is to find schemes that exploit the nonlinear dependencies between the candidate functions evaluations. Both problems are valuable research directions for future work.

For PC of nonlinear computation from noncolluding replicated databases, we presented a novel PC scheme and showed that the resulting PC rate equals the PC capacity as the message size grows for the case when the candidate functions are the independent messages and one arbitrary nonlinear function of these. Moreover, the PC rate approaches an outer bound on the PC capacity and thus becomes the capacity itself when the number of messages grows. Finally, we compared the outer bound and the achievable rate for the special case of PMC. Similar to the PPC scheme, the numerical result depicted a gap between the derived converse bound and the achievable rate. Closing this gap is an interesting direction for future work.

Finally, the insights provided by the results of M-PPIR problem motivates further exploration of practical setups. One direction for future research would consider the case where the number of desired messages from each class is *not* fixed. Another direction could be for the case where the user has some side-information in the form of messages from the dataset and wishes to retrieve any other set of messages from multiple desired classes. This direction is motivated by the close connection between PIR with side information (PIR-SI) and index coding (IC). More specifically, PIR-SI is related to private broadcasting [82] and blind index coding (BIC) [83], where the side-information is considered to be unknown to the server. We keep the question whither a similar connection can be established between PPIR and PICOD for future works.

APPENDIX A

PROOF OF LEMMA 1

In this appendix, we prove the independence of answers from k databases forming an information set as given in Lemma 1 of Chapter 3. The proof of Lemma 1 uses the linear independence of the columns of a generator matrix of $\mathscr{C}$ corresponding to an information set. Consider an information set $\mathcal{I}$ of the $[n, k]$ linear storage code $\mathscr{C}$, $|\mathcal{I}| = k$. The content of the databases indexed by $\mathcal{I}$, i.e., $\Big(\boldsymbol{C}_j,\ j \in \mathcal{I}\Big) = \mathbf{C}|_{\mathcal{I}}$, can be written as

$$\big((\mathbf{W}^{(1)})^{\mathsf{T}}|\ldots|(\mathbf{W}^{(f)})^{\mathsf{T}}\big)^{\mathsf{T}}\mathsf{G}^{\mathscr{C}}|_{\mathcal{I}} \sim \big((\mathbf{W}^{(1)})^{\mathsf{T}}|\ldots|(\mathbf{W}^{(f)})^{\mathsf{T}}\big)^{\mathsf{T}}, \tag{A.1}$$

where by the construction of any $[n, k]$ linear storage code, if $\mathcal{I}$ is an information set of the code $\mathscr{C}$, then $\mathsf{G}^{\mathscr{C}}|_{\mathcal{I}}$ is a $k \times k$ invertible matrix. (a) follows from [16, Lem. 1] and the fact that the messages are chosen independently and uniformly at random from $\mathbb{F}_p^{\beta\times k}$. Therefore, the content of any k databases forming an information set is statistically equivalent to the stored messages. Given that the symbols of the messages are independent, then the k columns of $\mathbf{C}|_{\mathcal{I}}$ are also statistically independent. Finally, since $A_j^{(v)}$, $j \in \mathcal{I}$, are deterministic functions (that are composed of the μ candidate linear combinations) of independent random variables $\{\boldsymbol{C}_j \colon j \in \mathcal{I}\}$ and $\mathcal{Q}$, $\{A_j^{(v)},\ j \in \mathcal{I}\}$ are statistically independent, and equation (3.4) follows.

Now, given that the candidate linear functions are evaluated element-wise over independent and uniformly distributed symbols, the symbols of each linear combination are also independent and identically distributed (i.i.d.), i.e., for $\mathbf{X}^{(v)} = \big(X_1^{(v)}, \ldots, X_{\mathsf{L}}^{(v)}\big)$, $X_1^{(v)}, \ldots, X_{\mathsf{L}}^{(v)}$ are i.i.d. according to a prototype random variable $X^{(v)}$.

Moreover, due to the commutative property of linear functions, linear computation over linearly-encoded symbols is equivalent to linear encoding of the linear

function evaluations. As a result, we can extend the argument of statistical equivalence to linear function evaluations over coded symbols. In other words, the evaluations of linear functions over the content of any k databases, forming an information set $\mathcal{I}$, are statistically equivalent to the evaluations of linear functions over the stored messages. Presenting $\mathbf{X}^{(v)} = \left(X_1^{(v)}, \ldots, X_{\mathsf{L}}^{(v)}\right)$ in the form $\mathbf{X}^{(v)} = \left(X_{i,j}^{(v)}\right)$, $i \in [\beta]$, $j \in [k]$, we have

$$\left((\mathbf{X}^{(1)})^\intercal | \ldots | (\mathbf{X}^{(\mu)})^\intercal\right)^\intercal \mathsf{G}^{\mathscr{C}}|_{\mathcal{I}} \sim \left((\mathbf{X}^{(1)})^\intercal | \ldots | (\mathbf{X}^{(\mu)})^\intercal\right)^\intercal. \tag{A.2}$$

Finally, since we can consider that the storage encoding is applied on individual function evaluations, conditioning on any subset of function evaluations $\mathbf{X}^{\mathcal{V}}$, $|\mathcal{V}| = \mu_v$, is equivalent to reducing the problem to the private computation of $\mu - \mu_v$ linear combinations. That is, $A_j^{(v)}$, $j \in \mathcal{I}$, are deterministic functions (that are composed of the $\mu - \mu_v$ candidate linear combinations) of independent random variables $\{\boldsymbol{C}_j : j \in \mathcal{I}\}$ and $\mathcal{Q}$, and $\{A_j^{(v)},\ j \in \mathcal{I}\}$ are still statistically independent. Hence, the statistical independence argument of equation (3.5) follows.

APPENDIX B

PROOF OF LEMMA 3

In this appendix, we prove Lemma 3 as presented in Section 3.2. Since each linear function $\mathbf{X}^{(v)} = \big(X_1^{(v)}, \ldots, X_{\mathsf{L}}^{(v)}\big)$, $v \in [\mu]$, contains L i.i.d. symbols, it is clear that $\forall\, l \in [\mathsf{L}]$,

$$\mathsf{H}\big(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}\big) = \mathsf{L}\,\mathsf{H}\big(X_l^{(1)}, \ldots, X_l^{(\mu)}\big), \text{ and} \tag{B.1}$$

$$\mathsf{H}\big(\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}\big) = \mathsf{L}\,\mathsf{H}\big(W_l^{(1)}, \ldots, W_l^{(f)}\big). \tag{B.2}$$

Let $\mathcal{J} \triangleq \{j_1, \ldots, j_h\}$ for some $h \in [r]$. We have

$$\begin{aligned}
&\Pr\Big[X_l^{(i_1)}, \ldots, X_l^{(i_h)}\Big] \\
=&\sum_{w_l^{\mathcal{J}^{\mathrm{c}}}} \Pr\big[W_l^{\mathcal{J}^{\mathrm{c}}} = w_l^{\mathcal{J}^{\mathrm{c}}}\big] \cdot \Pr\Big[X_l^{(i_1)}, \ldots, X_l^{(i_h)} \Big| W_l^{\mathcal{J}^{\mathrm{c}}} = w_l^{\mathcal{J}^{\mathrm{c}}}\Big] \\
=&\sum_{w_l^{\mathcal{J}^{\mathrm{c}}}} \Pr\big[W_l^{\mathcal{J}^{\mathrm{c}}} = w_l^{\mathcal{J}^{\mathrm{c}}}\big] \cdot \Pr\Big[W_l^{(j_1)}, \ldots, W_l^{(j_h)} \Big| W_l^{\mathcal{J}^{\mathrm{c}}} = w_l^{\mathcal{J}^{\mathrm{c}}}\Big] &\text{(B.3)} \\
=&\sum_{w^{\mathcal{H}^{\mathrm{c}}}} \Pr\big[W_l^{\mathcal{J}^{\mathrm{c}}} = w_l^{\mathcal{J}^{\mathrm{c}}}\big] \Big(\frac{1}{p}\Big)^h = \Big(\frac{1}{p}\Big)^h, &\text{(B.4)}
\end{aligned}$$

where equation (B.3) follows from the fact that there is a linear transformation between $X_l^{(i_1)}, \ldots, X_l^{(i_h)}$ and $W_l^{(j_1)}, \ldots, W_l^{(j_h)}$, and equation (B.4) holds since $W_l^{(j_1)}, \ldots, W_l^{(j_h)}$ are i.i.d. over $\mathbb{F}_p$. Hence, $\mathsf{H}\big(X_l^{(i_1)}, \ldots, X_l^{(i_h)}\big) = h$ (in p-ary units), which completes the proof.

APPENDIX C

PROOF OF LEMMA 4

Here we present the main components needed for the proof of Lemma 4 as presented in Section 3.3.3, however the detailed derivations are a direct application of the proof of [43, Lem. 1, Sec. V-B] and thus are omitted. The proof of [43, Lem. 1] is adapted to our setup with the following substitutions.

Let $\mathcal{L} \triangleq \{\ell_1, \ldots, \ell_r\} \subseteq [\mu]$ be the set of candidate linear combination indices corresponding to a basis of the row space of the linear combination coefficient matrix $\mathsf{V}_{\mu \times f}$, where $r = \text{rank}(\mathsf{V}) \leq \min\{\mu, f\}$. Then $X_l^{(\ell_1)}, \ldots, X_l^{(\ell_r)}$ satisfy $\mathsf{H}\big(X_l^{(\ell_1)}, \ldots, X_l^{(\ell_r)}\big) = \mathsf{H}\big(X_l^{[\mu]}\big)$, $\forall\, l \in [\mathsf{L}]$. Assume, without loss of generality, that the rows of the coefficient matrix are ordered such that the first r rows constitute the row basis, i.e., $(X_l^{(1)}, \ldots, X_l^{(r)}) = (X_l^{(\ell_1)}, \ldots, X_l^{(\ell_r)})$, $l \in [\mathsf{L}]$. Note that we can represent the candidate functions evaluations $\big(X_l^{(1)}, \ldots, X_l^{(\mu)}\big)$ in terms of the basis candidate functions evaluations $\big(X_l^{(\ell_1)}, \ldots, X_l^{(\ell_r)}\big)$ for $l \in [\mathsf{L}]$ with a deterministic linear mapping $\hat{\mathsf{V}}_{\mu \times r}$ of size $\mu \times r$ as $\big(X_l^{(1)}, \ldots, X_l^{(\mu)}\big)^{\mathsf{T}} = \hat{\mathsf{V}}_{\mu \times r}\big(X_l^{(\ell_1)}, \ldots, X_l^{(\ell_r)}\big)^{\mathsf{T}}$. As a result, we have $(\hat{\boldsymbol{v}}_1^{\mathsf{T}}, \ldots, \hat{\boldsymbol{v}}_r^{\mathsf{T}})^{\mathsf{T}} = \mathsf{I}_r$, where I_r is the $r \times r$ identity matrix and $\hat{\boldsymbol{v}}_i$ is the i-th row vector of the deterministic linear mapping matrix $\hat{\mathsf{V}}_{\mu \times r}$.

First, consider the case where the desired function evaluation index $v = 1$. Consider the queries corresponding to undesired τ-sums, i.e., τ-sums that do not involve any symbols from the desired function evaluation $\mathbf{U}^{(1)}$. There are $\binom{\mu-1}{\tau}$ different τ-sum types corresponding to such queries which can be divided into two groups as follows.

- Group 1: $\binom{\mu-1}{\tau} - \binom{\mu-r}{\tau}$ τ-sum types for which the corresponding τ-sums involve at least one element from the set $\{\mathbf{U}^{(2)}, \mathbf{U}^{(3)}, \ldots, \mathbf{U}^{(r)}\}$.
- Group 2: $\binom{\mu-r}{\tau}$ τ-sum types for which the corresponding τ-sums do not involve any element from the set $\{\mathbf{U}^{(2)}, \mathbf{U}^{(3)}, \ldots, \mathbf{U}^{(r)}\}$.

Let $q(U^{(v_{[\tau]})})$ denote a τ-sum as defined in Definition 6, in Section 3.3.1, after performing the sign assignment process, i.e.,

$$q(U^{(v_{[\tau]})}) \triangleq \sum_{\ell=1}^{\tau} (-1)^{\ell-1} U^{(v_\ell)}, \tag{C.1}$$

where $v_{[\tau]} = \{v_1, \ldots, v_\tau\} \subseteq [\mu]$, $v_1 < \cdots < v_\tau$, are the indices of the functions evaluations, and where the segment indices and the database index are suppressed to simplify the notation. Let the *type* of the τ-sum be presented by the set of distinct indices of functions evaluations involved in the τ-sum, i.e., the type of $q(U^{(v_{[\tau]})})$ is represented by $v_{[\tau]} = \{v_1, \ldots, v_\tau\}$. The key idea is to show that the symbols of the queries corresponding to Group 2 are deterministic linear functions of the queries corresponding to Group 1 when the symbols of the desired function evaluation $\mathbf{U}^{(1)}$ are known. Now, let $q_0 \triangleq q(U^{(v_{[\tau]})})$, where $r < v_1 < \cdots < v_\tau$, denote an arbitrary query corresponding to Group 2. Specifically, we need to show that, when the symbols of $\mathbf{U}^{(1)}$ queried by the given database are known, i.e., successfully decoded, the query q_0 can be written as a linear function of $\binom{\tau+r-1}{\tau} - 1$ queries corresponding to Group 1. These $\binom{\tau+r-1}{\tau} - 1$ queries contain elements of the row basis functions evaluations and elements included in the τ-sum of q_0 and comprise all the τ-sums of types corresponding to the subsets of size τ of $\mathcal{I} \triangleq [2:r] \cup v_{[\tau]}$, except the type of q_0, i.e., $\{v_1, \ldots, v_\tau\}$. Now, let $\tilde{\mathcal{Q}} \triangleq \left\{q(U^{(\hat{i}_{[\tau]})}) : \hat{i}_{[\tau]} \in \mathcal{T}\right\}$ be a set of queries where there is exactly one query corresponding to each of the $\binom{\tau+r-1}{\tau} - 1$ τ-sum types of Group 1, where $\mathcal{T} \triangleq \left\{\hat{i}_{[\tau]} = \{\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_\tau\} \subset \mathcal{I} \colon \hat{i}_{[\tau]} \neq v_{[\tau]}\right\}$. Finally, assume, without loss of generality, that the subsets of distinct indices $\hat{i}_{[\tau]} \in \mathcal{T}$ are ordered in natural lexicographical order, i.e., $\hat{i}_1 < \hat{i}_2 < \cdots < \hat{i}_\tau$.

Next, from the deterministic linear mapping between the candidate functions evaluations, $\hat{\mathsf{V}}_{\mu \times r}$, we have $U_*^{(v_\ell)} = \hat{v}_{v_\ell,1} U_*^{(1)} + \cdots + \hat{v}_{v_\ell,r} U_*^{(r)}$, $\ell \in [\tau]$, where $(\hat{v}_{v_\ell,1}, \ldots, \hat{v}_{v_\ell,r}) = \hat{\boldsymbol{v}}_{v_\ell}$. Now, we need to show that q_0 is a linear function of the

queries of $\tilde{\mathcal{Q}}$ as follows:

$$q_0 = \sum_{\hat{i}_{[\tau]} \in \mathcal{T}} h(U^{(\hat{i}_{[\tau]})}) q(U^{(\hat{i}_{[\tau]})}), \tag{C.2}$$

where $h(U^{(\hat{i}_{[\tau]})})$ is a linear coefficient calculated as a function of the deterministic linear mapping coefficients represented by the matrix

$$\hat{\mathsf{V}}^*_{(r-1)\times\tau} = \begin{pmatrix} \hat{v}_{v_1,2} & \hat{v}_{v_2,2} & \cdots & \hat{v}_{v_\tau,2} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{v}_{v_1,r} & \hat{v}_{v_2,r} & \cdots & \hat{v}_{v_\tau,r} \end{pmatrix} \tag{C.3}$$

as outlined in [43, Sec. V-B]. Given the above problem setup, notation, and definitions, one can verify that equation (C.2) holds for all queries corresponding to Group 2 (refer to [43, Sec. V-B] for the detailed derivation). Thus, a number of $\binom{\mu-r}{\tau}$ query types in Group 2 are redundant and can be removed from the query set.

APPENDIX D

PROOF OF THEOREM 4

In this appendix, we prove the converse bound on the PPC rate presented in Theorem 4 of Section 4.2. As previously mentioned, the proof follows similarly to the converse proof of Theorem 2 of Section 3.2. Denote the set of all queries by $\mathcal{Q} \triangleq \big\{Q_j^{(v)} \colon v \in [\mu], j \in [n]\big\}$. It can be shown that for both problems of coded PLC and PPC that use an MDS-PIR capacity-achieving storage code,

$$\mathsf{H}\big(A_{[n]}^{(v)} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathcal{Q}\big) \geq \frac{k}{n}\,\mathsf{H}\big(\mathbf{X}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}\big) + \frac{k}{n}\,\mathsf{H}\big(A_{[n]}^{(v')} \,\big|\, \mathbf{X}^{\mathcal{V}}, \mathbf{X}^{(v')}, \mathcal{Q}\big), \tag{D.1}$$

where $\mathcal{V} \subseteq [\mu]$ is arbitrary, $v \in \mathcal{V}$, and $v' \in [\mu] \setminus \mathcal{V}$.[15]

Next, since there are in total μ function evaluations, by Definition 9 we can recursively use equation (D.1) $r-1$ times with $\mathcal{L} = \{\ell_1, \ldots, \ell_r\} \subseteq [\mu]$ to obtain

$$\begin{aligned}
\mathsf{H}\big(A_{[n]}^{(\ell_1)} \big|\, \mathbf{X}^{(\ell_1)}, \mathcal{Q}\big) &\geq \sum_{v=1}^{r-1} \Big(\frac{k}{n}\Big)^{v} \mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})} \big|\, \mathbf{X}^{\{\ell_1,\ldots,\ell_v\}}\big) \\
&\qquad + \Big(\frac{k}{n}\Big)^{r-1} \mathsf{H}\big(A_{[n]}^{(\ell_r)} \big|\, \mathbf{X}^{\{\ell_1,\ldots,\ell_r\}}, \mathcal{Q}\big) \\
&\geq \sum_{v=1}^{r-1} \Big(\frac{k}{n}\Big)^{v} \mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})} \,\big|\, \mathbf{X}^{\{\ell_1,\ldots,\ell_v\}}\big),
\end{aligned} \tag{D.2}$$

where equation (D.2) follows from the nonnegativity of entropy. Note that in [44], the authors claim that the general converse for the DPIR problem strongly depends on the chosen permutation of the indices of the candidate functions. Here, we also make a similar observation and assume that the order of indices $\{\ell_1, \ldots, \ell_r\}$ is the permutation that maximizes the summation term of equation (3.7) and consider that $\mathbf{X}^{(\ell_1)}$ is the polynomial evaluation with the minimum entropy, i.e., $\mathsf{H}\big(\mathbf{X}^{(\ell_1)}\big) = \mathsf{L}\,\mathsf{H}_{\min}^{(\mathrm{B})}$.

[15] Similar derivations can be found in, e.g., [8], [52], [77].

Now,

$$\begin{aligned}
\mathsf{L}\,\mathsf{H}(X^{(\ell_1)}) &= \mathsf{H}\big(\mathbf{X}^{(\ell_1)}\big) \\
&\overset{(a)}{=} \mathsf{H}\big(\mathbf{X}^{(\ell_1)}\big|\mathcal{Q}\big) - \underbrace{\mathsf{H}\big(\mathbf{X}^{(\ell_1)} \,\big|\, A_{[n]}^{(\ell_1)}, \mathcal{Q}\big)}_{=0} \\
&= \mathsf{I}\big(\mathbf{X}^{(\ell_1)}; A_{[n]}^{(\ell_1)}\big|\mathcal{Q}\big) \\
&= \mathsf{H}\big(A_{[n]}^{(\ell_1)}\big|\mathcal{Q}\big) - \mathsf{H}\big(A_{[n]}^{(\ell_1)}\big|\,\mathbf{X}^{(\ell_1)}, \mathcal{Q}\big) \\
&\leq \mathsf{H}\big(A_{[n]}^{(\ell_1)}\big|\mathcal{Q}\big) - \sum_{v=1}^{r-1}\Big(\frac{k}{n}\Big)^v \mathsf{H}\big(\mathbf{X}^{(\ell_{v+1})}\big|\,\mathbf{X}^{(\ell_1)}, \ldots, \mathbf{X}^{(\ell_v)}\big), \qquad \text{(D.3)}
\end{aligned}$$

where (a) holds since any message is independent of the queries $\mathcal{Q}$, and knowing the answers $A_{[n]}^{(\ell_1)}$ and the queries $\mathcal{Q}$, one can determine $\mathbf{X}^{(\ell_1)}$, and equation (D.3) follows directly from equation (3.7).

Finally, the converse proof is completed by showing that

$$\begin{aligned}
\mathsf{R} &= \frac{\mathsf{L}\,\mathsf{H}_{\min}}{\sum_{j=1}^{n}\mathsf{H}\big(A_j^{(\ell_1)}\big)} \overset{(a)}{\leq} \frac{\mathsf{L}\,\mathsf{H}_{\min}}{\mathsf{H}\big(A_{[n]}^{(\ell_1)}\big)} \overset{(b)}{\leq} \frac{\mathsf{L}\,\mathsf{H}_{\min}}{\mathsf{H}\big(A_{[n]}^{(\ell_1)} \,\big|\, \mathcal{Q}\big)} \\
&\leq \frac{\mathsf{H}_{\min}}{\mathsf{H}_{\min}^{(\mathrm{B})} + \sum_{v=1}^{r-1}\big(\frac{k}{n}\big)^v \mathsf{H}\big(X^{(\ell_{v+1})} \,\big|\, X^{(\ell_1)}, \ldots, X^{(\ell_v)}\big)}, \qquad \text{(D.4)}
\end{aligned}$$

where (a) holds because of the chain rule of entropy, (b) is due to the fact that conditioning reduces entropy, and we apply equation (D.3) to obtain equation (D.4).

APPENDIX E

PROOF OF LEMMA 5

In this appendix, we prove the redundancy elimination lemma for the PPC schemes of Chapter 4 as stated in Lemma 5, Section 4.3.4. The proof of Lemma 5 relies on two arguments as follows.

(**i**) For the first round $\tau = 1$, we can directly eliminate redundant 1-sum types based on both the linear and the nonlinear dependencies between the μ candidate polynomial functions evaluations and the f independent messages. As a result, we have a total of $\mu - f$ redundant 1-sum types regardless of the desired polynomial evaluation.

(**ii**) For $\tau > 1$, we can represent the PPC problem as an allied PLC problem over the monomial basis of the polynomial candidate set. Let $\{\ell_1, \ldots, \ell_s\} \subseteq [\mu]$ be the set of indices that correspond to the monomial basis, where, for simplicity, $s \triangleq \mathsf{M}_g^{\mathrm{c}}(f)$. Then, $X_l^{(\ell_1)}, \ldots, X_l^{(\ell_s)}$ satisfy $\mathsf{H}\big(X_l^{(\ell_1)}, \ldots, X_l^{(\ell_s)}\big) = \mathsf{H}\big(X_l^{[\mu]}\big)$, $\forall\, l \in [\mathsf{L}]$. Without loss of generality, we can order the candidate polynomial functions by monomials first and then according to their degree, i.e., $(X_l^{(1)}, \ldots, X_l^{(s)}) = (X_l^{(\ell_1)}, \ldots, X_l^{(\ell_s)})$, $\forall\, l \in [\mathsf{L}]$. Accordingly, the candidate functions evaluations are represented in terms of the monomial basis evaluations with a deterministic linear mapping $\hat{\mathsf{V}}_{\mu \times \mathsf{M}_g^{\mathrm{c}}(f)}$ of size $\mu \times \mathsf{M}_g^{\mathrm{c}}(f)$, for all $l \in [\mathsf{L}]$, as

$$\big(X_l^{(1)}, \ldots, X_l^{(\mu)}\big)^{\mathsf{T}} = \hat{\mathsf{V}}_{\mu \times s} \big(X_l^{(\ell_1)}, \ldots, X_l^{(\ell_s)}\big)^{\mathsf{T}}.$$

Moreover, we have $(\hat{\boldsymbol{v}}_1^{\mathsf{T}}, \ldots, \hat{\boldsymbol{v}}_{\mathsf{M}_g^{\mathrm{c}}(f)}^{\mathsf{T}})^{\mathsf{T}} = \mathsf{I}_{\mathsf{M}_g^{\mathrm{c}}(f)}$, where $\mathsf{I}_{\mathsf{M}_g^{\mathrm{c}}(f)}$ is the $\mathsf{M}_g^{\mathrm{c}}(f) \times \mathsf{M}_g^{\mathrm{c}}(f)$ identity matrix and $\hat{\boldsymbol{v}}_i$ is the i-th row vector of the polynomial coefficient matrix $\hat{\mathsf{V}}_{\mu \times \mathsf{M}_g^{\mathrm{c}}(f)}$. With this mapping, one can show that for a desired polynomial indexed by $v = 1$, the types of τ-sums corresponding to undesired queries, i.e., τ-sums that do not involve any symbols from the desired function evaluation $\mathbf{U}^{(1)}$ can be divided into two groups as follows.

- Group 1: $\binom{\mu-1}{\tau} - \binom{\mu-\mathsf{M}_g^{\mathrm{c}}(f)}{\tau}$ τ-sum types for which the corresponding τ-sums involve at least one element from the set $\{\mathbf{U}^{(2)}, \mathbf{U}^{(3)} \ldots, \mathbf{U}^{(\mathsf{M}_g^{\mathrm{c}}(f))}\}$.
- Group 2: $\binom{\mu-\mathsf{M}_g^{\mathrm{c}}(f)}{\tau}$ τ-sum types for which the corresponding τ-sums do not involve any element from the set $\{\mathbf{U}^{(2)}, \mathbf{U}^{(3)}, \ldots, \mathbf{U}^{(\mathsf{M}_g^{\mathrm{c}}(f))}\}$,

such that the symbols of the queries corresponding to Group 2 are functions of the symbols of the queries corresponding to Group 1 when the symbols of the desired function evaluation are known. Thus, a number of $\binom{\mu-\mathsf{M}_g^{\mathrm{c}}(f)}{\tau}$ query types

in Group 2 are redundant and can be removed from the query set. Accordingly, with the above mapping to an allied PLC problem, we have presented the main component needed to prove the second argument. Then, the result follows directly from Lemma 4 of Chapter 3 and can be seen as a direct application of the proof of [43, Lem. 1, Sec. V-B] (see Appendix C for more details).

APPENDIX F

PROOF OF LEMMA 9

In this appendix, we prove an upper bound on the conditional mutual information stated in Lemma 9 of Section 6.2.2. We start the proof of the simplest case[16] where $\lambda = 1$ and $\eta = 1$ as a special case of Lemma 9. Then extend the proof to $\lambda \geq 1$ and $\eta \geq 1$.

Proof: For $\lambda = 1$ and $\eta = 1$, we have

$$
\begin{aligned}
&\mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; Q_{[n]}^{(1)} A_{[n]}^{(1)} \,\Big|\, \boldsymbol{W}^{(\theta_{1,1})}\Big) \\
&\overset{(a)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; Q_{[n]}^{(1)} A_{[n]}^{(1)} \boldsymbol{W}^{(\theta_{1,1})}\Big) \\
&\overset{(b)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; Q_{[n]}^{(1)} A_{[n]}^{(1)}\Big) + \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; \boldsymbol{W}^{(\theta_{1,1})} \,\Big|\, Q_{[n]}^{(1)} A_{[n]}^{(1)}\Big)}_{=0} \\
&\overset{(c)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; A_{[n]}^{(1)} \,\Big|\, Q_{[n]}^{(1)}\Big) + \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[\theta_{1,2}:f]}; Q_{[n]}^{(1)}\Big)}_{=0} \\
&= \mathsf{H}\big(A_{[n]}^{(1)} \,\big|\, Q_{[n]}^{(1)}\big) - \mathsf{H}\big(A_{[n]}^{(1)} \,\big|\, Q_{[n]}^{(1)} \boldsymbol{W}^{[\theta_{1,2}:f]}\big) \\
&\overset{(d)}{\leq} \mathsf{H}\big(A_{[n]}^{(1)}\big) - \mathsf{H}\big(\boldsymbol{W}^{(\theta_{1,1})} A_{[n]}^{(1)} \,\big|\, Q_{[n]}^{(1)} \boldsymbol{W}^{[\theta_{1,2}:f]}\big) + \underbrace{\mathsf{H}\big(\boldsymbol{W}^{(\theta_{1,1})} \,\big|\, A_{[n]}^{(1)} Q_{[n]}^{(1)} \boldsymbol{W}^{[\theta_{1,2}:f]}\big)}_{=0} \\
&\overset{(e)}{\leq} \mathsf{D} - \mathsf{H}\big(\boldsymbol{W}^{(\theta_{1,1})} A_{[n]}^{(1)} \,\big|\, Q_{[n]}^{(1)} \boldsymbol{W}^{[\theta_{1,2}:f]}\big) \\
&\overset{(f)}{=} \frac{\mathsf{L}}{\mathsf{R}} - \mathsf{H}\big(\boldsymbol{W}^{(\theta_{1,1})} \,\big|\, Q_{[n]}^{(1)} \boldsymbol{W}^{[\theta_{1,2}:f]}\big) - \underbrace{\mathsf{H}\big(A_{[n]}^{(1)} \,\big|\, Q_{[n]}^{(1)} \boldsymbol{W}^{(\theta_{1,1})} \boldsymbol{W}^{[\theta_{1,2}:f]}\big)}_{=0} \\
&= \frac{\mathsf{L}}{\mathsf{R}} - \mathsf{L} = \mathsf{L}\left(\frac{1}{\mathsf{R}} - 1\right)
\end{aligned}
$$

where

(a) follows from the independence between the messages as given by equation (6.2) and the independence of the messages and the queries as stated in equation (6.4);

[16] Note that, for the special case of ($\lambda = 1$, $\eta = 1$), the proof technique is similar to [7, Lem 5]. However, we restate these steps here with our notation to facilitate understanding the general proof.

(b) follows from the chain rule of mutual information and the independence of the messages as given by equation (6.2);

(c) follows from the independence between the messages and the queries as stated in equation (6.4);

(d) follows from the fact that conditioning reduces entropy and the correctness condition of equation (6.11);

(e) follows from the chain rule of entropy and Definition 13;

(f) follows fact that the answer strings are a deterministic function of the queries the stored messages as stated in equation (6.5).

Next, we extend the argument for $\lambda \geq 1$ and $\eta \geq 1$ as follows. Recall that $\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \triangleq \{\boldsymbol{W}^{(\theta_{1,1})}, \boldsymbol{W}^{(\theta_{1,2})}, \ldots, \boldsymbol{W}^{(\theta_{1,\lambda})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,\lambda})}\}$, and $\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}} \triangleq \boldsymbol{W}^{[\theta_{1,\lambda+1}:\theta_{2,1}-1]} \cup \boldsymbol{W}^{[\theta_{2,\lambda+1}:\theta_{3,1}-1]} \cup \ldots \cup \boldsymbol{W}^{[\theta_{\eta-1,\lambda+1}:\theta_{\eta,1}-1]} \cup \boldsymbol{W}^{[\theta_{\eta,\lambda+1}:f]}$. Accordingly, we have

$$
\begin{aligned}
&\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{[\eta]} A_{[n]}^{[\eta]} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\overset{(a)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{[\eta]} A_{[n]}^{[\eta]} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\overset{(b)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{[\eta]} A_{[n]}^{[\eta]}\Big) + \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \,\Big|\, Q_{[n]}^{[\eta]} A_{[n]}^{[\eta]}\Big)}_{=0} \\
&\overset{(c)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_{[n]}^{[\eta]} \,\Big|\, Q_{[n]}^{[\eta]}\Big) + \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{[\eta]}\Big)}_{=0} \\
&= \mathsf{H}\big(A_{[n]}^{[\eta]} \,\big|\, Q_{[n]}^{[\eta]}\big) - \mathsf{H}\big(A_{[n]}^{[\eta]} \,\big|\, Q_{[n]}^{[\eta]} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big) \\
&\overset{(d)}{\leq} \mathsf{H}\big(A_{[n]}^{[\eta]}\big) - \mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} A_{[n]}^{[\eta]} \,\big|\, Q_{[n]}^{[\eta]} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big) + \underbrace{\mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \,\big|\, A_{[n]}^{[\eta]} Q_{[n]}^{[\eta]} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big)}_{=0} \\
&\overset{(e)}{\leq} \mathsf{D} - \mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} A_{[n]}^{[\eta]} \,\big|\, Q_{[n]}^{[\eta]} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big) \\
&\overset{(f)}{=} \frac{\eta\lambda\mathsf{L}}{\mathsf{R}} - \mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \,\big|\, Q_{[n]}^{[\eta]} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big) - \underbrace{\mathsf{H}\big(A_{[n]}^{[\eta]} \,\big|\, Q_{[n]}^{[\eta]} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big)}_{=0} \\
&= \frac{\eta\lambda\mathsf{L}}{\mathsf{R}} - \eta\lambda\mathsf{L} = \eta\lambda\mathsf{L}\left(\frac{1}{\mathsf{R}} - 1\right)
\end{aligned}
$$

where

(a) follows from the independence of the messages as given by equation (6.2) and the independence of the messages and the queries as stated in equation (6.4);

(b) follows from the chain rule of mutual information and the independence of the messages as given by equation (6.2);

(c) follows from the independence of the queries and the messages as stated in equation (6.4);

(d) follows from the fact that conditioning reduces entropy and from the correctness condition of equation (6.11);

(e) follows from the chain rule of entropy and Definition 13;

(f) follows the fact that the answer strings are a deterministic function of the queries and the stored messages as stated in equation (6.5).

■

APPENDIX G

PROOF OF LEMMA 10

In this appendix, we prove a lower bound on the conditional mutual information stated in Lemma 10 of Section 6.2.2. We first proof of the simplest case where $\lambda = 1$ and $\eta = 1$ then extend the same argument for $\lambda \geq 1$ and $\eta \geq 1$. Before starting the proof, recall that for $\gamma \in [\Gamma]$, $\boldsymbol{W}^{\theta_{[\gamma],1}} \triangleq \{\boldsymbol{W}^{(\theta_{1,1})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\gamma,1})}\}$,

Proof: Let $\lambda = 1$, $\eta = 1$, and $\gamma \in [2 : \Gamma]$.

$$
\begin{aligned}
& n\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; Q_{[n]}^{(\gamma-1)} A_{[n]}^{(\gamma-1)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \geq \sum_{j=1}^{n} \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; Q_{j}^{(\gamma-1)} A_{j}^{(\gamma-1)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \overset{(a)}{=} \sum_{j=1}^{n} \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; Q_{j}^{(\gamma)} A_{j}^{(\gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \overset{(b)}{=} \sum_{j=1}^{n} \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; A_{j}^{(\gamma)} \,\Big|\, Q_{j}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \overset{(c)}{=} \sum_{j=1}^{n} \mathrm{H}\big(A_{j}^{(\gamma)} \,\big|\, Q_{j}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}}\big) - \underbrace{\mathrm{H}\big(A_{j}^{(\gamma)} \,\big|\, Q_{j}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}} \boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}\big)}_{=0} \\
& \geq \sum_{j=1}^{n} \mathrm{H}\big(A_{j}^{(\gamma)} \,\big|\, Q_{[n]}^{(\gamma)} A_{[j-1]}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}}\big) \\
& \overset{(c)}{=} \sum_{j=1}^{n} \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; A_{j}^{(\gamma)} \,\Big|\, Q_{[n]}^{(\gamma)} A_{[j-1]}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& = \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; A_{[n]}^{(\gamma)} \,\Big|\, Q_{[n]}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \overset{(b)}{=} \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; A_{[n]}^{(\gamma)} Q_{[n]}^{(\gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \overset{(d)}{=} \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; A_{[n]}^{(\gamma)} Q_{[n]}^{(\gamma)} \boldsymbol{W}^{(\theta_{\gamma,1})} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big) \\
& \qquad\qquad - \underbrace{\mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; \boldsymbol{W}^{(\theta_{\gamma,1})} \,\Big|\, A_{[n]}^{(\gamma)} Q_{[n]}^{(\gamma)} \boldsymbol{W}^{\theta_{[\gamma-1],1}}\Big)}_{=0} \\
& = \mathrm{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; \boldsymbol{W}^{(\theta_{\gamma,1})} \,\big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}}\big) + \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma-1],1}}; A_{[n]}^{(\gamma)} Q_{[n]}^{(\gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}} \boldsymbol{W}^{(\theta_{\gamma,1})}\Big) \\
& \overset{(e)}{=} \mathrm{H}\big(\boldsymbol{W}^{(\theta_{\gamma,1})}\big) + \mathrm{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma],1}}; A_{[n]}^{(\gamma)} Q_{[n]}^{(\gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma-1],1}} \boldsymbol{W}^{(\theta_{\gamma,1})}\Big)
\end{aligned}
$$

$$\overset{(f)}{=} \mathsf{L} + \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\gamma],1}}; A_{[n]}^{(\gamma)} Q_{[n]}^{(\gamma)} \,\Big|\, \boldsymbol{W}^{\theta_{[\gamma],1}}\Big)$$

where

- (a) follows from the privacy constraint of equation (6.10);
- (b) follows from the independence between the messages as given by equation (6.2) and the independence between the messages and the queries as stated in equation (6.4);
- (c) follows from the fact that the answer strings are a deterministic function of the queries and a the stored messages as stated in equation (6.5).
- (d) follows from the chain rule of mutual information, the independence of the messages as given by equation (6.2), and the correctness condition of equation (6.11); particularly, $\mathsf{H}\big(\boldsymbol{W}^{(\theta_{\gamma,1})} \,\big|\, Q_{[n]}^{(\gamma)} A_{[n]}^{(\gamma)}\big) = 0$;
- (e) follows from the independence of the messages as stated in equation (6.2);
- (f) follows from the chain rule of mutual information and the fact that each message consists of L independent and identically distributed symbols as given by equation (6.1).

Next, we extend the argument for $\lambda \geq 1$ and $\eta \geq 1$ as follows. Recall that $\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \triangleq \{\boldsymbol{W}^{(\theta_{1,1})}, \boldsymbol{W}^{(\theta_{1,2})}, \ldots, \boldsymbol{W}^{(\theta_{1,\lambda})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,\lambda})}\}$, and $\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}} \triangleq \boldsymbol{W}^{[\theta_{1,\lambda+1}:\theta_{2,1}-1]} \cup \boldsymbol{W}^{[\theta_{2,\lambda+1}:\theta_{3,1}-1]} \cup \ldots \cup \boldsymbol{W}^{[\theta_{\eta-1,\lambda+1}:\theta_{\eta,1}-1]} \cup \boldsymbol{W}^{[\theta_{\eta,\lambda+1}:f]}$.

Let $\Omega_1, \Omega_2 \in \mathfrak{S}$, such that $\Omega_1 \cap \Omega_2 = \phi$, without loss of generality, assume that $\Omega_1 = [\eta]$ and $\Omega_2 = [\eta+1 : 2\eta]$. Then

$$\begin{aligned}
&n\,\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_{[n]}^{\Omega_1} A_{[n]}^{\Omega_1} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\geq \sum_{j=1}^{n} \mathsf{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_j^{\Omega_1} A_j^{\Omega_1} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
&\overset{(a)}{=} \sum_{j=1}^{n} \mathsf{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; Q_j^{\Omega_2} A_j^{\Omega_2} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
&\overset{(b)}{=} \sum_{j=1}^{n} \mathsf{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_j^{\Omega_2} \,\big|\, Q_j^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
&\overset{(c)}{=} \sum_{j=1}^{n} \mathsf{H}\big(A_j^{\Omega_2} \,\big|\, Q_j^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) - \underbrace{\mathsf{H}\big(A_j^{\Omega_2} \,\big|\, Q_j^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big)}_{=0}
\end{aligned}$$

$$
\begin{aligned}
&\geq \sum_{j=1}^{n} \mathsf{H}\big(A_j^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} A_{[j-1]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
&\overset{(c)}{=} \sum_{j=1}^{n} \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_j^{\Omega_2} \,\Big|\, Q_{[n]}^{\Omega_2} A_{[j-1]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) + \underbrace{\mathsf{H}\big(A_j^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} A_{[j-1]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}\big)}_{=0} \\
&= \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} \,\Big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\overset{(b)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\overset{(d)}{=} \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta+1:2\eta],[\lambda]}} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
&\qquad\qquad - \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; \boldsymbol{W}^{\theta_{[\eta+1:2\eta],[\lambda]}} \,\Big|\, A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big)}_{=0} \\
&= \mathsf{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; \boldsymbol{W}^{\theta_{[\eta+1:2\eta],[\lambda]}} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) + \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{\theta_{[\eta+1:2\eta],[\lambda]}}\Big) \\
&\overset{(e)}{=} \mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta+1:2\eta],[\lambda]}}\big) + \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[2\eta],[\lambda]}}; Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[2\eta],[\lambda]}}\Big) \\
&\overset{(f)}{=} \eta\lambda\mathsf{L} + \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[2\eta],[\lambda]}}; Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[2\eta],[\lambda]}}\Big)
\end{aligned}
$$

where

- (a) follows from the privacy constraint of equation (6.10);
- (b) follows from the independence between the messages as given by equation (6.2) and the independence between the messages and the queries as stated in equation (6.4);
- (c) follows from the fact that the answer strings are a deterministic function of the queries and the stored messages as stated in equation (6.5);
- (d) follows from the chain rule of mutual information, the independence of the messages as stated in equation (6.2), and the correctness condition of equation (6.11); particularly, $\mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta+1:2\eta],[\lambda]}} \,\big|\, Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2}\big) = 0$;
- (e) follows from the independence of the messages as given by equation (6.2);
- (f) follows from the chain rule of mutual information and the fact that each message consists of L independent and identically distributed symbols as given by equation (6.1).

■

APPENDIX H

PROOF OF LEMMA 11

In this appendix, we prove a lower bound on the conditional mutual information stated in Lemma 11 of Section 6.3.2. Before starting the proof, recall that $\boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \triangleq \{\boldsymbol{W}^{(\theta_{1,1})}, \boldsymbol{W}^{(\theta_{1,2})}, \ldots, \boldsymbol{W}^{(\theta_{1,\lambda})}, \boldsymbol{W}^{(\theta_{2,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,1})}, \ldots, \boldsymbol{W}^{(\theta_{\eta,\lambda})}\}$, and $\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}} \triangleq \boldsymbol{W}^{[\theta_{1,\lambda+1}:\theta_{2,1}-1]} \cup \boldsymbol{W}^{[\theta_{2,\lambda+1}:\theta_{3,1}-1]} \cup \ldots \cup \boldsymbol{W}^{[\theta_{\eta-1,\lambda+1}:\theta_{\eta,1}-1]} \cup \boldsymbol{W}^{[\theta_{\eta,\lambda+1}:f]}$.

Proof: We start the proof with $\Omega_1 = [\eta]$ and $\Omega_2 = [\Gamma - \eta + 1 : \Gamma]$. For $\eta \geq \frac{\Gamma}{2}$, we have $\Omega_1 \cap \Omega_2 = [\Gamma - \eta + 1 : \eta]$

$$
\begin{aligned}
& n\,\mathrm{I}\Big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; Q_{[n]}^{\Omega_1} A_{[n]}^{\Omega_1} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
& \geq \sum_{j=1}^{n} \mathrm{I}\big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; Q_j^{\Omega_1} A_j^{\Omega_1} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
& \overset{(a)}{=} \sum_{j=1}^{n} \mathrm{I}\big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; Q_j^{\Omega_2} A_j^{\Omega_2} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
& \overset{(b)}{=} \sum_{j=1}^{n} \mathrm{I}\big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; A_j^{\Omega_2} \,\big|\, Q_j^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
& \overset{(c)}{=} \sum_{j=1}^{n} \mathrm{H}\big(A_j^{\Omega_2} \,\big|\, Q_j^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) - \underbrace{\mathrm{H}\big(A_j^{\Omega_2} \,\big|\, Q_j^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}\big)}_{=0} \\
& \geq \sum_{j=1}^{n} \mathrm{H}\big(A_j^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} A_{[j-1]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) \\
& \overset{(c)}{=} \sum_{j=1}^{n} \mathrm{I}\Big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; A_j^{\Omega_2} \,\Big|\, Q_{[n]}^{\Omega_2} A_{[j-1]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) + \underbrace{\mathrm{H}\big(A_j^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} A_{[j-1]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}\big)}_{=0} \\
& = \mathrm{I}\Big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} \,\Big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
& \overset{(b)}{=} \mathrm{I}\Big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
& \overset{(d)}{=} \mathrm{I}\Big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\Gamma-\eta+1:\Gamma],[\lambda]}} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big) \\
& \qquad\qquad - \underbrace{\mathrm{I}\Big(\boldsymbol{W}^{[f]\backslash\theta_{[\eta],[\lambda]}}; \boldsymbol{W}^{\theta_{[\Gamma-\eta+1:\Gamma],[\lambda]}} \,\Big|\, A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\Big)}_{=0}
\end{aligned}
$$

$$
\begin{aligned}
&= \mathsf{I}\big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; \boldsymbol{W}^{\theta_{[\Gamma-\eta+1:\Gamma],[\lambda]}} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) + \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\eta],[\lambda]}}; A_{[n]}^{\Omega_2} Q_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}} \boldsymbol{W}^{\theta_{[\Gamma-\eta+1:\Gamma],[\lambda]}}\Big) \\
&= \mathsf{H}\big(\boldsymbol{W}^{\theta_{[\Gamma-\eta+1:\Gamma],[\lambda]}} \,\big|\, \boldsymbol{W}^{\theta_{[\eta],[\lambda]}}\big) + \mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\Gamma],[\lambda]}}; Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2} \,\Big|\, \boldsymbol{W}^{\theta_{[\Gamma],[\lambda]}}\Big) \\
&\overset{(f)}{=} \mathsf{H}\big(\boldsymbol{W}^{\theta_{[\eta+1:\Gamma],[\lambda]}}\big) + \underbrace{\mathsf{I}\Big(\boldsymbol{W}^{[f]\setminus\theta_{[\Gamma],[\lambda]}}; A_{[n]}^{\Omega_2} \,\Big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\Gamma],[\lambda]}}\Big)}_{=0} \\
&\overset{(g)}{=} \lambda \mathsf{L}(\Gamma - \eta)
\end{aligned}
$$

where

- (a) follows from the privacy constraint of equation (6.10);
- (b) follows from the independence between the messages as given by equation (6.2) and the independence between the messages and the queries of equation (6.4);
- (c) follows from the fact that the answer is a deterministic function of the queries and the stored messages as stated in equation (6.5);
- (d) follows from the chain rule of mutual information, the independence of the messages as stated in equation (6.2), and the correctness condition of equation (6.11); particularly, $\mathsf{H}\big(\boldsymbol{W}^{\theta_{[\Gamma-\eta+1:\Gamma],[\lambda]}} \,\big|\, Q_{[n]}^{\Omega_2} A_{[n]}^{\Omega_2}\big) = 0$;
- (f) follows from the independence of the messages as given by equation (6.2); the second term equals zero due to the independence of the messages and the queries following equation (6.4) and the fact that the answer strings are a deterministic function of the queries and a *sufficient* number of messages from each classes, i.e., combining equations (6.5) and (6.6) we have

$$
\begin{aligned}
&\mathsf{H}\big(A_{[n]}^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{\theta_{[\Gamma],[\lambda]}}\big) = \mathsf{H}\big(A_{[n]}^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{[\theta_{1,1}:\theta_{1,\lambda}]} \boldsymbol{W}^{\theta_{[2:\Gamma],[\lambda]}}\big) \\
&= \mathsf{H}\big(A_{[n]}^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{[\theta_{1,1}:\theta_{2,1}-1]} \boldsymbol{W}^{[\theta_{2,1}:\theta_{2,\lambda}]} \boldsymbol{W}^{\theta_{[3:\Gamma],[\lambda]}}\big) \\
&= \vdots \\
&= \mathsf{H}\big(A_{[n]}^{\Omega_2} \,\big|\, Q_{[n]}^{\Omega_2} \boldsymbol{W}^{[f]}\big) = 0;
\end{aligned}
$$

- (g) follows from the fact that each message consists of L independent and identically distributed symbols as stated with equation (6.1).

■

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th Annu. IEEE Symp. Found. Comp. Sci. (FOCS)*, Milwaukee, WI, USA, Oct. 23–25, 1995, pp. 41–50.

[2] ——, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–982, Nov. 1998.

[3] W. Gasarch, "A survey on private information retrieval," *Bull. Eur. Assoc. Theor. Comput. Sci. (EATCS)*, no. 82, pp. 72–107, Feb. 2004.

[4] S. Yekhanin, "Private information retrieval," *Commun. ACM*, vol. 53, no. 4, pp. 68–73, Apr. 2010.

[5] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. S. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[6] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 212–236, Jan. 2014.

[7] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.

[8] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.

[9] ——, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6842–6862, Oct. 2018.

[10] ——, "The capacity of private information retrieval from Byzantine and colluding databases," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1206–1219, Feb. 2019.

[11] K. Banawan, B. Arasli, Y.-P. Wei, and S. Ulukus, "The capacity of private information retrieval from heterogeneous uncoded caching databases," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3407–3416, 2020.

[12] Z. Chen, Z. Wang, and S. A. Jafar, "The capacity of T-private information retrieval with private side information," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4761–4773, Aug. 2020.

[13] A. Heidarzadeh, B. Garcia, S. Kadhe, S. El Rouayheb, and A. Sprintson, "On the capacity of single-server multi-message private information retrieval with side information," in *Proc. 56th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2–5, 2018.

[14] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, "Private information retrieval with side information," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2032–2043, Apr. 2020.

[15] S. Kumar, H.-Y. Lin, E. Rosnes, and A. Graell i Amat, "Achieving maximum distance separable private information retrieval capacity with linear codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4243–4273, Jul. 2019.

[16] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2361–2370, Apr. 2018.

[17] ——, "Multiround private information retrieval: Capacity and storage overhead," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5743–5754, Aug. 2018.

[18] ——, "The capacity of symmetric private information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 322–329, Jan. 2019.

[19] S. Song and M. Hayashi, "Capacity of quantum private information retrieval with multiple servers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 1727–1731.

[20] R. Tandon, "The capacity of cache aided private information retrieval," in *Proc. 55th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 3–6, 2017, pp. 1078–1082.

[21] Q. Wang, H. Sun, and M. Skoglund, "The capacity of private information retrieval with eavesdroppers," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3198–3214, may 2019.

[22] Y.-P. Wei, K. Banawan, and S. Ulukus, "The capacity of private information retrieval with partially known private side information," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 8222–8231, Dec. 2019.

[23] Y.-P. Wei and S. Ulukus, "The capacity of private information retrieval with private side information under storage constraints," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2023–2031, Apr. 2020.

[24] Y.-P. Wei, B. Arasli, K. Banawan, and S. Ulukus, "The capacity of private information retrieval from decentralized uncoded caching databases," *Information*, vol. 10, no. 12, p. 372, Nov. 2019.

[25] A. Heidarzadeh, B. Garcia, S. Kadhe, S. El Rouayheb, and A. Sprintson, "On the capacity of single-server multi-message private information retrieval with side information," in *Proc. 56th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2–5, 2018, pp. 180–187.

[26] S. P. Shariatpanahi, M. J. Siavoshani, and M. A. Maddah-Ali, "Multi-message private information retrieval with private side information," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Guangzhou, China, Nov. 25–29, 2018, pp. 1–5.

[27] S. Li and M. Gastpar, "Single-server multi-message private information retrieval with side information: The general cases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 21–26, 2020, pp. 1083–1088.

[28] A. Heidarzadeh and A. Sprintson, "The linear capacity of single-server individually-private information retrieval with side information," Feb. 2022, arXiv:2202.12229v1 [cs.IT]. [Online]. Available: https://arxiv.org/abs/2202.12229

[29] N. B. Shah, K. V. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun. 29 – Jul. 4, 2014, pp. 856–860.

[30] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, and C. Hollanti, "Private information retrieval from coded storage systems with colluding, Byzantine, and unresponsive servers," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3898–3906, Jun. 2019.

[31] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, China, Jun. 14–19, 2015, pp. 2842–2846.

[32] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7081–7093, Nov. 2018.

[33] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM J. Appl. Algebra Geom.*, vol. 1, no. 1, pp. 647–664, Nov. 2017.

[34] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," in *Proc. 30th Annu. ACM Symp. Theory Comput. (STOC)*, Dallas, TX, USA, May 23–26, 1998, pp. 151–160.

[35] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway, "Locally random reductions: Improvements and applications," *J. Cryptology*, vol. 10, no. 1, pp. 17–36, 1997.

[36] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th Annu. ACM Symp. Theory Comput. (STOC)*, Chicago, IL, USA, May 02–04, 1988, pp. 351–371.

[37] A. Beimel, Y. Ishai, E. Kushilevitz, and I. Orlov, "Share conversion and private information retrieval," in *Proc. 27th Annu. Conf. Comput. Complexity*, Porto, Portugal, Jun. 26–29, 2012, pp. 258–268.

[38] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[39] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[40] S. Yekhanin, "Locally decodable codes," *Found. Trends Theor. Comput. Sci.*, vol. 6, no. 3, pp. 139–255, 2010.

[41] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proc. 17th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 3, San Francisco, CA, USA, Mar. 29 – Apr. 2, 1998, pp. 1257–1264.

[42] ——, "Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2825–2830, 2006.

[43] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.

[44] Z. Chen, Z. Wang, and S. A. Jafar, "The asymptotic capacity of private search," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4709–4721, Aug. 2020.

[45] D. Karpuk, "Private computation of systematically encoded data with colluding servers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2112–2116.

[46] N. Raviv and D. A. Karpuk, "Private polynomial computation from Lagrange encoding," *IEEE Trans. Inf. Forens. Secur.*, vol. 15, pp. 553–563, 2020.

[47] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *Proc. Iran Workshop Commun. Inf. Theory (IWCIT)*, Tehran, Iran, Apr. 25–26, 2018, pp. 1–6.

[48] Y. Yakimenka, H.-Y. Lin, and E. Rosnes, "On the capacity of private monomial computation," in *Proc. Int. Zurich Sem. Inf. Commun. (IZS)*, Zurich, Switzerland, Feb. 26–28, 2020, pp. 31–35.

[49] A. Heidarzadeh and A. Sprintson, "Private computation with side information: The single-server case," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 1657–1661.

[50] ——, "Private computation with individual and joint privacy," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 21–26, 2020, pp. 1112–1117.

[51] H.-Y. Lin, S. Kumar, E. Rosnes, and A. Graell i Amat, "Asymmetry helps: Improved private information retrieval protocols for distributed storage," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Guangzhou, China, Nov. 25–29, 2018.

[52] ——, "On the fundamental limit of private information retrieval for coded distributed storage," Aug. 2018, arXiv:1808.09018v1 [cs.IT]. [Online]. Available: https://arxiv.org/abs/1808.09018

[53] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, A.-L. Horlemann-Trautmann, D. Karpuk, and I. Kubjas, "t-private information retrieval schemes using transitive codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2107–2118, Apr. 2019.

[54] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and A. S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc. 22nd Int. Conf. Artif. Intell. & Statist. (AISTATS)*, Okinawa, Japan, Apr. 16–18, 2019.

[55] Z. Chen, Z. Wang, and S. A. Jafar, "The asymptotic capacity of private search," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2122–2126.

[56] L. Song and C. Fragouli, "Content-type coding," in *Proc. Int. Symp. Netw. Coding (NetCod)*, Sydney, NSW, Australia, June 22–24 June, 2015, pp. 31–35.

[57] S. Brahma and C. Fragouli, "Pliable index coding," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6192–6203, 2015.

[58] T. Liu and D. Tuninetti, "Tight information theoretic converse results for some pliable index coding problems," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 2642–2657, 2020.

[59] ——, "Decentralized pliable index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 532–536.

[60] ——, "Private pliable index coding," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 25–28, 2019, pp. 1–5.

[61] ——, "Secure decentralized pliable index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 21–26, 2020, pp. 1729–1734.

[62] ——, "Optimal linear coding schemes for the secure decentralized pliable index coding problem," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Riva del Garda, Italy, Apr. 11–15, 2021, pp. 1–5.

[63] T. Jiang and Y. Shi, "Sparse and low-rank optimization for pliable index coding," in *Proc. 6th IEEE Glob. Conf. Signal Inf. Process. Proc. (GlobalSIP)*, Anaheim, CA, USA, Nov. 26–29, 2018, pp. 331–335.

[64] L. Song, "A binary randomized coding scheme for pliable index coding with multiple requests," in *Proc. 10th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Hong Kong, China, Dec. 3–7, 2018, pp. 1–5.

[65] S. Sasi and B. S. Rajan, "Code construction for pliable index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 527–531.

[66] L. Ong, B. N. Vellambi, and J. Kliewer, "Optimal-rate characterisation for pliable index coding using absent receivers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 522–526.

[67] P. Krishnan, R. Mathew, and S. Kalyanasundaram, "Pliable index coding via conflict-free colorings of hypergraphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 12–20, 2021, pp. 214–219.

[68] I. Samy, R. Tandon, and L. Lazos, "On the capacity of leaky private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 1262–1266.

[69] H.-Y. Lin, S. Kumar, E. Rosnes, A. Graell i Amat, and E. Yaakobi, "The capacity of single-server weakly-private information retrieval," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 415–427, Mar. 2021.

[70] ——, "Multi-server weakly-private information retrieval," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1197–1219, Feb. 2022.

[71] T. Guo, R. Zhou, and C. Tian, "On the information leakage in private information retrieval systems," *IEEE Trans. Inf. Forens. Secur.*, vol. 15, pp. 2999–3012, 2020.

[72] G. Smith, "On the foundations of quantitative information flow," in *Proc. 12th Int. Conf. Found. Softw. Sci. Comput. Struct. (FoSSaCS)*, York, U.K., Mar. 22–29, 2009, pp. 288–302.

[73] G. Barthe and B. Köpf, "Information-theoretic bounds for differentially private mechanisms," in *Proc. 24th IEEE Comput. Secur. Found. Symp. (CSF)*, Cernay-la-Ville, France, Jun. 27–29, 2011, pp. 191–204.

[74] I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1625–1657, Mar. 2020.

[75] Copyright © 2022, IEEE. Reprinted, with permission, from S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private linear computation for noncolluding coded databases," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 847–861, Mar. 2022.

[76] J. Radhakrishnan, "Entropy and counting," in *Computational Mathematics, Modelling and Algorithms*, J. C. Misra, Ed. New Delhi, India: Narosa Publishing House, 2003, pp. 146–168.

[77] J. Xu and Z. Zhang, "On sub-packetization and access number of capacity-achieving PIR schemes for MDS coded non-colluding servers," *Science China Inf. Sci.*, vol. 61, no. 10, pp. 100 306:1–100 306:16, Oct. 2018.

[78] S. A. Obead and J. Kliewer, "Achievable rate of private function retrieval from MDS coded databases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2117–2121.

[79] Copyright © 2022, IEEE. Reprinted, with permission, from S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private polynomial computation for noncolluding coded databases," IEEE J. Sel. Areas Commun., 2022, to be published. doi:10.1109/TIFS.2022.3166667.

[80] R. G. L. D'Oliveira and S. El Rouayheb, "One-shot PIR: Refinement and lifting," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2443–2455, Apr. 2020.

[81] Copyright © 2019, IEEE. Reprinted, with permission, from S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "On the capacity of private nonlinear computation for replicated databases," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 25–28, 2019, pp. 1–5.

[82] M. Karmoose, L. Song, M. Cardone, and C. Fragouli, "Private broadcasting: An index coding approach," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 25–30, 2017, pp. 2543–2547.

[83] D. T. Kao, M. A. Maddah-Ali, and A. S. Avestimehr, "Blind index coding," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2076–2097, 2016.