Private Information Retrieval from MDS Coded Data in Distributed Storage Systems

Razan Tajeddine, Salim El Rouayheb
ECE Department, IIT, Chicago
Emails: rtajeddi@hawk.iit.edu, salim@iit.edu

***Abstract*—We consider the problem of providing privacy, in the private information retrieval (PIR) sense, to users requesting data from a distributed storage system (DSS). The DSS uses an (n, k) Maximum Distance Separable (MDS) code to store the data reliably on unreliable storage nodes. Some of these nodes can be spies which report to a third party, such as an oppressive regime, which data is being requested by the user. An information theoretic PIR scheme ensures that a user can satisfy its request while revealing, to the spy nodes, no information on which data is being requested. A user can achieve PIR by downloading all the data in the DSS. However, this is not a feasible solution due to its high communication cost. We construct PIR schemes with low download communication cost. When there is $b = 1$ spy node in the DSS, we construct PIR schemes with download cost $\frac{1}{1-R}$ per unit of requested data ($R = k/n$ is the code rate), achieving the information theoretic limit for linear schemes. The proposed schemes are universal since they depend on the code rate, but not on the generator matrix of the code. When there are $2 \leq b \leq n-k$ spy nodes, we devise linear PIR schemes that have download cost equal to $b + k$ per unit of requested data.**

I. Introduction

Consider the following scenario. A group of online peers (storage nodes) want to collaborate together to form a peer-to-peer (p2p) distributed storage system (DSS) to store and share files reliably, while ensuring information theoretic private information retrieval (PIR). The PIR [1], [2] property allows a user (possibly one of the peers) to download a file while revealing no information about which file is being downloaded. We are mainly motivated by the following two applications: 1) A DSS that protects users from surveillance and monitoring, for instance from an oppressive regime. The people (peers) collectively contribute to storing the data and making it pervasively available online. But, some peers could be spies for the regime. They could turn against their "neighbors" and report to the oppressor the identity of users accessing some information deemed to be anti-regime (blogs, photos, videos, etc.), leading to their persecution; 2) A DSS that protects the personal information of users, such as gender, age group, disease, etc., which can be inferred from their file access history. This information can potentially be used to target them with unwanted advertisement, or even affect them adversarially in other areas, such as applications to health insurance or bank loans. In this respect, the studied DSS can provide an infrastructure, at least in theory, over which applications, such as cloud storage and social networking, can be run with a privacy guarantee for the users.

We suppose the DSS is formed of n peers or nodes. Peers can be temporarily offline or can leave the system at any time. The data is stored redundantly in the system to guarantee its durability and availability. We assume that the DSS uses an (n, k) maximum distance separable (MDS) code that can tolerate $n-k$ simultaneous node failures. A certain number of nodes in the DSS, say b, whose identities are unknown to the users or the system, are spies and can report the user requests to the oppressor, or sell this information to interested third parties. The user can always achieve PIR by asking to download all the files in the DSS. However, this solution is not feasible due to its high communication cost, and more efficient solutions have been studied in the PIR literature [3]–[9] assuming the data is *replicated* in the system. The next example illustrates a PIR scheme with efficient communication cost that can be run on MDS coded data.

Example 1: Consider a DSS formed of $n = 4$ nodes that stores m files $(a_i, b_i), a_i, b_i \in GF(3^\ell), i = 1, 2, \dots, m$. The DSS uses an $(n, k) = (4, 2)$ MDS code over $GF(3)$ to store the files. Nodes $1, \dots, 4$ store respectively, $a_i, b_i, a_i + b_i, a_i + 2b_i, i = 1, \dots, m$. Suppose the user is interested in retrieving file f, i.e., (a_f, b_f), which can equally likely be any of the m files. To this end, the user generates a random vector $U = (u_1, \dots, u_m)$ with components chosen independently and uniformly at random from the underlying base field $GF(3)$. It sends the query vector $Q = U$ to nodes 1 and 2 and $Q = U + V_f$ to nodes 3 and 4, where V_f is the all zero vector of length m with a 1 in the f^{th} entry. Upon receiving the user's request, each node in the DSS returns to the user the projection of all its data on the received query vector. For instance, suppose that the user wants file 1. Then, nodes $1, \dots, 4$ return the following symbols from $GF(3^\ell)$, $I_1, I_2, a_1 + b_1 + I_1 + I_2, a_1 + 2b_1 + I_1 + 2I_2$, where $I_1 = \sum_{i=1}^m u_i a_i$ and $I_2 = \sum_{i=1}^m u_i b_i$ are thought of as "interference" terms. The returned information form an invertible linear system and the user can decode a_1 and b_1. Assume that the DSS contains $b = 1$ spy node. Then, the proposed scheme achieves PIR since the query vector to each node is statistically independent of the file index f. However, if the spy node, say node 1, knows the query vector of another node, say node 3, it may be able to pin down which file the user wanted, by computing $V_f = Q - U$. However, we assume that the spy node does not have access to the queries coming to the other regular (non-spy) nodes, and PIR is indeed achieved here. This PIR scheme downloads 4 symbols to retrieve a file

978-1-5090-1806-2/16/$31.00 ©2016 IEEE

of size 2 symbols. We say that the communication price of privacy $cPoP = 4/2 = 2$ for this scheme, which does not depend on the number of files in the system.

Replication-based PIR: PIR was first introduced in the seminal papers of Chor et al. in [1], [2] followed by significant amount of research in this area [3]–[7], [10], [11]. The classical model considers a binary database of length m and a user that wishes to retrieve privately the value of a bit (a record) in it, while minimizing the total communication cost including the upload (query) and download phase. Chor et al. showed that if there is one server storing the database, the user has to download the whole database in order to achieve information theoretic PIR. However, when the database is replicated on n non-colluding (non-cooperating) servers (nodes), they devised a PIR scheme with total, upload and download, communication cost, $O((n^2 \log n)m^{1/n})$ (and $O(m^{1/3})$ for the special case of $n = 2$). In the past few years, there has been significant progress in developing PIR protocols with total communication cost that is subpolynomial in the size of the database [10]–[12]. PIR in a computational sense was shown to be achievable with a single server (no replication) in [13] assuming the hardness of quadratic residuosity problem. PIR schemes on databases that are replicated but not perfectly synchronized were studied in [14].

Coded PIR: The original model studied in PIR assumes that the entire data is replicated on each node. PIR on coded data was studied in the literature on Batch Codes [15], where the data is coded to allow parallel processing leading to amortizing the PIR communication cost over multiple retrievals. Recently, the PIR problem in DSSs that use erasure codes was initiated in [8], where it was shown that one extra bit of download is sufficient to achieve PIR assuming the number of servers n to be exponential in the number of files. Bounds on the information theoretic tradeoff between storage and download communication cost for coded DSSs were derived in [9]. The setting when nodes can be byzantine (malicious) was considered in [16] and robust PIR schemes were devised using locally decodable codes. In [17], methods for transforming a replication-based PIR scheme into a coded-based PIR scheme with the same communication cost were studied.

Contributions: Motivated by the two DSS applications mentioned earlier, we draw the following distinctions with the previous literature on coded PIR: (i) To the best of our knowledge, all the previous work on coded PIR, except for [9], assumes that the code is used to encode together data from different files (records). However, the model here is different, since in DSS applications only data chunks belonging to the same file are encoded together (as done in Example 1); (ii) The work in [9] studies fundamental limits on the costs of coded PIR. Here, we provide explicit constructions of PIR scheme with efficient communication cost.

In comparison with the classical literature on replication-based PIR, we make the following observations: (i) We focus only on the number of downloaded bits (or symbols) in the communication cost of a PIR scheme. The motivation is that typically in DSSs, the size of a file is relatively larger than the total number of files; (ii) We assume that a regular non-spy node believes in the privacy cause and does not reveal the queries it receives to any other node. Only the spy nodes cooperate (collude) and share their queries in the hope of determining the requested file. This justifies the constraint later that at most b nodes can collude, and that the PIR scheme should ensure privacy even if such collusions occur.

In the model we study, we assume that the MDS code parameters (n, k) are given and depend on the desired reliability level for the data. Therefore, they are not design parameters that can be chosen to optimize the efficiency of the PIR scheme. However, the code itself may have to be designed jointly with the PIR scheme. A PIR scheme incurs many overheads on the DSS, including communication cost, computations [6], and connectivity (user contacts n instead of k nodes, as seen in Example 1). However, we measure here the efficiency of a PIR scheme only by its total download communication cost, which we refer to as the *communication Price of Privacy (cPoP). The cPoP is the total amount of downloaded data per unit of retrieved file.* The following questions naturally arise here: (1) What is the minimum achievable cPoP for given n, k and b? (2) How to efficiently construct codes and PIR schemes that achieve optimal cPoP? (3) Do the code and PIR scheme have to be designed jointly to achieve optimum cPoP? The last question addresses the problem of whether reliability and PIR could be addressed separately in a DSS. Moreover, it may have practical implications on whether data already existing in coded form needs to be re-encoded to achieve PIR with minimum cPoP.

In this paper, we make progress towards answering the last two questions and provide constructions of efficient PIR schemes for querying MDS coded data. Specifically, we make the following contributions: (i) For $b = 1$, i.e., a single spy node, we construct linear PIR scheme with $cPoP = \frac{1}{1-R}$ ($R = k/n$ is the code rate), thus achieving the lower bound on cPoP for linear schemes in [9]; (ii) For $2 \leq b \leq n - k$, we construct linear PIR schemes with $cPoP = b + k$. While the minimum cPoP in this regime is unknown, the constructed schemes have a cPoP that does not depend on m, the number of files in the system. An important property of the scheme for $b = 1$ is its *universality*. It depends only on n, k and b, but not on the generator matrix of the code. Moreover, both of these schemes can be constructed for any given MDS code, i.e., it is not necessary to design the code jointly with the PIR scheme. This implies that b does not have to be a rigid system parameter. Each user can choose their own value of b to reflect its desired privacy level, at the expense of a higher $cPoP$. The DSS can serve all the users simultaneously storing the same encoded data, i.e., without having to store different encodings for different values of b.

II. Model

DSSs: Consider a distributed storage system (DSS) formed of n storage nodes indexed from 1 to n. The DSS stores m files, $X^1, \ldots, X^m$, of equal sizes. The DSS uses WLOG a systematic(n, k) MDS code over $GF(q)$ to store the data

	node 1	node 2	...	node k	node $k+1$	...	node n
file 1	x_{11}^1	x_{21}^1	$\dots$	x_{k1}^1	$\lambda_{1,k+1}x_{11}^1 + \cdots + \lambda_{k,k+1}x_{k1}^1$	$\dots$	$\lambda_{1n}x_{11}^1 + \cdots + \lambda_{kn}x_{1k}^1$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$x_{1\alpha}^1$	$x_{2\alpha}^1$	$\dots$	$x_{k\alpha}^1$	$\lambda_{1,k+1}x_{1\alpha}^1 + \cdots + \lambda_{k,k+1}x_{k\alpha}^1$	$\dots$	$\lambda_{1n}x_{1\alpha}^1 + \cdots + \lambda_{kn}x_{k\alpha}^1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
file m	x_{11}^m	x_{21}^m	$\dots$	x_{k1}^m	$\lambda_{1,k+1}x_{11}^1 + \cdots + \lambda_{k,k+1}x_{k1}^m$	$\dots$	$\lambda_{1n}x_{11}^m + \cdots + \lambda_{kn}x_{1k}^m$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$x_{1\alpha}^m$	$x_{2\alpha}^m$	$\dots$	$x_{k\alpha}^m$	$\lambda_{1,k+1}x_{1\alpha}^m + \cdots + \lambda_{k,k+1}x_{k\alpha}^m$	$\dots$	$\lambda_{1n}x_{1\alpha}^1 + \cdots + \lambda_{kn}x_{k\alpha}^m$

TABLE I: The layout of the encoded symbols of the m files in the DSS.

redundantly and achieve reliability against $n-k$ node failures. We assume that each file, $X^i, i = 1, \dots, m$, is divided into α stripes, and each stripe is divided into k blocks. We represent the file $X^i = [x_{jl}^i], j = 1, \dots, k, l = 1, \dots, \alpha$, as a $k \times \alpha$ matrix, with symbols from the finite field $GF(q^\ell)$. The j^{th} row of the matrix X^i is stored on the systematic node $j, j = 1, \dots, k$. Each stripe of each file is encoded separately using the same systematic MDS code with a $k \times n$ generator matrix $\Lambda = [\lambda_{ij}]$ with elements in $GF(q)$. Since the code is systematic, the square submatrix of Λ formed of the first k columns is the identity matrix. The encoded data is stored on the DSS as shown in Table I. We assume that the user knows this layout table, i.e., it knows the coding coefficients for each node. We denote by $W_i \in GF(q^\ell)^{m\alpha}$ the column vector representing all the data on node i.

PIR: Suppose the user wants file X^f, where f is chosen uniformly at random from the set $[m] = \{1, \dots, m\}$. To retrieve file X^f, the user sends requests to the nodes, among which there are b collaborating spy nodes. The goal is to devise a PIR scheme that allows the user to decode X^f, while revealing no information, in an information theoretic sense, about f to the spy nodes. The user does not know the identity of the spy nodes. Otherwise, it can avoid contacting them. The spy nodes can collaborate together and analyze the different requests they receive from the user in order to identify the requested file. However, as explained in the introduction, the spy nodes do not have access to the requests coming to the regular nodes in the system. Under this setting, we are interested in linear PIR schemes.

Definition 1: A PIR scheme is linear over $GF(q)$, and of dimension d, if it consists of the following two stages.

1. Request stage: Based on which file the user wants, it sends requests to a subset of nodes in the DSS. The request to node i takes the form of a $d \times m\alpha$ query matrix Q_i over $GF(q)$.

2. Download stage: Node i responds by sending the projection of its data onto Q_i, i.e., $R_i = Q_i W_i \in GF(q^\ell)^d$.

We think of each query matrix Q_i as formed of d sub-queries corresponding to each of its d rows. Moreover, we think of the response of node i as formed of d sub-responses corresponding to projecting the node data on each row of Q_i.

Definition 2 (Information theoretic PIR): A linear PIR scheme achieves (perfect) information theoretic PIR iff $H(f|Q_j, j \in \gamma) = H(f)$, for all sets $\gamma \in [n], |\gamma| = b$ and any number of files m[1]. Here, $H(.)$ denotes the entropy function.

The objective is to design a linear PIR scheme that (i) allows the user to decode its requested file X^f and (ii) achieves information theoretic PIR with a low cPoP that does not depend on m. In the classical literature on PIR, the communication cost includes both the number of bits exchanged during the request and download stages. In DSSs, we assume that the content of the file dominates the total communication cost, i.e., ℓ is much larger than m. Therefore, we will only consider the download communication cost, which we will refer to as the communication price of privacy (cPoP).

Definition 3 (cPoP): The communication Price of Privacy (cPoP) of a PIR scheme is the ratio of the total number of bits sent from the nodes to the user during the download stage to size of the requested file.

III. Main Results

In this section, we state our two main results. The proof of Theorem 1 is given in Section V. The proof of Theorem 2 is omitted and can be found in the extended version [18].

Theorem 1: Consider a DSS using an (n, k) MDS code over $GF(q)$, with $b = 1$ spy node. Then, the linear PIR scheme over $GF(q)$ described in Section IV achieves perfect PIR with $cPoP = \frac{1}{1-R}$, where $R = k/n$.

The existence of PIR schemes over large fields that can achieve $cPoP = \frac{1}{1-R}$ for $b = 1$ follows from Theorem 4 in [9]. We prove Theorem 1 by providing an explicit construction of the linear PIR scheme. The proposed PIR construction is over same field over which the code is designed and is universal in the sense that it depends only on the parameters n, k and b and not on the generator matrix of the code.

Theorem 2: Consider a DSS using an (n, k) MDS code over $GF(q)$, with possibly $2 \leq b \leq n-k$ spy nodes. Then, there exists an explicit linear PIR scheme over the same field that achieves perfect PIR with $cPoP = b + k$.

IV. PIR scheme construction for $b = 1$

We describe here the PIR scheme referred to in Theorem 1. We assume WLOG that the MDS code is systematic. The PIR scheme requires the number of stripes $\alpha = n - k$ and the dimension $d = k$. Using the division algorithm, we write $\alpha = \beta k + r$ where, β and r are integers and $0 \leq r < k$ and $\beta \geq 0$.

[1]We require the PIR scheme to be applicable to any number of files m. This is related to the concept of strong achievability in [9].

Stripes	Sys. nodes 1	2	3	4	Parity nodes 5	6	7	8	9	10	11	12	13	14	15
r 1	1	2	3	4											
2	2	3	4	1											
3	3	4	1	2											
k 4					1	1	1	1							
5					2	2	2	2							
6					3	3	3	3							
7					4	4	4	4							
k 8									1	1	1	1			
9									2	2	2	2			
10									3	3	3	3			
11									4	4	4	4			
					k				k						

TABLE II: Example of the retrieval pattern for $(n, k) = (15, 4)$. The $\alpha \times n$ entries of the table correspond to the $\alpha \times n$ coded symbols of the wanted file. All entries with same number, say j (also given the same color) are privately retrieved in the j^{th} sub-query. Note that there are $k = 4$ nodes, including the last $r = 3$ nodes, in every sub-query, that do not have any retrieved symbols. The responses of these nodes are used to decode the "interference" from all the files, needed to confuse the nodes about what is being requested. This interference is then cancelled out from the other sub-responses in order to decode the desired file symbols in each sub-query.

The scheme consists of the user sending a $d \times m\alpha$ query matrix Q_i to each node $i, i = 1, \ldots, n$. To form the query matrices, the user generates a $d \times m\alpha$ random matrix $U = [u_{ij}]$, whose elements are chosen uniformly at random from $GF(q)$, the same field over which the MDS code is defined. The query matrices are then defined as follows:

$$Q_i = U + V_i, \quad i = 1, \ldots, n - r, \tag{1}$$
$$Q_i = U, \quad i = n - r + 1, \ldots, n. \tag{2}$$

U is the random component of the query aimed at confusing the node about the request, whereas V_i is a deterministic matrix that depends on the index f of the requested file and its role is to ensure that the user can decode the file. The matrices V_i are 0-1 matrices of dimensions $d \times m\alpha$. A "1" in the $(j, l)^{th}$ position of V_i implies that, during the j^{th} subquery, the l^{th} symbol on node i is being retrieved privately. The matrices V_i are designed to satisfy the following two conditions:

1) in each sub-query, a new systematic symbol of X^f is retrieved from each of the first r systematic stripes, and
2) βk new coded symbols are retrieved from β different stripes, i.e., k symbols per stripe.

Based on these desired retrieval patterns, we choose

$$V_1 = \left[\mathbf{0}_{k\times(f-1)\alpha} \;\vdots\; \begin{matrix} I_{r\times r} \\ \mathbf{0}_{(k-r)\times r} \end{matrix} \;\vdots\; \mathbf{0}_{k\times\beta k} \;\vdots\; \mathbf{0}_{k\times(m-f)\alpha} \right], \tag{3}$$

and $V_i, i = 2, \ldots, k$, is obtained from matrix V_{i-1} by a single downward cyclic shift of its row vectors. As for the parity nodes $i = sk + 1, \ldots, sk + k$, $s = 1, \ldots, \beta$, we choose

$$V_i = \left[\mathbf{0}_{k\times(f-1)\alpha+r+(s-1)k} \;\vdots\; I_{k\times k} \;\vdots\; \mathbf{0}_{k\times(\beta-s)k+(m-f)\alpha} \right]. \tag{4}$$

Example 2 (Retrieval pattern):
Consider a DSS using an $(n, k) = (15, 4)$ MDS code. Therefore, we have $d = k = 4$ sub-queries to each node. Also, the number of stripes is $\alpha = n - k = 11$. This gives $\beta = 2$ and $r = 3$. Table II gives the retrieval pattern of the PIR scheme.

Stripes	Sys. nodes 1	2	Parity nodes 3	4	5
1	1	2			
2			1	1	
3			2	2	

TABLE III: Retrieval pattern for a (5,2) code.

Example 3 (Decoding):

Consider a DSS using a $(5, 2)$ MDS code with generator matrix $\Lambda = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 & 3 \end{pmatrix}$, over $GF(5)$. Suppose the DSS is storing $m = 3$ files, X^1, X^2, X^3, following the layout in Table I. Our goal is to construct a linear scheme that achieves perfect PIR against $b = 1$ spy node, with $cPoP = \frac{1}{1-R} = \frac{5}{3}$. The construction in Section IV gives $\alpha = n - k = 3$ and $d = k = 2$. Therefore, we get $\beta = 1$ and $r = 1$. Suppose WLOG that the user wants file X^1, i.e., $f = 1$. The user generates an 2×9 random matrix $U = [u_{ij}]$, whose elements are chosen uniformly at random from $GF(5)$. For the nodes $1, \ldots, 4$, the query matrix $Q_i = U + V_i$, and $Q_5 = U$. Therefore, following Table III we have

$$Q_1 = \begin{bmatrix} u_{11}+1 & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} & u_{17} & u_{18} & u_{19} \\ u_{21} & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} & u_{27} & u_{28} & u_{29} \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} & u_{17} & u_{18} & u_{19} \\ u_{21}+1 & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} & u_{27} & u_{28} & u_{29} \end{bmatrix},$$

$$Q_3 = \begin{bmatrix} u_{11} & u_{12}+1 & u_{13} & u_{14} & u_{15} & u_{16} & u_{17} & u_{18} & u_{19} \\ u_{21} & u_{22} & u_{23}+1 & u_{24} & u_{25} & u_{26} & u_{27} & u_{28} & u_{29} \end{bmatrix},$$

$$Q_4 = \begin{bmatrix} u_{11} & u_{12}+1 & u_{13} & u_{14} & u_{15} & u_{16} & u_{17} & u_{18} & u_{19} \\ u_{21} & u_{22} & u_{23}+1 & u_{24} & u_{25} & u_{26} & u_{27} & u_{28} & u_{29} \end{bmatrix}.$$

This construction achieves perfect PIR, since the only information any node i knows about f is through the query matrix Q_i, which is random and independent of f. Next, we want to illustrate how the user can decode the file symbols. Each node sends back the length 2 vector, $R_i = (r_{i1}, r_{i2}) = Q_i W_i, i = 1, \ldots, 5$, to the user. Consider the sub-responses of the 5 nodes to the first sub-query. They form the following linear system:

$$x^1_{11} + I_1 = r_{11} \tag{5}$$
$$I_2 = r_{21} \tag{6}$$
$$x^1_{12} + x^1_{22} + I_1 + I_2 = r_{31} \tag{7}$$
$$x^1_{12} + 2x^1_{22} + I_1 + 2I_2 = r_{41} \tag{8}$$
$$I_1 + 3I_2 = r_{51}, \tag{9}$$

where $I_l = U_1 W_l, l = 1, 2$, and U_1 is the first row of U.

The user can first decode I_1 and I_2 from (6) and (9). Then, canceling out the values of I_1 and I_2 from the remaining equations, the user can solve for x^1_{11}, x^1_{12} and x^1_{22}. Similarly, the user can obtain x^1_{21}, x^1_{13} and x^1_{23} from the sub-responses to the second sub-query. This PIR scheme downloads 2 symbols from each server. Therefore, it has a $cPoP = \frac{10}{6} = \frac{5}{3}$, which matches the bound in Theorem 1.

V. Proof of Theorem 1

We prove Theorem 1 by showing that the scheme described in Section IV has the following properties.

Decodability: The scheme consists of $d = k$ sub-queries. We focus on the i^{th} sub-query, corresponding to the i^{th} row of the query matrices. From (1), (2), (3) and (4), we can find the sub-responses of the nodes. The systematic nodes $l, l = 1, \dots, k$, return the following sub-responses to the i^{th} sub-query,

$$x_{i1}^f + I_l \quad l=i \tag{10}$$

$$I_l \quad l=(i+1)_k, \dots, (i+k-r)_k \tag{11}$$

$$x_{l(i+k+1-l)_k}^f + I_l \quad l=(i+k-r+1)_k, \dots, (i+k-1)_k, \tag{12}$$

where $I_l = U_i W_l$, $l = 1, \dots, k$, U_i is the i^{th} row of U, and with the following notation: $(a)_b = a \mod b$ if $a \neq b$, and $(b)_b = b$. The parity nodes, $l = n - r + 1, \dots, n$, return the following r sub-responses, respectively, to the i^{th} sub-query,

$$\lambda_{1l} I_1 + \lambda_{2l} I_2 + \dots + \lambda_{kl} I_k, \quad l = n - r + 1, \dots, n. \tag{13}$$

The parity nodes, $l = sk + 1, \dots, sk + k$, for $s = 1, \dots, \beta$, return the following sub-responses to the i^{th} sub-query,

$$\lambda_{1l} x_{1,r+(s-1)k+i}^f + \lambda_{2l} x_{2,r+(s-1)k+i}^f + \dots + \lambda_{kl} x_{k,r+(s-1)k+i}^f$$
$$+ \lambda_{1l} I_1 + \lambda_{2l} I_2 + \dots + \lambda_{kl} I_k,$$
$$l = sk + 1, \dots, sk + k. \tag{14}$$

Equations (11) and (13) form a linear system of k equations in the k unknowns $I_1, \dots, I_k$. The linear equations in this system correspond to the k columns of the code generator matrix Λ having indices $(i+1)_k, \dots, (i+k-r)_k$ and $n-r+1, \dots, n$. Since Λ is a generator matrix of an (n, k) MDS code, any $k \times k$ square submatrix of Λ is full rank. Therefore, (11) and (13) form an invertible system, and the user can decode $I_1, \dots, I_k$. By canceling out these terms from (10) and (12) the user can decode r different systematic symbols from its wanted file. Similarly, from (14) the user decodes k coded symbols from each of β different stripes. Since the code is MDS, the k coded symbols in every stripe can be used to decode the k systematic symbols corresponding to that stripe. In total, the user can decode $\beta k + r = \alpha$ symbols of its wanted file per sub-query. But, there are k sub-queries and, by the design of the query matrices V_i, the retrieved symbols are different in each sub-query. Therefore, the user is able to decode all the $\alpha \times k$ symbols in its wanted file.

Privacy: Since $b = 1$, the only way a node i can learn information about f is from its own query matrix Q_i. By, construction Q_i is statistically independent of f and this scheme achieves perfect privacy.

cPoP: Every node $i \in [n]$ responds with $d = k$ symbols. Therefore, the total number of symbols downloaded by the user is kn. Therefore, $cPoP = \frac{kn}{k(n-k)} = \frac{1}{1-R}$.

VI. Conclusion

We studied the problem of constructing PIR schemes with low communication cost for requesting data from a DSS that uses MDS codes. Some nodes in the DSS may be spies who will report to a third party, such as an oppressive regime, which data is being requested by a user. The objective is to allow the user to obtain its requested data without revealing any information on the identity of the data to the spy nodes. We constructed PIR schemes against one spy node that achieve the information theoretic limit on the download communication cost for linear schemes. An important property of these schemes is their universality since they depend on the code rate, but not on the MDS code itself. When there are more than one spy node, we devised PIR schemes that have download cost independent of the total size of the data in the DSS.

References

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *IEEE Symposium on Foundations of Computer Science*, pp. 41–50, 1995.

[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] 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.

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

[5] A. Beimel and Y. Ishai, "Information-theoretic private information retrieval: A unified construction," in *Automata, Languages and Programming*, pp. 912–926, Springer, 2001.

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

[7] 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.*, pp. 261–270, IEEE, 2002.

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

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

[10] 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*, STOC '15, (New York, NY, USA), pp. 577–584, ACM, 2015.

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

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

[13] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," in *FOCS*, p. 364, IEEE, 1997.

[14] G. Fanti and K. Ramchandran, "Multi-server private information retrieval over unsynchronized databases," in *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*, pp. 437–444, Sept 2014.

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

[16] 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*, pp. 222–239, Springer, 2014.

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

[18] R. Tajeddine, S. El Rouayheb, "Private Information Retrieval from MDS Coded data in Distributed Storage Systems (extended version)," 2016. http://www.ece.iit.edu/~salim/PIRMDS.pdf.