This article has been accepted for publication in IEEE BITS the Information Theory Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MBITS.2024.3368349

A Guided Walk Through Coded Private Information Retrieval

Rafael G. L. D'Oliveira*, Salim El Rouayheb†
*SMSS, Clemson University, USA, rdolive@clemson.edu
†ECE, Rutgers University, USA, salim.elrouayheb@rutgers.edu

***Abstract*—We present an accessible introduction to the field of coded private information retrieval (PIR), aiming to make the subject approachable to newcomers. We start by presenting the fundamental concepts of PIR and its history, concentrating on the information-theoretic setting. We then explore secret sharing schemes and their connection to Reed-Solomon codes, showing how these can be utilized to create PIR schemes.**

We further show how the performance of these PIR schemes can be enhanced using a refining and lifting technique. We conclude by highlighting some open questions in the field. This article serves as a useful starting point for anyone interested in gaining an understanding of research in the field of coded PIR.

***Index Terms*—private information retrieval (PIR), coded PIR, secret sharing**

I. Introduction

The purpose of private information retrieval (PIR) [1, 2] is to allow users to privately access specific data from a database. This is interesting in a variety of scenarios, such as downloading a movie without sharing personal preferences with a streaming service, or doing a private search for financial or medical information. Thus, PIR addresses a fundamental need for privacy in today's digital age.

In recent times, there has been a surge of interest in applying PIR to coded databases. Coded databases store data in redundant format, allowing data to be recovered in the event of a failure or data loss. PIR for coded databases [3] seeks to enable a user to retrieve a specific piece of information from a coded database while preserving the privacy of their queries. This poses a new challenge, as coded databases introduce complexities not encountered in the conventional setting. Although this research area is still in its infancy, it is expected to gain more traction in the future as the demand for privacy in coded storage systems increases.

In this article, we present an accessible introduction to information-theoretic coded PIR, with the aim of rendering the subject more approachable to newcomers. We consider the setting where M files are distributed among N servers as an (N, K)-maximum distance separable code. A user wishes to privately retrieve one of the M files, ensuring that no T colluding servers can learn anything (hence the information-theoretic) about the desired file selection. The user's objective is to minimize the communication cost.

As will be shown, when dealing with large file sizes, the communication cost is mainly dictated by the download cost [4]. Following typical practice in the literature, we focus on maximizing the download rate, defined as the ratio between the desired information and the total downloaded information. The optimal download rate is called the PIR capacity.

We approach PIR through the perspective of secret sharing [5]–[7]. Secret sharing is a technique used to distribute a secret among a group of participants so that only specific subsets of participants can reconstruct the secret while others cannot. In principle, this technique can be applied to PIR when the databases are replicated (i.e., when $K = 1$), by considering the query for the desired file as the secret to be shared. This secret-sharing approach to replicated PIR is asymptotically optimal when the number of files M approaches infinity [8]. Furthermore, an optimal scheme for finite M can be obtained from the secret sharing method through a process called refinement and lifting [9], resulting in a capacity-achieving download rate of $\frac{(N-T)N^{M-1}}{N^M-T^M}$.

The same overarching technique is applicable to coded PIR ($K > 1$), but an analogue generalization of secret sharing is required [10]. Combining it with the same refinement and lifting[1] procedure as before, we

[1]For coded PIR, some extra assumptions [11, Appendix B] must be added to the refinement and lifting procedure [9].

obtain a coded PIR scheme that achieves a download rate of $\frac{(N-K-T+1)N^{M-1}}{N^M-(K+T-1)^M}$. This scheme achieves capacity when $T=1$, i.e., when there are no collusions [12]. For general T, the scheme represents the best known general family of schemes.[2] It has also been proven to be optimal under certain constraints [11]. Nonetheless, the capacity of PIR with MDS coded data and collusion remains an open question.

II. Notation

We denote the finite field with a prime q of elements by $\mathbb{F}_q$.[3] The M-dimensional coordinate vector space over $\mathbb{F}_q$ is then denoted by $\mathbb{F}_q^M$. We denote the standard basis of $\mathbb{F}_q^M$ by $\{e_1,\dots,e_M\}$ where $e_i \in \mathbb{F}_q^M$ is the vector with a 1 in the i-th coordinate and a zero in all other coordinates. We denote the inner product in $\mathbb{F}_q^M$ by $\langle\cdot,\cdot\rangle : \mathbb{F}_q^M \times \mathbb{F}_q^M \to \mathbb{F}_q$.

If X is a discrete random variable which takes values in an alphabet $\mathcal{X}$, then the Shannon entropy of X is

$$H(X) = -\sum_{x\in\mathcal{X}} \Pr[X=x]\log(\Pr[X=x]),$$

where we set the basis of our logarithms to be 2. The entropy of X conditioned on a discrete random variables Y taking values from an alphabet $\mathcal{Y}$ is

$$H(X|Y) = -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_{X,Y}(x,y)\log\left(\frac{P_{X,Y}(x,y)}{P_Y(y)}\right),$$

where we denote $P_{X,Y}(x,y)=\Pr[X=x,Y=y]$ and $P_Y(y)=\Pr[Y=y]$. The mutual information of X and Y is $I(X;Y)=H(X)-H(X|Y)=H(Y)-H(Y|X)$.

III. Private Information Retrieval

In PIR, a user seeks to retrieve the i-th entry of a database $D\in\mathbb{F}_q^M$, without revealing the specific entry of interest. Suppose the entire dataset D is stored on a single server. One straightforward solution to this problem is for the user to download the entire dataset D. By doing so, the user obtains the desired entry $D_i\in\mathbb{F}_q$, while the server remains ignorant of which entry the user was interested in. However, the communication cost for this approach can be quite high, as it requires the transfer of the entire database. The goal of PIR is to retrieve the intended information privately while minimizing the communication cost. Thus, can the user do better than downloading the whole database?

In the case of a single server, the answer is no – at least if the user wants to guarantee perfect information-theoretic privacy [2]. To see why, consider a scheme where the user does not download some coordinate $D_k\in\mathbb{F}_q$. It is then clear to the server that the user was not interested in D_k, which in turn reveals some information about the desired coordinate i, namely that it is not equal to k.

Therefore, in order to achieve performance improvements beyond simply downloading the entire database, certain restrictions must be imposed on the server or servers that hold the data. One possibility is to limit the computational power of the servers, which allows for the use of cryptographic protocols, leading to the concept of computational PIR [14]. Another alternative is to allow the user to have some additional information, or side information, about the database, which is known as PIR with side information [15]. We consider the scenario where the data is distributed among many servers, which are prohibited from colluding too much (at least not all servers, as this would be equivalent to the single-server case). Let us now present what is arguably the simplest non-trivial PIR scheme.

Example 1 (Simple PIR Scheme)**.** Suppose that there are $N=2$ servers which do not collude with each other. Each has a copy of the same database $D\in\mathbb{F}_q^M$ containing M files (each a single symbol of $\mathbb{F}_q$). Suppose that a user wants to retrieve a private file $D_i\in\mathbb{F}_q$ without revealing any information about which file, i.e., the private information is not the file D_i, but the index $i\in\{1,\dots,M\}$ of the desired file. The user proceeds as follows.

- **Step 1:** Generate a random query vector $R\in\mathbb{F}_q^M$ uniformly at random.
- **Step 2:** The user transmits the query $X_1=R$ to Server 1 and $X_2=e_i+R$ to Server 2, where e_i is the vector with entry one in the i-th coordinate, and zero in all other coordinates.
- **Step 3:** Each Server j computes the inner product $Y_j=\langle X_j,D\rangle$ and transmits it to the user.
- **Step 4:** The user decodes $D_i=\langle e_i,D\rangle=Y_2-Y_1$.

This scheme is private since what each server observes, X_j, is uniformly distributed. In information theoretic terms, the mutual information $I(i;X_j)=0$,

[2]For some restricted parameters, the scheme described in [13] outperforms the scheme we describe here.

[3]For simplicity, we consider only fields with a prime amount of elements. The results can easily be extended to general finite fields.

This article has been accepted for publication in IEEE BITS the Information Theory Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MBITS.2024.3368349

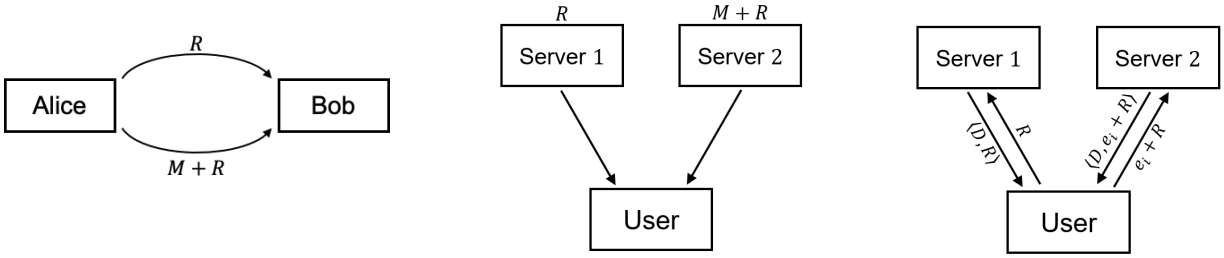

Fig. 1: We show how the one time pad can be applied to secure multipath communication, secure storage, and private information retrieval. On the left, Alice wants to send a confidential message $M \in \mathbb{F}_q$ to Bob. To do this, Alice uniformly generates a random key $R \in \mathbb{F}_q$. Alice transmits the key R through one channel and the padded message $M + R$ through another. Bob can decode the confidential message M by subtracting the key R from the padded message $M + R$. As long as an eavesdropper can only see one of the transmitted messages, the confidential message M is secure. In the middle, a user wants to use two servers to store a confidential file $M \in \mathbb{F}_q$. To do this, the User uniformly generates a random key $R \in \mathbb{F}_q$. The User stores the key R in Server 1 and the padded file $M + R$ in Server 2. The user can retrieve the confidential message M by subtracting the key R from the padded message $M + R$. As long as the servers do not collude, the confidential message M is secure. On the right, a user wants to privately retrieve the i-th file D_i of a database $D \in \mathbb{F}_q^M$ containing M files which is replicated between two servers. To do this, the User uniformly generates a random key $R \in \mathbb{F}_q^M$. The User transmits the query R to Server 1 and the query $e_i + R$ to Server 2, where $e_i \in \mathbb{F}_q^M$ is the vector with a one in the i-th entry and zeroes in every other entry. Server 1 computes the inner product $\langle D, R\rangle \in \mathbb{F}_q$ and transmits it to the User. Server 2 computes the inner product $\langle D, e_i + R\rangle \in \mathbb{F}_q$ and transmits it to the User. The user retrieves the file $D_i = \langle D, e_i\rangle$ by subtracting the response of Server 1 from that of Server 2. As long as the servers do not collude, the private index i is secure.

i.e., the private index i is independent of any query X_j. The communication cost of the scheme is as follows.

- **Upload:** the user uploads M symbols $X_j \in \mathbb{F}_q$ to each server, totaling $2M$ symbols uploaded.
- **Download:** the user downloads a symbol $Y_j \in \mathbb{F}_q$ from each server, totaling 2 symbols downloaded.

Thus, the total communication cost is $2M+2$ symbols.

Example 1 considers the case where each file consists of a single symbol. In many applications, however, files might contain multiple symbols. In this setting, when the file size is allowed to be arbitrarily large, the cost of the query data uploaded is negligible in comparison to the cost of downloading [4]. This is because the original query can be reused for each segment of the file, requiring only a single upload. To illustrate this point, we present the following example.

Example 2 (Simple PIR Scheme for Large Files)**.** Suppose each file is a q-ary vector of length s (thus $s = 1$ in Example 1). Let $D = (D(1), \ldots, D(s)) \in \mathbb{F}_q^{sM}$ denote the full database, where $D(k) \in \mathbb{F}_q^M$ corresponds to the k-th symbol of each file. The user proceeds as follows.

1) **Step 1:** Generate a random query vector $r \in \mathbb{F}_q^M$ uniformly at random.
2) **Step 2:** The user transmits the query $X_1 = R$ to Server 1 and $X_2 = e_i + R$ to Server 2, where e_i is the vector with entry one in the i-th coordinate, and zero in all other coordinates.
3) **Step 3:** Each Server j computes the inner products $Y_j^k = \langle X_j, D(k)\rangle$ and transmits it to the user.
4) **Step 4:** The user decodes the file he wants as $D_i = (Y_2^1 - Y_1^1, \ldots, Y_2^s - Y_1^s)$.

The privacy of this scheme follows directly from that of Example 1. The communication cost of this modified scheme is as follows.

- **Upload:** M symbols $X_j \in \mathbb{F}_q$ to each server, totaling $2M$ symbols uploaded.
- **Download:** s symbols $Y_j^k \in \mathbb{F}_q$ from each server totaling $2s$ symbols downloaded.

Thus, the total communication cost is $2M+2s$ symbols.

As we observed in Example 2, the length s of the file only affects the download cost, making it the dominant factor for large files. Consequently, much attention has been devoted to devising schemes with optimal download costs.

A common metric used to evaluate such schemes is the download rate, defined as the ratio of the desired file's length to the total amount of symbols downloaded. For the scenario described in Example 2, the download rate is equal to $\frac{s}{2s} = \frac{1}{2}$.

The information-theoretic capacity of PIR is the reciprocal of the infimum achievable download cost per symbol of the desired message. In [8], it was shown that the capacity of PIR with M files replicated among N servers, with at most T servers colluding, is given by $\frac{(N-T)N^{M-1}}{N^M-T^M}$. For instance, in the scenario described in Example 2, with $N = 2$ non-colluding ($T = 1$) servers, the capacity equals $\frac{2^{M-1}}{2^M-1}$. We note that while the capacity depends on the number of files, the download rate for the scheme in Example 2 does not. However, as the number of files grows, the capacity rapidly approaches the download rate $\frac{1}{2}$ of the aforementioned scheme. Consequently, the scheme in Example 2 is asymptotically optimal.

In what follows, we discuss the use of the ideas behind the one-time pad and its generalization, secret sharing [5, 6], to extend the scheme in Example 2. This will result in the development of asymptotically optimal PIR schemes for arbitrary numbers of files M and servers N, with at most T collusions.

IV. The One-Time Pad and Secret Sharing

The approaches presented in Examples 1 and 2 can be likened to a well-established cryptographic technique, known as the one-time pad[4], devised to protect the exchange of information between two participants. Imagine a scenario where a user, Alice, intends to transmit a confidential message $M \in \mathbb{F}_q$ to a recipient, Bob, over a communication channel susceptible to the interception by an eavesdropper, Eve. The one-time pad method operates as described below.

Algorithm 1 (One-Time Pad)**.**

- **Step 1:** Alice and Bob share a random symbol $R \in \mathbb{F}_q$.
- **Step 2:** Alice encrypts the message M by computing $X = M + R$ and transmits X to Bob.
- **Step 3:** Upon receipt of X, Bob decrypts the message by computing $M = X - R$ using the shared random symbol R.

In 1948, Shannon published a seminal paper [17] in which he introduced the concept of perfect security for encryption schemes.[5] A cryptographic scheme is said to be perfectly secure if the mutual information between the message M and the received signal X is zero, that is, if M and X are statistically independent. Shannon proved that the one-time pad encryption satisfies this criterion of perfect security. To demonstrate this, we calculate the mutual information as follows.

$$\begin{aligned} I(M;X) &= H(X) - H(X|M) \\ &= H(X) - H(M+R|M) \\ &= H(X) - H(R) \\ &= \log(|\mathbb{F}_q|) - \log(|\mathbb{F}_q|) \\ &= 0 \end{aligned}$$

The first equality follows from the definition of mutual information. The second from substituting $X = M+R$. The third from the fact that the uncertainty about the sum of two random variables conditioned on one of them is that of the other. The fourth from the fact that X and R are uniformly distributed in $\mathbb{F}_q$. This confirms that the one-time pad is indeed perfectly secure.

The one-time pad can be readily extended for use in secure multipath communication, secure distributed storage, and, more importantly to our discussion, to private information retrieval, as shown in Figure 1.

For secure distributed storage, the idea is to store the encrypted message $X_1 = M + R$ in one server and the "secret key" $X_2 = R$ in another server. Consequently, the private message M can only be retrieved by downloading both X_1 and X_2 from their respective servers. Since $I(X_i; M) = 0$ for both $i = 1$ and $i = 2$, the scheme is perfectly secure, provided that the servers do not exchange information with each other. For PIR, the schemes presented in Examples 1 and 2 are a direct application of the one-time pad. In the absence of privacy constraints, the user would simply transmit the query e_i to obtain $D_i = \langle D, e_i \rangle$. Thus, to obtain privacy, the user employs the random pad R

[4]We refer to [16] for an interesting discussion on the history behind the invention of the one-time pad.

[5]Shannon published an earlier version of this paper in 1945 [18] which was classified. Interestingly, this precedes Shannon's *other* seminal paper [19].

This article has been accepted for publication in IEEE BITS the Information Theory Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MBITS.2024.3368349

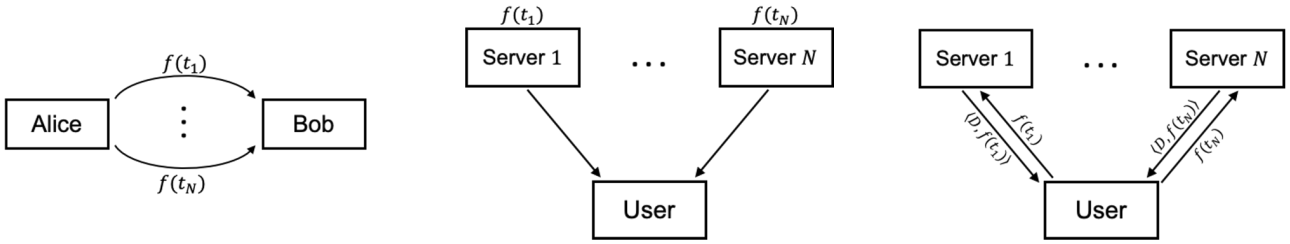

Fig. 2: We show how secret sharing can be applied to secure multi-path communication, secure storage, and private information retrieval. The schemes are analogous to those described in Figure 1.

to conceal the query e_i from the servers. To construct asymptotically optimal PIR schemes for any number of servers N, with up to T collusions, we explore a generalization of the one-time pad called secret sharing.

V. Secret Sharing

Secret sharing [5, 6] is a protocol that enables the distribution of a confidential piece of information, referred to as the "secret," among a group of participants in such a manner that only specific subsets of participants can reconstruct the secret, while others cannot. We illustrate this concept through the following example.

Consider a bank with a safe that should only be opened in the presence of at least three out of five managers. The safe is secured with a secret password that should remain accessible only to any group of three of the five managers. To enforce this security measure, each manager receives a "share" of this secret password. These shares are distributed in such a way that any combination of three can be used to decipher the secret password, but any fewer than three will not yield information about the secret password. This problem can be solved by what is known as threshold secret sharing.

Example 3 ($(5, 3)$-Threshold Secret Sharing)**.**

- **Step 1:** Choose $R_1, R_2 \in \mathbb{F}_q$ uniformly at random.
- **Step 2:** If $S \in \mathbb{F}_q$ is the secret, define the polynomial $f(t) = S + R_1 t + R_2 t^2$.
- **Step 3:** Each manager i receives the share $f(i)$.

As a result, every manager has a single evaluation of the parabolic function $f(t) = S + R_1 t + R_2 t^2$. Since a parabola is uniquely determined by three distinct points, any three managers can collaborate to deduce the coefficients of f and consequently reveal the secret password S. However, if fewer than three managers try to find the secret, it is impossible to determine the polynomial f; for any two evaluations i_1 and i_2, the mutual information $I(S; f(i_1), f(i_2)) = 0$, i.e., any two evaluations are statistically independent of the secret.

By expressing the shares available to each manager as a vector, we can formulate the following linear system.

$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 2 \\ 1 & 4 & 2 \\ 1 & 5 & 4 \end{pmatrix} \begin{pmatrix} S \\ R_1 \\ R_2 \end{pmatrix}$$

In this matrix representation, each manager's share corresponds to a single row. Given any three rows, the linear system's solution can be determined uniquely. However, if fewer than three rows are present, the system becomes underdetermined, and it can be shown that any potential solution is equally probable. Since six distinct evaluations are needed, the number of elements in the field q must be strictly larger than the number of shareholders. The general threshold secret sharing scheme is as follows.

Algorithm 2 ($(N, T + 1)$-Threshold Secret Sharing)**.**

- The secret is an element $S \in \mathbb{F}_q$.
- The number of shareholders is N ($N < q$).
- The minimum number of shareholders required to decode the secret is $T + 1$.
- **Step 1:** Choose T random keys $R_1, \ldots, R_T \in \mathbb{F}_q$ uniformly at random.
- **Step 2:** Define the polynomial $f(t) = S + R_1 t + \ldots + R_T t^T$.
- **Step 3:** Each shareholder i receives the share $f(i)$.

The fundamental concept behind $(N, T+1)$-threshold secret sharing is that the polynomial f of degree T is uniquely determined by $T + 1$ evaluation points. How-

ever, due to the presence of random keys $R_1, \ldots, R_T$, fewer than $T+1$ evaluation points make f indeterminate, as it could represent any polynomial. It is essential to emphasize that the random keys R_j must be uniformly distributed. This highlights an advantage of working over finite fields, as there is no uniform distribution over the real numbers, for example.

Shamir's secret sharing can be readily applied to the scenarios we discussed previously, multipath communication, secure distributed storage, and private information retrieval. We illustrate this in Figure 2.

In the scenario of PIR with $N = 5$ servers, where at most $T = 2$ servers are allowed to collude, the $(5, 3)$-threshold secret sharing scheme can be used. By replacing the secret S with the query vector e_i, the query polynomial $f(t) = e_i + R_1 t + R_2 t^2$ is obtained. The user then sends $f(j)$ to each server j. Each server computes the inner product $\langle f(j), D\rangle$ and sends the result to the user. This can be viewed as evaluating the polynomial $h(t) = \langle e_i, D\rangle + \langle R_1, D\rangle t + \langle R_2, D\rangle t^2$, where $D_i = \langle e_i, D\rangle$ in one of its coefficients. Since $h(t)$ has degree 2, it can be uniquely determined by three evaluations. Thus, the user can obtain D_i by contacting any three of the servers, providing a PIR scheme that is secure against any two servers colluding. Furthermore, since only three evaluations are needed, this scheme also offers resilience to up to $7 - 3 = 4$ erasures or 2 errors. The download rate for this scheme is $\frac{1}{7}$.

In the original formulation of the PIR problem, the user is primarily concerned with maintaining privacy and not resilience to erasures or errors. In such cases, one can opt for a ramp secret sharing scheme to improve the download rate at the cost of reduced resilience.

In the ramp secret sharing scheme [7], the main idea is to include more secrets as coefficients in the polynomial used for the secret sharing scheme. For example, in a $(5, 3)$-threshold secret sharing scheme, the polynomial $f(t) = S + R_1 t + R_2 t^2$ is utilized, where S is the secret and R_1, R_2 are random keys. However, if the user intends to contact all $N = 5$ servers with no erasures or errors, they can instead consider the polynomial $g(t) = S_1 + S_2 t + S_3 t^2 + R_1 t^3 + R_2 t^4$, where now we have three secrets. With two or fewer evaluations, nothing can be learned about the coefficients of g. However, from three evaluations up to five, the user will gradually learn more information about the secrets until at $N = 5$ evaluations the user will be able to fully decode all three secrets. This results in a communication rate of $\frac{3}{5}$, which is an improvement over the $(N, T+1)$-threshold secret sharing scheme at the cost of noise resilience.

The general ramp secret sharing scheme is as follows.

Algorithm 3 ($(N, B, T+1)$-Ramp Secret Sharing)**.**

- The secrets are elements $S_1, \ldots, S_{B-T+1} \in \mathbb{F}_q$.
- The number of shareholders is N ($N < q$).
- The minimum threshold required to learn something about the secrets is $T+1$.
- The threshold to uniquely determine all secrets is $B+1$. ($T \leq B < N$)
- **Step 1:** Choose T random keys $R_1, \ldots, R_T \in \mathbb{F}_q$ uniformly at random.
- **Step 2:** Define the polynomial $f(t) = S_1 + S_2 t + \ldots + S_{B-T+1} t^{B-T} + R_1 t^{B-T+1} + \ldots + R_T t^B$.
- **Step 3:** Each shareholder i receives the share $f(i)$.

Using an $(N, B, T+1)$ ramp secret sharing scheme, we can ensure privacy against T colluding servers, achieve tolerance to $B - T$ erasures, and correct up to $\frac{B-T}{2}$ errors. It is important to note that, if privacy is not a concern, this scheme can function as an error-correcting code by setting $T = 0$ and $B = N - 1$. In this configuration, the scheme is equivalent to a Reed-Solomon code [20, 21].

In the context of the classical PIR setting (no erasures or errors, and replicated databases) with N servers and a maximum of T collusions, employing an $(N, N-T, T)$ ramp secret sharing scheme can yield a PIR scheme with a rate of $1 - \frac{T}{N}$. This rate is asymptotically optimal. However, the capacity of PIR for these parameters [8], given by $\frac{(N-T)N^{M-1}}{N^M - T^M}$, depends on the number of messages M. Our next goal is to demonstrate how to attain optimal capacity-achieving schemes. These optimal schemes can be derived from the secret sharing schemes through a technique known as refinement and lifting. Moreover, the same technique will later be utilized for coded PIR.

VI. Refining and Lifting PIR Schemes

In [8], it was shown that for the case of replicated data, the secret sharing schemes approach to PIR does not give the optimal scheme. In [9] it was shown how secret sharing schemes can be converted to optimal schemes that perform the same as those presented in [8]. The approach works by taking the secret sharing scheme as an input, "refining" it so that it can be

improved for $M = 2$ files, and then "lifting" the scheme to accommodate a general number of files M.

A. Refining

We begin by showing how to refine the scheme in Example 1. This scheme can be represented by Table I.

Server 1	Server 2
R	$e_i + R$

TABLE I: This table represents the scheme where a query R is sent to Server 1 and a query $e_i + R$ is sent to Server 2. Server 1 responds by sending the inner product $\langle D, R\rangle$ of the query he receives with his dataset D. In the same way, Server 2 responds with $\langle D, e_i+R\rangle$.

The entries below the servers denote the queries, and the servers respond with the inner product of these queries and the database. So, for example, Server 1 is queried with R and responds with $\langle D, R\rangle$ and Server 2 is queried with $e_i + R$ and responds with $\langle D, e_i + R\rangle$.

Example 4. Consider the PIR setting with $N = 2$ servers where a user wants to download a single file privately out of a total of $M = 2$ files. Denote the files by $W^1, W^2 \in \mathbb{F}_q^2$. Thus, the database $D = (W^1, W^2) \in \mathbb{F}_q^4$. Without loss of generality, suppose that the user is interested in the first file W^1.[6]

One can directly apply the scheme in Example 1 to obtain a PIR scheme which solves the problem with a download rate of $\frac{1}{2}$. Consider, however, the "refined" scheme in Table II.

Server 1	Server 2
a_1 b_1	$a_2 + b_1$

TABLE II: The refinement of Example 1.

The queries are chosen as follows.

- Queries $a_1, a_2 \in \text{span}(e_1, e_2) \subseteq \mathbb{F}_q^4$ are chosen uniformly at random, subject to being linearly independent.
- The query $b_1 \in \text{span}(e_3, e_4) \subseteq \mathbb{F}_q^4$ is chosen uniformly at random subject to being linearly independent, i.e., such that $b_1 \neq 0$ in this case.

Thus, a_1 queries only the first message, b_1 queries only the second message, and $a_2 + b_1$ queries both messages. From the answer $\langle D, a_1\rangle$ to the first query a_1, the user can obtain a linear combination of the two entries W_1^1 and W_2^1 in W^1. Furthermore, by subtracting the answer $\langle D, b_1\rangle$ from the answer $\langle D, a_2 + b_1\rangle$, the user can obtain $\langle D, a_2 + b_1\rangle - \langle D, b_1\rangle = \langle D, a_2\rangle$, thus obtaining another combination of the two entries W_1^1 and W_2^1 in W^1. Since these two combinations are linearly independent, the user is able to decode W^1.

The scheme is private with respect to Server 1 because the queries he observes, a_1 and b_1, are symmetric with respect to either file W^1 or W^2, i.e., from the server's point of view he observes two nonzero queries, one which queries the first file and one which queries the second file. The scheme is private with respect to Server 2 because again, the query $a_2 + b_1$ is again symmetric with respect to either file.

The download rate for the scheme is $\frac{2}{3}$, as the user obtains the two symbols in $W^1 \in \mathbb{F}_q^2$ by downloading a total of three symbols. This rate is larger than the secret sharing scheme's $\frac{1}{2}$ and is, indeed, the optimal download rate achievable for its parameters.

Note that b_1 in Example 4 plays a similar role as R in Example 1. The main idea behind refinement is to use the same structure of the secret sharing scheme but substituting the random keys with linearly independent queries to the message the user is not interested in.

Let us consider an example with collusion.

Example 5. Consider the PIR setting with $N = 3$ servers where a user wants to download a single file privately out of a total of $M = 2$ files. Denote the files by $W^1, W^2 \in \mathbb{F}_q^3$. Thus, the database $D = (W^1, W^2) \in \mathbb{F}_q^6$. Also, assume that at most $T = 2$ servers collude. Without loss of generality, suppose that the user is interested in the first file W^1. One can directly apply the following secret sharing scheme to the problem.

Server 1	Server 2	Server 3
R_1	R_2	$e_i + R_1 + R_2$

TABLE III: Secret sharing PIR scheme for Example 5.

This scheme achieves a download rate of $\mathcal{R} = \frac{1}{3}$. The refinement of this scheme is shown in Table IV.

[6] If the user is interested in another file, just rename the files.

Server 1	Server 2	Server 3
a_1	a_2	$a_3 + b_1 + b_2$
b_1	b_2	

TABLE IV: The refinement of Table III.

The queries are chosen as follows.

- Queries $a_1, a_2, a_3 \in \text{span}(e_1, e_2, e_3) \subseteq \mathbb{F}_q^6$ are chosen uniformly at random, subject to being linearly independent.
- The queries $b_1, b_2 \in \text{span}(e_4, e_5, e_6) \subseteq \mathbb{F}_q^6$ are chosen uniformly at random subject to being linearly independent.

Note that decoding the refined scheme is equivalent to decoding the secret sharing scheme. And the privacy of the refined scheme follows straightforwardly from that of the secret sharing scheme. The download rate of the refined scheme is then $\frac{3}{5}$. This rate is larger than the secret sharing scheme's $\frac{1}{3}$ and is, indeed, the optimal download rate achievable.

In general, we can refine an $(N, N-T, T)$-ramp secret sharing scheme with download rate $1 - \frac{T}{N}$ to obtain a refined PIR scheme with optimal download rate $\frac{(N-T)N}{N^2-T^2}$.

B. *Lifting*

Now, we explain how to "lift" a refined PIR scheme to get the optimal scheme for any number of files. First, let us lift the scheme from Example 4 to three files.

Example 6. Consider the PIR setting with $N = 2$ servers where a user wants to download a single file privately out of a total of $M = 3$ files. Denote the files by $W^1, W^2, W^3 \in \mathbb{F}_q^4$.[7] Thus, the database $D = (W^1, W^2, W^3) \in \mathbb{F}_q^{12}$. Without loss of generality, suppose that the user is interested in the first file W^1. In Table VI we show a direct extension of the scheme in Table II.

Server 1	Server 2
a_1	$a_2 + b_1$
b_1	$a_3 + c_1$
c_1	

TABLE V: Direct extension of Table II.

[7]The size of each file is now four instead of two, as in Example 4. In general, the more we lift, the larger the files must be. As stated in Section III, we are interested in the regime with large files, as this is the regime where the download rate dominates.

In Table VI, the queries a_1, a_2, a_3 only query the first file, b_1 only query the second file, and c_1 only query the third file. However, the issue with this scheme is that it is not private. The reason for this is that Server 2 receives two queries, one of them querying the first and second files, $a_2 + b_1$, and the other querying the first and third files, $a_3 + c_1$. This asymmetry with respect to the first file reveals to Server 2 that the user is interested in the first file.

This can be solved by adding an additional query to Server 2 of the form $b_2 + c_2$ and, indeed, this would suffice for privacy. However, we can improve the download rate by using this query to obtain an extra bit of information. This can be achieved by querying $a_4 + b_2 + c_2$ to Server 1. Thus, we obtain the lifted PIR scheme in Table VI.

Server 1	Server 2
a_1	$a_2 + b_1$
b_1	$a_3 + c_1$
c_1	$b_2 + c_2$
$a_4 + b_2 + c_2$	

TABLE VI: Lifting Table II to three files.

The queries are chosen as follows.

- Queries $a_1, \ldots, a_4 \in \text{span}(e_1, \ldots, e_4) \subseteq \mathbb{F}_q^{12}$ are chosen uniformly at random, subject to being linearly independent.
- Queries $b_1, b_2 \in \text{span}(e_5, \ldots, e_8) \subseteq \mathbb{F}_q^{12}$ are chosen uniformly at random subject to being linearly independent.
- Queries $c_1, c_2 \in \text{span}(e_9, \ldots, e_{12}) \subseteq \mathbb{F}_q^{12}$ are chosen uniformly at random subject to being linearly independent.

The privacy of the scheme follows from the symmetry with respect to each file from both server's point of view. The download rate for the scheme is $\frac{4}{7}$. This rate is larger than the secret sharing scheme's $\frac{1}{2}$ and is, indeed, the optimal download rate achievable. Note, however, that it is worse than the refined download rate for two files $\frac{3}{5}$. Indeed, this is generally true, i.e., as we increase the number of files, the optimal scheme will perform worse, quickly approaching the rate of the secret sharing scheme.

Let us understand how we go from the scheme in Table II to that of Table VI. The main idea comes from the secret sharing scheme in Table I. This secret sharing scheme essentially tells us that a user can utilize a *noise*

query R from one server to hide a *useful* query e_i by sending a *mixed* query $e_i + R$ to another server. The same idea is repeated in the scheme in Table II where the noise query b_1 is used to hide the useful query a_2 by sending the mixed query $a_2 + b_1$ to Server 2.

The same idea is used again in Table VI. The *noise* queries b_1 and c_1 are used to hide useful queries a_2 and a_3 via mixed queries $a_2 + b_1$ and $a_3 + c_1$. But now we have a noise query that is the mixture of two messages $b_2 + c_2$, which can also be used to hide a useful query a_4 through the mixed query $a_4 + b_2 + c_2$.

We now present a systematic way to describe what is going on. But first we need to define what a d-query is. A d-query is a sum of d queries, each querying a specific file. For example, in Table VI, a_1 is a 1-query, $a_3 + c_1$ is a 2-query, and $a_4 + b_2 + c_2$ is a 3-query.

Consider the scheme in Table II. We represent its structure by means of the *symbolic matrix* $\mathcal{S}_2 = \begin{pmatrix} 1 & 2 \end{pmatrix}$.[8] Each column of $\mathcal{S}_2$ corresponds to a server. A value of 1 in the first column represents sending all possible combinations of 1-queries of every message to Server 1, and a value of 2 in the second column represents sending all combinations of 2-queries of every message to Server 2. Thus, from $\mathcal{S}_2$ we obtain the following structure.

Server 1	Server 2
a_1	$a_2 + b_2$
b_1	

TABLE VII: Query structure obtained from $\mathcal{S}_2$ for a PIR scheme with two non-colluding servers and two files.

The next step is to impose relations between the queries that the user is not interested in. Assuming that the user is interested in the first file, this would correspond to a lack of interest in the queries b_1 and b_2. The relations are taken to mirror those of the secret sharing scheme in Table 1, i.e., $b_2 = b_1$. Performing this substitution and then requiring all queries to be uniformly chosen but linearly independent, we obtain the optimal scheme for two messages in Table II.

The scheme in Table VI has the symbolic matrix $\mathcal{S}_3 = \begin{pmatrix} 1 & 2 \\ 3 & \end{pmatrix}$, where we omit zeroes. It is straightforward to check that the same approach outlined for $\mathcal{S}_2$ is able to retrieve the scheme in Table VI from $\mathcal{S}_3$.

Thus, for Table VI, the 1-query in Server 1 gives rise to the 2-queries in Server 2. In the same way, the 2-queries in Server 2 gives rise to the 3-query in Server 1. The extension to a general scheme is straightforward. For $M = 4$ files, the structure of the scheme is $\mathcal{S}_4 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$. Applying the same reasoning before, we obtain the following scheme.

Server 1	Server 2
a_1	$a_2 + b_1$
b_1	$a_3 + c_1$
c_1	$a_4 + d_1$
d_1	$b_2 + c_2$
	$b_3 + d_2$
	$c_3 + d_3$
$a_5 + b_2 + c_2$	$a_8 + b_4 + c_4 + d_4$
$a_6 + b_3 + d_2$	
$a_7 + c_3 + d_3$	
$b_4 + c_4 + d_4$	

TABLE VIII: Query structure obtained from $\mathcal{S}_4$ for a PIR scheme with two non-coluding servers and four files.

Here, the queries are chosen analogously to previous examples. This scheme is optimal for $M = 4$ files and has a download rate of $\frac{8}{15}$. The symbolic matrix for $\mathcal{S}_M$ is then a matrix where the first column consists of all odd positive integers less than or equal to M and the right column of all even ones.

Let us now show what happens when there are collusions. The scheme in Table IV has the symbolic matrix $\mathcal{S}_2 = \begin{pmatrix} 1 & 1 & 2 \end{pmatrix}$. This occurs because, for $T = 2$ collusions, the secret sharing scheme in Table III now needs to utilize $T = 2$ noise queries to obtain a useful query. Thus, two 1's appear in the symbolic matrix $\mathcal{S}_2$ to obtain a 2. To lift this to $M = 3$ messages, we need to have two 2's in the symbolic matrix $\mathcal{S}_3$ to obtain a 3. This can be done by having the second row of $\mathcal{S}_3$ be the same as the first row, but shifted. The final product is $\mathcal{S}_3 = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \\ 3 & & \end{pmatrix}$. One can then obtain the following scheme.

[8] In general, the symbolic matrix depends on the amount of servers and collusion. We omit this dependency to avoid overloading our notation.

Server 1	Server 2	Server 3
a_1	a_2	$a_3+b_1+b_2$
b_1	b_2	$a_4+c_1+c_2$
c_1	c_2	b_3+c_3
a_5	$a_7+b_4+b_5$	a_6
b_4	$a_8+c_4+c_5$	b_5
c_4	b_6+c_6	c_5
$a_9+b_3+b_6+c_3+c_6$		

TABLE IX: Query structure obtained from $\mathcal{S}_3$ for a PIR scheme with three servers, of which at most two collude, and two files.

The queries are chosen analogously to previous examples. This scheme is optimal for $M=3$ files and has a download rate of $\frac{9}{19}$.

In general, for N servers with at most T collusions, the symbolic matrix for $M=2$ files takes the form $\mathcal{S}_2=(\overbrace{1,\ldots,1}^{T},\overbrace{2,\ldots,2}^{N-T})$.The symbolic matrix for $M=3$ files is then obtained analogously to the examples above. If we divide the total amount queries with an a_*, given by N^{M-1}, and divide it by the total amount of queries $\frac{N^M-r^M}{N-r}$, we obtain the optimal download rate $\frac{(N-T)N^{M-1}}{N^M-T^M}$.

VII. Coded PIR

In this section, we discuss coded PIR schemes from multiple servers that may potentially collude. Contrary to a replicated database, the data storage employs a fixed erasure code, typically a Maximum Distance Separable (MDS) code. We present a detailed example below.

Example 7 (Coded Data)**.** Consider a scenario where $M=2$ messages, $(X_1,X_2),(Y_1,Y_2)\in\mathbb{F}_5^2$ are stored using a $(4,2)$-MDS code as illustrated in Table X.

Server 1	Server 2	Server 3	Server 4
X_1	X_2	X_1+X_2	X_1+2X_2
Y_1	Y_2	Y_1+Y_2	Y_1+2Y_2

TABLE X: Data stored in a $(4,2)$-MDS code.

Suppose that the user is interested in the first message, $X=(X_1,X_2)\in\mathbb{F}_5^2$, and that at most $T=2$ servers collude. A secret-sharing-like scheme designed to allow PIR on the data coded as in Table X, as proposed in [22], is presented in Table XII.

Server 1	Server 2	Server 3	Server 4
R	S	$R+S$	e_1+R+2S

TABLE XI: Query structure for Example 7.

The queries in Table XI are defined as follows:

- The vectors $R,S\in\mathbb{F}_5^2$ are uniformly and independently distributed.
- The vector e_1 is the first vector of the standard basis of $\mathbb{F}_5^2$.

This scheme is private, as the queries for any two servers are uniformly and independently distributed.

Let us move on to decoding. If we denote the data in Server i by the vector $D_i\in\mathbb{F}_5^2$, the first server's response is $\langle D_1,R\rangle=X_1R_1+Y_1R_2$, the second is $\langle D_2,S\rangle=X_2S_1+Y_2S_2$, the third is

$$\langle D_3,R+S\rangle=(X_1+X_2)(R_1+S_1)+(Y_1+Y_2)(R_2+S_2),$$

and the fourth is

$$\langle D_4,e_1+R+2S\rangle=(X_1+2X_2)(1+R_1+2S_1)+(Y_1+2Y_2)(R_2+2S_2).$$

A direct computation then gives us

$$0=\langle D_1,R\rangle+3\langle D_2,S\rangle+3\langle D_3,R+S\rangle+\langle D_4,R+2S\rangle. \quad (1)$$

Thus, from the servers' responses we obtain

$$X_1+2X_2=\langle D_1,R\rangle+3\langle D_2,S\rangle+3\langle D_3,R+S\rangle+\langle D_4,e_1+R+2S\rangle.$$

Thus, the user is able to privately obtain a linear combination of X_1 and X_2. To decode, the user needs another combination. This can be obtained by repeating the scheme but with the e_1 vector appearing in the third query as shown in Table XII.

Server 1	Server 2	Server 3	Server 4
R'	S'	$e_1+R'+S'$	$R'+2S'$

TABLE XII: Query structure for Example 7.

The queries in Table XII are defined as follows:

- The vectors $R',S'\in\mathbb{F}_5^2$ are uniformly and independently distributed.
- The vector e_1 is the first vector of the standard basis of $\mathbb{F}_5^2$.

The user then computes

$$X_1 + X_2 = \langle D_1, R\rangle + 3\langle D_2, S\rangle + 3\langle D_3, e_1 + R + S\rangle + \langle D_4, R + 2S\rangle.$$

With the knowledge of the values of both X_1+2X_2 and X_1+X_2, the user can now obtain the first message $X = (X_1, X_2)$. To retrieve two desired symbols, the user had to download eight symbols in total. Consequently, the rate of this PIR scheme is $\mathcal{R} = \frac{1}{4}$.

The key equation in Example 7 is (1). Explanations on how to construct queries which satisfy these types of equations can be found in [10]. The refining approach described in Section VI-A can also be used for coded data, as shown in the next example.

Example 8 (Refined Scheme on Coded Data)**.** To refine the scheme in Example 7 we need the files to be larger. Now $X_1, X_2, Y_1, Y_2 \in \mathbb{F}_5^2$. Thus, the first message is now $(X_1, X_2) \in \mathbb{F}_5^4$. Refining the scheme in Example 7 we get the following scheme.

Server 1	Server 2	Server 3	Server 4
a_1	a_2	$a_1 + a_2$	$2a_1 + a_2 + b_1 + 2b_2$
b_1	b_2	$b_1 + b_2$	

TABLE XIII: Query structure for Example VII.

The queries in Table XIII are defined as follows:

- The queries $a_1, a_2 \in \text{span}(e_1, e_2) \subseteq \mathbb{F}_5^2$ are chosen uniformly at random subject to being linearly independent.
- The queries $b_1, b_2 \in \text{span}(e_3, e_4) \subseteq \mathbb{F}_5^2$ are chosen uniformly at random subject to being linearly independent.

This scheme is private since for any two colluding servers, the queries querying each file are indistinguishable. For example, if Servers 1 and 3 collude, they observe two linearly independent 1-queries of the first file, a_1 and $a_1 + a_2$, and two linearly independant 1-queries of the second file, b_1 and b_1+b_2. Alternatively, if Servers 3 and 4 collude, they observe a 1-query of each file, a_1+a_2 and b_1+b_2, and a 2-query $2a_1+a_2+b_1+2b_2$ that is such that the part that queries the first file $2a_1+a_2$ is linearly independant from the 1-query $a_1 + a_2$ and the part that queries the second file $b_1 + 2b_2$ is linearly independant from the 1-query $b_1 + b_2$.

To decode, we note that the queries which query the second file satisfy the same key identity as (1), i.e.,

$$0 = \langle D_1, b_1\rangle + 3\langle D_2, b_2\rangle + 3\langle D_3, b_1 + b_2\rangle + \langle D_4, b_1 + 2b_2\rangle.$$

This combination can then be used by the user to obtain from the fourth server

$$\langle X_1 + 2X_2, 2a_1 + a_2\rangle = \langle D_1, b_1\rangle + 3\langle D_2, b_2\rangle + 3\langle D_3, b_1 + b_2\rangle + \langle D_4, 2a_1 + a_2 + b_1 + 2b_2\rangle.$$

Combining this with $\langle X_1, a_1\rangle$, obtained from the first server, $\langle X_2, a_2\rangle$, obtained from the second server, and $\langle X_1+2X_2, 2a_1+a_2\rangle$, obtained from the third server, the user can obtain all four $\langle X_i, a_j\rangle$, for $i, j \in \{1, 2\}$. Since a_1 and a_2 are linearly independent, the user can obtain the first message X. To retrieve four desired symbols, the user had to download seven symbols in total. Thus, the rate of this PIR scheme is $\frac{4}{7}$.

In general, any secret-sharing-like scheme like that in [10] can be refined as described in Example . The lifting approach described in Section VI-B can then be readily applied for coded data. The scheme in Example VII has a symbolic matrix of $\mathcal{S}_2 = (1, 1, 1, 2)$, i.e. Servers 1-3 get 1-queries, and then Server 4 gets a 2-query. This symbolic matrix can be lifted to three messages in the same way as described in Section VI-B, following the rule that three k-queries are needed to obtain a $(k+1)$-query. Thus, $\mathcal{S}_3 = \begin{pmatrix} 1 & 1 & 1 & 2 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 1 & 1 \\ 3 & & & \end{pmatrix}$ obtaining the query structure in Table XIV.

Server 1	Server 2	Server 3	Server 4
a_1	a_2	a_3	$a_4 + b_4$
b_1	b_2	b_3	$a_5 + c_4$
c_1	c_2	c_3	$b_5 + c_5$
a_7	a_8	$a_9 + b_9$	a_6
b_7	b_8	$a_{10} + c_9$	b_6
c_7	c_8	$b_{10} + c_{10}$	c_6
a_{13}	$a_{14} + b_{14}$	a_{11}	a_{12}
b_{13}	$a_{15} + c_{14}$	b_{11}	b_{12}
c_{13}	$b_{15} + c_{15}$	c_{11}	c_{12}
$a_{16} + b_{16} + c_{16}$			

TABLE XIV: Lifted version of Table XIII for 3 files.

This scheme achieves a download rate of $\frac{16}{37}$.

Details of the precise choices of the query distributions are shown in [11, Appendix B]. The main idea is to choose b's and c's so that they satisfy an equation analogous to (1). The a's have to be chosen carefully so that they are indistinguishable from the b's and c's to any $T = 2$ colluding servers, but also so that the first message can be decoded.

Thus, our approach to T-secure coded PIR where the private data consisting of M files is split among N servers utilizing an (N, K)-MDS code, can be split into two parts. An algebraic one, which involves finding an equation of the form (1). And a combinatorial one, which involves refining and lifting. The algebraic equation tells us how many useless queries r can be transformed into useful queries $N - r$. We call r the chain size of the PIR scheme. The smaller r is, the higher the communication rate. Indeed, the communication rate is given by $\frac{(N-r)(N^{M-1})}{N^M - r^M}$. The best known value for the chain is $r = K + T - 1$ obtained by the secret-sharing-like scheme in [10].

VIII. Final Remarks

The best known general family of schemes for (N, K)-MDS coded PIR with N servers, of which at most T collude, and M files, can be obtained by refining and lifting [9], the secret-sharing-like schemes presented in [10]. These schemes achieve a download rate of $\frac{(N-K-T+1)N^{M-1}}{N^M-(K+T-1)^M}$ and are capacity achieving when $K = 1$ [8] or $T = 1$ [12]. These schemes also achieve capacity under certain restrictions [11]. In general, however, they are not capacity achieving [13]. Indeed, the capacity of PIR with MDS coded data and collusion remains an open question.

An interesting line of research is to generalize the scheme presented in [13]. Another line is to extend the application of the refine and lift technique to other settings of the PIR such as those described in Section IX, e.g., to the setting in [23]. Overall, the goal of this article is to present an accessible introduction to information-theoretic coded PIR. Many problems remain unsolved, and we believe that new players might be able to bring insight into the field.

IX. Related Work

The literature on PIR is now vast and spans both the computer science (CS) and Information Theory (IT) community. Our goal in this section is to give references to the reader to the main trends and line of work in the PIR literature, which can be used as pointers for further investigation.

PIR was first introduced in the seminal papers of Chor et al. in [1, 2] and has since received significant attention, originally from the CS community (see, e.g., [24]–[32]) and later in the IT community. In the CS formulation of the PIR problem both the upload and the download communication cost count, while in the IT formulation only the download matters. This results from the fact that in CS the goal is to retrieve a single bit, while IT considers applications where one wants to retrieve a large file, and thus the upload cost is negligible. Moreover, in the original formulation, information-theoretic privacy was sought after, and the data was assumed to be replicated on multiple non-colluding servers. The objective was to devise PIR schemes with a low total communication cost (upload + download). The initial scheme by Chor et al. [1, 2] had total communication cost of order $O((N^2 \log N)M^{1/N})$ (and $O(M^{1/3})$ for the special case of $N = 2$), where m is the length of the database. The PIR scheme with the best communication cost was $O(M^{1/2N-1})$ [33] until recently when a series of works constructed PIR with total communication cost that is sub-polynomial in M [27, 29, 31, 34]. The idea is to represent the retrieval operation as an evaluation of a carefully designed multivariate polynomial. In terms of lower bounds on the total communication cost, very little is known beyond small improvements on the straightforward bound of $\log M$ [35, 36]. The other overheads of PIR have received less attention. For example, schemes that amortize the computational complexity of PIR across many queries have been devised based either on pre-computations [25] or on batch codes [37].

Another line of work focused on computational privacy for PIR [14, 38], which is based on the assumed hardness of certain mathematical problems. Computational PIR schemes with low communication cost were first constructed in [14]. What was surprising there is that these schemes do not require data replication and can be used with a single server. The downside was that they suffered from a high computational overhead [39].

The past ten years have witnessed renewed interest in PIR. One of the drivers of this interest is recent progress on novel codes for distributed storage. A natural question thus arises here: How to design an efficient PIR scheme that can work with data that is stored on multiple servers in coded and not only

replicated form, which is the main question we tried to address in this article. Many research groups have worked on tackling different aspects of this problem, which we refer to as coded PIR. Previous work explored the savings in communication and storage costs and the trade-offs between the two, brought about by using codes instead of replication [3, 4, 40].

A distinguishing aspect of the work on coded PIR is the focus on minimizing the download cost, as justified earlier this paper. One can then define the PIR capacity, which is, informally, the supremum downloaded rate for privately retrieving a file. Unlike fundamental limits on the total communication cost, significant progress has been made in characterizing the PIR capacity. Originally, the work of Sun and Jafar [8] was able to characterize the PIR capacity for replicated data on N servers and T collusion. A result that was later generalized to other settings, e.g., MDS coded data [12], multi-message PIR in [41], symmetric PIR in [42, 43] and multi-round PIR [44], and private search [45].

MDS codes, and in particular Reed-Solomon (RS) codes, are the most popular codes used to achieve reliability and fault-tolerance in distributed storage. Therefore, an important question here is to design PIR schemes that can work with MDS coded data. As a first approach, one can separate the reliability constraint from the privacy constraint. This can be achieved by assuming the MDS code is fixed and designing efficient coded PIR schemes, as done in the examples above. The work of [22, 46] presented explicit PIR schemes for linear MDS coded data. In the case of no collusion, these codes achieve the asymptotic PIR capacity for MDS coded data. Freij-Hollanti et al. presented in [10] PIR schemes with improved rates, namely $\frac{N-(K+T-1)}{N}$, which can withstand any number of colluding servers and where the data is stored using Generalized Reed-Solomon codes.

In [10] it was conjectured that the MDS coded PIR capacity that can protect against T colluding servers, with $1 \leq T \leq N-K$ is $\frac{(N-K-T+1)N^{M-1}}{N^M-(K+T-1)^M}$. This conjecture was later disproved in [47]. In [9] it was shown that PIR schemes that can achieve asymptotic capacity can be transformed into ones that achieve the conjectured rate through the refine and lift operations described in this paper. Interestingly, one of the immediate results of the refine and lift operations is that the first capacity-achieving schemes of [8] can be obtained as refining and lifting the well-known Shamir secret sharing scheme. The capacity of PIR with MDS coded data and collusion is still open.

The work on MDS-coded PIR has also motivated the development of PIR schemes that can work with codes beyond MDS [48]–[50]. Note that this separation between code and PIR design is not optimal. It was shown in [51] that one can achieve PIR schemes with a lower download cost by jointly designing the storage code and the PIR scheme.

Another line of work investigated the role that side information [15] or caching [52] can play in PIR. In [15], it was shown that single-server PIR can be achieved without having to download everything (i.e., a $\frac{1}{M}$ rate). In this case, the PIR capacity was shown to be $\frac{S+1}{M}$, where S is the number of messages that the user has as side information. Different variants on this setting and information-theoretic bounds were further investigated in [53]–[55].

Other studies have also looked at relaxing privacy requirements in PIR to improve performance [56, 57]. In this setting, the case where the data are stored as MDS codes has also been considered [58].

In terms of applications in practice, the PIR with side information schemes in [15] were combined with computational PIR and implemented in a real-world system at Google [59]. Recent applications have focused on the use of PIR to achieve privacy in federated learning applications [60, 61].

The concept of PIR was generalized to private computations, where one wants to compute a private function of the data. The case of linear functions was considered in [62], that of polynomials in [62, 63], and for multi-message in [64]. The fundamental bounds of private computations were studied in [65]. Connections between PIR and secure distributed matrix multiplication (SDMM) [66] were shown in [67, 68]. In the other direction, SDMM techniques utilizing a combinatorial tool called *degree tables* [69, 70] were utilized to obtain secret-sharing-like schemes for MDS coded PIR [71].

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*. IEEE Computer Society, 1995, pp. 41–41.

[2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.

[3] N. B. Shah, K. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 856–860.

[4] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 2842–2846.

[5] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[6] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, International Workshop on*. IEEE Computer Society, 1979, pp. 313–313.

[7] G. R. Blakley and C. Meadows, "Security of ramp schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1984, pp. 242–268.

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

[9] R. G. L. D'Oliveira and S. El Rouayheb, "One-shot PIR: Refinement and lifting," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2443–2455, 2019.

[10] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 647–664, 2017.

[11] L. Holzbaur, R. Freij-Hollanti, J. Li, and C. Hollanti, "Toward the capacity of private information retrieval from coded and colluding servers," *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 517–537, 2021.

[12] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.

[13] H. Sun and S. A. Jafar, "Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti et al." *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1000–1022, 2017.

[14] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 1997, pp. 364–364.

[15] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, "Private information retrieval with side information," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2032–2043, 2019.

[16] S. M. Bellovin, "Frank miller: Inventor of the one-time pad," *Cryptologia*, vol. 35, no. 3, pp. 203–222, 2011.

[17] C. E. Shannon, "Communication theory of secrecy systems," *The Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[18] ——, "A mathematical theory of cryptography," *Bell System Technical Memo MM 45-110-02*, 1945.

[19] ——, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[20] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[21] R. J. McEliece and D. V. Sarwate, "On sharing secrets and Reed-Solomon codes," *Communications of the ACM*, vol. 24, no. 9, pp. 583–584, 1981.

[22] R. Tajeddine and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 1411–1415.

[23] Z. Jia and S. A. Jafar, "X-secure T-private information retrieval from MDS coded storage with byzantine and unresponsive servers," *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7427–7438, 2020.

[24] S. Yekhanin, "Private information retrieval," *Communications of the ACM*, vol. 53, no. 4, pp. 68–73, 2010.

[25] A. Beimel, Y. Ishai, and T. Malkin, "Reducing the servers computation in private information retrieval: PIR with preprocessing," in *Advances in Cryptology—CRYPTO 2000*. Springer, 2000, pp. 55–73.

[26] A. Beimel, Y. Ishai, and E. Kushilevitz, "General constructions for information-theoretic private information retrieval," *Journal of Computer and System Sciences*, vol. 71, no. 2, pp. 213–247, 2005.

[27] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond, "Breaking the o (n/sup 1/(2k-1)/) barrier for information-theoretic private information retrieval," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE, 2002, pp. 261–270.

[28] W. Gasarch, "A survey on private information retrieval," in *Bulletin of the EATCS*. Citeseer, 2004.

[29] Z. Dvir and S. Gopi, "2-server PIR with sub-polynomial communication," *arXiv preprint arXiv:1407.6692*, 2014.

[30] D. Woodruff and S. Yekhanin, "A geometric approach to information-theoretic private information retrieval," in *Twentieth Annual IEEE Conference on Computational Complexity, 2005.* IEEE, 2005, pp. 275–284.

[31] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *Journal of the ACM (JACM)*, vol. 55, no. 1, p. 1, 2008.

[32] ——, "Private information retrieval," *Communications of the ACM*, vol. 53, no. 4, pp. 68–73, 2010.

[33] A. Ambainis, "Upper bound on the communication complexity of private information retrieval," in *Automata, Languages and Programming*. Springer, 1997, pp. 401–407.

[34] K. Efremenko, "3-query locally decodable codes of subexponential length," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1694–1703, 2012.

[35] S. Wehner and R. De Wolf, "Improved lower bounds for locally decodable codes and private information retrieval," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2005, pp. 1424–1436.

[36] A. A. Razborov and S. Yekhanin, "An omega (nˆ 1/3) lower bound for bilinear group based private information retrieval," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 2006, pp. 739–748.

[37] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 262–271.

[38] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 402–414.

[39] R. Sion and B. Carbunar, "On the computational practicality of private information retrieval," in *In Proceedings of the Network and Distributed Systems Security Symposium*, 2007.

[40] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 2852–2856.

[41] K. Banawan and S. Ulukus, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Transactions on Information Theory*, vol. 64, no. 10, pp. 6842–6862, 2018.

[42] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 322–329, 2018.

[43] Q. Wang and M. Skoglund, "On PIR and symmetric PIR from colluding databases with adversaries and eavesdroppers," *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 3183–3197, 2018.

[44] H. Sun and S. A. Jafar, "Multiround private information retrieval: Capacity and storage overhead," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5743–5754, 2018.

[45] Z. Chen, Z. Wang, and S. A. Jafar, "The asymptotic capacity of private search," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4709–4721, 2020.

[46] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, 2018.

[47] H. Sun and S. A. Jafar, "Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by freij-hollanti et al." *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1000–1022, 2017.

[48] D. Augot, F. Levy-Dit-Vehel, and A. Shikfa, "A storage-efficient and robust private information retrieval scheme allowing few servers," in *Cryptology and Network Security*. Springer, 2014, pp. 222–239.

[49] S. Kumar, H.-Y. Lin, E. Rosnes, and A. G. i Amat, "Achieving maximum distance separable private information retrieval capacity with linear codes," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4243–4273, 2019.

[50] H.-Y. Lin, S. Kumar, E. Rosnes, and A. G. i Amat, "Asymmetry helps: Improved private information retrieval protocols for distributed storage," in *2018 IEEE Information Theory Workshop (ITW)*. IEEE, 2018, pp. 1–5.

[51] H. Sun and C. Tian, "Breaking the MDS-PIR capacity barrier via joint storage coding," *Information*, vol. 10, no. 9, p. 265, 2019.

[52] R. Tandon, "The capacity of cache aided private information retrieval," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 1078–1082.

[53] Y.-P. Wei, K. Banawan, and S. Ulukus, "Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 3215–3232, 2018.

[54] S. Li and M. Gastpar, "Single-server multi-user private information retrieval with side information," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1954–1958.

[55] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "The role of coded side information in single-server private information retrieval," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 25–44, 2021.

[56] H.-Y. Lin, S. Kumar, E. Rosnes, A. G. i Amat, and E. Yaakobi, "Weakly-private information retrieval," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 1257–1261.

[57] I. Samy, R. Tandon, and L. Lazos, "On the capacity of leaky private information retrieval," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 1262–1266.

[58] A. O. Orvedal, H.-Y. Lin, and E. Rosnes, "Weakly-private information retrieval from MDS-coded distributed storage," *arXiv preprint arXiv:2401.09412*, 2024.

[59] S. Patel, G. Persiano, and K. Yeo, "Private stateful information retrieval," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1002–1019.

[60] Z. Jia and S. A. Jafar, "X-secure t-private federated submodel learning with elastic dropout resilience," *IEEE Transactions on Information theory*, vol. 68, no. 8, pp. 5418–5439, 2022.

[61] S. Vithana and S. Ulukus, "Private read update write (pruw) in federated submodel learning (fsl): Communication efficient schemes with and without sparsification," *IEEE Transactions on Information Theory*, 2023.

[62] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private polynomial function computation for noncolluding coded databases," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1800–1813, 2022.

[63] N. Raviv and D. A. Karpuk, "Private polynomial computation from lagrange encoding," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 553–563, 2019.

[64] A. Gholami, K. Wan, H. Sun, M. Ji, and G. Caire, "On multi-message private computation," *arXiv preprint arXiv:2305.05332*, 2023.

[65] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2018.

[66] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[67] R. G. L. D'Oliveira, S. El Rouayheb, D. Heinlein, and D. Karpuk, "Notes on communication and computation in secure distributed matrix multiplication," in *2020 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2020, pp. 1–6.

[68] Z. Jia and S. A. Jafar, "On the capacity of secure distributed batch matrix multiplication," *IEEE Transactions on Information Theory*, vol. 67, no. 11, pp. 7420–7437, 2021.

[69] R. G. L. D'Oliveira, S. El Rouayheb, and D. Karpuk, "Gasp codes for secure distributed matrix multiplication," *IEEE Transactions on Information Theory*, vol. 66, no. 7, pp. 4038–4050, 2020.

[70] R. G. L. D'Oliveira, S. El Rouayheb, D. Heinlein, and D. Karpuk, "Degree tables for secure distributed matrix multiplication," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 907–918, 2021.

[71] F. Kazemi, N. Wang, R. G. L. D'Oliveira, and A. Sprintson, "Degree tables for private information retrieval," in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022, pp. 1–8.