Bounds on the Capacity of PIR over Graphs

Bar Sadeh*, Yujie Gu†, and Itzhak Tamo*

*Department of Electrical Engineering–Systems, Tel-Aviv University, Tel-Aviv 39040, Israel.
†Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan.

Abstract

In the private information retrieval (PIR) problem, a user wants to retrieve a file from a database without revealing any information about the desired file’s identity to the servers that store the database. In this paper, we study the PIR capacity of a graph-based replication system, in which each file is stored on two distinct servers according to an underlying graph. This paper aims to provide upper and lower bounds to the PIR capacity of graphs via various graph properties. In particular, we provide several upper bounds on the PIR capacity that apply to all graphs. We further improve the bounds for specific graph families (which turn out to be tight in certain cases) by utilizing the underlying graph structure. For the lower bounds, we establish optimal rate PIR retrieval schemes for star graphs via edge-coloring techniques. Lastly, we provide an improved PIR scheme for complete graphs, which implies an improved general lower bound on all graphs’ PIR capacity.

1 Introduction

Private information retrieval (PIR), introduced by Chor et al. [6, 7], aims at enabling users to efficiently retrieve a file from public databases while not revealing any information on the identity of the desired file (privacy requirement). In the classical settings of [6], there are $K$ files that are replicated across $N$ non-communicating servers and a user that generates and sends $N$ queries to the servers, one for each server. Upon receiving its query, a server responds truthfully with an answer to the user, who can then use the answers to retrieve the desired file (reliability requirement).

In the PIR problem, one aims at constructing efficient retrieval schemes, where the total communication cost measures efficiency during the retrieval scheme, i.e., the total number of bits sent by the user and the servers. This was, in fact, the main figure of merit in the original paper [6], see also [1, 9, 11]. On the other hand, in [8, 20], the main figure of merit was only the download cost (the cost of sending the answers by the servers), motivated by the understanding that the query size (measured in bits) is negligible when compared with the current file sizes and therefore also the answers’ size. In this work, we adopt this approach and...
define the maximum ratio between the retrieved file size and the download cost of a retrieval scheme to be the PIR capacity.

In recent years the PIR problem has attracted a tremendous amount of attention from the research community. Among many results, the PIR capacity in the classical settings, where each server holds a replica of the database, was determined in [20]. Moreover, many variants of the PIR problem were also considered and studied. For instance, the PIR problem under colluding servers, where some servers are allowed to communicate and share the queries between themselves [10, 19, 21]; the coded PIR problem [4, 8, 23] where the files are not replicated but rather encoded using linear codes and then stored across the servers; the symmetric PIR problem [22, 24] where both the user and databases' privacy should be preserved, and many more.

Although coding techniques for storage systems have been extensively developed in recent decades, and even some of them have found their way to real-world systems, still many system designers favor replication over coding as a means to provide resiliency to data loss due to reasons such as simplicity of implementation, file updates, and availability (see [2, 12, 16]). Nevertheless, with the current amounts of stored data, databases are far too large to be stored on a single server. Hence, it is impractical to replicate the whole database across all servers fully [17]. Therefore, practically every server stores only a part of the entire database, and each file is replicated among several servers. Such a replication system is modeled by a hypergraph whose vertices represent the servers, and its hyper-edges represent the files, where a hyperedge (a file) contains the vertices (servers) that store this file. Such a system is also called a (hyper-) graph-based replication system [17], and if each file is stored on exactly \(r\) servers, it is called an \(r\)-replication system. We note that the classical PIR settings, where each server stores the entire database, are just a special case of this problem.

In this work, we consider the PIR problem for graph-based systems, i.e., a 2-replication system modeled by a simple graph). This problem and its extensions have been considered in several recent works. In [17], Raviv et al. who initially introduced this model, studied the resiliency of the system against colluding servers. In [5], Banawan et al. explored the information-theoretic graph-based PIR capacity in the non-colluding model; however, their graph replication model is different from the model considered in this paper. Indeed, in [5] the files are the graph’s vertices, and each server is an edge that connects between the files the server stores. Hence, by assumption, each server stores only two files, whereas no such constraint is imposed in our model. In [13], Jia et al. studied the hypergraph-based PIR problem with additional properties that both the user’s privacy and the security of the stored data are protected against colluding servers.

Despite these works, the graph-based PIR problem is far from being understood, even for the simplest model of non-colluding servers. In this work, we try to bridge that gap and explore connections between this problem and the underlying graph structure. In particular, we derive two incomparable general upper bounds on the PIR capacity for general graphs via linear programming. Then, we further improve these bounds and derive tight upper bounds for star and Hamiltonian vertex-transitive graphs. This is achieved by utilizing the underlying
structure of the specific graph at hand. We complement these upper bounds with the following lower bounds. For star graphs, we construct an optimal rate PIR retrieval scheme using hypergraph edge-coloring. Lastly, we provide an improved PIR scheme for complete graphs, which implies an improved general lower bound on the PIR capacity for all graphs. For the reader’s convenience, we summarize our results in the following table.

| Graph on N vertices | Capacity Bounds | Reference |
|---------------------|-----------------|-----------|
| Star graph $S_N$    | $\mathcal{C}(S_N) = \Theta(N^{-\frac{1}{2}})$ | Theorem 16 |
| Complete graph $K_N$| $\frac{2^{N-1}}{2N-1} \cdot \frac{1}{N} \leq \mathcal{C}(K_N) \leq \frac{2}{N+1}$ | Theorem 20, Corollary 9 |
| General graph $G$    | $\mathcal{C}(K_N) \leq \mathcal{C}(G) \leq \min \left\{ \frac{\Delta}{|W|}, \frac{1}{\nu(G)}, \frac{L}{N} \right\}$ | Theorem 1, Corollary 6 |

Table 1: Summary of the main results. For the definitions and notations see Section 2, and for the upper bound on a general graph, see Theorem 1 and Corollary 6.

The remainder of this paper is organized as follows. The problem statement and notations are given in Section 2. The upper bounds on the PIR capacity are provided in Section 3. In Section 4 we present a capacity-achieving PIR scheme for star graphs. In Section 5 we construct a PIR retrieval scheme for complete graphs. We conclude in Section 6 with concluding remarks and some open questions.

2 Problem statement and notations

For positive integers $a < b$ let $[a, b] = \{a, a + 1, \ldots, b\}$ and $[a] = \{1, \ldots, a\}$. For elements $A_i, i \in I$ with indices in a set $I$ and a subset $J \subseteq I$ we let $A_J = \{A_i : i \in J\}$.

We consider the 2-replication private information retrieval problem in the non-colluding setting, meaning that each file in the system is replicated and stored on exactly two servers, any two servers store at most one common file, and servers do not share any information between themselves. Let $S = \{S_1, S_2, \ldots, S_N\}$ denote $N$ non-colluding servers and $W = \{W_1, W_2, \ldots, W_K\}$ denote $K$ independent files. Each $W_i \in \mathbb{F}_2^L$ is a binary vector of length $L$ chosen uniformly at random from all the vectors of $\mathbb{F}_2^L$, therefore

$$H(W_1) = \ldots = H(W_K) = L$$

$$H(W_1, \ldots, W_K) = H(W_1) + \cdots + H(W_K) = KL.$$

This model naturally can be described by a simple graph (i.e., a graph with no parallel edges and loops) $G = (S, \mathcal{W})$, whose vertex set is the set of servers $S$, and the edges are the files $\mathcal{W}$. A file $W_k \in \mathcal{W}$ is also viewed as the edge $\{S_i, S_j\}$ if the file is stored on servers $S_i$ and $S_j$. We denote by $W_{S_i} = \{W \in \mathcal{W} : S_i \in W\}$ the set of files stored on server $S_i$, and by $W_{i,j}$ the file that is stored on servers $S_i, S_j$. Hence, an instance of the PIR problem is uniquely defined by a graph, and any graph gives rise to an instance of the above PIR problem.

A user wants to retrieve a file $W_\theta$ privately, where $\theta$ is chosen uniformly at random from the set $[K]$, in the sense that each server learns nothing about the requested file index $\theta$. To
that end, the user uses a random variable $R$ whose realization $r$ is kept private and generates $N$ queries $Q_1^{[θ,r]}, \ldots, Q_N^{[θ,r]}$, one for each server. The query $Q_i^{[θ,r]}$ is then sent to server $S_i$. Throughout the paper, when it is understood from the context, we shall omit the superscript $[θ,r]$ and simply write $Q_i$ instead of $Q_i^{[θ,r]}$. Let $Q = \{Q_n : n ∈ [N]\}$ denote the set of all queries generated by the user. Since the user has no information on the content of the files, the queries are independent of them, which means that

$$I(W_1, \ldots, W_K; Q_1, \ldots, Q_N) = 0.$$ 

Server $S_i$ upon receiving its query $Q_i$ replies with an answer $A_i$, which is a function of the query $Q_i$ and the files it holds $W_{S_i}$. Thus, for any $i ∈ [N]$

$$H(A_i|Q_i, W_{S_i}) = 0.$$ 

A PIR scheme has two formal requirements, reliability and privacy, described next.

**Reliability:** The user should be able to retrieve the desired file $W_{θ}$ from the received answers $A_i$ with zero probability of error, hence

$$H(W_{θ}|A_{[N]}, Q_{[N]}) = 0.$$ 

**Privacy:** Each server learns no information about the desired file index $θ$, i.e., for any $i ∈ [N]$

$$H(θ|Q_i, W_{S_i}) = H(θ) = \log(K).$$ 

The main figure of merit for the PIR problem is its efficiency, i.e., how many more bits are transmitted than the actual file size being retrieved. As in [8, 20], we assume that the file size $L$ is significantly larger than the number of files in the system and therefore also than the query size, which means that $H(W) ≫ H(Q)$ for any query $Q ∈ Q$ and file $W ∈ W$. Hence, the query upload cost is negligible compared with the download cost (the answers $A_i$), and therefore we will merely focus on the download cost. We define the PIR rate of a retrieval scheme $T$ as

$$R_{T,L} = \frac{L}{\sum_{i ∈ [N]} H(A_i)},$$

which is the ratio between the retrieved file size and the total number of bits incurred in the system due to the answers. We note that we will omit the subscripts $L$ and $T$ in the sequel. As usual, the PIR capacity of a graph $G$ is defined as the best possible rate for arbitrarily large file size, i.e., the capacity of $G$ is defined as

$$C(G) = \sup_{T,L} R_{T,L}.$$
3 Upper bounds for the PIR capacity

In this section, we derive several upper bounds on the PIR capacity of general graphs. We begin with a bound that was originally derived in [17], however, here, we show that this bound holds more generally in the model of non-colluding servers. Then, we proceed to derive new incomparable upper bounds for general graphs. Lastly, we present an improved bound for star and Hamiltonian vertex-transitive graphs.

These derived upper bounds should be compared with the known lower bounds on the PIR capacity. The best known lower bound for the PIR capacity of a general graph is $C(G) \geq 1/N$, which follows from a retrieval scheme given in [17, Section III]. In section 5 we will improve the lower bound for general graphs by improving the lower bound for complete graphs.

3.1 Bounding the capacity via linear programming

Before presenting the main theorem of this subsection and its proof, we recap the following standard definitions from graph theory. Let $G = (V, E)$ be a graph. A matching $M \subseteq E$ of $G$ is a subset of edges, such that no two edges have a common vertex. The matching number of $G$, denoted by $\nu(G)$ is the maximum size of a matching of $G$, i.e., $\nu(G) = \max\{|M| : M \text{ is a matching of } G\}$. Lastly, the degree of a vertex is the number of edges incident to it, and $\Delta$ the maximum degree of $G$ is the degree of the vertex with the greatest number of edges incident to it.

The following theorem provides a general upper bound on the PIR capacity as a function of various graph parameters.

**Theorem 1.** Let $G = (S, W)$ be a graph with maximum degree $\Delta$, then its PIR capacity satisfies

$$C(G) \leq \min\left\{\frac{\Delta}{|W|}, \frac{1}{\nu(G)}\right\}.$$  \hspace{1cm} (1)

The proof of Theorem 1 follows from the following observation and lemma.

**Observation 2.** For any answer $A_i$, $i \in [N]$, a requested file index $k \in [K]$ and a set $J \subseteq [K]

$$H(A_i|Q_i, W_J, \theta = k) = H(A_i|Q_i, W_j).$$  \hspace{1cm} (2)

Indeed, given the query $Q_i$, the answer $A_i$ is a function of the files stored on the $i$-th server, independent of the requested file $\theta = k$.

**Lemma 3.** Let $i, j \in [N]$ be two distinct servers that share the $k$-th file $W_k \in W$, then

$$H(A_i) + H(A_j) \geq H(A_i|W \setminus \{W_k\}, Q) + H(A_j|W \setminus \{W_k\}, Q) \geq L.$$  

Proof: The proof goes along the same lines as the proof of Lemma 1 in [5]. The first inequality holds since conditioning reduces entropy, hence we proceed to prove the second inequality.

$$L = H(W_k)$$
\begin{align*}
= H(W_k | W \setminus \{W_k\}, Q, \theta = k) \quad & \text{(3)} \\
= H(W_k | W \setminus \{W_k\}, Q, \theta = k) - H(W_k | W \setminus \{W_k\}, Q, \theta = k, A_{[N]}) \quad & \text{(4)} \\
= H(W_k | W \setminus \{W_k\}, Q, \theta = k) - H(W_k | W \setminus \{W_k\}, Q, \theta = k, A_i, A_j) \quad & \text{(5)} \\
= I(W_k; A_i, A_j | W \setminus \{W_k\}, Q, \theta = k) \quad & \text{(6)} \\
= H(A_i, A_j | W \setminus \{W_k\}, Q, \theta = k) \quad & \text{(7)}
\end{align*}

where (3) follows since \( W_k \) is independent of the remaining files, the queries and the requested file \( \theta = k \); (4) follows from the reliability requirement, i.e., \( W_k \) can be retrieved from the queries \( Q, \theta = k \) and the answers \( A_{[N]} \); (5) holds since the answers \( A_{[N]} \setminus \{A_i, A_j\} \) are a function of \( Q \) and \( W \setminus \{W_k\} \); (6) holds since \( H(A_i, A_j | W, Q, \theta = k) = 0 \) and lastly, (7) follows from (2). \( \blacksquare \)

Next, we provide the proof of Theorem 1, which is similar to the proof of [17, Lemma 8]. We note that Theorem 1 is, in fact, a strengthening of [17, Lemma 8] since we show that the same result holds without the assumptions [17] that the servers may collude and share their queries amongst themselves.

We recall the following definition from graph theory. The incidence matrix \( I(G) \) of a graph \( G \) is a \( |V(G)| \times |E(G)| \) binary matrix in which rows correspond to vertices and columns correspond to edges, and an entry contains 1 if and only if the respective vertex is incident with the respective edge, otherwise it is zero.

**Proof of Theorem 1.** Consider the rate of some PIR scheme for \( G \)

\[
R = \frac{L}{\sum_{i \in [N]} H(A_i)} = \frac{1}{\sum_{i \in [N]} (H(A_i)/L)} = \frac{1}{\sum_{i \in [N]} \mu_i} = \frac{1}{\mathbb{1}_N \cdot \mu^T},
\]

where \( \mu_i \triangleq H(A_i)/L \) for any \( i \in [N] \), \( \mu \triangleq (\mu_1, \ldots, \mu_N) \), and \( \mathbb{1}_N \) is the all-ones vector of length \( N \). According to Lemma 3 if servers \( i \) and \( j \) are adjacent, then \( \mu_i + \mu_j \geq 1 \). Hence, we can get an upper bound on the PIR rate via the reciprocal of the optimal value of the following linear program.

\[
\min \quad \mathbb{1}_N \cdot \mu^T, \quad \text{subject to} \quad I(G)^T \cdot \mu^T \geq \mathbb{1}_K \quad \text{and} \quad \mu \geq 0.
\]

(8) Its dual problem is

\[
\max \quad \mathbb{1}_K \cdot \eta^T, \quad \text{subject to} \quad I(G) \cdot \eta^T \leq \mathbb{1}_N \quad \text{and} \quad \eta \geq 0,
\]

(9) where \( \eta \) is a vector of length \( K \). By the primal-dual theory, any feasible solution to (8) is a lower bound for (9). Below we provide two feasible solutions to (9).

(S1) It is readily verified that \( \eta = \frac{1}{\Delta} \cdot \mathbb{1}_K \) is a feasible solution of (9), where \( \Delta \) is the maximum degree in \( G \). Therefore, \( \mathbb{1}_K \cdot \eta^T = K/\Delta \), which implies that the PIR rate is at most \( \Delta/K \).
Let $M \subseteq W$ be a maximum matching of $G$, i.e., $|M| = \nu(G)$. Let $\eta = (\eta_1, \ldots, \eta_K) \in \{0, 1\}^K$ be an indicator vector of the set $M$, i.e., $\eta_i = 1$ if and only if $W_i \in M$. Again, it is readily seen that $\eta$ is a feasible solution of (9) and therefore as before the PIR rate is at most $1/\nu(G)$.

This completes the proof of the two upper bounds.

We now turn to present different types of graphs and their capacity bounds obtained by Theorem 1. The examples show that the two upper bounds from Theorem 1 are incomparable in general, as seen in the first and last examples.

Examples 4.

1) Let $G$ be a regular graph of degree $\Delta$ with an odd number of vertices $N$, and $K = \frac{\Delta \cdot N}{2}$ edges. It follows that $\nu(G) \leq \frac{N - 1}{2}$. Hence

$$\frac{1}{N} \leq C(G) \leq \min \left\{ \frac{2}{\nu(G)} \right\} = \frac{2}{N} < \frac{1}{\nu(G)}.$$

Note that this upper bound is, in fact, tight for some regular graphs, as can be verified by a graph that is a perfect matching on $N$ vertices for an even $N$. However, this upper bound is not tight in general for regular graphs, and in Section 3.3 we provide a tighter bound for a specific type of regular graphs.

2) Let $G = K_{N,M}$ be the complete bipartite graph over two sets of vertices of size $N$ and $M$ ($N \leq M$). Then, $K = N \cdot M$, $\Delta = M$, $\nu(G) = N$ and

$$\frac{1}{M + N} \leq C(G) \leq \frac{1}{N}.$$

It is an interesting open question to determine the exact PIR capacity of this graph.

3) Let $G = W_{2N}$ be the wheel graph on $2N$ vertices, formed by connecting a single vertex to all vertices of a cycle of length $2N - 1$. It is easy to verify that $K = 4N - 2$, $\Delta = 2N - 1$ and $\nu(G) = N$. Hence

$$\frac{1}{2N} \leq C(G) \leq \min \left\{ \frac{1}{2}, \frac{1}{N} \right\} = \frac{1}{N}.$$

This lower bound is not tight, since by merging a capacity achievable scheme for the cycle graph with the scheme for the star graph given in Theorem 16 we get a PIR scheme with a rate better than $\frac{1}{2N}$.

3.2 Bounding the PIR capacity via a system of inequalities

The bounds given in Section 3.1 on the PIR capacity are by no means optimal and tight in the general case. Therefore, one needs to use further properties of the specific graph at hand to obtain tighter bounds. In this section, we lower bound the amount of information sent by a specific server by some function of the amount of information sent by its neighbors. This
allows us to describe an optimization problem that upper bounds the scheme rate and whose solution depends on the specific graph structure.

Then, we will focus on a specific graph, the star graph, and apply the optimization problem to improve the upper bound for the star graph obtained using Theorem 1. The reason for considering the star graph is twofold. First, the star graph has a very simple structure, and it is one of the simplest types of graphs for which determining the PIR capacity is still non-trivial. Second, the results developed in this section may be used for general graphs since any graph is a union of star graphs of possibly different sizes. Indeed, every vertex and its set of neighbors form a star graph.

The main result of this section follows from the following theorem.

**Theorem 5.** Given a PIR scheme for a graph \( G = (S, W) \), then for every server \( S \in S \) of degree \( \delta \) and neighbors set \( N(S) = \{S_1, \ldots, S_\delta\} \) the following holds

\[
H(A_S) \geq \sum_{i=1}^\delta \max \left\{ 0, L - \sum_{j=i}^\delta H(A_{S_j}) \right\}.
\]

It immediately follows that the best lower bound is attained when the \( H(A_{S_j}) \) are ordered in a descending order, i.e., \( H(A_{S_1}) \geq H(A_{S_2}) \geq \ldots \geq H(A_{S_\delta}) \).

**Proof:** For \( i \in [\delta] \) let \( W_i \) be the file stored on \( S \) and \( S_i \). The answer \( A_S \) from server \( S \) satisfies

\[
\begin{align*}
H(A_S) & \geq H(A_S|Q) \\
& = H(A_S|Q, W \setminus W_\delta) \\
& \geq I(A_S; W_\delta|Q, W \setminus W_\delta) \\
& = \sum_{i=1}^\delta I(A_S; W_i|Q, W \setminus W_\delta, W_{[i-1]}) \\
& = \sum_{i=1}^\delta \left( H(A_S|Q, W \setminus W_\delta, W_{[i-1]}, \theta = i) - H(A_S|Q, W \setminus W_\delta, W_{[i]}, \theta = i) \right) \\
& = \sum_{i=1}^\delta I(A_S; W_i|Q, W \setminus W_\delta, W_{[i-1]}, \theta = i) \\
& = \sum_{i=1}^\delta \left( H(W_i|A_S, Q, W \setminus W_\delta, W_{[i-1]}, \theta = i) - H(W_i|A_S, Q, W \setminus W_\delta, W_{[i-1]}, \theta = i) \right) \\
& = \sum_{i=1}^\delta L - H(W_i|A_S, Q, W \setminus W_\delta, W_{[i-1]}, \theta = i)
\end{align*}
\]
where (10) follows since $A_S$ is independent of $W \setminus W_δ$; (11) follows from the chain rule for mutual information; (12) follows from (2); (13) follows since $W_i$ is independent of $\{W_{[i-1]}, Q, W \setminus W_δ, \theta\}$ and (14) follows since the entropy of $W_i$ given other random variables is at most $L$.

Next, we turn to derive an upper bound on $H(W_i|A_S, Q, W \setminus W_δ, W_{[i-1]}, \theta)$. For every $i \in [δ]$ let $A_{i,δ} = \{A_S, \ldots , A_S\}$.

\[H(W_i|A_S, Q, W \setminus W_δ, W_{[i-1]}, \theta = i)\]
\[\leq H(W_i|A_S, Q, W \setminus W_δ, W_{[i-1]}, \theta = i, A_{i,δ})\]
\[+ H(A_{i,δ}|A_S, Q, W \setminus W_δ, W_{[i-1]}, \theta = i)\]
\[= H(A_{i,δ}|A_S, Q, W \setminus W_δ, W_{[i-1]}, \theta = i)\]
\[\leq \sum_{j=i}^δ H(A_{S_j}),\]  

(17)

where (15) follows from the inequality $H(X|Y) \leq H(X|Y, Z) + H(Z|Y)$; (16) follows from the reliability requirement of the PIR scheme and (17) holds since conditioning reduces entropy and by the chain rule. Finally, by combining (14) and (17) the result follows.

The following corollary follows immediately from Theorem 5 and it provides a general bound on the PIR capacity of graphs. For a server $S \in S$ of degree $δ_S$ let $u^S_1, \ldots , u^S_{δ_S}$ be its neighbors.

Corollary 6. The PIR capacity of a graph $G = (S, W)$ is at most $C(G) \leq L/X$, where $X$ is the solution of the following minimization problem

\[\min \sum_{S \in S} x_S, \text{ subject to } x_S \geq \sum_{k=1}^{δ_S} \max \left\{0, L - \sum_{j=k}^{δ_S} x^{S_j}\right\} \text{ for every server } S \in S.\]  

(18)

Proof: Consider a PIR scheme for $G$, then by Theorem 5 one can verify that the $H(A_S)$’s for $S \in S$ are a feasible solution for (18) and therefore $X \leq \sum_{S \in S} H(A_S)$. Then

\[C(G) = \frac{L}{\sum_{S \in S} H(A_S)} \leq \frac{L}{X},\]

and the result follows.

Solving the minimization problem (18) might be difficult for general graphs. Therefore, obtaining an informative (closed-form) upper bound for the PIR capacity might not be possible. Hence, to obtain a closed-form upper bound, it might be easier to apply the result of Theorem 5 for specific graph families, as we shall do next for the star graph.

A star (graph) $S_N = (S, W)$ is a simple graph on $N$ vertices with one vertex, say $S_N$, with degree $\deg(S_N) = N - 1$ and the other $N - 1$ vertices are of degree 1. Hence, $K = |W| = N - 1$ and $W_i = \{S_i, S_N\}$ for $i \in [K]$. The matching number is $\nu(S_N) = 1$ and the maximal degree
Δ = N − 1, then by Theorem 4 \( C(S_N) \leq 1 \), which is uninformative, since the PIR capacity of any graph is at most 1. In the next theorem, we invoke Theorem 5 to obtain a significantly tighter upper bound on the star graph’s PIR capacity. In fact, the bound will be shown to be tight up to a multiplicative factor via an achievable scheme (Theorem 16).

**Theorem 7.** The PIR capacity of the star graph \( S_N \) satisfies \( C(S_N) \leq O(N^{-\frac{1}{2}}) \).

**Proof:** Consider a star based PIR retrieval scheme with \( N \) servers and \( K = N − 1 \) files. Let \( S_N \) be the unique vertex of degree \( N − 1 \), and without loss of generality assume that \( H(A_1) \geq H(A_2) \geq \ldots \geq H(A_{N−1}) \). Let \( t \) be the largest integer in \( [N−1] \) such that \( H(A_{N−t}) \leq \frac{L}{t} \), and note that \( t \geq 1 \), since \( H(A_{N−1}) \leq \frac{L}{1} \). Then, by Theorem 5

\[
H(A_N) \geq \sum_{i=1}^{N−1} \max \left\{ 0, L - \sum_{j=i}^{N−1} H(A_j) \right\}
\]

\[
\geq \sum_{i=N−t}^{N−1} \max \left\{ 0, L - \sum_{j=i}^{N−1} H(A_j) \right\}
\]

\[
= tL - \sum_{i=N−t}^{N−1} \sum_{j=i}^{N−1} H(A_{N−t})
\]

\[
\geq tL - \frac{t(t+1)}{2} \cdot \frac{L}{t}
\]

\[
= tL - \frac{t+1}{2} \cdot \frac{L}{t}
\]  

(19)

(20)

where (19) follows by the definition of \( t \). Notice that also by the definition of \( t \) that

\[
H(A_1) \geq H(A_2) \geq \ldots \geq H(A_{N−(t+1)}) > \frac{L}{t+1}.
\]  

(21)

Then, by combining (20) and (21)

\[
\sum_{i=1}^{N} H(A_i) \geq H(A_N) + \sum_{i=1}^{N−t−1} H(A_i)
\]

\[
\geq tL - \frac{t+1}{2} \cdot \frac{L}{t} + \frac{N−t−1}{t+1} \cdot \frac{L}{t+1}
\]

\[
\geq L \cdot \min_{t\in[N−1]} \left\{ \frac{t}{2} + \frac{N}{t+1} - \frac{3}{2} \right\}
\]

\[
\geq \left( \sqrt{2N} - 2 \right) L,
\]

where the last inequality follows since the function \( f(x) = x/2 + N/(x + 1) - 3/2 \) attains its minimum at \( x_{\min} = \sqrt{2N} − 1 \). Therefore the PIR rate satisfies

\[
\frac{L}{\sum_{i=1}^{N} H(A_i)} \leq \frac{L}{\left( \sqrt{2N} - 2 \right) L} = \frac{1}{\sqrt{2N} - 2},
\]

as required. ■
3.3 Bounding the capacity for Hamiltonian vertex-transitive graphs

This section proves an improved upper bound on the PIR capacity for Hamiltonian vertex-transitive graphs, which turns out to be tight for cycle graphs. We first recall the needed definitions. A graph $G$ is called Hamiltonian if it possesses a cycle that traverses every vertex in the graph, and it is called vertex-transitive graph if its automorphism group $\text{Aut}(G)$ is transitive on its vertices, i.e., for any two vertices $u, v \in V(G)$ there exists an automorphism $f \in \text{Aut}(G)$ such that $f(v) = u$.

By Theorem 1, the PIR capacity of the family of regular graphs on $N$ vertices (which also includes the family vertex-transitive graphs on $N$ vertices) is at most $2/N$. In the following theorem, we prove a slightly tighter bound.

**Theorem 8.** Let $G = (S, W)$ be a Hamiltonian vertex-transitive graph, then

$$\mathcal{C}(G) \leq \frac{2}{N+1}. \quad (22)$$

Before proceeding to prove Theorem 8 we remark the following. First, the upper bound (22) is tight for certain graphs. Indeed, it was shown in [5] that the PIR capacity of the cycle graph $C_N$ on $N$ vertices is $\mathcal{C}(C_N) = \frac{2}{N+1}$. Second, since the complete graph is vertex-transitive, we have the following immediate corollary, which provides a slightly tighter upper bound than the one obtained by Theorem 1 for complete graphs. In Section 5 we provide PIR schemes (lower bounds) for complete graphs.

**Corollary 9.** The capacity of the complete graph $K_N$ on $N$ vertices satisfies

$$\mathcal{C}(K_N) \leq \frac{2}{N+1}.$$ 

To prove Theorem 8 we will need the following lemma.

**Lemma 10.** In a vertex-transitive graph-based replication system, for any achievable rate $R$, there exists a scheme with a rate $R$ that satisfies

$$H(A_i|Q) = H(A_j|Q), \quad \forall i, j \in [N], \quad (23)$$

$$H(A_i|W \setminus \{W_{i,j}\}, Q) \geq \frac{L}{2}, \quad \forall i, j \in [N]. \quad (24)$$

The proof of Lemma 10 is deferred to Appendix A.

**Proof of Theorem 8** Let $(S_1, W_1, S_2, W_2, \ldots, S_{N}, W_{N}, S_1)$ be a Hamiltonian cycle in $G = (S, W)$. By Lemma 10 if $R$ is an achievable rate, then there exists a PIR scheme with $R$ satisfying also (23) and (24). Then

$$L = H(W_N)$$

$$= H(W_N|Q, \theta = N) \quad (25)$$

$$= H(W_N|Q, \theta = N) - H(W_N|A_{[N]}, Q, \theta = N) \quad (26)$$
\[ I(W_N; A_i|Q, \theta = N) = H(A_i|Q, \theta = N) - H(A_i|W_N, Q, \theta = N) \leq \sum_{i \in [N]} H(A_i|Q, \theta = N) - H(A_i|W_N, Q, \theta = N) = \sum_{i \in [N]} H(A_i|Q) - H(A_i|W_N, Q, \theta = N) \leq \sum_{i \in [N]} H(A_i|Q) - H(A_i|W_N, Q, \theta = N), \]

where (25) follows since the queries \( Q \) and the identity of the required file are independent of the required file content; (26) follows from the reliability requirement of the PIR scheme; (27) follows from Observation 2; and (28) is due to (23) in Lemma 10.

Rearranging the above inequality yields

\[ NH(A_1|Q) \geq L + H(A_1|W_N, Q, \theta = N) \geq L + \sum_{j=1}^{N-1} H(A_j|W_N, Q, A_{[j-1]}, \theta = N) \geq L + \sum_{j=1}^{N-1} H(A_j|W_N, Q, W \setminus \{W_j\}, \theta = N) \geq L + \sum_{j=1}^{N-1} H(A_j|Q, W \setminus \{W_j\}, \theta = N) \geq L + \sum_{j=1}^{N-1} \frac{L}{2} = \frac{(N + 1)L}{2}, \]

where (29) follows from the non-negativity of entropy; (30) follows from the chain rule of entropy; (31) follows since conditioning reduces entropy; (32) follows since the file \( W_j \) is stored only on servers \( S_j, S_{j+1} \) and therefore the answers \( A_{[j-1]} \) are deterministic functions of \( Q \) and \( W \setminus \{W_j\} \); (33) follows from the fact that \( W_N \in W \setminus \{W_j\} \) for \( j \in [N - 1] \); and (34) follows from (24) in Lemma 10 and Observation 2. Therefore, the PIR rate satisfies

\[ R = \frac{L}{\sum_{i \in [N]} H(A_i)} \leq \frac{L}{\sum_{i \in [N]} H(A_i|Q)} = \frac{L}{NH(A_1|Q)} \leq \frac{2}{N+1}. \]

\[ \square \]

**Remark 11.** Although the result seems to apply to very limited graph families, i.e., the graph’s requirements are very strong, this is not precise. The question of whether every connected
vertex-transitive graph has a Hamiltonian path or cycle is a famous open problem in graph theory, which is sometimes referred to as Lovász conjecture (see [14]). In fact, all the known examples of connected vertex-transitive graphs are Hamiltonian, except for four known examples: the Petersen graph, the Coxeter graph, and two graphs which are some variants of these two.

4 Retrieval schemes for star-based PIR system

In this section, we present retrieval schemes for the star graph. We begin with a simple scheme (Theorem 12) with rate $\Omega(N^{-1})$. Then, building on the ideas given in the first scheme, we proceed to present a scheme with an improved rate, which is, in fact, optimal up to a multiplicative factor. More precisely, the scheme has a rate $\Omega(N^{-1/2})$ which matches the upper bound given in Theorem 7. The precise rate achieved by the first scheme is stated in the next theorem.

**Theorem 12.** The star graph’s PIR capacity $S_N$ is at least $\mathcal{C}(S_N) \geq 2/N$.

To construct the retrieval scheme, we shall use the notion of an edge-coloring of a graph. A (proper) edge-coloring of a graph with possibly self-loops is a coloring of the graph edges, such that incident edges (i.e., edges that share a common vertex) have distinct colors. Formally, given a graph $G = (V, E)$, an edge-coloring of it with $T$ colors is a mapping $c : E \rightarrow [T]$ such that for any vertex $i \in V$ and $j, j'$ two distinct neighbors of $i$, $c(i, j) \neq c(i, j')$. Clearly, any edge-coloring requires a number of colors which is at least the maximum degree of $G$.

We will make use of the well-known Baranyai’s Theorem given below.

**Theorem 13 (Baranyai’s theorem).** [3, 18] Let $X$ be a set of size $N$. Given $t|N$, there exist $(N-1)/(t-1)$ partitions of $X$ into $t$-sets such that each $t$-subset of $X$ occurs in exactly one of these partitions.

**Remark 14.** For the special case of $t = 2$ and even $N$, Theorem 13 provides a partition of the edges of the complete graph on $N$ vertices to $N - 1$ perfect matchings.

Let $G$ be the complete graph on $N$ vertices (including self-loops), i.e., each vertex is connected to all other vertices, including itself; therefore, the degree of each vertex is $N$. We will show next by invoking Theorem 13 that $G$ admits an edge-coloring with exactly $N$ colors, which is clearly optimal.

**Lemma 15.** The graph $G$ admits an edge-coloring with $N$ colors.

**Proof:** Assume first that the number of vertices $N$ is even. By Theorem 13 the non-loop edges of $G$ can be partitioned into $N - 1$ perfect matchings, $M_1, \ldots, M_{N-1}$. These partitions define the color classes of the edge-coloring as follows. Color the edge $(i, j)$ for $i \neq j$ by the color $l$ if matching $M_l$ is the unique matching that contains this edge. Next, color all the self-loops by the color $N$. It is easy to verify that, indeed, this is a proper edge-coloring of $G$. 


Next, assume that $N$ is odd, and consider the complete graph (without self-loops) with vertex set $[N+1]$. Again, by Theorem 13 the edges of the graph can be partitioned to $N$ perfect matchings $M_i$. Define the coloring of $G$ as follows. For distinct vertices $i, j$ of $G$ color the edge $(i, j)$ by the color $l$ if matching $M_l$ is the unique matching that contains this edge. Lastly, for $i = 1, \ldots, N$ color the self-loop $(i, i)$ by color $l$ if edge $(i, N+1)$ is in matching $M_l$. It is easily verified that this is indeed a proper edge-coloring of $G$, which completes the proof.

Proof of Theorem 12: The result will follow from a PIR scheme with the claimed rate, given next. Suppose that each file $W = (W(1), \ldots, W(K)) \in \mathbb{F}_2^K$ is a binary vector of length $K = N - 1$. Let $G = (V, E)$ be the complete graph (including self-loops) on vertex set $V = [N - 1]$, and let $c : E \to [N - 1]$ be an edge-coloring of $G$ as given in Lemma 15. Next, we define the retrieval scheme.

(a) The user chooses a file index $\theta \in [K]$ and a permutation $\sigma : [K] \to [K]$ uniformly at random from the set of all permutations on the set $[K]$.

(b) The user generates $N$ queries $Q_i$ for $i = 1, \ldots, N$ and send them to the $N$ servers, as follows. To server $N$ it sends the permutation $\sigma$, whereas to server $1 \leq i < N$ it sends $\sigma(c(i, \theta))$, the random color assigned to the edge $(i, \theta)$ by the random permutation $\sigma$ and the coloring $c$. Hence, the queries take the following form

$$Q_i := Q_i(\theta, \sigma) = \begin{cases} \sigma(c(i, \theta)) & 1 \leq i < N, \\ \sigma & i = N. \end{cases}$$

(c) Server $i$ for $i = 1, \ldots, N$ replies with an answer $A_i$ as follows. For $1 \leq i < N$ the server returns the value of the $\sigma(c(i, \theta))$-th bit of the file $W_i$ it holds, i.e., it returns $W_i(Q_i)$. On the other hand, server $N$ for each distinct $m, n \in [K]$ sends the sum of the $\sigma(c(m, n))$-th bits of the files $W_m$ and $W_n$. To conclude, the reply $A_i$ takes the form

$$A_i(Q_i) = \begin{cases} W_i(\sigma(c(i, \theta))) & 1 \leq i < N, \\ \{W_m(\sigma(c((m, n)))) \oplus W_n(\sigma(c((m, n)))) : \forall m \neq n \in [K]\} & i = N. \end{cases}$$

Next, we show that the scheme has the claimed rate and satisfies the privacy and reliability requirements.

Rate: Recall that the file size is $K$ bits. On the other hand, each of the first $N - 1$ servers returns a single bit, and server $N$ returns $\binom{K}{2}$ bits; therefore, the retrieval rate of the scheme is

$$\frac{K}{\binom{K}{2} + (N - 1)} = \frac{2}{N}.$$  

Privacy: Regardless of the value of $\theta$ and the coloring $c$, since $\sigma$ is picked uniformly at random the $i$-th query $\sigma(c(i, \theta))$ for $i < N$ is uniformly distributed over $[K]$, and therefore
it reveals no information on \( \theta \). Similarly, the \( N \)-th query is completely independent of \( \theta \) and therefore the scheme is private.

**Reliability:** For any \( i \in [K] \), we will show that the user is able to retrieve the bit \( W_\theta(\sigma(c(i, \theta))) \). Since, when \( i \) ranges in \([K]\), \( \sigma(c(i, \theta)) \) ranges also in \([K]\) (here we use the property that \( c \) is a proper coloring with \( K \) colors), then the user is able to retrieve all the file bits, as needed.

For \( i = \theta \) the user receives the bit \( W_\theta(\sigma(c(\theta, \theta))) \) from server \( \theta \). For \( i \in [K], i \neq \theta \) the user receives the bit \( W_i(\sigma(c(i, \theta))) \) from server \( i \), whereas from server \( N \) it receives also the bit \( W_i(\sigma(c(i, \theta))) \oplus W_\theta(\sigma(c(\theta, \theta))) \). Therefore, by adding these two bits the user can recover the bit \( W_\theta(\sigma(c(\theta, \theta))) \), and the result follows.

The above scheme uses a coloring of the complete graph and provides a rate that decays linearly with the number of servers in the system. Next, we provide an improved scheme that uses the same idea of coloring; however, this time, the coloring will be of a hyper-graph, more specifically a Steiner system, defined below. The resulted scheme will be shown to be optimal up to a constant factor, i.e., it achieves the upper bound given in Theorem 7. Hence, we have the following theorem.

**Theorem 16.** The rate of the star graph satisfies \( C(S_N) = \Theta(N^{-\frac{1}{2}}) \).

The upper bound was proven in Theorem 7 while the lower bound will follow from a PIR scheme with the claimed rate. As already mentioned, the improved scheme will rely on a coloring of Steiner system, whose definition is given next.

**Definition 17.** A Steiner system \( S(t, k, n) \) is a set system \((X, B)\), where \( X \) is an \( n \)-element set, say \( X = [n] \), and \( B \subseteq \binom{X}{t} \) is a collection of \( k \)-element subsets (blocks) of \( X \), such that any \( t \)-element subset of \( X \) is contained in exactly one block of \( B \).

A Steiner system \( S(2, q, q^2) \) for a prime power \( q \) can be constructed from \( A^2(\mathbb{F}_q) \) the classical affine geometry of dimension 2. Indeed, set \( X = \mathbb{F}_q^2 \), i.e., all the ordered pairs \((x, y) \in \mathbb{F}_q^2 \) as elements; and blocks \( B \subseteq \binom{X}{2} \) to be all subsets of the form \( y = mx + b \) or \( x = a \) (for fixed \( a, m, b \in \mathbb{F}_q \) ), that is, blocks of the form

\[
\{(x, mx + b) : x \in \mathbb{F}_q \} \text{ for } m, b \in \mathbb{F}_q \text{ or } \{(a, y) : y \in \mathbb{F}_q \} \text{ for } a \in \mathbb{F}_q.
\]

It is clear that there are total of \( q^2 \) elements in \( X \), the size of each block \( B \in B \) is \( |B| = q \) and that the number of blocks is \( |B| = q(q + 1) \). In addition, for distinct elements \( x, y \in X \) there exists a unique block, denoted by \( B_{x, y} \in B \), that contains both \( x \) and \( y \) (this is simply the line that connects between \( x \) and \( y \)). Lastly, each element appears in exactly \( q + 1 \) blocks.

As we used an edge-coloring of a graph in the last construction, we will define a (proper block) coloring for a set system such that blocks that share an element have distinct colors. Formally, given a blocks set \( \mathcal{B} \) of \( X \), a coloring of \( \mathcal{B} \) with \( T \) colors is a mapping \( c : B \to [T] \) such that distinct colors are assigned to distinct blocks that share an element, i.e., if blocks \( B, B' \in \mathcal{B} \) are distinct blocks such that \( B \cap B' \neq \emptyset \), then \( c(B) \neq c(B') \). Clearly, any coloring of
the Steiner system $S(2, q, q^2)$ described above requires at least $q + 1$ colors, since each $x \in X$ is contained in exactly $q + 1$ blocks. The next lemma shows that the lower bound on the number of needed colors is tight, i.e., there is a coloring with only $q + 1$ colors.

**Lemma 18.** The Steiner system $S(2, q, q^2)$ admits a coloring with $q + 1$ colors.

**Proof:** Each block of the Steiner system $S(2, q, q^2)$ corresponds to a line in the affine plane $\mathbb{A}^2(F_q)$ with a slope $m \in F_q \cup \{\infty\}$, where we define the slope of a line parallel to the $y$-axis as $\infty$. We let the color of a block be its slope $m$. One can easily verify that every two distinct lines that share a point have distinct slopes; therefore, this is a proper coloring of the blocks. \[\square\]

Next, we proceed to describe our scheme and prove its properties.

**Proof of Theorem 16.** Recall that $S_N = (S, W)$ is a star graph with $N$ vertices where $S_i = \{W_i\}$ for any $1 \leq i \leq N - 1$, and $S_n = W$. First, assume that the number of files $K = N - 1 = q^2$ for a prime power $q$, and suppose that each file $W = (W(1), \ldots, W(q + 1)) \in F^{q+1}_2$ is a binary vector of length $q + 1$. Let $(X, B)$ for $X = [q^2]$ be the Steiner system $S(2, q, q^2)$ described above, and let $c : B \to [q+1]$ be the coloring as given in Lemma 18. We define the retrieval scheme as follows.

(a) The user chooses a file index $\theta \in [q^2]$, a random index $\gamma \in [q + 1]$ and a permutation $\sigma : [q + 1] \to [q + 1]$ uniformly at random from the set of all permutations on the set $[q + 1]$.

(b) The user generates $N$ queries $Q_i$ for $i = 1, \ldots, N$ and send them to the $N$ servers, as follows. To server $N$ it sends the permutation $\sigma$, to server $\theta$ it sends $\gamma$, whereas to every other server $1 \leq i < N, i \neq \theta$ it sends $\sigma(c(B_{i,\theta}))$, the random color assigned to the block that contains $i$ and $\theta$ by the permutation $\sigma$ and the coloring $c$. Hence, the queries take the following form

$$Q_i := Q_i(\theta, \gamma, \sigma) = \begin{cases} \sigma(c(B_{i,\theta})) & 1 \leq i < N, i \neq \theta, \\ \gamma & i = \theta, \\ \sigma & i = N. \end{cases}$$

(c) Server $i$ for $i = 1, \ldots, N$ replies with an answer $A_i$ as follows. For $1 \leq i < N$, the server returns the value of the $Q_i$-th bit of the file $W_i$ it holds. On the other hand, server $N$ for each block $B \in B$ sends the sum of the $\sigma(c(B))$-th bits of all the files $W_j, j \in B$. Formally, the reply $A_i$ takes the form

$$A_i(Q_i) = \begin{cases} W_i(\sigma(c(B_{i,\theta}))) & 1 \leq i < N, i \neq \theta, \\ W_i(\gamma) & i = \theta, \\ \{\bigoplus_{j \in B} W_j(\sigma(c(B))) : \forall B \in B\} & i = N. \end{cases}$$
In what follows, we prove that the scheme has the claimed rate and it satisfies the privacy and reliability requirements.

**Rate:** Recall that the file size is \( q + 1 \) bits. Each of the first \( N - 1 \) servers returns a single bit, while server \( N \) returns \( |B| = q(q + 1) \) bits, therefore the rate of the scheme is

\[
\frac{q + 1}{(N - 1) + q(q + 1)} \geq \frac{1}{2\sqrt{N - 1} + 1} = \Omega\left(\frac{1}{N^{\frac{1}{2}}}\right),
\]

where we used the fact that \( q^2 = N - 1 \).

**Privacy:** Similarly to the privacy proof of Theorem 12, for every \( 1 \leq i \leq N - 1 \) the \( i \)-th query \( Q_i \) is uniformly distributed in \( [q + 1] \) regardless of the value of \( \theta \) and the coloring \( c \), since both \( \sigma \) and \( \gamma \) are chosen uniformly at random. For the same reason \( Q_N \) is independent of \( \theta \), hence the scheme is private.

**Reliability:** Let \( B_1, \ldots, B_{q+1} \) be the \( q + 1 \) blocks that contain \( \theta \). For any \( i \in [q+1] \) we will show that the user is able to retrieve the bit \( W_\theta(\sigma(c(B_i))) \). Note that when \( i \) ranges in \( [q+1] \), \( \sigma(c(B_i)) \) ranges also in \( [q + 1] \) (here we use the property that \( c \) is a proper block coloring with \( q + 1 \) colors), then the user is able to retrieve all the file bits, as needed.

From each server \( j \neq \theta \) such that \( j \in B_i \), the user receives the bit \( W_j(\sigma(c(B_i))) \), whereas from server \( N \) it receives the bit

\[
\bigoplus_{j \in B_i \setminus \{\theta\}} W_j(\sigma(c(B_i))) + \bigoplus_{j \in B_i} W_j(\sigma(c(B_i))) = W_\theta(\sigma(c(B_i))).
\]

Then, by summing these \( q \) bits, we have

\[
\bigoplus_{j \in B_i \setminus \{\theta\}} W_j(\sigma(c(B_i))) + \bigoplus_{j \in B_i} W_j(\sigma(c(B_i))) = W_\theta(\sigma(c(B_i))).
\]

Next, we address the case where the number of files \( K \) is not a prime power. By the Bertrand-Chebyshev theorem [15], let \( q \) be a prime in the range \( \sqrt{K} \leq q \leq 2\sqrt{K} \), and define the scheme with \( q^2 \) files, where the files \( W_i \) for \( i = K + 1, \ldots, q^2 \) are dummy files, i.e., \( W_i = 0 \). It is easy to verify that the rate of this scheme is also \( \Omega\left(\frac{1}{N^{\frac{1}{2}}}\right) \), as needed.

**Remark 19.** Although it is tempting to assume that one can not improve the scheme’s rate by allowing different servers to transmit different amounts of information, this does not hold. In other words, it is possible to improve the scheme rate dramatically for certain graphs by allowing varying quantities of \( H(A_i) \)'s for different servers. Indeed, the above scheme for the star graph achieves the rate of \( \Omega\left(\frac{1}{N^{\frac{1}{2}}}\right) \) with non-constant answer size \( H(A_i) \). On the other hand, if one restricts the servers to transmit the same amount of information, i.e., \( H(A_i) = H(A_j) \) for every \( i, j \in [N] \), then by Theorem 6 it follows that for any scheme of the star graph \( H(A_i) \geq L/2 \), and therefore the rate is at most \( 2/N \), which is clearly much less than \( \Omega\left(\frac{1}{N^{\frac{1}{2}}}\right) \).

## 5 Retrieval schemes for the complete graph PIR system

Understanding the complete graph’s PIR capacity is of special importance since this graph contains the maximum number of edges (files) for a given number of servers. Furthermore, any
graph $G$ on $N$ vertices is a subgraph of the complete graph $K_N$. Therefore any PIR scheme for $K_N$ can be converted into a scheme for $G$. In other words, it holds that $\mathcal{C}(K_N) \leq \mathcal{C}(G)$, which is a general lower bound for any graph $G$.

By Remark 2 the PIR capacity of $K_N$ is at most $\frac{2}{N+1}$. Whereas for the lower bound the following is known. In [3] it was shown that $\mathcal{C}(K_3) = \frac{1}{2}$ and $\mathcal{C}(K_4) \geq \frac{3}{10}$. For general $N$ we have $\mathcal{C}(K_N) \geq 1/N$. In this section, we describe a general scheme that slightly improves the rate of $1/N$. More precisely, we provide a scheme with rate $\frac{2N-1}{2N-1} \cdot \frac{1}{N}$. Indeed, the improvement vanishes exponentially fast as the number of servers increases. However, the purpose of this construction is to provide an indication that the bound of $1/N$ is not optimal, and the upper bound of $\frac{2}{N+1}$ might be achievable with a more intricate scheme.

We begin by describing the queries each server receives and the answer it provides in return; then, we explain how to generate the queries. Recall that for distinct $i, j \in [N]$, $W_{i,j}$ is the unique file stored on servers $i, j$, and assume also that each file is a binary vector of length $L = 2^{N-1}$.

**Queries:** Each server $j$ receives a bijection

$$\sigma_j : \{P : P \subseteq N(j)\} \rightarrow [L]$$

from the family of all subsets of its neighbors $N(j)$ to the set of integers $[L]$.

**Answers:** Server $j$ returns $L - 1$ bits, one for each nonempty subset of its neighbors, calculated as follows. For a nonempty subset $P \subseteq N(j)$ of its neighbors, the sum of all the $\sigma_j(P)$-th bits of files $W_{j,v}, v \in P$ is computed, i.e., the bit

$$b^j_P = \bigoplus_{v \in P} (W_{j,v})_{\sigma_j(P)}.$$  

**Generating the queries:** Assume that $W_{i,i'}$ is the requested file, and let $\Omega$ be the family of subsets of $[N]$ that contain exactly one of the servers $i$ or $i'$, i.e.,

$$\Omega = \{P \subseteq [N] : |P \cap \{i, i'\}| = 1\}.$$  

Clearly, $|\Omega| = L$, and let $\pi : \Omega \rightarrow [L]$ be a bijection selected uniformly at random from the set of all possible bijections from $\Omega$ to $[L]$. Next, we describe how to construct for each server $j$ its bijection $\sigma_j$ as in (35). First, for $j \neq i, i'$ let $P \in \Omega$ be a set that contains $j$ and define

$$\sigma_j(P \setminus \{j\}) = \pi(P).$$

Note that there are exactly $L/2$ such sets $P$. Extend $\sigma_j$ arbitrarily to a bijection as in (35), and note that the resulted bijection $\sigma_j$ is distributed uniformly among all the possible bijections. Therefore, server $j$ does not learn anything upon receiving its query.

Next, for a server $a \in \{i, i'\}$ let $\overline{a}$ be such that $\{a, \overline{a}\} = \{i, i'\}$. Let $P \in \Omega$ be a set that contains $a$ and define

$$\sigma_{a,\overline{a}}(\{a, \overline{a}\} \cup P \setminus \{a\}) = \pi(P),$$

(38)
and extend $\sigma_a$ arbitrarily to a bijection as in (35). Similarly, it is easy to verify that the resulted bijection $\sigma_a$ is distributed uniformly among all the possible bijections, and therefore, server $a$ does not learn anything upon receiving its query.

The following theorem shows that the above PIR scheme has the claimed properties.

**Theorem 20.** The above scheme is a PIR scheme with rate $\frac{2^{N-1} - 1}{N} \cdot \frac{1}{L}$.

**Proof:**

**Reliability:** We will show that the user can compute every bit of the file $W_{i,i'}$. For $l \in [L]$ let $P \in \Omega$ be the unique set such that $\pi(P) = l$ and let $a = P \cap \{i, i'\}$. Then, upon receiving the answers from the servers, the user can compute the following bit

$$\bigoplus_{j \in P \setminus \{a\}, j \neq a} b^j_{P \setminus \{j\}} \oplus b^a_{\{\pi\} \cup P \setminus \{a\}} = \bigoplus_{j \in P \setminus \{j\}} \bigoplus_{v \in \{\pi\} \cup P \setminus \{a\}} (W_{j,v})_{\pi_{\{j\}}} (W_{a,v})_{\pi_a} \sigma_a(\pi_{P \setminus \{a\}})$$

where (39) follows from (36); and (40) follows from (37), (38) and the fact that $\pi(P) = l$. Note that the sum in (40) contains twice each $l$-th bit of each file $W_{j,v}, j, v \in P$, and therefore is cancelled out, unless $\{j, v\} = \{a, \overline{a}\}$, where in this case it appears exactly once. Therefore, (40) is equal to $(W_{i,i'})_l$, as needed.

**Privacy:** As already mentioned, each server receives a permutation distributed uniformly among all the possible permutations of the sets of subsets of its neighbors. Therefore, there is no information leakage.

**Rate:** Each server return $L - 1 = 2^{N-1} - 1$ bits, hence the total amount of bits sent by all servers is $N \cdot (2^{N-1} - 1)$, whereas the size of the file is $L = 2^{N-1}$. Therefore, the rate is $\frac{2^{N-1} - 1}{N} \cdot \frac{1}{L}$, as needed.

The following example demonstrates how to construct the described scheme for $K_3$.

**Example 21.** Let $S = \{S_1, S_2, S_3\}$, $W = \{W_{1,2}, W_{1,3}, W_{2,3}\}$ and suppose that each file $W \in W$ is a vector in $\mathbb{F}_2^3$. Without loss of generality assume that the required file is $\theta = W_{1,2}$ meaning that $\{i, i'\} = \{1, 2\}$ and $\Omega = \{\{1\}, \{2\}, \{1, 3\}, \{2, 3\}\}$. Assume that the randomly selected bijection $\pi : \Omega \to [L]$ is the bijection

$$\pi(\{1\}) = 1, \pi(\{2\}) = 2, \pi(\{1, 3\}) = 3, \pi(\{2, 3\}) = 4.$$

Next, we define the bijections $\sigma_j$’s. Since $3 \neq i, i'$, then by (37)

$$\sigma_3(\{1\}) = \pi(\{1, 3\}) = 3, \sigma_3(\{2\}) = \pi(\{2, 3\}) = 4.$$

Next, for $\sigma_1$ note that $a = 1, \overline{a} = 2$, hence by (38)

$$\sigma_1(\{2\}) = \pi(\{1\}) = 1 \text{ and } \sigma_1(\{2, 3\}) = \pi(\{1, 3\}) = 3.$$
Similarly,

\[ \sigma_2(\{1\}) = 2 \text{ and } \sigma_2(\{1, 3\}) = 4. \]

Finally, each of \( \sigma_1, \sigma_2, \sigma_3 \) is arbitrarily extended to a bijection as in (35). Each server returns 3 bits, but the user uses only two of them to retrieve the file \( W_{1,2} \). The needed bits from the three bits server \( S_1 \) returns are

\[ b_{\{2\}}^1 = W_{1,2}(1) \text{ and } b_{\{2,3\}}^1 = W_{1,2}(3) \oplus W_{1,3}(3). \]  

(41)

Similarly, from server \( S_2 \), the user uses the bits

\[ b_{\{1\}}^2 = W_{1,2}(2) \text{ and } b_{\{1,3\}}^2 = W_{1,2}(4) \oplus W_{2,3}(4), \]

and from server \( S_3 \)

\[ b_{\{1\}}^3 = W_{1,3}(3) \text{ and } b_{\{2\}}^3 = W_{2,3}(4). \]

The user now can retrieve the file \( W_{1,2} \) since

\[ W_{1,2}(1) = b_{\{2\}}^1, W_{1,2}(2) = b_{\{1\}}^2, W_{1,2}(3) = b_{\{2,3\}}^1 \oplus b_{\{1\}}^3, \text{ and } W_{1,2}(4) = b_{\{1,3\}}^2 \oplus b_{\{2\}}^3. \]

6 Concluding remarks and open questions

In this paper, we studied the PIR capacity of graph-based replication systems with non-colluding servers. We proved several upper bounds that rely on the underlying graph structure, which turns out to be tight in certain cases. For lower bounds, we used edge-coloring to establish PIR schemes for star graphs, which are optimal up to a constant factor. Also, we improved the known PIR schemes for complete graphs, which imply an improved lower bound for all graphs.

Despite these results, the PIR capacity of a graph is a parameter that is still far from being well-understood. Hence we conclude with several open questions.

1) Improve the upper (Corollary 9) and lower (Section 5) bounds on the PIR capacity of complete graphs.

2) Is it possible to extend the PIR schemes for star graphs given in Section 4 to other graph families, such as bipartite graphs, trees, regular graphs?

3) The assumption that each file is stored on only two servers is too constraining; therefore, a natural question is whether the results given in this paper can be generalized to hypergraphs.
References

[1] A. Ambainis, “Upper bound on the communication complexity of private information retrieval,” in Automata, Languages and Programming. Berlin, Germany: Springer-Verlag, 1997, pp. 401–407.

[2] 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

[3] Zs. Baranyai, “On the factorization of the complete uniform hypergraph,” in Infinite and finite sets (Proc. Intern. Coll. Keszthely, 1973). Amsterdam, Netherlands: North-Holland, pp. 91–108, 1975.

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

[5] K. Banawan and S. Ulukus, “Private information retrieval from non-replicated databases,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), pp. 1272–1276, 2019.

[6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private information retrieval,” in Proc. 36th Annu. Symp. Found. Comput. Sci., 1995, pp. 41–50.

[7] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, “Private information retrieval,” Journal of the ACM, vol. 45, no. 6, pp. 965–981, 1998.

[8] T. H. Chan, S.-W. Ho, and H. Yamamoto, “Private information retrieval for coded storage,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), pp. 2842–2846, 2015.

[9] Z. Dvir and S. Gopi, “2-server PIR with sub-polynomial communication,” in Proc. 47th Annu. ACM Symp. Theory Comput., 2015, pp. 577–584.

[10] 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, Jan. 2017.

[11] W. Gasarch, “A survey on private information retrieval,” Bulletin of the EATCS, vol. 82, pp. 72–107, 2004.

[12] 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

[13] Z. Jia and S. A. Jafar, “On the asymptotic capacity of X-secure T-private information retrieval with graph-based replicated storage,” IEEE Trans. Inf. Theory, vol. 66, no. 10, pp. 6280–6296, Oct. 2020.
[14] K. Kutnar and D. Marušič, “Hamilton cycles and paths in vertex-transitive graphs - Current directions,” *Discrete Mathematics*, vol. 309, no. 17, pp. 5491–5500, 2009.

[15] S. Ramanujan, “A proof of Bertrand’s postulate,” *J. Indian Math. Soc.*, vol. 11, pp. 181–182, 1919.

[16] 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)*, pp. 1510–1517, 2020.

[17] N. Raviv, I. Tamo, and E. Yaakobi, “Private information retrieval in graph-based replication systems,” *IEEE Trans. Inform. Theory*, vol. 66, no. 6, pp. 3590–3602, 2020.

[18] A. Schrijver and A.E. Brouwer, “Uniform hypergraphs”, in A. Schrijver (Ed.), *Packing and covering in combinatorics (Study week “Stapelen en overdekken”, Amsterdam, The Netherlands, June 5-9, 1978)*, Mathematical Centre Tracts, pp. 39–73, 1979. Stichting Mathematisch Centrum.

[19] H. Sun and S. A. Jafar, “Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti,” *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1000–1022, Feb. 2018.

[20] H. Sun and S. A. Jafar, “The capacity of private information retrieval,” in *IEEE Globecom*, pp. 1–6, Dec. 2016.

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

[22] H. Sun and S. A. Jafar, “The capacity of symmetric private information retrieval,” *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 322–329, 2018.

[23] 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.

[24] Q. Wang and M. Skoglund, “Symmetric private information retrieval from MDS coded distributed storage with non-colluding and colluding servers,” *IEEE Trans. Inform. Theory*, vol. 65, no. 8, pp. 5160–5175, Aug. 2019.
A Proof of Lemma 10

For an arbitrary scheme $T$ and a desired file with index $\theta$ (stored on servers, say $k$ and $l$), we denote by $Q^T_i(\theta)$ and $A^T_i$ the query sent to the $i$-th server and its answer, respectively. Also, we denote by $Q^T$ the set of queries generated by the user.

Let $T$ be a PIR scheme for a vertex transitive graph $G$. For an automorphism $f \in \text{Aut}(G)$ of $G$, let $f(\theta)$ be the index of the unique file stored on servers $f(k), f(l)$. Given the scheme $T$ and the automorphism $f$, one can construct the scheme $T_f$ to retrieve the file with index $\theta$, as follows. The user sends to server $i$ the query $Q^T_f_i := Q^T_{f(i)}(f(\theta))$ and the automorphism $f$, which in return replies with the answer $A^T_f_i := A^T_{f(i)}$.

**Scheme analysis:** This scheme can be thought of as applying the scheme $T$ on the graph $f(G)$ with a desired file with index $f(\theta)$. Note that the graph $f(G)$ is simply a relabeling of every vertex $i$ by $f(i)$. Then, by the reliability of $T$, the user can recover the file with index $f(\theta)$, which is the unique file stored on vertices $f(k), f(l)$. Since we just relabeled the vertices, this is also the unique file stored on vertices $k, l$ in the graph $G$, i.e., the file with index $\theta$, as needed.

The privacy requirement follows since we simply ran the original scheme $T$, which is assumed to be private, and clearly, the rate of the scheme $T_f$ is the same as of the scheme $T$.

Next, we define below a scheme $T'$ that uniformly selects one of the schemes $T_f$ for $f \in \text{Aut}(G)$ and uses it to retrieve the desired file. The exact details are as follows.

(a) The user chooses a file $\theta \in [K]$, and an automorphism $f \in \text{Aut}(G)$ uniformly at random.

(b) The user generates the queries $Q'^T := Q^T_f$, and sends them to the servers together with the automorphism $f$.

(c) Each server $i$ replies with an answer $A^T_f_i$.

Clearly, $T'$ is a scheme having the same rate as $T$. Let $i, j$ be two servers; then by the transitivity of the automorphism group, there exists a $g \in \text{Aut}(G)$ with $g(j) = i$, then

$$H(A^T_i | Q'^T) = \frac{1}{|\text{Aut}(G)|} \sum_{f \in \text{Aut}(G)} H(A^T_{f(i)} | Q'^T)$$

$$= \frac{1}{|\text{Aut}(G)|} \sum_{f \in \text{Aut}(G)} H(A^T_{f(i)} | Q^T_f (f(\theta)))$$

$$= \frac{1}{|\text{Aut}(G)|} \sum_{f \in \text{Aut}(G)} H(A^T_{f(g(j))} | Q^T_f (f(\theta)))$$

$$= \frac{1}{|\text{Aut}(G)|} \sum_{f \in \text{Aut}(G)} H(A^T_j | Q^T_f (f(\theta)))$$

$$= \frac{1}{|\text{Aut}(G)|} \sum_{f \in \text{Aut}(G)} H(A^T_j | Q^T_f (f(\theta)))$$

23
\[= H(A_j^{T'} | Q^{T'}) , \]

where (42) follows since \( f \in \text{Aut}(G) \) is chosen uniformly, and (43) follows since if \( f \) ranges over all the automorphisms so does \( fg \). This proves (23). Similarly, it can be shown that for any file \( W_{i,j} \)

\[H(A_i^{T'} | W \setminus \{ W_{i,j} \}) = H(A_j^{T'} | W \setminus \{ W_{i,j} \}). \tag{44} \]

By Lemma 3, we have

\[H(A_i^{T'} | W \setminus \{ W_{i,j} \}) + H(A_j^{T'} | W \setminus \{ W_{i,j} \}) \geq L. \tag{45} \]

Combining (44) and (45) gives \( H(A_i^{T'} | W \setminus \{ W_{i,j} \}) = H(A_j^{T'} | W \setminus \{ W_{i,j} \}) \geq \frac{L}{2} \). This completes the proof.