Private Information Retrieval in Graph-Based Replication Systems

Netanel Raviv, *Member, IEEE*, Itzhak Tamo, and Eitan Yaakobi, *Senior Member, IEEE*

***Abstract*— In a Private Information Retrieval (PIR) protocol, a user can download a file from a database without revealing the identity of the file to each individual server. A PIR protocol is called *t-private* if the identity of the file remains concealed even if t of the servers collude. Graph based replication is a simple technique, which is prevalent in both theory and practice, for achieving robustness in storage systems. In this technique each file is replicated on two or more storage servers, giving rise to a (hyper-)graph structure. In this paper we study private information retrieval protocols in graph based replication systems. The main interest of this work is understanding the collusion structures which emerge in the underlying graph. Our main contribution is a 2-replication scheme which guarantees perfect privacy from acyclic sets in the graph, and guarantees partial-privacy in the presence of cycles. Furthermore, by providing an upper bound, it is shown that the PIR rate of this scheme is at most a factor of two from its optimal value for regular graphs. Lastly, we extend our results to larger replication factors and to graph-based coding, a generalization of graph based replication that induces smaller storage overhead and larger PIR rate in many cases.**

***Index Terms*— Private information retrieval (PIR), distributed storage systems.**

I. Introduction

RECENT data breaches in major corporations have emphasized the need for privacy in the digital era. Among the many challenges that designers of distributed storage systems face is the ability to support *private information retrieval* (PIR) protocols. These protocols enable the user to retrieve an entry in the database, while concealing the identity of that entry from the servers. This paper studies PIR protocols in a particular common type of distributed storage systems.

Manuscript received March 4, 2019; revised September 12, 2019; accepted November 9, 2019. Date of publication November 22, 2019; date of current version May 20, 2020. The work of N. Raviv was supported in part by the Post-Doctoral Fellowship of the Center for the Mathematics of Information (CMI), California Institute of Technology. The work of I. Tamo was supported in part by the Israel Science Foundation (ISF) under Grant 1030/15 and Grant NSF-BSF 2015814. The work of E. Yaakobi was supported in part by the Israel Science Foundation (ISF) under Grant 1817/18. This work was presented in part at the International Symposium on Information Theory (ISIT), Vail, CO, USA, 2018.

N. Raviv was with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA. He is now with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA.

I. Tamo is with the Department of Electrical Engineering—Systems, Tel-Aviv University, Tel-Aviv 39040, Israel.

E. Yaakobi is with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel.

Communicated by A. Jiang, Associate Editor for Coding Theory.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2019.2955053

Coding for storage systems has developed tremendously in recent years. However, many system designers still favor replication techniques, over more involved ones, as a means to guarantee robustness against hardware failures [7], [14]. In spite of having high storage overhead and low failure resilience, replication is often preferred due to its simplicity of implementation. In addition, various types of replication systems are studied in theoretical research due to their real-world impact and ease of analysis [11], [21], [22], [32], [33]. However, since contemporary datasets are far too large to be stored on one machine, it is usually the case where every machine stores a small number of selected files from the dataset, each of which is replicated among geographically separated machines. In turn, such systems can be modeled as hypergraphs, where nodes represent storage servers and (hyper-)edges represent files. In these graphs, an edge is incident with a node if a copy of the respective file is stored on the respective server. Storage systems which broadly adhere to the above outline are called *graph-based replication systems*. A graph based replication system in which every file is replicated r times is called an r-*replication system*, and r is called its *replication factor*.

One of the most important metrics by which PIR protocols are measured is their *collusion resistance*. In its most simplistic form, a PIR protocol must guarantee perfect privacy against every individual server.[1] That is, it should be information-theoretically impossible for every individual server to infer any information regarding the identity of the requested file. The term collusion resistance measures the ability of a PIR protocol to perform beyond this baseline. That is, what is the maximum number of servers that still remain completely oblivious to the identity of the file, even if collusion among them is permitted. Traditionally, the term "collusion" stems from a mindset which considers the servers themselves as adversaries. Yet, the authors of this paper deem this interpretation obsolete, since it does not align with contemporary storage services. Instead, one can think of geographically separated servers as having independent security protocols, that must be individually broken by an adversary. In this case, the term "colluding servers" refers to a set of servers whose security was breached by an outside adversary, that can therefore observe their input and output. Normally, the term *t-privacy* of a given protocol indicates the maximum number of servers that cannot infer any information

[1] In some settings, only computational privacy is required, but this paper focus exclusively on perfect privacy.

regarding the identity of the file even if they collude; and in our alternative viewpoint, $t+1$ is the minimum number of individually-secured servers that must be breached by an adversary in order to infringe the perfect privacy of the protocol. Nevertheless, this alternative viewpoint is mathematically equivalent, and in our choice of terms we comply with the standard nomenclature.

Motivated by the prevalence of graph-based storage systems in the real world, in this paper we initiate a study about PIR protocols in such systems. Specifically, for a given number of servers, files, replication factor r, and a privacy parameter t, we wish to find an r-replication system which supports t-private PIR at the maximum possible rate, defined as the ratio between the size of the desired file and the amount of downloaded information. Since such systems are inherently non-uniform, in the sense that every server stores a different part of the dataset, one might expect that the collusion resistance will behave accordingly. Indeed, our results show that the right viewpoint for analyzing colluding sets is not their size, but rather the structure of their induced subgraph.

Consequently, our results also provide PIR protocols for a *given* r-replication system, in cases where one is not free to disperse the files as one pleases.[2] Specifically, we provide a PIR protocol for 2-replication systems and show that its PIR rate is at least half of its optimal value in many cases of interest. For larger replication factors we provide a simple scheme whose collusion resistance is less than the replication factor, and another scheme which enables larger collusion resistance by a reduction to the 2-replication case.

Further, we study an alternative *graph-based coding* approach, which generalizes graph-based coding, in which every file is coded by using an MDS code, and the resulting codeword symbols are dispersed as in graph-based replication systems. While this approach reduces the storage overhead and increases the PIR rate, our scheme requires a careful dispersion of files in order to guarantee nontrivial collusion resistance. The results in this paper, and graph-based coding in particular, call for future research and practical implementations, that would hopefully bring the vast PIR literature closer to realistic storage systems.

This paper is structured as follows. Preliminaries and previous works are discussed in Section II. Protocols and bounds for 2-replication systems are given in Section III, and larger replication factors are discussed in Section IV. Graph-based coding is discussed in Section V, and open problems for future research are discussed in Section VI.

II. Preliminaries

A. Problem Statement, Results, and Inherent Flaws

Given integers s, n, f, $r \geq 2$, and t, devise an r-replication system with s servers, n files of f symbols each, and a respective t-private PIR protocol. The figure of merit of any given protocol is its PIR rate, defined as the ratio between the amount of required information and the expected amount of downloaded information. Notice that a PIR algorithm must be random, and hence the amount of downloaded symbols might vary from one execution to another. Perfect privacy, however, must be maintained in all executions. Further, notice that every tuple of parameters is feasible at rate $1/n$, by downloading the entire dataset.

Our contributions in this paper are for the cases $r=2$ and $t \geq 2$ in Section III, a simple scheme for $2 \leq t < r$ in Subsection IV-A, and by reduction to $r=2$, for larger values of r and $t \geq r$ in Subsection IV-B. It will be clear in the sequel that our schemes rely heavily on the (hyper-)graph structure, and hence reduce to construction graphs with that structure. Our analysis also provides PIR protocols for *given* graphs, in cases where one cannot choose the dispersion of the files.

It is readily verified that if $r=2$, $t \geq 2$, and two servers store more than one file in common (a scenario that corresponds to the existence of parallel edges in the graph) then every PIR protocol must download all the mutual files from these two servers. This is an inherent flaw of graph-based replication, and to avoid trivialities, we restrict our attention to graphs with no parallel edges, and consequently to $n \leq \binom{s}{2}$. It will also be evident from our bound in Subsection III-B that for $t \geq 2$, the PIR rate must tend to zero as s tends to infinity (which is a necessity if one wishes n to tend to infinity, since $n \leq \binom{s}{2}$). This is also an inherent flaw of these systems. Under these unfortunate truths, we wish to maximize the PIR rate for a given set of parameters.

We comment that in general, we pay greater attention to the case $t \geq r$. In the case where $t < r$ and one can disperse files arbitrarily, one can store all information on r servers and apply standard PIR protocols for (fully) replicated storage (yet it remains open if this is optimal). If one is not free to disperse the file arbitrarily, the scheme in Subsection IV-A applies, and provides the same PIR rate as the $r=2$ case, but with no limitation on the number of files.

We also study graph-based coding, where instead of the parameter r above, one is given N and K, and is required to devise a t-private protocol for a system in which every file is coded with an (N,K) MDS code, and dispersed among N servers. These systems generalize graph-based replication, and induce lower storage overhead (i.e., the amount of redundant storage in the system) than r-replication systems if $K/N > 1/r$.

B. Previous Work

Originally defined in [8], the PIR problem has attracted a tremendous amount of research in the past two decades; and due to its tight connection with distributed storage, PIR enjoyed an increasing attention in the past few years. Since a comprehensive summary of previous works is beyond the scope of this paper, we list herein only a partial list of recent contributions, and elaborate on the most relevant ones.

The recent surge of interest in PIR, which addresses the problem from a distributed storage standpoint, includes the reduction of storage overhead by using error correcting codes in [12] and its improvement in [5]; obtaining secrecy by one

[2]This is the case, e.g., in cloud storage systems in which file dispersion is random, or in situations where the user does not own the data.

extra bit in [20] and its improvement in [6]; and an extensive line of works regarding achievability and capacity in various scenarios, such as multi-round, multi-message, symmetric, and with byzantine or colluding servers [2], [3], [23]–[26], [29]. This line of works is a natural extension of an earlier one in the computer science community, which addressed the problem in a more simplistic fashion. Namely, the dataset is assumed to be replicated in its entirety on all servers in the system, and the files are assumed to consist of a single bit. Furthermore, this problem is strongly connected to *locally decodable codes* [30], [31], and has seen a substantial progress recently [10].

All of the aforementioned works fall into either one of two extremes in the approach towards PIR. In one, the dataset in its entirety is stored in every server, and in the other it is coded by using an MDS code. The current work addresses a sweet spot between the two, that is strongly motivated by real-world applications [7], [14], as well as a plethora of storage models that were addressed in the past [11], [21], [22], [32], [33].

Nevertheless, two notions that are relevant to this work were recently addressed in the literature. First, one may consider the special case of graph-based replication in which the degree of every node in the graph (i.e., number of edges that are incident with it) is upper bounded by some parameter. Evidently, this special case is strongly connected to recent works [1], [28] that addressed the general coded PIR question in cases where each server is constrained to contain only a fraction of the entire dataset. Yet, [28] did not impose the particular replication structure that is fundamental to our approach, and more importantly, did not consider collusion. We also note that [1] contains bounds that are applicable in our case as well (specifically, by choosing $t = 2$ in [1, Thm. 2]), but are far more general, and as such provide weak bounds for our model.

Another notion that was previously studied is that of *collusion patterns* [16], [27]. In this variant, the system must guarantee collusion resistance against specific subsets of servers, rather than any subset up to a certain size. This notion bears some similarity to this work, since one may compel the vertices in these specific sets not to induce a subgraph which infringes privacy in our scheme. However, the approach and the results of these works is substantially different from ours, e.g., since [27] only discuss coded storage, and [16] discussed replication of the entire dataset in every server, and disjoint colluding sets.

Remark 1. *After preliminary versions of this paper were made publicly available, two follow-up pre-prints have been published online. In [4] the roles of the files and servers in the storage graph are reversed (i.e., vertices represent files and edges represent servers), and the special cases of the complete graph and the cycle graph are studied. Ref. [15] studies an additional constraint of X-security, where the data itself should be concealed from sets of at most X colluding servers. In addition, [15] characterizes the capacity in several cases, some of which by utilizing the size of the maximum 2-matching of the storage graph.*

C. Notations and Background

For a prime power q let $\mathbb{F}_q$ be the field with q elements. In a PIR protocol (not necessarily a graph-based one), a dataset $X = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top)^\top \in \mathbb{F}_q^{n \times f}$, which consists of n *files* $\{\mathbf{x}_i\}_{i=1}^n$ of f symbols each, is stored across s storage servers in a possibly coded manner.

The user wishes to download the file $\mathbf{x}_\phi$, where for the sake of the probabilistic analysis, ϕ is seen as uniformly distributed over $[n] \triangleq \{1, 2, \ldots, n\}$. To this end, the user uses randomness in order to generate queries $\mathbf{q}_1, \ldots, \mathbf{q}_s$, one for every server. In turn, server i replies with $\mathbf{a}_i$, that is a deterministic function of $\mathbf{q}_i$ and of the server's content. The protocol is called *t-private* if for every subset $\mathcal{T} \subseteq [s]$ of size at most t,

$$I(\{\mathbf{q}_j\}_{j \in \mathcal{T}}; \phi) = 0,$$

where I denotes mutual information. The *PIR rate* of the system is $f / \sum_{i \in [s]} |\mathbf{a}_i|$, i.e., the ratio between the size of the desired data and the amount of downloaded one, both measured in $\mathbb{F}_q$ symbols.[3]

In a graph-based replication system every file is replicated multiple times and each one of the copies is stored on a different server. If all files are replicated an identical number of times r, we say that it is an r-replication system, and r is its replication factor. In a 2-replication system a graph structure arises, in which nodes represent servers, edges represent files, and an edge is incident with a node if the respective file is stored on the respective server. Similarly, in r-replication systems for $r > 2$ an r-uniform hypergraph[4] structure arises, and in systems where every file is replicated a different number of times, a non-uniform hypergraph arises. Notice that for $r = 2$ a *multigraph* might arise, i.e., a graph in which two nodes can have multiple edges connecting them, in cases where there exist two servers that share more than one file in common. While our analysis does not exclude these cases, they result in poor collusion resistance (see Subsection II-A). Therefore, we restrict our attention to systems in which every two servers store at most one file in common (see Remark 8 for further discussion).

Graphs are denoted by $G = (E, V)$, where $E = \{e_1, e_2, \ldots\}$ and $V = \{v_1, v_2, \ldots\}$ are the sets of edges and vertices, respectively. Unless otherwise stated, all graphs in this paper are undirected, and hence, an edge is a subset of vertices (subset of size two in graphs, and of arbitrary size in hypergraphs). For a given graph G' we denote its set of edges by $E(G')$ and its set of vertices by $V(G')$. Since graphs represent storage systems in this paper, the terms *node*, *vertex*, and *server* are used interchangeably, and so does the terms *edge* and *file*.

For a graph G and a subset $\mathcal{S} \subseteq V(G)$ we denote by $G_\mathcal{S}$ the subgraph induced by $\mathcal{S}$, i.e., the graph which consists of

[3] As mentioned in Subsection II-A, PIR rate is normally defined as the ratio between the size of the file and the *expected* number of downloaded symbols, since those might depend on the randomness of the user. However, in all of our schemes the number of downloaded elements is not random, and hence we restrict our attention to this definition for the sake of simplicity.

[4] That is, a hypergraph in which all edges contain an identical number of nodes.

the nodes in $\mathcal{S}$ and all the edges in $E(G)$ such that both of their incident nodes are in $\mathcal{S}$. A *cycle* in G is a subgraph of G whose nodes are $\{v_i\}_{i=0}^{t-1}$ for some t, and whose edges are $\{v_i, v_{i+1 \bmod t}\}_{i=0}^{t-1}$, and these edges exist also in $E(G)$. An edge e is said to be *incident* with a vertex v, and vice versa, if $v \in e$. The set of edges in $E(G)$ that are incident with v are denoted by $\Gamma_G(v)$, where G is omitted if clear from context. The *incidence matrix* $I(G)$ of a graph G is a $|V(G)| \times |E(G)|$ zero-one matrix in which rows correspond to nodes and columns correspond to edges, and an entry contains 1 if and only if the respective vertex is incident with the respective edge. In the sequel, the well-known *Breadth First Search* (BFS) algorithm is used repeatedly, in graphs as well as in hypergraphs, and the uninformed reader is referred to [9].

In all subsequent protocols, the queries $\mathbf{q}_1, \ldots, \mathbf{q}_s$ are vectors in $\mathbb{F}_q^n$, i.e., they contain a field element for every file. However, since the servers contain only a portion of the files in the system, the user communicates only their support to the servers. We denote by Q the $s \times n$ matrix whose i'th row is $\mathbf{q}_i$ for every $i \in [s]$, and note that it is a random variable that depends on ϕ, and on the randomness at the user.

Since submatrices are used repeatedly, we define the following notation. For a matrix $A \in \mathbb{F}^{s \times n}$ and sets $\mathcal{S} \subseteq [s]$ and $\mathcal{N} \subseteq [n]$, let $A_{\mathcal{S},\mathcal{N}}$ be the submatrix of A that consists of the rows in $\mathcal{S}$ and the columns in $\mathcal{N}$. Further, let $A_{:,\mathcal{N}} \triangleq A_{[s],\mathcal{N}}$ and $A_{\mathcal{S},:} \triangleq A_{\mathcal{S},[n]}$. For vectors $\mathbf{a} \in \mathbb{F}_q^n$ and $\mathbf{b} \in \mathbb{F}_q^s$ we define $\mathbf{a}_\mathcal{N}$ and $\mathbf{b}_\mathcal{S}$ analogously. For convenience, we consider the rows and columns of a matrix $A_{\mathcal{S},\mathcal{N}}$ as indexed by $\mathcal{S}$ and $\mathcal{N}$, respectively, rather than by $[|\mathcal{S}|]$ and $[|\mathcal{N}|]$. For example, if $n = s = 4$ and $\mathcal{S} = \mathcal{N} = \{2, 3\}$, then $A_{\mathcal{S},\mathcal{N}}$ is a 2×2 matrix whose entries are indexed by $(2,2), (2,3), (3,2), (3,3)$. Since submatrices of Q are in strong correspondence with subgraphs of G, for every subgraph T of G (denoted $T \subseteq G$) we denote $Q^T \triangleq Q_{V(T),E(T)}$, and similarly, for every vector $\mathbf{v} \in \mathbb{F}_q^s$ we define $\mathbf{v}^T \triangleq \mathbf{v}_{V(T)}$.

By and large, we use lower-case letters $(a, b, c, \ldots)$ to denote scalars, boldface letters $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots)$ to denote vectors (all of which are row vectors), capital letters $(A, B, C, \ldots)$ to denote matrices or graphs, and calligraphic letters $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots)$ to denote sets. Finally, we use the standard notation $[N, K]_q$ to denote a linear code of length N and dimension K over $\mathbb{F}_q$.

III. Replication Factor Two

A. A PIR Protocol for 2-Replication Systems

The following scheme applies for $r = 2$ and any field $\mathbb{F}_q$ with at least three elements. Upon requiring file $\mathbf{x}_\phi$, the user randomly chooses a vector $\boldsymbol{\alpha} = (\alpha_i)_{i=1}^n \in (\mathbb{F}_q^*)^n$, a vector $\boldsymbol{\gamma} = (\gamma_i)_{i=1}^s \in (\mathbb{F}_q^*)^s$, and an element $h \in \mathbb{F}_q \setminus \{0, 1\}$, all uniformly at random, and defines

$$Q \triangleq \operatorname{diag}(\boldsymbol{\gamma}) \cdot I_\phi \cdot \operatorname{diag}(\boldsymbol{\alpha}),$$

where I_ϕ is obtained from the incidence matrix $I(G)$ by replacing the lower 1-entry in each column with -1, and then replacing the 1-entry in column ϕ by h.

Let $\mathbf{q}_j$, the query for server j, be the j-th row of Q. Clearly, to upload this row we only need to send the values of its nonzero entries, and hence the total upload complexity is $2n$. Each node responds with $\mathbf{a}_j = \mathbf{q}_j \cdot X$, and therefore the download complexity is sf, and the PIR rate is $1/s$. Note that node j can calculate the inner product since the support of $\mathbf{q}_j$ contains only the indices of the files available to it. Upon receiving the information from all s servers, the user has access to $QX = \operatorname{diag}(\boldsymbol{\gamma}) I_\phi \operatorname{diag}(\boldsymbol{\alpha}) X$. Then, by multiplying from the left by the matrix $\operatorname{diag}(\boldsymbol{\gamma})^{-1}$ and by the all ones vector $\mathbb{1}$, the user get

$$\begin{aligned}\mathbb{1} \cdot \operatorname{diag}(\boldsymbol{\gamma})^{-1} \operatorname{diag}(\boldsymbol{\gamma}) I_\phi \operatorname{diag}(\boldsymbol{\alpha}) X &= \mathbb{1} \cdot I_\phi \operatorname{diag}(\boldsymbol{\alpha}) X \\ &= (h-1)\alpha_\phi \mathbf{x}_\phi,\end{aligned}$$

and hence $\mathbf{x}_\phi$ can be recovered. We proceed with studying the collusion resistance of the suggested scheme. The following claim is a special case of a more general one that is given in the sequel (Theorem 5). Nevertheless, it is given here in its current form to maintain simplicity and flow, and its proof is sketched.

Proposition 2. *For any set of servers $\mathcal{S} \subseteq V$ such that $G_\mathcal{S}$ does not contain a cycle, we have that $I(\{\boldsymbol{q}_i\}_{i \in \mathcal{S}}; \phi) = 0$.*

Proof sketch. To prove the claim, we analyze the submatrix of queries that is seen by $\mathcal{S}$. For clarity, we omit zero columns from this matrix, as well as columns of weight one, since the latter ones are obviously purely random, and cannot cause leakage of information. Hence, the matrix we analyze is chosen according to the random variable $Q^{G_\mathcal{S}}$.

It is evident that every matrix which is chosen according to $Q^{G_\mathcal{S}}$ has support which is identical to that of $I(G)^{G_\mathcal{S}}$. In what follows we explain why *every* $|V(G_\mathcal{S})| \times |E(G_\mathcal{S})|$ matrix M whose support is identical to that of $I(G)^{G_\mathcal{S}}$ can be obtained by some choice of $\boldsymbol{\gamma}, \boldsymbol{\alpha}$, and h with identical probability, regardless of the value of ϕ. Consequently, this proves that no information regarding ϕ is leaked.

We calculate $\Pr(Q^{G_\mathcal{S}} = M)$ by an iterative process that follows a Breadth First Search (BFS) traversal on $G_\mathcal{S}$. Pick an arbitrary $v_i \in \mathcal{S}$, and fix the value of the corresponding γ_i (with probability one). Clearly, it follows that $\Pr(\gamma_i \cdot \alpha_j \cdot (I_\phi)_{i,j} = M_{i,j}) = (q-1)^{-1}$ for every $e_j \in \Gamma_{G_\mathcal{S}}(v_i)$ regardless of whether or not $(I_\phi)_{i,j}$ is the entry of I_ϕ which is multiplied by h. Having the values of α_j for every $e_j \in \Gamma_{G_\mathcal{S}}(v_i)$ fixed, we have that $\Pr(\gamma_{j'} \cdot \alpha_j \cdot (I_\phi)_{j',j} = M_{j',j}) = (q-1)^{-1}$ for the same reasons, where $v_{j'}$ is the other end of edge e_j (again, regardless of whether or not $(I_\phi)_{j',j}$ is the entry of I_ϕ which is multiplied by h). In other words, we have that fixing an entry in $\boldsymbol{\gamma}$ which corresponds to some $v \in V(G_\mathcal{S})$ compels us to fix the values in $\boldsymbol{\alpha}$ which correspond to all of $\Gamma_{G_\mathcal{S}}(v)$. In turn, fixing these entries of $\boldsymbol{\alpha}$ compels us to fix the values of $\boldsymbol{\gamma}$ at the other endpoints of the edges in $\Gamma_{G_\mathcal{S}}(v)$. Since $G_\mathcal{S}$ does not contain a cycle, we may proceed in a BFS fashion and have that every edge-node incidence in $G_\mathcal{S}$ reduces the overall probability of obtaining M by $(q-1)^{-1}$. Hence, every such matrix M is obtained with probability $(q-1)^{-|M|}$, where $|M|$ is the size of the support of M, and regardless of the value of ϕ. Hence, perfect privacy is guaranteed. □

It readily follows from Proposition 2 that given system parameters $(s, n, f, r = 2, t)$, it suffices to find an underlying

graph with s nodes n edges and girth (the size of smallest cycle) at least $t+1$; this is a fundamental challenge in graph theory (see Example 7 below). We now turn to study how gracefully the perfect privacy deteriorates if $\mathcal{S}$ contains one or more cycles, i.e., how much of ϕ's identity is revealed.

Proposition 3. *For any cycle $C = (V', E')$ in G, any matrix M in the support of the random variable Q^C is invertible if and only if $e_\phi \in E'$.*

Proof. Let $A \triangleq \operatorname{diag}(\boldsymbol{\gamma}_{V'})^{-1} M \operatorname{diag}(\boldsymbol{\alpha}_{E'})^{-1}$, and observe that $\operatorname{rank}(A) = \operatorname{rank}(M)$. If $\phi \notin E'$, then each column of A has two nonzero entries 1 and -1. Hence, $\mathbb{1}$ is in its left kernel, and thus $\operatorname{rank}(A) < c$, where $c \triangleq |V'| = |E'|$. Moreover, it is an easy exercise to show that any set of $c-1$ columns of A are linearly independent, and hence $\operatorname{rank}(A) = c-1$.

On the other hand if $\phi \in E'$, assume without loss of generality that A is of the form

$$A = \begin{pmatrix} * & & & & h \\ * & * & & & \\ & * & \ddots & & \\ & & \ddots & * & \\ & & & * & -1 \end{pmatrix},$$

where $*$ denotes a nonzero entry. Then, $\det A = (-1)^{c-1} h \cdot \det A_1 - \det A_2$, where A_1 (resp. A_2) is the bottom-left (resp. top-left) $(c-1) \times (c-1)$ submatrix of A. Notice that $\det A_1$ is the product of all $*$-entries in the sub-diagonal of A, and that $\det A_2$ is product of all $*$-entries in the main diagonal of A. Hence, since every pair of $*$-entries in any given column are negations of one another, it follows that $\det A_1 = (-1)^{c-1} \det A_2$. Thus, $\det A = (-1)^{2c-2} h \cdot \det A_2 - \det A_2 = (h-1) \det A_2 \neq 0$. □

Corollary 4. *A set $\mathcal{S} \subseteq V$ can narrow down the possible values of e_ϕ (and hence, of ϕ itself) to*

$$\mathcal{T} = \mathcal{T}(\mathcal{S}, \phi) \triangleq \left(\bigcap_{k=1}^{\ell} E(C_k) \right) \setminus \left(\bigcup_{k=1}^{\ell'} E(C'_k) \right), \quad (1)$$

where $C_1, \ldots, C_\ell$ are all cycles in $G_\mathcal{S}$ that contain[5] e_ϕ, and $C'_1, \ldots, C'_{\ell'}$ are all cycles in $G_\mathcal{S}$ that do not contain e_ϕ.

Proof. Let M be the matrix that is seen by $\mathcal{S}$; chosen according to the random variable $Q^{G_\mathcal{S}}$. By Proposition 3, the colluding servers can compute the rank of M^C for every cycle C in their induced subgraph, and deduce if $e_\phi \in E(C)$ accordingly. □

We now show that Corollary 4 is in some sense the best that the colluding servers can hope for. Formally, we show that conditioned by $e_\phi \in \mathcal{T}$, all respective possible queries are obtained with identical probability. The immediate conclusion is that out of the $\log n$ protected bits of ϕ, the information leakage if a set $\mathcal{S}$ colludes is precisely $\log n - \log |\mathcal{T}|$ (note that this expression might vary with ϕ). More formally, for every $\mathcal{S} \subseteq V$ we have

$$\begin{aligned} I(\{\mathbf{q}_j\}_{j\in\mathcal{S}}; \phi) &= H(\phi) - H(\phi | \{\mathbf{q}_j\}_{j\in\mathcal{S}}) \\ &= \log n - H(\log |\mathcal{T}(\mathcal{S}, \phi)|). \end{aligned}$$

To state the main theorem of this paper, whose proof is given in Appendix A, and of which Proposition 2 is a special case, we require the following definition. For $\mathcal{S} \subseteq V$ and $\mathcal{D} \subseteq E$, we say that a matrix in $\mathbb{F}_q^{|\mathcal{S}| \times |\mathcal{D}|}$ is $(\mathcal{S}, \mathcal{D})$-*compatible* with G ($(\mathcal{S}, \mathcal{D})$-compatible, for short) if its support coincides with that of $I(G)_{\mathcal{S},\mathcal{D}}$. This definition extends naturally to a subgraph $T \subseteq G$ where a matrix in $\mathbb{F}_q^{|V(T)| \times |E(T)|}$ is said to be T-compatible if it is $(V(T), E(T))$-compatible.

Theorem 5. *For every subgraph $T \subseteq G$, the support of the random variable $Q^T | \phi$ is the set of all matrices $A \in \mathbb{F}_q^{|V(T)| \times |E(T)|}$ such that*:

(a) *A is T-compatible with G; and*
(b) *for every cycle $C \subseteq T$,*

$$\operatorname{rank}(A^C) = \begin{cases} |E(C)| & \text{if } \phi \in E(C) \\ |E(C)| - 1 & \text{if } \phi \notin E(C) \end{cases}.$$

Furthermore, the random variable $Q^T | \phi$ is uniformly distributed on its support.

First, it is evident that the case where T is acyclic in Theorem 5 proves Proposition 2. Second, we have the following corollary.

Corollary 6. *For every set $\mathcal{S} \subseteq V$ and every two distinct values $\phi_1, \phi_2 \in [n]$ such that $\phi_2 \in \mathcal{T}(\mathcal{S}, \phi_1)$, the servers in $\mathcal{S}$ cannot infer if $\phi = \phi_1$ or $\phi = \phi_2$.*

Proof. Clearly, it suffices to prove that the random variables $Q^{G_\mathcal{S}} | (\phi = \phi_1)$ and $Q^{G_\mathcal{S}} | (\phi = \phi_2)$ are identical, i.e., the same queries are obtained with identical probabilities. Since both random variables are uniformly distributed on their support by Theorem 5, it suffices to prove that their supports are identical. Also by Theorem 5, it suffices to prove that the conditions (a) and (b) coincide in both cases. For (a) this claim is clear since it does not depend on the value of ϕ. For condition (b), we need to prove that $\phi_1 \in E(C)$ if and only if $\phi_2 \in E(C)$ for every cycle C in $G_\mathcal{S}$, which is precisely the meaning of $\phi_2 \in \mathcal{T}(\mathcal{S}, \phi_1)$. □

We now turn to present several choices of the graph G, and the resulting privacy of the PIR schemes. These examples are summarized in Table I.

Example 7.

1) *Taking G to be the Petersen graph (a 3-regular graph with 10 nodes, 15 edges, and girth 5) allows to store 15 files on 10 servers, 3 files on each, where any 4 servers cannot infer any information regarding ϕ. According to the structure of the Petersen graph, at least 8 servers are required to infer the exact identity of ϕ. The upload complexity is 30 field elements, and the download complexity is $10f$ field elements, i.e., the PIR rate is 0.1.*

[5]For $\ell = 0$ we define $\bigcap_{k=1}^{\ell} E(C_k) = E$, and for $\ell' = 0$ we define $\cup_{k=1}^{\ell'} E(C'_k) = \varnothing$.

TABLE I

DIFFERENT EXAMPLES FOR THE CHOICE OF G IN SECTION III. THE PARAMETER t STANDS FOR THE GUARANTEED t-PRIVACY OF THE SYSTEM AND d DENOTES THE FIXED DEGREE OF THE VERTICES IN THE GRAPH

	n	s	t	d	**PIR rate**
Petersen	15	10	4	3	$\frac{1}{10}$
Complete bipartite	Square	$2\sqrt{n}$	3	$\sqrt{n}$	$\frac{1}{2\sqrt{n}}$
Gen. polygons	$O(q^3)$	$O(q^2)$	5	$q+1$	$O(n^{-2/3})$
	$O(q^4)$	$O(q^3)$	7	$q+1$	$O(n^{-3/4})$
	$O(q^6)$	$O(q^5)$	11	$q+1$	$O(n^{-5/6})$
Murty	$p^{2m}(p^m+2)$	$2p^{2m}$	4	p^m+2	$O(n^{-2/3})$
Ramanujan	Any	$\frac{2n}{d}$	$O(\log n)$	Constant	$\frac{d}{2n}$

2) *Taking $G = (L \cup R, V)$ to be the complete bipartite graph, where $|L| = |R| = s/2$, allows to store $n = s^2/4$ files. To retrieve a file $\boldsymbol{x}_\phi$, the user downloads $\frac{s}{2} \cdot f$ field elements. The resulting system ensures perfect privacy against all sets $\mathcal{S} \subseteq L \cup R$ such that either $|S \cap L| \leq 1$ or $|S \cap R| \leq 1$, and in particular, all sets of size three.*
3) *Graphs of large (constant) girth g are particularly useful since all sets with at most $g-1$ nodes are cycle-free, and hence the resulting protocol is $(g-1)$-private. These can be obtained as incidence graphs of generalized polygons [21, Table I], of which Item 2 above is a special case. In particular, for prime power q, there exist explicit graphs with degree $q+1$ with $s \in \{O(q^2), O(q^3), O(q^5)\}$ (and hence $n \in \{O(q^3), O(q^4), O(q^6)\}$), where $g \in \{6, 8, 12\}$, respectively. The respective download complexities are $O(n^{2/3}) \cdot f$, $O(n^{3/4}) \cdot f$, and $O(n^{5/6}) \cdot f$.*
4) *Let $p \geq 5$ be a prime, and let m be a positive integer. The Murty graph [19] is a (p^m+2)-regular graph with $s = 2p^{2m}$ nodes, $n = p^{2m}(p^m+2)$ edges, and girth five. In the resulting system, a database of n files is stored on $O(n^{2/3})$ servers, $O(n^{1/3})$ files in each, and ensures perfect privacy against any four colluding servers. To retrieve a file, a user downloads $O(n^{2/3}) \cdot f$ field elements.*
5) *Ramanujan graphs (e.g., [18]) with n edges and constant degree have girth $O(\log n)$. Hence, the system is resilient against any $O(\log n)$ colluding servers, but require download of $\delta n f$ field elements for some $\delta \in (0, 1)$.*

Remark 8. *It is evident that the correctness of the scheme and its privacy guarantees hold also in cases where there exist two servers that store more than one file in common. However, in the resulting multigraph, these two servers form a cycle, and hence $t = 1$. However, the system designer may choose to disperse the files while ignoring the aforementioned restriction in order to increase the number of files in the system. Then, to maintain $t \geq 2$, the user will download all but one of the mutual files from any pair of servers, prior to executing the above scheme, which will induce a loss of rate.*

B. Bound on the PIR Rate

In this subsection we explore the limitations of PIR protocols for graph-based replication systems by proving a bound on the PIR rate. The resulting bound is particularly powerful for the important family of regular graphs, for which the bound is within a factor of two from the rate in Subsection III-A. In what follows, the maximum degree of a vertex in G is denoted by δ, and recall that we assume $t \geq 2$. This bound also applies in cases where servers can share more than one file in common.

Lemma 9. *For $r = 2$ the PIR rate is at most $\frac{\delta}{n}$.*

Proof. Let G be the induced graph, and let $\mu_i \geq 0$ be the amount of downloaded information (measured in fraction of f) from server i by the user in any execution of the algorithm. Clearly, it must be that[6] $\mu_i + \mu_j \geq 1$ for every edge $\{i, j\} \in E(G)$, since otherwise, servers i and j can infer that their mutual file (or files) is not required by the user, and hence the system is not 2-private. Further, the PIR rate of the system is $(\mathbb{1}_s \cdot \boldsymbol{\mu}^\top)^{-1}$, where $\mathbb{1}_s$ is the all 1's vector of length s and $\boldsymbol{\mu} \triangleq (\mu_1, \ldots, \mu_s)$. Hence, an upper bound on the PIR rate of the system is obtained from the optimal solution of the following linear program.

$$\min \ \mathbb{1}_s \cdot \boldsymbol{\mu}^\top, \text{ subject to } I(G)^\top \boldsymbol{\mu}^\top \geq \mathbb{1}_n \text{ and } \boldsymbol{\mu} \geq 0, \quad (2)$$

That is, the inverse of the optimum value of the objective function serves as an upper bound on the PIR rate of the system. In the following problem, which is called the *dual* of (2), $\boldsymbol{\eta}$ is a vector of n variables.

$$\max \ \mathbb{1}_n \cdot \boldsymbol{\eta}^\top, \text{ subject to } I(G)\boldsymbol{\eta}^\top \leq \mathbb{1}_s \text{ and } \boldsymbol{\eta} \geq 0. \quad (3)$$

According to the primal-dual theory [9, Sec. 29.4], any solution which is feasible for (3) provides a lower bound for (2). It is readily verified that $\boldsymbol{\eta} = \frac{1}{\delta} \cdot \mathbb{1}_n$ is a feasible solution for (3), and the objective function for this solution equals n/δ. Therefore, the PIR rate is bounded by δ/n. □

In cases where G is a regular graph, which are particularly interesting since they induce systems with balanced storage, the resulting bound equals $\frac{\delta}{n} = \frac{2\delta}{s\delta} = 2/s$. In some cases, it is possible to improve the bound δ/n for graphs which are not regular, and the details are given in Appendix C.

IV. ARBITRARY REPLICATION FACTORS

In this section we consider r-replication systems for $r \geq 2$, which are favored in practice due to their greater resilience to

[6]More broadly, $\mu_i + \mu_j$ must at least the number of mutual files of server i and server j.

simultaneous failures [7], [14]. First, for any integer $r \geq 2$, collusion resistance of $r-1$ can be attained by a simple scheme that is given below in Subsection IV-A. Then, we provide another scheme in Subsection IV-B, which guarantees larger collusion resistance by a reduction to the 2-replication case. The collusion resistance in the latter case will strongly depend on our ability to increase the girth by removing edges from a certain multigraph. To simplify the discussion, in this section we alleviate the requirement that every two servers share at most one file in common.

A. Replication Factor r and Collusion Resistance $r-1$

The user begins by choosing a uniformly random matrix $V \in \mathbb{F}_q^{r\times n}$, whose rows sum to $\mathbf{e}_\phi$, the ϕ'th unit vector of length n. Then, the user disperses the nr symbols of the matrix V to the queries $\{\mathbf{q}_i\}_{i=1}^s$ arbitrarily,[7] such that every server that stores a file $\mathbf{x}_j$ receives a unique entry from the j'th column of V. Namely, the r servers which store the r copies of a file $\mathbf{x}_j$ receive the symbols of the j'th column of V, one symbol per server. In turn, the servers respond with the respective linear combinations $\{\mathbf{a}_j = \mathbf{q}_j \cdot X\}_{j=1}^s$, and the user computes $\sum_{i=1}^s \mathbf{a}_i = \mathbf{e}_\phi \cdot X = \mathbf{x}_\phi$.

It is readily verified that the PIR rate is $1/s$, and that every set of $r-1$ servers can observe at most $r-1$ entries in every column of V. Hence, since these entries appear entirely random, the resulting scheme is $r-1$ private. Notice that there is no restriction on the number of files that can be stored in this system, nor there is a restriction on their dispersion.

B. Arbitrary Replication Factor by Reduction

In systems where files might be stored in more than two servers, one can obtain perfect privacy by "ignoring" all but two copies of every file that is replicated more than twice, in a sense that will be made clear shortly, and applying the scheme in Section III. Observe that choosing which copies to ignore may drastically affect the collusion resistance of the system, since each choice produces a different graph with different cycles. Nevertheless, this observation can in fact contribute to the security of the system by concealing the cycle structure of the resulting graph from an adversary. In what follows we formalize these intuitions and discuss the different aspects of the reduction to the 2-replication scheme.

Evidently, it is natural to consider an r-replication system for $r \geq 2$ (or in fact, any replication system) as a hypergraph, where each file corresponds to a hyperedge. Yet, for our purpose it is often more convenient to consider it as a *colored multigraph*. That is, instead of considering every file as a hyperedge, which is incident with the nodes that contain it, we consider a multigraph in which every edge carries a label (or a color) in $[n]$. Then, two servers are connected by an edge with label $i \in [n]$ if both of them contain a copy of $\mathbf{x}_i$. Clearly, given a hypergraph G, one can easily create the respective colored multigraph $\hat{G}$ by replacing hyperedge i with a clique whose edges are labelled by i. Notice that $\hat{G}$ can be a multigraph (i.e., contain parallel edges) since hyperedges

[7]This is possible since $\sum_{i=1}^s |\mathbf{q}_i| = \sum_{i=1}^s |\Gamma(i)| = rn$, where $|\mathbf{q}_i|$ is the length of $\mathbf{q}_i$.

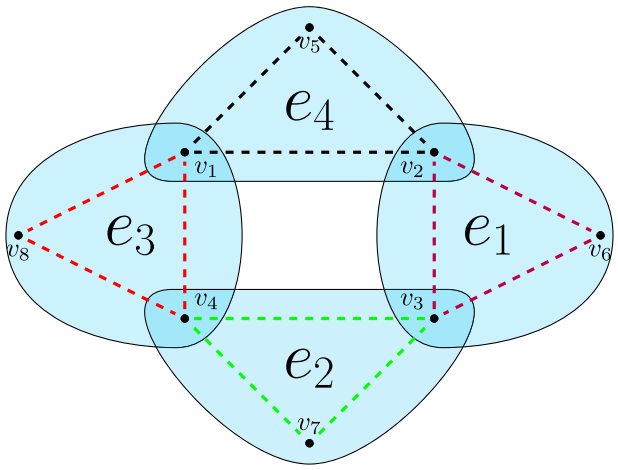

Fig. 1. A hypergraph G (in light blue) and its respective colored multigraph $\hat{G}$ (in dashed lines). The vertices $\{v_1, v_2, v_5\}$ contain a monochromatic cycle, but not a polychromatic one. The vertices $\{v_1, v_4, v_3, v_2, v_5\}$ contain a monochromatic cycle, and a polychromatic one.

can intersect in more than one node. An illustration of these definitions is given in Figure 1, which also demonstrates the natural notions of a *monochromatic* and *polychromatic* cycles, that will be useful in the sequel. In what follows we use G and $\hat{G}$ interchangeably.

Given a replication system with a respective multigraph $\hat{G}$, it is obvious that the user can choose any two copies of every file, and apply the scheme from Section III while ignoring the remaining copies. Formally, for a server i that stores a copy of $\mathbf{x}_j$ that is chosen to be ignored by the user, the user simply transmits a zero coefficient for $\mathbf{x}_j$, or omits that coefficient altogether. Further, the operation of ignoring all but two copies of every file corresponds to removing all but one of the edges of every color. Obviously, there are potentially many options to choose which edge to keep for every label, and every such choice can be described by a function $c : [n] \to E(\hat{G})$ such that the edge $c(i)$ is labelled by i, for every $i \in [n]$. For any such c, let $\hat{G}_c$ be the result of keeping the edges $\{c(i)\}_{i\in[n]}$, and removing the remaining ones. It is readily verified that the resulting scheme guarantees perfect privacy against colluding sets that do not contain a cycle in $\hat{G}_c$.

Clearly, if one can choose the file dispersion in the system as one pleases, then it is possible to first choose the dispersion of only two copies of each file, so that the resulting graph G' has a certain girth. Then, the remaining copies can be dispersed arbitrarily, and the PIR scheme is performed with respect to the function c that $c(i) \in E(G')$ for every i. However, if $\hat{G}$ is *given* to the user, finding a function c such that $\hat{G}_c$ has a large girth requires more care.

For a given $\hat{G}$ one can choose c at random. In spite of not having any clear minimum girth guarantee, this approach has the extra benefit of concealing the cycle structure from an adversary. For a given integer g, a function c such that $\hat{G}_c$ has girth g, if exists, can be found be deciding the feasibility of the following $\{0,1\}$-program. In this program, for $i \in [n]$ let E_i be the set of all 2-subsets $\{a, b\}$ of $[s]$ such that there exists an edge $\{a, b\}$ labelled by i.

- **Objective**: None.
- **Variables**: $\{x_{i,\{a,b\}} \mid i \in [n] \text{ and } \{a,b\} \in E_i\}$.
- **Constraints**:
  - $\sum_{\{a,b\}\in E_i} x_{i,\{a,b\}} = 1$ for all $i \in [n]$.
  - $\sum_{i|\{a,b\}\in E_i} x_{i,\{a,b\}} \leq 1$ for every $\{a,b\}$ such that there exists at least one edge $\{a,b\}$ in $\hat{G}$.
  - $\sum_{i|\{a,b\}\in E_i} x_{i,\{a,b\}} + \sum_{i|\{b,c\}\in E_i} x_{i,\{b,c\}} + \sum_{i|\{c,a\}\in E_i} x_{i,\{c,a\}} \leq 2$, for every $a,b,c \in [s]$ that contain at least one triangle in $\hat{G}$.
  - $\vdots$
  - $\sum_{j=1}^{g} \sum_{i|\{a_j,a_{(j+1) \bmod g}\in E_i\}} x_{i,\{a_j,a_{(j+1) \bmod g}\}} \leq g-1$ for every $a_0,\ldots,a_{g-1} \in [s]$ that contain at least one g-cycle in $\hat{G}$.

Clearly, the first set of constraints guarantees that exactly one edge is chosen for every file $i \in [n]$. The second set of constraints guarantees that the resulting choice does not contain 2-cycles, the next set guarantees that there are no triangles, and so on. Finally, we note that while solving this system for a general g is NP-hard, the special case $g = 2$ reduces to finding a maximum matching in a bipartite graph, a problem that can be solved efficiently.

V. Graph-Based Coding–Reducing the Storage Overhead at Improved PIR Rates

This section discusses storage systems in which every file is similarly stored on a small number of servers, but replication is generalized to arbitrary encoding. Hence, when employing an $[N,K]_q$ code with rate larger than $1/2$ (i.e., $K/N > 1/2$), we obtain an improvement over previous schemes in terms of storage overhead. Furthermore, it is shown that the resulting PIR rate is improved whenever $N - K > 1$. However, the (coded) file dispersion must follow a certain structure, and the resulting collusion patterns are in correspondence with *polychromatic cycles* (see Subsection IV-B and Figure 1), as will be explained next. Finally, we note that the scheme in this section is loosely inspired by ideas from [13] and [17].

Essentially, in the scheme of Section III, every file $\mathbf{x}_i$ is coded by using a repetition code of length 2 over the alphabet $\mathbb{F}_q^f$. Then, every symbol of the resulting codeword is stored on a different server. The scheme which is presented in this section generalizes this concept by employing codes other than the repetition code.

For integers N and K let $G \in \mathbb{F}_q^{K\times N}$ be a generator matrix of an $[N,K]_q$ MDS code $\mathcal{D}$. Consider every file $\mathbf{x}_i$ as an $(f/K) \times K$ matrix $(\mathbf{x}_{i,1}^\top,\ldots,\mathbf{x}_{i,K}^\top)$ over $\mathbb{F}_q$, and let $(\mathbf{x}_{i,1}^\top,\ldots,\mathbf{x}_{i,K}^\top)\cdot G \triangleq (\mathbf{y}_{i,1}^\top,\ldots,\mathbf{y}_{i,N}^\top)$, where the vectors $\{\mathbf{y}_{i,j}\}_{j=1}^N$ are called the *codeword symbols* of $\mathbf{x}_i$. Let $\mathcal{L}_1,\ldots,\mathcal{L}_N \subseteq [s]$ be disjoint nonempty subsets whose union is $[s]$ (and hence we must have $N \leq s$). Then, for every $i \in [n]$, disperse the N codeword symbols $\mathbf{y}_{i,1},\ldots,\mathbf{y}_{i,N}$ to the servers such that for every $j \in [N]$, the codeword symbol $\mathbf{y}_{i,j}$ is in exactly one server which belongs to $\mathcal{L}_j$. For example, one can think of a system in which the servers are partitioned to three disjoint subsets; the servers in the first subset contain the first halves of all files, the servers in the second contain the other half, and the servers in the third contain the sums of the two halves (see Example 12 and Example 13 which follow).

The above coding scheme gives rise to an N-uniform N-partite hypergraph in the following manner. Let $[s]$ be the set of vertices, and define hyperedges $e_1,\ldots,e_n$, such that e_i contains all servers that store either one of $\mathbf{y}_{i,1},\ldots,\mathbf{y}_{i,N}$. It is evident that the edges are of size N, and that the N parts of the hypergraph are the sets $\mathcal{L}_1,\ldots,\mathcal{L}_N$. Let G be this hypergraph, and let $\hat{G}$ be its respective colored multigraph, as described in Subsection IV-B.

We begin by presenting the PIR protocol for the special case $N - K = K$, and later extend it to other parameters by operating in *rounds*. Begin by choosing $\boldsymbol{\alpha} \in (\mathbb{F}_q^*)^n, \boldsymbol{\gamma} \in (\mathbb{F}_q^*)^s$, and $h \in \mathbb{F}_q \setminus \{0,1\}$ uniformly at random, and pick an arbitrary subset $\mathcal{K} \subseteq [N]$ of size K. Then, for every $m \in [N]$, a server $j \in [s]$ which belongs to $\mathcal{L}_m$ receives the following query.

$$(\mathbf{q}_j)_t = \begin{cases} \gamma_j \cdot \alpha_t \cdot h^{\delta(t,m)} & \text{if } j \text{ contains a codeword symbol of } \mathbf{x}_t \\ 0 & \text{else} \end{cases}, \quad (4)$$

where $\delta(t,m)$ is a Boolean indicator for the event "$m \in \mathcal{K}$ and $t = \phi$". Namely, the user transmits to server j the part of the vector $\gamma_j \cdot \boldsymbol{\alpha}$ that is relevant to it, where arbitrary K servers that store a codeword symbol of $\mathbf{x}_\phi$ are having the ϕ'th entry of $\gamma_j \cdot \boldsymbol{\alpha}$ multiplied by h. In turn, a server j in $\mathcal{L}_m$, which stores $\{\mathbf{y}_{\ell,m} | \ell \in \mathcal{L}\}$ for some $\mathcal{L} \subseteq [n]$, responds with $\mathbf{a}_j \triangleq \sum_{\ell\in\mathcal{L}} (\mathbf{q}_j)_\ell \cdot \mathbf{y}_{\ell,m}$. Having the responses $\{\mathbf{a}_i\}_{i=1}^s$, the user composes the following matrix.

$$\left(\sum_{j\in\mathcal{L}_1} \gamma_j^{-1}\mathbf{a}_j^\top,\ldots,\sum_{j\in\mathcal{L}_N} \gamma_j^{-1}\mathbf{a}_j^\top\right) = \underbrace{\sum_{j=1}^{n} \alpha_j(\mathbf{y}_{j,1}^\top,\ldots,\mathbf{y}_{j,N}^\top)}_{\triangleq Y} + \mathbf{e},$$

where for $m \in [N]$, the m'th column of $\mathbf{e}$ is

$$(\mathbf{e})_m = \begin{cases} \alpha_\phi(h-1)\mathbf{y}_{\phi,m} & \text{if } m \in \mathcal{K} \\ 0 & \text{else} \end{cases}.$$

Now, it is evident that every row in the matrix Y is a codeword in $\mathcal{D}$, whose minimum distance is $N-K+1$. Therefore, since $\mathbf{e}$ has at most K nonzero columns, and since $K = N - K$, a decoding algorithm[8] for $\mathcal{D}$ can extract $\mathbf{e}$ from the matrix that was composed by the user. At this point the user has obtained $\{\mathbf{y}_{\phi,m}\}_{m\in\mathcal{K}}$, that are sufficiently many codeword symbols of $\mathbf{x}_\phi$ in order to retrieve it. Therefore, the PIR rate of this scheme is $\frac{f}{s\cdot(f/K)} = \frac{K}{s} = \frac{N-K}{s}$. The proof of privacy will be given after the general description.

Notice that in the above scheme, $N-K$ codeword symbols of $\mathbf{x}_\phi$ are obtained, while K many of those are sufficient to retrieve $\mathbf{x}_\phi$. However, in cases where $N-K < K$, the scheme will not be successful, and in cases where $N - K > K$, the resulting scheme will not be exploited to its full potential.

Therefore, to address cases in which $K \neq N - K$, we retrieve *multiple* files in *rounds*, a standard practice in the

[8]Notice that the "error values" are in prescribed positions, and hence, an *erasure correction* algorithm suffices.

PIR literature (e.g., [13], [17]). That is, we assume that the user wishes to download $\mathbf{x}_{\phi_1}, \ldots, \mathbf{x}_{\phi_b}$ privately for some $b \geq 1$, and the protocol operates in $\ell \geq 1$ rounds. In each round, the user sends a query to every server, and receives responses from all servers. Specifically, we choose b and ℓ so that $Kb = \ell(N-K)$, i.e., $\ell \triangleq \frac{LCM(K,N-K)}{N-K}$ and $b \triangleq \frac{LCM(K,N-K)}{K}$. Prior to executing these rounds, the user fixes the following subsets of $[N]$

$$\begin{aligned} J^{(1)} &= J^{(1,1)} \cup J^{(1,2)} \cup \ldots \cup J^{(1,b)} \\ J^{(2)} &= J^{(2,1)} \cup J^{(2,2)} \cup \ldots \cup J^{(2,b)} \\ &\vdots \\ J^{(\ell)} &= J^{(\ell,1)} \cup J^{(\ell,2)} \cup \ldots \cup J^{(\ell,b)}, \end{aligned} \tag{5}$$

such that in every row, the sets in the union are pairwise disjoint, such that $|J^{(i)}| = N-K$ for every $i \in [\ell]$, and such that $|\cup_{i=1}^{s} J^{(i,j)}| = K$ for every $j \in [b]$. Intuitively, for $j \in [b]$ and $i \in [\ell]$, the set $J^{(j,i)}$ contains the indices of the codeword symbols of $\mathbf{x}_{\phi_j}$ that are retrieved during round i. The choice of such sets is easy, and is illustrated in Appendix B.

In each round i the user executes the aforementioned protocol (for the case $K = N-K$), where $J^{(i)}$ is used in lieu of the set $\mathcal{K}$. That is, the queries are defined as in (4), with the difference that $\delta(t,m)$ is a Boolean indicator for the event "there exists $j \in [b]$ such that $t = \phi_j$ and $m \in J^{(i,j)}$". Having obtained the responses from all servers in round i, the user computes

$$\left(\sum_{j \in \mathcal{L}_1} \gamma_j^{-1} \mathbf{a}_j^\top, \ldots, \sum_{j \in \mathcal{L}_N} \gamma_j^{-1} \mathbf{a}_j^\top \right) = \underbrace{\sum_{j=1}^{n} \alpha_j (\mathbf{y}_{j,1}^\top, \ldots, \mathbf{y}_{j,N}^\top)}_{\triangleq Y} + \mathbf{e}',$$

where for $m \in [N]$, the m'th column of $\mathbf{e}'$ is

$$(\mathbf{e}')_m = \begin{cases} \alpha_{\phi_j}(h-1)\mathbf{y}_{\phi_j,m} & \text{if } m \in J^{(i,j)} \\ 0 & \text{else} \end{cases}.$$

Since $|J^{(i)}| = N-K$, a decoding algorithm on the matrix Y can extract the values of $\mathbf{e}'$. Hence, according to the structures of the sets in (5), it follows that by the end of the ℓ'th round, the user has obtained the K codeword symbols $\{\mathbf{y}_{\phi_j,m}\}_{m \in \cup_i J^{(i,j)}}$ of $\mathbf{x}_{\phi_j}$ for every $j \in [b]$, and hence all the files $\{\mathbf{x}_{\phi_j}\}_{j=1}^{b}$ can be retrieved. The resulting PIR rate is

$$\frac{bf}{s \cdot (f/K) \cdot \ell} = b \cdot \frac{K}{s\ell} = \frac{\ell(N-K)}{K} \cdot \frac{K}{s\ell} = \frac{N-K}{s}.$$

Remark 10. *Roughly speaking, the scheme which is described in Section III is as a special case of the one in this section, where $K = 1$, $N = 2$, and $\mathcal{D} \triangleq \{(x,-x)|x \in \mathbb{F}_q\}$, and the resulting rate is indeed $\frac{N-K}{s} = \frac{1}{s}$. However, further simplification is possible for this particular choice of $\mathcal{D}$, since the process of extracting the error vector $\boldsymbol{e}$ reduces to multiplying by $\mathbb{1}$ from the left. Hence, the partitioning of the servers to subsets $\{\mathcal{L}_j\}_{j=1}^{N}$ is not required.*

Proposition 11. *A set $\mathcal{S} \subseteq V$ that contains no polychromatic cycles in $\hat{G}$ gains no information about $\phi_1, \ldots, \phi_b$.*

Proof. For $\mathcal{S}$ that does not contain a polychromatic cycle, let $\mathcal{R} \subseteq [n]$ be the set of hyperedges in G that have two or more vertices in $\mathcal{S}$. Similar to Proposition 2, we analyze the matrix which is chosen according to the random variable $Q_{\mathcal{S},\mathcal{R}}$. Clearly, every matrix which is chosen according to $Q_{\mathcal{S},\mathcal{R}}$ is $(\mathcal{S},\mathcal{R})$-compatible with G, and we show that the inverse is also true.

Let $M \in \mathbb{F}_q^{|\mathcal{S}|\times|\mathcal{R}|}$ be a matrix which is $(\mathcal{S},\mathcal{R})$-compatible with G. Fix some $v_i \in \mathcal{S}$ as the starting point of the BFS algorithm, and choose an arbitrary value for γ_i (with probability 1). Once γ_i is fixed, it is evident that $\Pr(\gamma_i \cdot \alpha_j \cdot h^\delta = M_{i,j}) = (q-1)^{-1}$ for every hyperedge e_j that is incident with v_i regardless of the value of the Boolean indicator δ.

Notice that the only mutual element of these hyperedges is v_i, since otherwise, a polychromatic cycle of length two would exist in $\hat{G}$. Therefore, once α_j is fixed for such a hyperedge e_j, we have that $\Pr(\gamma_\ell \cdot \alpha_j \cdot h^\delta = M_{\ell,j}) = (q-1)^{-1}$ for every ℓ such that $v_\ell \in e_j \cap \mathcal{R}$, again, regardless of δ. Proceeding in a BFS fashion, we have that each node-hyperedge incidence reduces the overall probability of obtaining M by a multiplicative factor of $(q-1)^{-1}$. Since $\mathcal{S}$ does not contain a polychromatic cycle, no discrepancy is encountered, which concludes the proof. □

Example 12. *Consider $s = 12$, and let $\mathcal{D}$ be the parity code $\{(x,y,x+y)|x,y \in \mathbb{F}_q\}$, and hence $N = 3$ and $K = 2$. Also, let $\mathcal{L}_1 = \{1,\ldots,4\}$, $\mathcal{L}_2 = \{5,\ldots,8\}$, and $\mathcal{L}_3 = \{9,\ldots,12\}$. Consider the following 16 hyperedges.*

$$\begin{array}{cccc} \{1,5,9\} & \{2,5,10\} & \{3,5,11\} & \{4,5,12\} \\ \{1,6,10\} & \{2,6,11\} & \{3,6,12\} & \{4,6,9\} \\ \{1,7,11\} & \{2,7,12\} & \{3,7,9\} & \{4,7,10\} \\ \{1,8,12\} & \{2,8,9\} & \{3,8,10\} & \{4,8,11\} \end{array}$$

It is readily verified that every two distinct edges intersect in at most one node, and hence, there are no polychromatic cycles of length 2. The resulting system is 2-private, has storage overhead 1.5, and its PIR rate is 1/12.

Example 13. *Generalizing the previous example, let s be any integer divisible by 3, let $\mathcal{D}$ be the parity code, and let $\mathcal{L}_1 = \{1,\ldots,s/3\}$, $\mathcal{L}_2 = \{s/3+1,\ldots,2s/3\}$, and $\mathcal{L}_3 = \{2s/3+1,\ldots,s\}$. Let $\mathcal{M}_1,\ldots,\mathcal{M}_{s/3}$ be* edge-disjoint maximum matchings[9] *in a complete bipartite graph H whose one side is $\mathcal{L}_2$, and the other is $\mathcal{L}_3$. Notice that $|\mathcal{M}_i| = s/3$ for every i, and consider the following hyperedges.*

$$\begin{aligned} &\{\{1,a,b\}|\{a,b\} \in \mathcal{M}_1\}, \quad \{\{2,a,b\}|\{a,b\} \in \mathcal{M}_2\}, \\ &\qquad \ldots, \quad \{\{s/3,a,b\}|\{a,b\} \in \mathcal{M}_{s/3}\} \end{aligned}$$

[9]Recall that a matching is a subset of disjoint edges. A *maximal* matching is a matching such that any edges that is added to it violates the disjointness of its edges. A *maximum* matching is a matching of the largest possible cardinality. It is readily verified that a complete bipartite graph $K_{m,m}$ contains m disjoint maximum matchings.

We claim that any two of the above hyperedges intersect in at most one node. Assuming otherwise we have $|\{a_1, a_2, a_3\} \cap \{b_1, b_2, b_3\}| = 2$ *for some integers* a_i *and* b_i*. If* $a_1 = b_1$*, it follows that the edges* $\{a_2, a_3\}$ *and* $\{b_2, b_3\}$ *in* H *share a vertex, even though they both belong to* $\mathcal{M}_{a_1}$*, a contradiction. If* $a_1 \neq b_1$*, it follows that the matchings* $\mathcal{M}_{a_1}$ *and* $\mathcal{M}_{b_1}$ *both contain the edge* $\{a_2, a_3\} = \{b_2, b_3\}$*, another contradiction.*

Therefore, the resulting system is 2-private, accommodates $n = s^2/9$ *files, incurs storage overhead of* 1.5*, and has PIR rate of* $1/s$*. For comparison, considering the full graph on* s *nodes and applying the scheme in Section III provides a 2-private system with* $n = (s^2+s)/2$ *files, storage overhead* 2*, and comparable PIR rate* $1/s$*.*

VI. Discussion and Open Questions

In this paper we initiated a study of private information retrieval for a specific storage model that is widely used in practice, and widely studied in theoretical research. In order to improve our understanding of this model, and in order to improve its applicability to real-world systems, we suggest the following research directions.

1) Close the gap between achievable PIR rate in Subsection III-A and the upper bound in Subsection III-B.
2) Improve the collusion resilience in systems with arbitrary replication factors.
3) Construct families of dense graphs in which $\mathcal{T}(\mathcal{S}, \phi)$ (1) is large for every $\mathcal{S} \subseteq [s]$ and every ϕ.
4) Study graceful degradation for replication factors larger than two.
5) Find PIR schemes for 2-replication systems that guarantee collusion resistance against cycles, and are nontrivial (i.e., download less than the entire dataset).

Appendix A
Proof of the Main Theorem

The proof of Theorem 5 requires two auxiliary lemmas (Lemma 14 and Lemma 15), and then is proved in two parts (Lemma 16 and Lemma 17).

Lemma 14. *Let* $C \subseteq G$ *be a cycle with* c *edges, and let* $M \in \mathbb{F}_q^{c \times (c-1)}$ *be a matrix which is* $(V(C), E(C) \setminus \{j\})$*-compatible, where* j *is the maximum index of an edge in* $E(C)$*. Then, there exist precisely* $q-1$ *vectors* $\boldsymbol{a} \in \mathbb{F}_q^c$ *such that* $M' \triangleq (M|\boldsymbol{a}) \in \mathbb{F}_q^{c \times c}$ *is* $(V(C), E(C))$*-compatible and* $\operatorname{rank}(M') = c - 1$*.*

Proof. First, observe that since $C \setminus \{j\}$ is a tree, and since M is $(V(C), E(C) \setminus \{j\})$-compatible with G, it follows that $\operatorname{rank} M = c - 1$. Hence, the added vector $\mathbf{a}$ must be in $\operatorname{colspan}(M)$, i.e.,

$$\mathbf{a} = \sum_{k \in E(C) \setminus \{j\}} m_k \mathbf{c}_k, \tag{6}$$

where the $\mathbf{c}_k$'s are the columns of M and the m_k's are coefficients from $\mathbb{F}_q$. Furthermore, since M' must be compatible with G, the column $\mathbf{a}$ must contain nonzero entries precisely in row i_1 and row i_2, that correspond to the two vertices incident with edge j. Hence, since each row $k \in V(C) \setminus \{i_1, i_2\}$ of M contains precisely two nonzero entries in some columns k_1 and k_2, it follows that intersecting the column span of M with $N_k \triangleq \{\mathbf{x} = (x_i)_{i=1}^c \in \mathbb{F}_q^c | x_k = 0\}$ reduces the degrees of freedom in (6) by 1, since it renders any one of $\{m_{k_1}, m_{k_2}\}$ to be a linear function of the other. Therefore,

$$\dim(X) = (c-1) - (c-2) = 1, \text{ where}$$
$$X \triangleq \operatorname{colspan}(M) \bigcap \left(\bigcap_{k \in V(C) \setminus \{i_1, i_2\}} N_k \right).$$

Since any nonzero vector in X is a suitable candidate for $\mathbf{a}$, the claim follows. □

Lemma 15. *If an edge* $e \in E(G)$ *is on a cycle in* G*, then there exists a BFS ordering of* $E(G)$ *for which* e *is a back edge.*

Proof. Denote $e_\phi = \{v_f, v_g\}$ and choose $v_d \in V(G)$ which maximizes $\operatorname{dist}(v_g, v_d)$, where distance between two vertices is defined as the number of edges in the shortest path between them. Without loss of generality, assume that $\operatorname{dist}(v_g, v_d) \geq \operatorname{dist}(v_f, v_d)$, and consider a BFS run which begins at v_d. Partition $V(G)$ to layers $L_1, L_2, \ldots$ according to their distance from v_d, and recall that edges inside each layer are always back edges. Hence, if e_ϕ is inside a layer, we are done. Otherwise, assume that v_f is in L_i for some i, and hence v_g is in L_{i+1}. Since e_ϕ is on a cycle, there exists another edge e' from a node $v' \in L_i$ to v_g. Hence, in cases where v' pops out of the queue before v_f, e_ϕ will indeed be a back edge. It is readily verified that the order of insertion of discovered vertices in the same layer is arbitrary, and hence there exists a BFS run in which v' predates v_f, and the claim follows. □

We now turn to prove Theorem 5 in two parts.

Lemma 16. *For every subgraph* $T \subseteq G$*, the support of the random variable* $Q^T|\phi$ *is the set of all matrices* $A \in \mathbb{F}_q^{|V(T)| \times |E(T)|}$ *such that*:

(a) A *is* T*-compatible with* G*; and*
(b) *for every cycle* $C \subseteq T$

$$\operatorname{rank}(A^C) = \begin{cases} |E(C)| & \text{if } \phi \in E(C) \\ |E(C)| - 1 & \text{if } \phi \notin E(C) \end{cases}.$$

Proof. For simplicity assume that $2|q$, but other cases can be proved similarly. By the definition of $Q^T|\phi$, it is evident that (a) is necessary, and according to Proposition 3, it follows that (b) is necessary. In what follows, it is shown that (a) and (b) are also sufficient. To this end, let $A \in \mathbb{F}_q^{|V(T)| \times |E(T)|}$ be a matrix which satisfies (a) and (b), and it is shown that there exists a choice of $\boldsymbol{\alpha}, \boldsymbol{\gamma}$, and h for which $Q^T|\phi$ produces A.

Consider a BFS run on T, and number $V(T)$ and $E(T)$ according to their discovery times. That is, let $v_1, \ldots, v_{|V(T)|}$ be the vertices of T sorted by their discovery times, and let $e_1, \ldots, e_{|E(T)|}$ be the edges of T sorted by their discovery times. Also, assume that if $e_\phi \in E(T)$, and e_ϕ closes a cycle, then it is a back edge (see Lemma 15).

The values of $\boldsymbol{\alpha}, \boldsymbol{\gamma}$, and h which produce A are determined according to this BFS ordering, as follows.

First, fix an arbitrary value in $\mathbb{F}_q^*$ for γ_1. Then, since v_1 is incident with the edges $e_1,\ldots,e_{|\Gamma(v_1)|}$, we fix the values of $\alpha_1,\ldots,\alpha_{|\Gamma(v_1)|}$ as $\alpha_i \triangleq A_{v_1,e_i}/\gamma_1, i \in \{1,\ldots,|\Gamma(v_1)|\}$. Then, for $v_2,\ldots,v_{|\Gamma(v_1)|+1}$, that are the end vertices of $e_1,\ldots,e_{|\Gamma(v_1)|}$, respectively, we fix $\gamma_i = A_{v_i,e_{i-1}}/\alpha_{i-1}, i \in \{2,\ldots,|\Gamma(v_1)|+1\}$. If e_ϕ is not on a cycle in T, and e_ϕ happens to be, say, e_1, then we can obviously choose $\alpha_2 \triangleq A_{v_2,e_1}/(\gamma_1 \cdot h)$, where h is arbitrary (the case where e_ϕ lies on a cycle is treated in the sequel). Clearly, this process goes on unhindered as long as a back edge is not discovered.

Once a back edge $e_b = \{v_c, v_d\}, b \neq \phi$ is discovered, we have that γ_c, γ_d were already determined in earlier stages of the algorithm. Hence, we ought to show that there exists α_b for which

$$\alpha_b = \frac{A_{v_c,e_b}}{\gamma_c}, \text{ and} \qquad \alpha_b = \frac{A_{v_d,e_b}}{\gamma_d}. \tag{7}$$

To this end, let C be a cycle which is discovered in whole when e_b is discovered and let c be its number of edges. Further, let $M \triangleq A^{C\setminus\{e_b\}}$, i.e., the partial matrix of A which corresponds to the subgraph $C \setminus \{e_b\}$. Similarly, let $N \triangleq \operatorname{diag}(\boldsymbol{\gamma}_{V(C)}) I^{C\setminus\{e_b\}} \operatorname{diag}(\boldsymbol{\alpha}_{E(C)\setminus\{e_b\}})$ be the matrix which corresponds to the choice of entries in $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ up until e_b is discovered. By the correctness of the algorithm so far, it follows that $M = N$. Moreover, both M and N are $(V(C), E(C)\setminus\{j\})$-compatible, and by the definition of A, the submatrix A^C is C-compatible, and its rank is $c-1$. According to Lemma 14 there exist precisely $(q-1)$ columns $\mathbf{c}_1,\ldots,\mathbf{c}_{q-1}$ that extend M (and also N) to a C-compatible matrix of rank $c-1$, one of which is A^C. Further, it is evident that the matrix $\operatorname{diag}(\boldsymbol{\gamma}_{V(C)}) I^C \operatorname{diag}(\boldsymbol{\alpha}_{E(C)})$, for *any* of the $(q-1)$ possible values of $\alpha_b \in \mathbb{F}_q^*$, results in a C-compatible matrix of rank $c-1$ as well. Therefore, there exists a 1-1 correspondence between the possible values of α_b and $\mathbf{c}_1,\ldots,\mathbf{c}_{q-1}$. Since one of $\mathbf{c}_1,\ldots,\mathbf{c}_{q-1}$ is the actual e_b'th column of A^C, it follows that there exists a unique value of $\alpha_b \in \mathbb{F}_q^*$ which satisfies (7).

If e_ϕ lies on a cycle C' in T, we denote $e_\phi \triangleq \{v_f, v_g\}$. Since e_ϕ is a back edge, we have that γ_g and γ_f were determined in earlier steps of the algorithm. Hence, we must find $\alpha_\phi \in \mathbb{F}_q^*$ and $h \in \mathbb{F}_q \setminus \{0,1\}$ for which

$$h\gamma_g\alpha_\phi = A_{v_g,e_\phi} \tag{8}$$

$$\gamma_f\alpha_\phi = A_{v_f,e_\phi}.. \tag{9}$$

Clearly, the choice $\alpha_\phi \triangleq A_{v_f,e_\phi}/\gamma_f$ satisfies (9), and consequently, $h \triangleq \frac{A_{v_g,e_\phi}}{\gamma_g\alpha_\phi}$ satisfies (8). We are only left to show that this value for h is neither 0 nor 1. First, it is obviously nonzero as a product of nonzero terms. Second, if $h = 1$ happens to be the answer, we have by Proposition 3 that $A^{C'}$ is rank-deficient, in contradiction with condition (b). □

Lemma 17. *For every $T \subseteq G$, the random variable $Q^T|\phi$ is uniformly distributed on its support.*

Proof. Let A be a matrix in the support of $Q^T|\phi$. By following the proof of Lemma 16, we have that once γ_1 is fixed, and as long as a back edge is not discovered, every edge-node incidence reduces the overall probability of obtaining A by $(q-1)^{-1}$. In addition, every back edge which is not e_ϕ reduces the probability of obtaining A by $(q-1)^{-1}$ due to (7), instead of by $(q-1)^{-2}$ for tree edges.[10] Finally, if e_ϕ lies on a cycle, it reduces the overall probability by $\frac{1}{q-1}$ due to (9) and by $\frac{1}{q-2}$ due to (8). Therefore, we have the following, where u denotes the number of edge-node incidences in T, and k denotes the number of back edges in a BFS run (which is identical in every run of a BFS algorithm).

- If e_ϕ is not on a cycle in T then

$$\Pr((Q^T|\phi) = A) = \left(\frac{1}{q-1}\right)^{u-k}.$$

- If e_ϕ is on a cycle in T then

$$\Pr((Q^T|\phi) = A) = \left(\frac{1}{q-1}\right)^{u-k} \cdot \frac{1}{q-2}.$$ □

Appendix B
Choice of Sets

The process of choosing the sets $\{J^{(j,i)}\}_{(j,i)\in[r]\times[b]}$ in (5) is very simple, and is best illustrated by the following examples.

Example 18. *Assume that $N-K=4$ and $K=6$, which implies that $r=3$ and $b=2$. Consider the following matrix*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & & \\ 2 & 2 & & & 1 & 1 \\ & & 2 & 2 & 2 & 2 \end{pmatrix},$$

which naturally corresponds to the sets

$$J^{(1,1)} = \{1,2,3,4\} \qquad J^{(1,2)} = \varnothing$$
$$J^{(2,1)} = \{5,6\} \qquad J^{(2,2)} = \{1,2\}$$
$$J^{(3,1)} = \varnothing \qquad J^{(3,2)} = \{3,4,5,6\}.$$

As another example, in which $N-K \geq K$, we may consider the following.

Example 19. *Assume that $N-K=6$ and $K=4$, which implies that $r=2$ and $b=3$. Consider the following matrix*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 \\ 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix}$$

which naturally corresponds to the sets

$$J^{(1,1)} = \{1,2,3,4\} \qquad J^{(2,1)} = \varnothing$$
$$J^{(1,2)} = \{5,6\} \qquad J^{(2,2)} = \{1,2\}$$
$$J^{(1,3)} = \varnothing \qquad J^{(2,3)} = \{3,4,5,6\}.$$

Appendix C
Improved Bound

In this section it is shown that the upper bound in Lemma 9 can be improved in some cases. Consider the dual linear program in (3). We begin by presenting a feasible vector $\boldsymbol{\rho}$,

[10] An edge which is not a back edge in a BFS ordering is called a tree edge.

for which the value of the objective $\mathbb{1}_n \cdot \boldsymbol{\eta}^\top$ is often larger than $\frac{n}{\delta}$. Let $\boldsymbol{\rho} = (\rho_i)_{i=1}^n \in \mathbb{R}^n$ be such that

$$\rho_i \triangleq \frac{1}{\max\{\deg(a_i), \deg(b_i)\}},$$

where a_i and b_i are the vertices incident with edge e_i, for every $i \in [n]$. That is, for every $i \in [n]$ the value of ρ_i is the inverse of the maximum among the degrees of the vertices that are incident with the edge e_i.

It is now shown that $\boldsymbol{\rho}$ is a feasible vector for (3). Since $\boldsymbol{\rho} \geq 0$, it suffices to show that $I(G) \cdot \boldsymbol{\rho}^\top \leq \mathbb{1}_s$. For $j \in [s]$ let $I(G)_j$ be the j'th row of $I(G)$, which is a zero-one vector indicating the edges that are incident with node v_j, and denote the neighboring vertices of v_j by $u_1, \ldots, u_{\deg(v_j)}$. Hence, we have

$$\begin{aligned} I(G)_j \cdot \boldsymbol{\rho}^\top &= \sum_{k=1}^{\deg(v_j)} \frac{1}{\max\{\deg(v_j), \deg(u_k)\}} \\ &\leq \sum_{k=1}^{\deg(v_j)} \frac{1}{\deg(v_j)} = 1, \end{aligned}$$

and therefore $I(G) \cdot \boldsymbol{\rho}^\top \leq \mathbb{1}_s$. Recall that the value of the objective function for this feasible vector is

$$\mathbb{1}_n \cdot \boldsymbol{\rho}^\top = \sum_{i=1}^{n} \frac{1}{\max\{\deg(a_i), \deg(b_i)\}} \triangleq \mu,$$

and hence an immediate bound on the PIR rate is μ^{-1}.

This bound is stronger than Lemma 9 for graphs that are far from being regular; e.g., where there exist one node of high degree, whereas the remaining nodes are of low degree. For example, consider graphs which contain an ℓ-regular subgraph on $s-1$ nodes for some constant ℓ, and an additional vertex of degree $s-1$. These graphs satisfy $n = (s-1)(1+\ell/2)$ and $\delta = s-1$. The bound resulting Lemma 9 is $1/(1+\ell/2)$, a constant, whereas

$$\mu^{-1} = \frac{1}{(s-1) \cdot \frac{1}{s-1} + \frac{(s-1)\ell}{2} \cdot \frac{1}{\ell+1}} = \frac{2(\ell+1)}{2(\ell+1) + \ell(s-1)}$$

goes to zero as s grows.

REFERENCES

[1] M. A. Attia, D. Kumar, and R. Tandon, "The capacity of private information retrieval from uncoded storage constrained databases," May 2018, *arXiv:1805.04104*. [Online]. Available: https://arxiv.org/abs/1805.04104

[2] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," Sep. 2016, *arXiv:1609.08138*. [Online]. Available: https://arxiv.org/abs/1609.08138

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

[4] K. Banawan and S. Ulukus, "Private information retrieval from non-replicated databases," Dec. 2018, *arXiv:1901.00004*. [Online]. Available: https://arxiv.org/abs/1901.00004

[5] S. Blackburn and T. Etzion, "PIR array codes with optimal PIR rates," Sep. 2016, *arXiv:1609.07070*. [Online]. Available: https://arxiv.org/abs/1609.07070

[6] S. Blackburn, T. Etzion, and M. B. Paterson, "PIR schemes with small download complexity and low storage requirements," Sep. 2016, *arXiv:1609.07027*. [Online]. Available: https://arxiv.org/abs/1609.07027

[7] *Apache Cassandra 2.1 for DSE, Data replication*. Accessed: Dec. 4, 2019. [Online]. Available: https://docs.datastax.com/en/cassandra/2.1/cassandra/architecture/architectureDataDistributeReplication_c.html

[8] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. IEEE 36th Annu. Found. Comput. Sci. (FOCS)*, Oct. 1995, pp. 41–50.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[10] Z. Dvir and S. Gopi, "2-server PIR with subpolynomial communication," in *Proc. 47th Annu. ACM Symp. Theory Comput. (STOC)*, 2015, pp. 577–584.

[11] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep./Oct. 2010, pp. 1510–1517.

[12] A. Fazeli, A. Vardy, and E. Yaakobi, "PIR with low storage overhead: Coding instead of replication," May 2015, *arXiv:1505.06241*. [Online]. Available: https://arxiv.org/abs/1505.06241

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

[14] Hadoop Distributed File System (HDFS). *Architecture Guide-Data Replication*. Accessed: Dec. 4, 2019. [Online]. Available: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#Data+Replication

[15] Z. Jia and S. A. Jafar, "On the asymptotic capacity of X-secure T-private information retrieval with graph based replicated storage," Apr. 2019, *arXiv:1904.05906*. [Online]. Available: https://arxiv.org/abs/1904.05906

[16] Z. Jia, H. Sun, and S. A. Jafar, "The capacity of private information retrieval with disjoint colluding sets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[17] D. Karpuk, "Private computation of systematically encoded data with colluding servers," Jan. 2018, *arXiv:1801.02194*. [Online]. Available: https://arxiv.org/abs/1801.02194

[18] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[19] U. S. R. Murty, "A generalization of the Hoffman–Singleton graph," *ARS Combin.*, vol. 7, pp. 191–193, 1979.

[20] N. B. Shah, K. V. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2014, pp. 856–860.

[21] N. Silberstein and T. Etzion, "Optimal fractional repetition codes based on graphs and designs," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4164–4180, Aug. 2015.

[22] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

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

[24] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," May 2016, *arXiv:1605.00635*. [Online]. Available: https://arxiv.org/abs/1605.00635

[25] H. Sun and S. A. Jafar, "Multiround private information retrieval: Capacity and storage overhead," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5743–5754, Aug. 2018.

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

[27] 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 *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1908–1912.

[28] R. Tandon, M. Abdul-Wahid, F. Almoualem, and D. Kumar, "Private information retrieval from storage constrained databases–coded caching meets PIR," Nov. 2017, *arXiv:1711.05244*. [Online]. Available: https://arxiv.org/abs/1711.05244

[29] Q. Wang and M. Skoglund, "Symmetric private information retrieval for MDS coded distributed storage," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

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

[31] S. Yekhanin, *Locally Decodable Codes and Private Information Retrieval Schemes*. Berlin, Germany: Springer-Verlag, 2010.

[32] L. Yohananov and E. Yaakobi, "Codes for graph erasures," May 2017, *arXiv:1705.02639*. [Online]. Available: https://arxiv.org/abs/1705.02639

[33] L. Yohananov and E. Yaakobi, "Codes for erasures over directed graphs," Sep. 2017, *arXiv:1709.04701*. [Online]. Available: https://arxiv.org/abs/1709.04701

Netanel Raviv (S'15–M'17) received the B.Sc. degree in mathematics and computer science and the M.Sc. and Ph.D. degrees in computer science from the Technion—Israel Institute of Technology, Israel, in 2010, 2013, and 2017, respectively. He was a Post-Doctoral Scholar with the Center for the Mathematics of Information (CMI), California Institute of Technology. He is an Assistant Professor with the Department of Computer Science and Engineering, McKelvey School of Engineering, Washington University in Saint Louis. His research interests include applications of coding theory to computation, privacy, and storage. He was awarded the IBM Ph.D. Fellowship from 2015 to 2016, the First Prize of the Feder Family Competition for best student work in communication technology, and the Lester–Deutsche Post-Doctoral Fellowship.

Itzhak Tamo received the B.A. degree in mathematics, and the B.Sc. and Ph.D. degrees in electrical engineering from Ben-Gurion University, Israel, in 2008 and 2012, respectively. From 2012 to 2014, he was a Post-Doctoral Researcher with the Institute for Systems Research, University of Maryland, College Park. Since 2015, he has been a Senior Lecturer with the Electrical Engineering Department, Tel-Aviv University, Israel. His research interests include storage systems and devices, coding, information theory, and combinatorics. He was a corecipient (with Zhiying Wang and Jehoshua Bruck) of the IEEE Communication Society Data Storage Technical Committee 2013 Best Paper Award. He received the 2015 IEEE Information Theory Society Paper Award along with A. Barg. In 2018, he received the Krill Prize.

Eitan Yaakobi (S'07–M'12–SM'17) received the B.A. degree in computer science and mathematics and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. He is an Associate Professor with the Computer Science Department, Technion—Israel Institute of Technology. From 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology, and the Center for Memory and Recording Research, University of California. His research interests include information and coding theory with applications to nonvolatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He received the Marconi Society Young Scholar Award in 2009 and the Intel Ph.D. Fellowship from 2010 to 2011.