Private Information Retrieval over Random Linear Networks

Razane Tajeddine*, Antonia Wachter-Zeh†, Camilla Hollanti*

* Department of Mathematics and Systems Analysis, Aalto University School of Science, Espoo, Finland

Emails: {razane.tajeddine, camilla.hollanti}@aalto.fi

† Institute for Communications Engineering, Technical University of Munich, Germany

Email: antonia.wachter-zeh@tum.de

Abstract

In this paper, the problem of providing privacy to users requesting data over a network from a distributed storage system (DSS) is considered. The DSS, which is considered as the multi-terminal destination of the network from the user's perspective, is encoded by a maximum rank distance (MRD) code to store the data on these multiple servers. A private information retrieval (PIR) scheme ensures that a user can request a file without revealing any information on which file is being requested to any of the servers. In this paper, a novel PIR scheme is proposed, allowing the user to recover a file from a storage system with low communication cost, while allowing some servers in the system to collude in the quest of revealing the identity of the requested file. The network is modeled as a random linear network, *i.e.*, all nodes of the network forward random (unknown) linear combinations of incoming packets. Both error-free and erroneous random linear networks are considered.

I. Introduction

Privacy is a major concern for Internet or network users. Whenever a user downloads a file from a server, they reveal their interest in the requested file. Private information retrieval (PIR) allows a user to hide the identity of their requested file from the servers. PIR was first introduced in [1], where the data is assumed to be replicated on multiple servers and the user is able to retrieve the file she wants privately. It was also proved that if the data is stored on a single server, then the only way to achieve PIR in the *information theoretic* sense, *i.e.*, privacy with no restrictions on the computational power of the server, is to download all the files.

A large body of literature on PIR appeared after its introduction, mostly focusing on minimizing the communication cost. PIR schemes with subpolynomial communication cost were constructed for multiple servers in [2]–[4] and later for two servers in [5], calculating the communication cost as the sum of the upload and the download cost.

The work of R. Tajeddine and C. Hollanti was supported by the Academy of Finland, under Grants No. 276031, 282938, and 303819, and by the Technical University of Munich Institute for Advanced Study, funded by the German Excellence Initiative and the EU 7th Framework Programme under Grant Agreement No. 291763, via a Hans Fischer Fellowship. The work of A. Wachter-Zeh was supported by the Technical University of Munich Institute for Advanced Study, funded by the German Excellence Initiative and European Union 7th Framework Programme under Grant Agreement No. 291763 and the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Grant No. WA3907/1-1.

Later on, much work was done to construct PIR schemes aiming to reduce the download cost, assuming the upload cost to be negligible with respect to the download cost. This assumption is based on the fact that the size of the uploaded query vectors depends only on the *number* of files in the system, while the size of the downloaded response vectors depends on the *size* of the files. More precisely, for a single sub-query, the query vector to a node consists of m symbols in $\mathbb{F}_q$, where m is the number of files and $\mathbb{F}_q$ is the q-element finite field, while the response vector from one node is 1 symbol in $\mathbb{F}_q^w$, where w is the length of the file. In distributed storage systems (DSSs), and in the information-theoretic reformulation of this problem [6], the size of the files is assumed to be arbitrarily large, thus making the number of the files negligible with respect to the size of the files, *i.e.*, w is much larger than m. Therefore, the download cost dominates the total communication cost. These assumptions reflect practical scenarios such as storing and retrieving videos.

In more recent work on PIR, there is growing interest in studying PIR for coded data. In [7], it was shown that when the data is coded on an exponentially large number of servers, only one extra downloaded bit is needed to achieve privacy. The same low download complexity with a linear number of servers was achieved in [8]. Furthermore, a method to transform any linear replication-based PIR scheme into a scheme for coded data while minimizing the storage overhead was proposed in [9]. In addition to various constructions, the fundamental limits on the download cost of PIR schemes have been characterized for replicated data in [10] and for coded data in [11].

The servers can be either considered to not communicate with each other and are therefore only aware of the query they received from the user, referred to as a *non-colluding* system, or, it can be assumed that any set of maximum t servers are communicating in an effort to figure out the requested file's identity, referred to as a *colluding* system. We refer to the latter as t-*PIR*, meaning that any subset of up to t servers is allowed to collude. The capacity for t-PIR has been established in [10]. For coded t-PIR the capacity is still unknown, but a first scheme for t-PIR on coded servers employing a maximum distance separable (MDS) code was described in [12]. In [13], this scheme was extended to a wider set of parameters and an algebraic framework was established, also resulting in a conjecture for the coded t-PIR capacity, which was disproved for a two-file ($m = 2$) system in [14]. Asymptotically ($m \to \infty$), the conjectured capacity [13] matches the capacity of *symmetric* PIR [15]–[18], where the user can exclusively decode only the requested file. The work in [13] was later extended to non-MDS codes in [19], and arbitrary linear codes were also considered in [20]. Symmetric PIR schemes over storage systems with Byzantine and colluding servers are discussed [21]. In [22], the capacity of PIR schemes over a replicated storage systems with Byzantine and colluding servers was found. Noisy PIR over a storage system where the messages are replicated on multiple servers was discussed in [23].

In the present work, we construct a PIR scheme for a network with colluding nodes, where the data is encoded using a maximum rank distance (MRD) code [24]. The user sends the queries over a network of nodes to the destination nodes, which constitute a DSS, and the DSS servers send their responses over the random network to the user, such that the identity of the requested file remains secret.

We model the network as in random linear network coding (RLNC), a concept that was introduced after the seminal observation [25] that forwarding linear combinations of the incoming packets at each node (called *network coding*) instead of just forwarding packets (called *routing*), increases the throughput. In RLNC [26], the structure of the network is unknown to the sink(s) and receiver(s) and might even change from time to time. Each node of the network forwards a random linear combination of its incoming packets and the goal of a certain receiver is to recover some or all of the transmitted packets. When the packets are modeled as vectors over a finite field and we see the set of transmitted packets as a linear subspace, then, in the error-free case, the received packets form a subspace of the transmitted packets. Also, one injected erroneous packet might propagate widely in the network and destroy all of the received packets, but the received subspace still has a large intersection with the transmitted one. This observation by Kötter and Kschischang [26] motivated the use of *subspace codes* for error-correction in RLNC. *Lifted MRD codes* are special subspace codes that were proposed for RLNC in [24] and are efficiently decodable [27]–[29]. In this paper, we use MRD codes as the query and storage codes and lift the code before transmitting the query/response over the network to be able to guarantee PIR while coping with errors.

Contributions: The main contributions of this paper are as follows:

- To the best of the authors' knowledge, PIR schemes over a random linear network with data encoded using MRD codes is considered for the first time in this paper. We take this step towards a more practical model, compared to the earlier literature that assumes direct links between the user and the storage system when retrieving a file.
- Two PIR schemes are given, one assuming an error-free channel, and another assuming a network with errors , where the errors are considered in the uplink channel, as well as the downlink. To the best of the authors' knowledge uplink errors were not considered in any preceding work on PIR.
- The achieved PIR rate for an error-free network achieves the asymptotic PIR capacity when the field size is sufficiently large. For the network with errors, the scheme achieves the conjectured asymptotic PIR capacity given in [30].

II. Preliminaries

A. Notation

The following table provides an overview of the nomenclature used in this paper.

Let q be a power of a prime and let $\mathbb{F}_q$ denote the finite field of order q and $\mathbb{F}_{q^s}$ its extension field of order q^s. We use $\mathbb{F}_q^{s\times n}$ to denote the set of all $s \times n$ matrices over $\mathbb{F}_q$ and $\mathbb{F}_{q^s}^n = \mathbb{F}_{q^s}^{1\times n}$ for the set of all row vectors of length n over $\mathbb{F}_{q^s}$.

Rows and columns of $s \times n$-matrices are indexed by $1, \dots, s$ and $1, \dots, n$, respectively. Denote the set of integers $[a, b] = \{i : a \leq i \leq b\}$. By $\mathrm{rk}_q(A)$ and $\mathrm{rk}_{q^s}(A)$, we denote the rank of a matrix A over $\mathbb{F}_q$, respectively $\mathbb{F}_{q^s}$.

The following proposition will be useful in this work:

Proposition 1. *(Probability of full rank, [31]) Let $A \in \mathbb{F}_q^{\kappa\times\kappa}$ be a square matrix with elements chosen uniformly at random from $\mathbb{F}_q$. Then*

$$P = \mathbb{P}(\mathrm{rk}_q(A) = \kappa) \geq \left(1 - \frac{1}{q}\right)^{\kappa} .$$

B. Rank-Metric Codes and Gabidulin Codes

By utilizing the vector space isomorphism $\mathbb{F}_{q^s} \cong \mathbb{F}_q^s$, there is a bijective map from vectors $a \in \mathbb{F}_{q^s}^n$ to matrices $A \triangleq \Phi(a) \in \mathbb{F}_q^{s\times n}$. The rank distance between a and $b \in \mathbb{F}_{q^s}^n$ is the rank of the difference of the two matrix representations:

$$d_{\mathrm{R}}(a, b) \triangleq \mathrm{rk}_q(a - b) = \mathrm{rk}_q(A - B).$$

NOMENCLATURE

m	Number of files
l	Number of servers
n	Number of sub-servers in an $[n,k,d]_q^{\mathsf{R}}$ MRD code
k	Dimension of the codeword in an $[n,k,d]_q^{\mathsf{R}}$ MRD code
d	Minimum distance of an $[n,k,d]_q^{\mathsf{R}}$ code
t	Number of colluding nodes
$\rho = \frac{n}{l}$	Columns stored on a server
β	Number of subdivisions / stripes
X	Set of files stored in the system
X^f	File requested by the user
R_{PIR}	PIR rate
Y_j	Data stored on server j
$Q^{f,(i)}$	Query matrix sent in round i to retrieve file X^f
$Q^{f,(i)}_{j,rec}$	Query matrix received by server j in round i to retrieve file X^f
$R^{f,(i)}_{j,rec}$	Received response matrix from server j in round i
sk	File size
ϵ	Number of errors introduced in the network
τ	Number of erasures in the network
μ	$m\beta/s$

An $[n,k,d]_q^{\mathsf{R}}$ code $\mathcal{C}$ over $\mathbb{F}_{q^s}$ is a linear rank-metric code, *i.e.*, it is a linear subspace of $\mathbb{F}_{q^s}^n$ of dimension k and minimum rank distance d. For linear codes with $n \leq s$, the Singleton-type upper bound [32], [33] implies that $d \leq n-k+1$. If $d = n-k+1$, the code is called a *maximum rank distance* (MRD) code.

Gabidulin codes [33] are a special class of rank-metric codes. A linear Gabidulin code $\mathcal{G}(n,k)$ over $\mathbb{F}_{q^s}$ of length $n \leq s$ and dimension k is defined by evaluating degree-restricted linearized polynomials:

$$\begin{aligned}\mathcal{G}(n,k) = \{&(f(\alpha_0), f(\alpha_1), \dots, f(\alpha_{n-1})) : \\ &f(z) = f_0 z^{q^0} + f_1 z^{q^1} + \cdots + f_{k-1} z^{q^{k-1}}\},\end{aligned}$$

where $\alpha_0, \alpha_1, \dots, \alpha_{n-1} \in \mathbb{F}_{q^s}^n$ have to be linearly independent over $\mathbb{F}_q$. Gabidulin codes are MRD codes, *i.e.*, $d = n-k+1$, cf. [33], [34].

C. System Model

Assume a network with multiple servers, where the stored data $X \in \mathbb{F}_{q^s}^{m \times k}$ is formed by m files $X^1, \dots, X^m \in \mathbb{F}_q^{s \times k}$, where each file X^i is subdivided into β stripes, where each stripe is assumed to be on a separate row, such that

$$X = \Phi^{-1} \begin{pmatrix} X^1 \\ X^2 \\ \vdots \\ X^m \end{pmatrix} \in \mathbb{F}_{q^s}^{m\beta \times k},$$

where Φ^{-1} denotes the one-to-one mapping from $s \times k$ matrices in $\mathbb{F}_q$ to vectors in $\mathbb{F}_{q^s}^k$ (*i.e.*, the inverse mapping of Φ). Each row of the data $X \in \mathbb{F}_{q^s}^{m\beta \times k}$, *i.e.*, each stripe, is encoded with a $\mathcal{G}(n,k)$ code over $\mathbb{F}_{q^s}$ such that we obtain $m\beta$ Gabidulin codewords, forming the rows of a matrix $Y \in \mathbb{F}_{q^s}^{m\beta \times n}$.

Assume that the data is stored on l servers where n is divisible by l. We divide the matrix Y into l blocks of $\rho \triangleq n/l$ columns and store each such block on a separate server. By $Y_j \in \mathbb{F}_{q^s}^{m\beta \times \rho}$, we denote the j^{th} block of Y, stored on server j. The servers are assumed to be honest-but-curious, and at most t of them may collude, *i.e.*, share the queries that they receive with each other. Fig. 1 illustrates the system model.

Note that according to the setup of this problem, each server is storing ρ columns of Y, which means the privacy constraint should take into account that each server acts as ρ colluding servers. For simplicity, we view the system as a system of n sub-servers with up to ρt collusions. Technically, the collusion in this setting is not full ρt collusion as not any ρt sub-servers can collude. The problem of constructing PIR schemes with certain collusion patterns is considered in [35].

Throughout this paper, we will use superscripts to denote the file index, and subscripts to denote the server index.

D. *PIR Scheme and PIR Rate*

We assume that the data is stored on l servers as described earlier. PIR allows the user to retrieve a file $X^f \in \mathbb{F}_q^{s \times k}$, $f \in \{1, \dots, m\}$, from the database without revealing the identity f of the file to any of the servers. A *linear PIR scheme* is a scheme over $\mathbb{F}_q$, consisting of the following stages:

- Query stage: When the user wants to retrieve a file X^f, she sends a query matrix $Q^f \in \mathbb{F}_q^{n \times m\beta}$ to the servers. We divide the query matrix into l blocks of ρ rows each, where block j is denoted by $Q_j^f \in \mathbb{F}_q^{\rho \times m\beta}$, and is sent to server j.
- Response stage: Upon receiving the query, server j responds to the user by projecting its stored data onto the query matrix. The response matrix, R_j^f, is the lifted diagonal matrix consisting of the diagonal elements of the product of the matrices Q_j^f and Y_j, defined formally in section III.

We next define the PIR rate which is the ratio of the number of downloaded symbols from the network if no privacy is required and the total number of downloaded symbols with privacy assumption, *i.e.*, the number of downloaded information

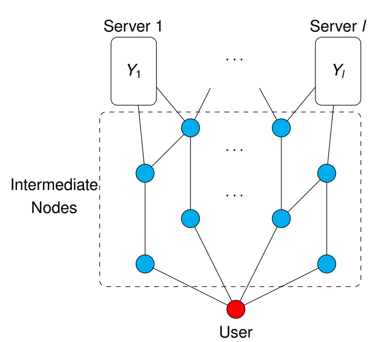

Fig. 1: Network Model: The user is connected to the servers via intermediate nodes. When data is sent from the user to the servers, or from server to user, the packets go through a random network using the intermediate nodes. The intermediate nodes compute random linear combinations of the received packets and send them forward. The intermediate network nodes are considered to be non-colluding and to simply relay data. We also assume the typical RLNC model in which all packets of one "shot" arrive at the same time, and therefore no memory or data handling is needed.

symbols and the total number of the downloaded symbols. The number of information symbols is the size of the file, *i.e.*, sk, and we assume the user downloads δ symbols from each server.

Definition 1 (PIR rate)**.** *The* PIR rate R_{PIR} *is*

$$R_{\mathsf{PIR}} = \frac{sk}{l\delta}.$$

E. The Star Product

The star product scheme for constructing a PIR scheme on a storage system that is encoded with a Generalized Reed–Solomon (GRS) code was first introduced in [13]. In that work, the star product between two codewords is the element-by-element product between the symbols of two codewords. The reason for choosing the star product as such is that the star product between two GRS codes results in another GRS code. In this work, however, we define the star product differently, as the storage system is encoded using a Gabidulin code, namely, such that the star product between two Gabidulin codes is also a Gabidulin code.

Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^s}$ be linearly independent over $\mathbb{F}_q$ which implies $n \leq s$. Let the storage code $\mathcal{C}$ be a $\mathcal{G}(n,k)$ Gabidulin code, *i.e.*:

$$\begin{aligned}\mathcal{C} = \{&(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})) : \\ &f(z) = f_0 z^{q^0} + f_1 z^{q^1} + \cdots + f_{k-1} z^{q^{k-1}}\}.\end{aligned}$$

Now, let the query code $\mathcal{D}$ be a $\mathcal{G}(n,t)$ Gabidulin code, where

$$\begin{aligned}\mathcal{D} = \{&(g(\alpha_0), g(\alpha_1), \ldots, g(\alpha_{n-1})) : \\ &g(z) = g_0 z^{q^0} + g_1 z^{q^1} + \cdots + g_{t-1} z^{q^{t-1}}\}.\end{aligned}$$

The *star product* of $\mathcal{C}$ and $\mathcal{D}$ is then defined by:

$$\begin{aligned}\mathcal{C} \star \mathcal{D} = \{&(h(\alpha_0), h(\alpha_1), \ldots, h(\alpha_{n-1})) : \\ &h(z) = f(g(z)), \\ &f(z) = f_0 z^{q^0} + f_1 z^{q^1} + \cdots + f_{k-1} z^{q^{k-1}}, \\ &g(z) = g_0 z^{q^0} + g_1 z^{q^1} + \cdots + g_{t-1} z^{q^{t-1}}\},\end{aligned}$$

which is a $\mathcal{G}(n, k+t-1)$ Gabidulin code.

Example 1. *Consider a storage system over* $\mathbb{F}_{2^5}$ *with primitive element* α *and primitive polynomial* $z^5 + z^2 + 1$*. Let* $\mathcal{C}$ *be a* $\mathcal{G}(5,3)$ *Gabidulin code defined by:*

$$\begin{aligned}\mathcal{C} = \{&(f(1), f(\alpha), f(\alpha^2), f(\alpha^3), f(\alpha^4)) : \\ &f(z) = f_0 z + f_1 z^2 + f_2 z^4\},\end{aligned}$$

$$G_{C\star D} = \begin{array}{c} z^1 \\ z^2 \\ z^4 \\ z^8 \end{array} \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ 1 & \alpha^2 & \alpha^4 & \alpha^3+\alpha & \alpha^3+\alpha^2+1 \\ 1 & \alpha^4 & \alpha^3+\alpha^2+1 & \alpha^3+\alpha^2+\alpha & \alpha^4+\alpha^3+\alpha+1 \\ 1 & \alpha^3+\alpha^2+1 & \alpha^4+\alpha^3+\alpha+1 & \alpha^4+\alpha^3+\alpha^2+\alpha & \alpha \end{pmatrix}.$$

Fig. 2: Generator matrix of the code $\mathcal{C}\star\mathcal{D}$ in Example 1.

and $\mathcal{D}$ be a $\mathcal{G}(5,2)$ Gabidulin code defined by:

$$\mathcal{D} = \{(g(1), g(\alpha), g(\alpha^2), g(\alpha^3), g(\alpha^4)) :$$
$$g(z) = g_0 z + g_1 z^2\}.$$

Now the code $\mathcal{C}\star\mathcal{D}$ is a $\mathcal{G}(5,4)$ Gabidulin code defined by:

$$\begin{aligned} \mathcal{C}\star\mathcal{D} = \big\{&(h(1), h(\alpha), h(\alpha^2), h(\alpha^3), h(\alpha^4)) : \\ &h(z) = f(g(z)) \\ &\quad = f_0(g_0)z + (f_0(g_1) + f_1(g_0))z^2 + \\ &\quad (f_1(g_1) + f_2(g_0))z^4 + f_2(g_1)z^8\big\}. \end{aligned}$$

A generator matrix of the code $\mathcal{C}\star\mathcal{D}$ is given in Figure 2. Clearly, this definition of the star product works for any linear rank-metric code.

F. Random Linear Network Coding

In this paper, data is transmitted over a random linear network, *i.e.*, each node computes linear combinations of the received packets and forwards the linear combination, see [26] for more details. Additionally, erroneous packets can be inserted on any edge of the network. The random network channel is therefore modeled by

$$R = AQ + N_e,$$

where the rows of $Q \in \mathbb{F}_{q^s}^{n\times m\beta}$ denote the packets that should be transmitted, $A \in \mathbb{F}_{q_N}^{n\times n}$ denotes the channel matrix, $N_e \in \mathbb{F}_{q_N}^{n\times m\beta}$ the overall error matrix and the rows of $R \in \mathbb{F}_{q^s}^{n\times 1}$ denote the packets at the receiver side. We assume, for simplicity, that the field size, q_N, of the entries of the network matrices is equal to q^s. This assumption is easy to generalize as long as the characteristics match, *i.e.*, $\mathrm{char}(\mathbb{F}_{q_N}) = \mathrm{char}(\mathbb{F}_{q^s})$.

Remark 1. *Throughout this paper, the results will not depend on the topology of the network. This is in line with the assumption of random linear network coding, see* e.g. *the seminal papers [24], [26]. Thus, our results hold for any network and the network structure does not have to be known, neither to the user nor to the servers. Even more, the structure is allowed to change during different uses of the network.*

III. PIR over an Error-Free Network

Query: In the following, we assume that s divides m. We assume the user wants to retrieve the file X^f, $f \in \{1, \ldots, m\}$, while hiding the identity f from the storage system as defined in Section II-C. For this purpose, the files will be subdivided into $\beta = n - k - \rho t + 1$ stripes, and a file will be retrieved in k rounds of queries. For query i, the user will generate $t\rho$ random vectors of length $m\beta$, denoted by $u_1^{(i)}, \ldots, u_{t\rho}^{(i)} \in \mathbb{F}_q^{m\beta}$, such that

$$U^{(i)} = \begin{pmatrix} u_1^{(i)} \\ \vdots \\ u_{t\rho}^{(i)} \end{pmatrix}^\top, \tag{1}$$

where $U^{(i)} \in \mathbb{F}_q^{m\beta \times t\rho}$. These vectors are then mapped onto $t\rho$ vectors of size $\mu = m\beta/s$ in $\mathbb{F}_{q^s}$. The vectors are chosen uniformly at random so that the queries received by any $t\rho$ servers reveal no information about the index of the requested file.

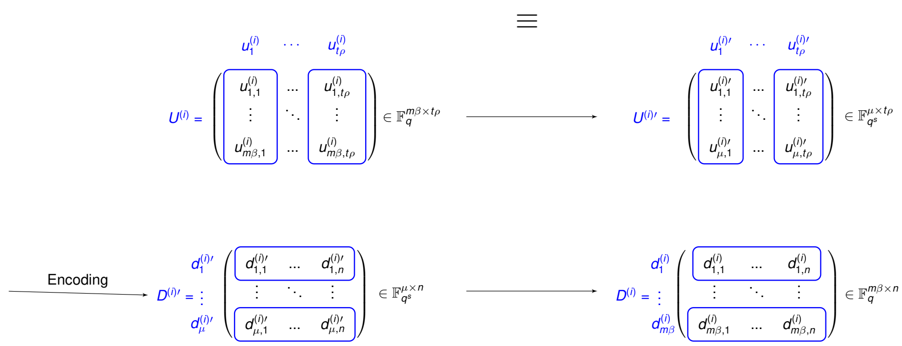

Fig. 3: Encoding random vectors into the query. $t\rho$ random vectors of length $m\beta$ are generated, then mapped onto vectors of size $m\beta/s$. Those vectors are then encoded into codewords of a $\mathcal{G}(n, t\rho)$ Gabidulin code. In turn, those codewords are then mapped onto $m\beta$ codewords to form matrix $D^{(i)}$.

The user then generates the matrix $D^{(i)\prime} \in \mathbb{F}_{q^s}^{\mu \times n}$ by encoding the μ rows of $U^{(i)\prime}$ to μ codewords of the $\mathcal{G}(n, t\rho)$ Gabidulin code $\mathcal{D}$. Those codewords are then mapped onto $m\beta$ codewords of $\mathcal{D}$ to form the matrix $D^{(i)} \in \mathbb{F}_q^{m\beta \times n}$, as shown in Fig. 3.

Afterwards, the user forms the deterministic matrix $E^{f,(i)} \in \mathbb{F}_q^{n \times m\beta}$ (see (2)), which is used to retrieve the file X^f. In one query round, the user can retrieve β stripes of the requested file, as discussed in [30]. Therefore, the matrix $E^{f,(i)}$ is chosen such that this matrix adds a 1 to the randomly generated matrix $D^{(i)}$ in β positions of the required file X^f. Let e^f be the matrix of size $\beta \times m\beta$, such that

$$e^f = \begin{pmatrix} \mathbf{0}_{\beta(f-1)} & I_{\beta \times \beta} & \mathbf{0}_{\beta(m-f)} \end{pmatrix}.$$

W.l.o.g., the user can choose the matrix $E^{f,(i)}$ such that

$$E^{f,(i)} = \begin{pmatrix} \mathbf{0}_{i-1 \times m\beta} \\ e^f \\ \mathbf{0}_{n-\beta-i+1 \times m\beta} \end{pmatrix}, \tag{2}$$

and hides it using the random matrix $D^{(i)}$ by adding

$$D_Q^{f,(i)} = D^{(i)\top} + E^{f,(i)} \in \mathbb{F}_q^{n \times m\beta}, \tag{3}$$

where $D^{(i)\top}$ is the transpose of matrix $D^{(i)}$. The user then divides $D_Q^{f,(i)}$ into l blocks, where each block, $D_{Q_j}^{f,(i)}, j = 1, \ldots, l$, consists of $\rho = n/l$ rows of $D_Q^{f,(i)}$. Afterwards, she lifts every matrix, $D_{Q_j}^{f,(i)}$, (*i.e.*, appends an identity matrix) to obtain the query matrices

$$Q_j^{f,(i)} = \left(\begin{array}{c:c} I_{\rho\times\rho} & D_{Q_j}^{f,(i)} \end{array} \right) \in \mathbb{F}_q^{\rho\times(\rho+m\beta)} \tag{4}$$

for $j = 1, \ldots, l$.

Query transmission: The user sends $Q_j^{f,(i)}$ to server j, $j = 1, \ldots, l$. The network is random (cf. Section II-F) and we assume that server j receives

$$Q_{j,rec}^{f,(i)} = \left(\begin{array}{c:c} A_j^{(i)} & A_j^{(i)} D_{Q_j}^{f,(i)} \end{array} \right) \in \mathbb{F}_{q^s}^{\rho\times(\rho+m\beta)},$$

where $A_j^{(i)} \in \mathbb{F}_{q^s}^{\rho\times\rho}$ is the random channel transfer matrix in the network from the user to server j in round i, and $D_{Q_j}^{f,(i)}$ is block j of the matrix $D_Q^{f,(i)}$, *i.e.*, the ρ rows of $D_Q^{f,(i)}$ sent to server j.

Server response: The server projects its data on the query matrix received, and returns to the user the following matrix

$$R_j^{f,(i)} = \left(\begin{array}{c:c:c} I_{\rho\times\rho} & A_j^{(i)} & Y_j \star (A_j^{(i)} D_{Q_j}^{f,(i)}) \end{array} \right) \in \mathbb{F}_{q^s}^{\rho\times(2\rho+m\beta)},$$

where Y_j is block j of Y, as defined in Section II-C.

The user then receives the response

$$R_{j,rec}^{f,(i)} = \left(\begin{array}{c:c:c} A_j^{(i)\prime} & A_j^{(i)\prime} A_j^{(i)} & A_j^{(i)\prime} (Y_j \star (A_j^{(i)} D_{Q_j}^{f,(i)}) \end{array} \right) \\ \in \mathbb{F}_{q^s}^{\rho\times(2\rho+m\beta)}$$

from server j, after being transmitted through the network where $A_j^{(i)\prime} \in \mathbb{F}_{q^s}^{\rho\times\rho}$ denotes the $\rho\times\rho$ random channel transfer matrix from server j to the user in round i.

Information retrieval: From the responses of the different servers, the user can then form the matrix $R_{rec}^{f,(i)} \in \mathbb{F}_{q^s}^{n\times(2n+m\beta)}$.

$$R_{rec}^{f,(i)} = \left(\begin{array}{c:c:c} A^{(i)\prime} & A^{(i)\prime} A^{(i)} & A^{(i)\prime} (Y \star (A^{(i)} D_Q^{f,(i)})) \end{array} \right) \\ \in \mathbb{F}_{q^s}^{n\times(2n+m\beta)}$$

where

$$A^{(i)} := \begin{pmatrix} A_1^{(i)} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & A_{l-1}^{(i)} & 0 \\ 0 & \cdots & 0 & A_l^{(i)} \end{pmatrix} \in \mathbb{F}_{q^s}^{n\times n}$$

and

$$A^{(i)\prime} := \begin{pmatrix} A_1^{(i)\prime} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & A_{l-1}^{(i)\prime} & 0 \\ 0 & \cdots & 0 & A_l^{(i)\prime} \end{pmatrix} \in \mathbb{F}_{q^s}^{n\times n}$$

Remark 2. *Note that [26] deals with single sources only. For multiple sources, in the standard network coding problem, it is not easy to design subspace codes that can be "distributed" to the different sources and combined to one codeword at the receiver side. For example in [36], the problem was solved only for the special case of three sources. In our scenario, however, the servers that respond to the queries do not encode an* arbitrary *matrix. They transmit the matrix $R_j^{f,(i)}$. Once the user obtains all matrices $R_{j,rec}^{f,(i)}$ these matrices can be combined into the matrix $R_{rec}^{f,(i)}$.*

Remark 3. *If the overall channel transfer matrices $A^{(i)}$ and $A^{(i)\prime}$ have full rank, then, the user receives $A^{(i)\prime}(Y \star D_Q^{f,(i)}) = A^{(i)\prime}Y \star A^{(i)}D^{(i)} + A^{(i)\prime}Y \star A^{(i)}E^{f,(i)}$, which is a codeword from a $\mathcal{G}(n, k + t\rho - 1)$ code with β errors in known locations, which can be treated as erasures. Since the $\mathcal{G}(n, k + t\rho - 1)$ code has minimum rank distance $n - k - t\rho + 2$, it can correct any $\beta = n - k - t\rho + 1$ (rank) erasures and the reconstruction of $Y \star E^{f,(i)}$,* i.e., *β stripes of file X^f, can be done exactly as in a usual Gabidulin decoding problem.*

Example 2. *Let m denote the number of files. The data, $X \in \mathbb{F}_{q^s}^{m\times 2}$, is encoded using a $\mathcal{G}(3, 2)$ Gabidulin code $\mathcal{C}$ over $\mathbb{F}_{2^3}$, where*

$$\mathcal{C} = \{(f(1), f(\alpha), f(\alpha^2)) : f(z) = f_0 z + f_1 z^2\}.$$

Hence, its corresponding generator matrix is

$$G_C = \begin{pmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha^2 + \alpha \end{pmatrix} \in \mathbb{F}_{q^s}^{2\times 3}.$$

We consider a network with $l = 3$ servers, where every column Y_j of

$$Y = XG_C \in \mathbb{F}_{q^s}^{m\times 3}$$

is stored on a server.

The goal is to construct a PIR scheme where the user wants file X^f from X, while keeping the identity f of the file hidden from the servers, and the servers do not collude, i.e., *$t = 1$. The number of subdivisions in this case is $\beta = n-k-t\rho+1 = 1$. Assume the user wants file X^1. To this end, we use a $\mathcal{G}$(3,1) code as the query code,*

$$\mathcal{D} = \{(f(1), f(\alpha), f(\alpha^2)) : f(z) = f_0 z\}$$

whose generator matrix is

$$G_D = \begin{pmatrix} 1 & \alpha & \alpha^2 \end{pmatrix} \in \mathbb{F}_{q^s}^{1\times 3}.$$

Two rounds are needed to retrieve the full file X^1, hence, let matrix $e^1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}$. Then, $E^{1,(1)}$ is the $3 \times m$ matrix $E^{1,(1)} = \begin{pmatrix} e^1 \\ \mathbf{0}_{n-1\times m} \end{pmatrix}$, and $E^{1,(2)} = \begin{pmatrix} \mathbf{0}_{1\times m} \\ e^1 \\ \mathbf{0}_{n-2\times m} \end{pmatrix}$ which the user uses to decode the file X^1, as defined in (2)*. The matrix $E^{1,(i)}$ is added to the random matrix $D^{(i)}$, which is chosen uniformly at random from the code $\mathcal{D}$, as shown in* (3)*.*

The query matrix is then lifted,

$$Q_j^{1,(i)} = \left(\begin{array}{c:c} 1 & D_{Q_j}^{1,(i)} \end{array} \right),$$

where $D_{Q_j}^{1,(i)}$ *is the* j^{th} *row of* $D_Q^{1,(i)}$*, generated in round* i*, for* $i = 1, 2$*.*

In round i*, the user sends row* j *of matrix* $Q^{f,(i)}$ *to server* j*. Assume that server* j *receives*

$$Q_{j,rec}^{f,(i)} = \left(\begin{array}{c:c} a_j^{(i)} & a_j^{(i)} D_{Q_j}^{1,(i)} \end{array} \right),$$

where $a_j^{(i)} \in \mathbb{F}_{2^3}$ *is a scalar introduced by the network in round* i*.*

There are two cases a server can encounter:

- $a_j^{(i)} = 0$*: In this case, server* j *receives the all-zero vector and sends nothing back to the user.*
- $a_j^{(i)} \neq 0$*: In this case, server* j *will calculate the star product of its contents* Y_j *with the received query matrix,* $Q_{j,rec}^{f,(i)}$*, and lifts the obtained matrix to form its response*

$$R_j^{1,(i)} = \left(\begin{array}{c:c:c} 1 & a_j^{(i)} & a_j^{(i)} D_{Q_j}^{1,(i)} \end{array} \right),$$

which is sent to the user. The user receives

$$R_{j,rec}^{1,(i)} = \left(\begin{array}{c:c:c} {a_j}^{(i)\prime} & {a_j}^{(i)\prime} a_j^{(i)} & {a_j}^{(i)\prime} a_j^{(i)} D_{Q_j}^{1,(i)} \end{array} \right)$$

from server j*,* $j = 1, 2, 3$*, and forms the overall matrix*

$$R^{1,(i)} = \left(\begin{array}{c:c:c} A^{(i)\prime} & A^{(i)\prime} A^{(i)} & A^{(i)\prime} (Y \star A^{(i)} D_{Q_1}^{1,(i)}) \end{array} \right)$$

where $A^{(i)}$ *and* $A^{(i)\prime}$ *are the* 3×3 *diagonal matrices with diagonal elements* $(a_1^{(i)}, a_2^{(i)}, a_3^{(i)})$ *and* $(a_1^{(i)\prime}, a_2^{(i)\prime}, a_3^{(i)\prime})$*, respectively, as defined in Section II-D.*

The user may now face two different scenarios:

  - *All coefficients* ${a_j}^{(i)\prime} \neq 0$ *for all* $j \in \{1, 2, 3\}$*. In this case, the user can decode the matrix*

$$\left(Y \star D_{Q_1}^{1,(i)} \right) = \left(\begin{array}{c} Y \star D^{(i)\top} + Y \star E^{1,(i)} \end{array} \right).$$

  The matrix $Y \star D^{(i)\top}$ *is a codeword from a* $\mathcal{G}(3, 2)$ *code* $\mathcal{C} \star \mathcal{D}$*, since* Y *is a codeword from a* $\mathcal{G}(3, 2)$ *code and* $D^{(i)}$ *is from a* $\mathcal{G}(3, 1)$ *code. Therefore, the user can decode one rank erasure in* $\mathcal{C} \star \mathcal{D}$*. In our setting, the matrix* $E_j^{1,(i)}$ *is the deterministic matrix introduced by the user in order to decode the desired file. The positions of the errors are known, so they can be viewed as erasures, thus allowing the user to decode* $Y \star E^{1,(i)}$*.*
  - *On the other hand, if at least one of the coefficients* ${a_j}^{(i)\prime} = 0$*, then* $A^{(i)\prime}(Y \star D^{(i)\top})$ *becomes a codeword in a* $\mathcal{G}(2, 2)$ *Gabidulin code that is not able to correct any errors or erasures. Thus, the user will not be able to decode the desired file. This case is equivalent to the case where* ${a_j}^{(i)} = 0$*. For instance, assuming* ${a'_2}^{(i)} = 0$*, the user*

obtains $R^{1,(i)} = \begin{pmatrix} A^{(i)\prime} & A^{(i)\prime}A^{(i)} & A^{(i)\prime}(Y \star A^{(i)}D_{Q_1}^{1,(i)}) \end{pmatrix}$ *such that the second row is the all-zero vector, and one can observe that the user cannot decode the file from this response matrix.*

In this example, the PIR rate is $\frac{1}{3}$ *with error probability* $1-(1-\frac{1}{8})^{12} = 1-\left(\frac{7}{8}\right)^{12} \approx 0.798582762$*. Nevertheless, the error probability approaches* 0 *when* q *is sufficiently large (cf. Prop. 1).*

In the following example, we assume that the servers store 2 columns of the data Y, and allow 2 servers to collude. Moreover, it is shown that if erasures occur in the network, partial recovery of the user's requested file is still possible.

Example 3. *Assume that the data* $X \in \mathbb{F}_{q^s}^{m\times 3}$ *is encoded using a* $\mathcal{G}(8,3)$ *Gabidulin code* $\mathcal{C}$*, with generator matrix* G_C *over* $\mathbb{F}_{2^8}$*. The data is then stored on* $l=4$ *servers, such that each server stores* $\rho = 2$ *columns of* $Y = XG_C$*. Let* Y_j *be the block of* Y *stored on server* j*.*

Assume also that any $t=2$ *servers can collude. Since every server is storing two columns of the data, as discussed in II-C, this can be seen as a case of* 4*-collusion. The goal is to construct a PIR scheme with* 4*-collusion that allows a user to retrieve a file* X^f *from the servers, without revealing the file identity to any of the servers. The files are subdivided into* $\beta = n-k-t\rho+1 = 2$ *stripes, and* $k=3$ *rounds of queries are sent. In round* i*, the query code*

$$\mathcal{D} = \{d = (g(\alpha_0), g(\alpha_1)) : g(z) = g_0 z + g_1 z^2\} \in \mathbb{F}_{2^8}^{\mu\times 8}$$

is used to encode $t\rho = 4$ *random vectors* $u_1^{(i)\prime}, u_2^{(i)\prime}, u_3^{(i)\prime}, u_4^{(i)\prime} \in \mathbb{F}_2^{2m}$ *and form the random matrix* $D^{(i)} \in \mathbb{F}_2^{m\beta\times n}$*.*

Afterwards, the $3\times 2m$ *deterministic matrix* $E^{f,(i)}$*, chosen as explained in Section II-D, is added to the random matrix* $D^{(i)}$ *to form* $D_Q^{f,(i)} = D^{(i)\top} + E^{f,(i)}$*.*

The query matrix to server j *is then*

$$Q_j^f = \begin{pmatrix} I_{2\times 2} & D_{Q_j}^{f,(i)} \end{pmatrix},$$

where $D_{Q_j}^{f,(i)}$ *is the* j^{th} *block, consisting of* 2 *rows of* $D_Q^{f,(i)}$*.*

For instance, assume that the user wants file X^1*. Then in the first round,*

$$E^{1,(1)} = \begin{pmatrix} e^1 \\ \mathbf{0}_{6\times 2m} \end{pmatrix}.$$

The user sends $Q_j^{1,(i)}$, i.e., *block* j *of the query matrix* $Q^{1,(i)}$*, consisting of* j *rows, to server* j*. Server* j *receives the matrix*

$$Q_{j,rec}^{1,(i)} = \begin{pmatrix} A_j^{(i)} & A_j^{(i)} D_{Q_j}^{1,(i)} \end{pmatrix},$$

where $A_j^{(i)} \in \mathbb{F}_{q^s}^{2\times 2}$ *is the* 2×2 *matrix introduced by the network in round* i*. Then, the server sends back the matrix*

$$R_j^{1,(i)} = \begin{pmatrix} I_{2\times 2} & A_j^{(i)} & Y_i \star (A_j^{(i)} D_{Q_j}^{1,(i)}) \end{pmatrix},$$

and the user receives the matrix

$$R_{j,rec}^{1,(i)} = \begin{pmatrix} A_j^{(i)\prime} & A_j^{(i)\prime}A_j^{(i)} & A_j^{(i)\prime}(Y_j \star (A_j^{(i)} D_{Q_j}^{1,(i)}) \end{pmatrix}$$

where $A_j^{(i)\prime}$ is the 2×2 matrix introduced by the network from server j to the user. Then the user can construct the matrix

$$R_{rec} = \left(\begin{array}{c:c:c} A^{(i)\prime} & A^{(i)\prime}A^{(i)} & A^{(i)\prime}(Y \star (A^{(i)}D_Q^{1,(i)})) \end{array} \right),$$

where $A^{(i)\prime} = \begin{pmatrix} A_1^{(i)\prime} & 0 & 0 & 0 \\ 0 & A_2^{(i)\prime} & 0 & 0 \\ 0 & 0 & A_3^{(i)\prime} & 0 \\ 0 & 0 & 0 & A_4^{(i)\prime} \end{pmatrix}$ *and* $A^{(i)} = \begin{pmatrix} A_1^{(i)} & 0 & 0 & 0 \\ 0 & A_2^{(i)} & 0 & 0 \\ 0 & 0 & A_3^{(i)} & 0 \\ 0 & 0 & 0 & A_4^{(i)} \end{pmatrix}$*, as defined in Section II-D.*

We see that $\mathcal{C} \star \mathcal{D}$ is a $\mathcal{G}(8,6)$ Gabidulin code, which can correct up to 2 erasures. The matrix $E^{f,(i)}$ adds 2 errors with known locations to the code, i.e., *erasures. Thus, the user is able to decode the file X^f if both matrices $A_j^{(i)\prime}$ and $A_j^{(i)}$ have full rank in all rounds with a PIR rate $R_{\mathsf{PIR}} = \frac{\mathrm{rk}_{q^s}(A^{(i)\prime}Y \star A^{(i)}E^{f,(i)})}{n} = \frac{1}{4}$. Following Proposition 1, this happens with probability $\left((1-\frac{1}{q^s})^{16}\right)^3 = (1-\frac{1}{q^s})^{19}$.*

On the other hand, assume that the matrices $A^{(i)}$ and $A^{(i)\prime}$ have a rank deficiency, such that $\mathrm{rk}_{q^s}(A^{(i)\prime}(Y \star A^{(i)}D_Q^{f,(i)})) = 7$. The user can then still decode 1 stripe of the required file if the erasures from the network occur on the links where the errors from the matrix $E^{f,(i)}$ are added, i.e., *the rank deficiency occurs in the matrix $A^{(i)\prime}(Y \star A^{(i)}E^{f,(i)})$.*

Let P_1 denote the probability of this event.[1]

$$\begin{aligned} P_1 = \mathbb{P}\Big(&\mathrm{rk}_{q^s}\left(A^{(i)\prime}(Y \star A^{(i)}D_Q^{f,(i)})\right) = 7 \\ &\textit{and } \mathrm{rk}_{q^s}\left(A^{(i)\prime}(Y \star A^{(i)}E^{f,(i)})\right) = 1\Big) \\ = &\binom{2}{1}\left(\left(1-\frac{1}{q^s}\right)^7\left(\frac{1}{q^s}\right)\right)^2 \\ &+ 2\left[\binom{2}{1}\left(\left(1-\frac{1}{q^s}\right)^7\left(\frac{1}{q^s}\right)\right)\left(1-\frac{1}{q^s}\right)^8\right] \\ \approx &\ 0.015. \end{aligned}$$

Then, the user is able to decode 1 stripe, with PIR rate $R_{\mathsf{PIR}} = \frac{1}{8}$, with error probability $1-P_1$. Thus, the the average realized download rate is

$$P_2 \cdot \frac{2}{8} + P_1 \cdot \frac{1}{8} \approx 0.24$$

where P_2 is the probability that both $A^{(i)}$ and $A^{(i)\prime}$ have full rank in round i, i.e., $P_2 = \left(1-\frac{1}{q^s}\right)^{16} \approx 0.94$.

According to Prop. 1, the probability of matrices $A^{(i)}$ and $A^{(i)\prime}$ are simultaneously full rank in all query rounds $i = 1,\dots,k$ is $\left(1-\frac{1}{q^s}\right)^{2n+k}$. Together with the results of the previous examples, we can conclude the following theorems. Since the scheme from one round is the same for all other rounds, we will drop the superscript (i) in the following.

Theorem 1. *In the above setting, the user can decode the file X^f from the k responses from rounds $i = 1,\dots,k$, R_{rec}^f, with PIR rate*

$$R_{\mathsf{PIR}} = \left(1 - \frac{k+t\rho-1}{n}\right),$$

[1]This is the sum of two terms: The first is the probability that both $A^{(i)}$ and $A^{(i)\prime}$ have an erasure in the same place and in one of the two places where $E^{f,(i)}$ is nonzero, and the second is the probability that one of the matrices has an erasure in a position where $E^{f,(i)}$ is nonzero and the other is full rank.

with error probability $1-\left(1-\frac{1}{q^s}\right)^{2n+k}$.

Proof. Decodability. Assume that $\text{rk}_{q^s}(A)=\text{rk}_{q^s}(A')=n$, *i.e.*, A and A' are full rank, which happens in all k rounds with probability $\left(1-\frac{1}{q^s}\right)^{2n+k}$, as given in Prop. 1. The responses R^f_{rec} could then be reduced to

$$Y\star D+Y\star E^f,$$

where $Y\star D^f_Q$ is a codeword from $\mathcal{G}(n,k+t\rho-1)$ Gabidulin code, and $\text{rk}_{q^s}(Y\star E^f)=n-k-t\rho+1$. Therefore, the user can decode $Y\star E^f$, which gives one part from each stripe in each round (this follows from the way the matrix E^f is chosen). Moreover, the choice of E^f ensures that the user retrieves at most one part of any stripe from a single sub-server, *i.e.,* the user does not retrieve redundant information about a certain stripe. Hence, after k query rounds, the user will have k independent equations for each stripe, and therefore, can recover the full file X^f.

Privacy. Since the random matrix D is a $\mathcal{G}(n,t\rho)$ Gabidulin code, from which the codewords are chosen uniformly at random, the query received by any $t\rho$ servers is random, and the server cannot know where a 1 was added. □

Remark 4. *It is easy to see that the error probability in the above theorem approaches* 0 *when the field size grows. Moreover, the PIR rate matches the conjectured capacity [13].*

The above theorem is restricted to the case where no erasures occur in the network. However, as was shown in Example 3, the rank deficiency of $A'(Y\star AD^f_Q)$ might reduce the rank of $A'(Y\star AE^f)$ that is received by the user as well. In such cases, the user can still decode $\text{rk}_{q^s}(A'(Y\star AE^f))$ stripes from the requested file, X^f. For instance, as shown in Example 3, the user can still be able to decode one stripe of the requested file, despite the rank deficiency of the matrix $A'(Y\star AD^f_Q)$.

In the following, we consider the rate given from a single query round i. Since the probabilistic rate is the same for all query rounds, $i=1,\cdots,k$, we discuss what follows for a single round and drop the superscript (i). In the first stage of querying, if no erasures occur, then the user can still retrieve β parts of the requested file X^f. On the other hand, assume that $\text{rk}_{q^s}(A'(Y\star AD^f_Q))-\text{rk}_{q^s}(A'(Y\star AE^f))=k+t\rho-1$, and $\text{rk}_{q^s}(A'(Y\star AE^f))=\delta<\beta$, the user can then retrieve δ parts of the file. However, the user has to send a second stage of queries to retrieve $(\beta-\delta)$ new parts of the file X^f, which can substitute the erased parts. To do that, the user will subdivide the stripes again into β sub-stripes, then use the same scheme as in the first stage to retrieve a part of each sub-stripe. Here it is important to note that each sub-server should be asked at most once about any stripe. This means that, in the second stage of querying, the matrix E^f should be chosen such that, to retrieve stripe ℓ, a 1 is never added to the columns $\ell,\dots,\ell+k-1$ of D, since these will give information already acquired about stripe ℓ in other rounds, *i.e.,* they give redundant information. Assume that no erasures occur in the second stage, then, along with the responses from the first stage, the user now has k independent equations about all sub-stripes of the file X^f. On the other hand, if erasures occur in stage 2, this process will be followed another time. This will be done recursively until all the file X^f is recovered. We state this more formally in Theorem 2 below.

Denote

$$\begin{aligned}&\mathbb{P}(\mathrm{rk}_{q^s}(A'(Y\star AD_Q^f))-\mathrm{rk}_{q^s}\left(A'(Y\star AE^f)\right)=k+t\rho-1,\\&\quad\text{and }\ \mathrm{rk}_{q^s}\left(A'(Y\star AE^f)\right)=\delta)=:P_\delta.\end{aligned}$$

Theorem 2. *The user can decode δ stripes of the file X^f from the responses R^f_{rec} with a PIR rate*

$$R_{\mathsf{PIR}}=\frac{\delta}{n}\,, \tag{5}$$

with error probability $1-P_\delta$.

Proof. Decodability. Since $Y\star D^f$ is a codeword in a $\mathcal{G}(n,k+t\rho+1)$ Gabidulin code, it is able to tolerate $n-(k+t\rho-1)$ erasures. Therefore, the user can decode the codeword $Y\star D^f$, whenever $\mathrm{rk}_{q^s}(A'(Y\star AD_Q^f))-\mathrm{rk}_{q^s}(A'(Y\star AE^f))=k+t\rho-1$. Additionally, assume that $\mathrm{rk}_{q^s}(A'(Y\star AE^f))=\delta$. Then the user is able to retrieve δ stripes of the requested file X^f. As the probability of the joint event that $\mathrm{rk}_{q^s}(A'(Y\star AD_Q^f))-\mathrm{rk}_{q^s}(A'(Y\star AE^f))=k+t\rho-1$ and $\mathrm{rk}_{q^s}(A'(Y\star AE^f))=\delta$ is P_δ, the PIR rate is

$$R_{\mathsf{PIR}}=\frac{\delta}{n}\,, \tag{6}$$

with error probability $1-P_\delta$. The average realized download rate is

$$\sum_{\delta=1}^{\beta}\frac{P_\delta\cdot\delta}{n}.$$

Privacy. Privacy is achieved for the same reason as in Theorem 1.

□

IV. PIR ON A NETWORK WITH ERRORS

Errors in a random network happen, *e.g.,* due to malicious nodes injecting erroneous packets or due to congestion. In this section, we will consider such an erroneous network. The network is assumed to introduce up to ϵ errors and up to τ erasures to the sent packets.

From [24], it is known that errors of rank ϵ can be corrected if $2\epsilon\le\tau+d-1$, where d is the minimum rank distance of the code. For this purpose, each file is subdivided into $\beta=n-k-\rho t-2\epsilon-\tau+1$ stripes. The matrix $D^{(i)}$ is constructed in the same manner as in section II-D. As for the matrix $E^{f,(i)}$, it is chosen as a codeword of a rank metric code $\mathcal{E}^f$ such that the response $\mathcal{C}\star(\mathcal{D}+\mathcal{E}^f)$ is again a Gabidulin code, which follows from the techniques used in [30]. Specifically, we start with E^f being the $n\times m\beta$ all-zero matrix. In round i, for all $\delta\le\lceil\frac{i\beta}{k}\rceil$, the vector $e_\delta^{f,(i)}$ is taken to be the vector with all zeros and a single 1 at position $\beta(f-1)+\delta$, which is the vector requesting the stripe δ of the file X^f. The vector $e_\delta^{f,(i)}$ is encoded using a $\mathcal{G}(n,1)$ code $\mathcal{E}^f=\{g(\alpha_0),\cdots,g(\alpha_{n-1}):g(z)=g_0z^{q^{i\beta-\delta k+k+\rho t-1}}\}$ and added to row $(f-1)m\beta+\delta$ of $E^{f,(i)}$.[2]

[2]For more specific details on how the scheme is constructed, see [30].

Remark 5. *In the above scheme, the storage code $\mathcal{C}$ is a Gabidulin code of dimension k. Hence, we pick $\mathcal{D}$ to be a Gabidulin code of dimension $t\rho$, so that $\mathcal{C} \star \mathcal{D}$ is a Gabidulin code of dimension $t\rho + k - 1$ (see Section II-E). Thus, in round i, $\mathcal{C} \star (\mathcal{D} + \mathcal{E}^f)$ is a $\mathcal{G}(n, k + t\rho + \beta - 1)$ Gabidulin code, padded with parts retrieved from previous rounds $1, \ldots, i-1$, with degrees $> n-1$, where degrees $q^{t\rho+k}$ and higher of the evaluation polynomial consist of parts of the file requested by the user. From $\mathcal{C} \star (\mathcal{D} + \mathcal{E}^f)$, the user can interpolate to retrieve the higher powers, thus retrieving β parts of the file she requires.*

We now assume that $2\epsilon + \tau \leq d - 1$, where d is the minimum distance of the code $\mathcal{C} \star (\mathcal{D} + \mathcal{E}^f)$. Then $\mathrm{rk}_{q^s}(N_e) = \epsilon$ errors and τ erasures can be corrected. In this case, the underlying PIR scheme would be similar to the scheme in [30], with the ability to correct ϵ errors and τ erasures. For the rest of this section, we assume that $2\epsilon + \tau \leq d - 1$. In the current setting, the query code is constructed assuming a certain number of errors and erasures introduced by the network, thus, in this section, the probability of error is not taken into account.

We discuss the results for a single round i, and drop the superscript (i). In the following, we distinguish two cases, namely the case where no errors occur on the uplink, and some errors occur on the downlink, and the case where some errors occur on the uplink as well. As opposed to the work done in [21]–[23], we consider PIR over a random linear network, where the messages are coded over the servers, not replicated. Moreover, as opposed to all previous works on PIR, we consider uplink errors in this work.

A. Downlink errors

Let us first assume that no errors are introduced on the uplink, but errors may be introduced on the downlink. In this case, it follows from [24] that the code can correct ϵ errors and τ erasures as long as $2\epsilon + \tau \leq d - 1$. Since $\mathcal{D}$ and $\mathcal{E}^f$ are chosen such that $\mathcal{C} \star (\mathcal{D} + \mathcal{E}^f)$ is a $\mathcal{G}(n, k + t\rho + \beta - 1)$ Gabidulin code, the user can decode and retrieve the requested file when $2\epsilon + \tau \leq d - 1 = n - k - t\rho - \beta$, where d is defined as the minimum distance of the code $\mathcal{C} \star (\mathcal{D} + \mathcal{E}^f)$. This can be summed up in the following theorem.

Theorem 3. *Assume that the queries are sent through a network where no errors occur, and the responses are sent through a network with errors. Moreover, assume that*

$$2\epsilon + \tau \leq d - 1.$$

Then, the user can decode the errors and erasures introduced by the network, and consequently, can decode the requested file with rate $R_{\mathsf{PIR}} = \frac{\beta}{n}$.

Proof. This follows directly from [24], where it was shown that if $\mathcal{C}$ is a Gabidulin code with minimum distance d, then the errors introduced to the system will still allow for correct decodability when

$$2\epsilon + \tau \leq d - 1.$$

Since $\mathcal{D}$ and $\mathcal{E}^f$ were chosen such that $(Y \star (D^\top + E^f))$ is a codeword in a Gabidulin code with minimum distance $d \geq 2\epsilon + \tau + 1$, thus the responses received by the user form a Gabidulin code that can correct up to $(d-1)/2$ errors. $\square$

B. Uplink errors

Assume that the queries are now sent through a network that introduces errors. In this case, the errors will be received by the servers. Since the dimension of the matrix sent to the servers is small enough to protect the user's privacy, the servers will have no capability to correct any errors. Hence, the servers will project the received matrix, including the errors, on their stored data and send it back to the user. We can see that an erasure that happens on the uplink would translate to an erasure on the downlink, so we can account for the total number of erasures as τ.

Consider an error Z_j of rank r_Z introduced on the channel going to server $j, j = 1, \dots, l$, such that a transformation matrix B_j is applied to Z_j. Let $D^f_{Q_j} = D_j^\top + E_j^f$. Therefore, server j receives the query matrix with additional errors:

$$\begin{aligned} Q^f_{j,rec} &= \left(\begin{array}{c:c} A_j + B_j Z_j & A_j D^f_{Q_j} + B_j Z_j \end{array} \right) \\ &= \left(\begin{array}{c:c} \hat{A}_j & \hat{Q}_j \end{array} \right), \end{aligned}$$

where $\hat{Q}_j = A_j D^f_{Q_j} + B_j Z_j$ is the erroneous received query, and $\hat{A}_j = A_j + B_j Z_j$.

The server then projects the query matrix received on its stored data and then appends the matrix with an identity matrix. The server then returns to the user a double-lifted matrix

$$R^f_j = \left(\begin{array}{c:c:c} I_{\rho\times\rho} & \hat{A}_j & Y_j \star \hat{Q}_j \end{array} \right),$$

where Y_j is the corresponding j^{th} block of Y.

The user receives the response

$$R^f_{j,rec} = \left(\begin{array}{c:c:c} A'_j + N_{e,j} & A'_j \hat{A}_j + N_{e,j} & A'_j (Y_j \star \hat{Q}_j) + N_{e,j} \end{array} \right)$$

from server j transmitted through the network, where $N_{e,j}$ represents the errors introduced on the downlink from server j to the user in round i.

From the received responses from all the servers, the user forms the matrix

$$R^f_{rec} = \left(\begin{array}{c:c:c} A' + N_e & A'\hat{A} + N_e & A'(Y \star \hat{Q}) + N_e \end{array} \right),$$

where

$$A' = \begin{pmatrix} A'_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A'_l \end{pmatrix}, \hat{A} = \begin{pmatrix} \hat{A}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{A}_l \end{pmatrix},$$

and

$$N_e = \begin{pmatrix} N_{e,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{e,l} \end{pmatrix}.$$

From the query construction, it is known that $Y \star AD^f_{Q_j}$ is a codeword of a rank metric code $\mathcal{G}(n, k + t\rho + \beta - 1)$, but the problem is that the received word, $Y \star (AD^f_{Q_j} + BZ)$, is not a codeword of a rank metric code. However, $Y \star (D^f_{Q_j} + BZ)$ can be seen as the codeword, $Y \star D^f_{Q_j}$, of the rank metric code $\mathcal{C} \star (\mathcal{D} + \mathcal{E}^f)$, in addition to the noise introduced on the downlink, N_e, and some noise function $F(BZ)$ of $AD^f_{Q_j}$ along with the errors BZ. Let $\mathrm{rk}_{q^s}(F(BZ)) \leq \epsilon_q$, and let $\epsilon = \epsilon_q + \epsilon_r$. We note here that ϵ is an upper bound to the rank of the errors introduced, and τ is an upper bound to the number of erasures in the system.

Now, if $n - k - t\rho - \beta \geq \tau + 2\epsilon$, the user will be able to decode the requested file X^f with a rate $\frac{\beta}{n}$.

In general, if errors occur on both links, we have the following theorem:

Theorem 4. *Assume that the queries and responses are sent through a network where errors occur. Moreover, assume that $n \geq \tau + 2\epsilon + k + t\rho + \beta$. The user can then decode the errors, and thus retrieve the requested file, X^f, with a rate $\geq \frac{n-k-t\rho-2\epsilon-\tau+1}{n} = \frac{\beta}{n}$.*

Proof. The proof of this theorem follows similarly to the proof of Theorem 3. The inequality in the rate expression is due to the fact that full $t\rho$ collusion is assumed while only certain $t\rho$ sub-servers can collude.

□

V. Conclusion

In this paper, we consider a distributed storage system such that the data is encoded using an MRD code and stored on servers. The user and servers communicate over a random linear network. The user sends queries to the servers across the random network, such that the queries give no information about the file requested by the user to any server. It is further assumed that t nodes collude in an effort to figure out the index of the file requested by the user, and the network is assumed to introduce erasures and errors into the queries and the responses. In this work, we have constructed a PIR scheme that allows the user to retrieve a file from a DSS over such a network privately. The scheme achieves asymptotically high PIR rates as $q \to \infty$. In this work, we consider the intermediate nodes in the network as dummy relays that only receive, combine and forward packets without storing data. However, this can be generalized in future work, as the intermediate nodes can be considered as other users that might collude or trade the information with third parties.

References

[1] 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.

[2] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *Journal of the ACM (JACM)*, vol. 55, no. 1, pp. 1–16, 2008.

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

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

[5] Z. Dvir and S. Gopi, "2 server PIR with sub-polynomial communication," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, ser. STOC '15. New York, NY, USA: ACM, 2015, pp. 577–584. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746546

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

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

[8] S. R. Blackburn and T. Etzion, "PIR array codes with optimal PIR rates," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2658–2662.

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

[10] H. Sun and S. A. Jafar, "The capacity of private information retrieval with colluding databases," in *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*. IEEE, 2016, pp. 941–946.

[11] 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.

[12] 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*, 2018.

[13] 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.

[14] H. Sun and S. A. Jafar, "Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti et al," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1893–1897.

[15] Q. Wang and M. Skoglund, "Symmetric private information retrieval for MDS coded distributed storage," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[16] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," in *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2016, pp. 1–5.

[17] Q. Wang and M. Skoglund, "Secure symmetric private information retrieval from colluding databases with adversaries," *arXiv preprint arXiv:1707.02152*, 2017.

[18] ——, "Symmetric private information retrieval for MDS coded distributed storage," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[19] 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 Transactions on Information Theory*, 2018.

[20] S. Kumar, H.-Y. Lin, E. Rosnes, and A. Graell I Amat, "Achieving private information retrieval capacity in distributed storage using an arbitrary linear code," *arXiv preprint arXiv:1712.03898*, 2017.

[21] Q. Wang and M. Skoglund, "On PIR and symmetric PIR from colluding databases with adversaries and eavesdroppers," *IEEE Transactions on Information Theory*, 2018.

[22] K. Banawan and S. Ulukus, "The capacity of private information retrieval from byzantine and colluding databases," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 1206–1219, 2019.

[23] ——, "Noisy private information retrieval: On separability of channel coding and information retrieval," *arXiv preprint arXiv:1807.05997*, 2018.

[24] D. Silva, F. R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE transactions on information theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[25] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Aug. 2000.

[26] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[27] G. Richter and S. Plass, "Error and erasure decoding of Rank-Codes with a modified Berlekamp-Massey algorithm," in *ITG Conf. Source Channel Coding (SCC)*, 2004.

[28] A. Wachter-Zeh, V. Afanassiev, and V. Sidorenko, "Fast decoding of Gabidulin codes," *Des. Codes Cryptogr.*, vol. 66, no. 1, pp. 57–73, Jan. 2013.

[29] S. Puchinger and A. Wachter-Zeh, "Sub-quadratic decoding of Gabidulin codes," in *2016 IEEE International Symposium on Information Theory (ISIT)*, Jul. 2016.

[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 Transactions on Information Theory*, 2019.

[31] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.

[32] P. Delsarte, "Bilinear forms over a finite field with applications to coding theory," *J. Combinatorial Theory Ser. A*, vol. 25, no. 3, pp. 226–241, 1978.

[33] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Inf. Transm.*, vol. 21, no. 1, pp. 3–16, 1985.

[34] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inform. Theory*, vol. 37, no. 2, pp. 328–336, Mar. 1991.

[35] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. El Rouayheb, "Private information retrieval schemes for coded data with arbitrary collusion patterns," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1908–1912.

[36] W. Halbawi, T. Ho, and I. Duursma, "Distributed Gabidulin codes for multiple-source network error correction," in *International Symposium on Network Coding (NetCod)*, 2014.