Private Proximity Retrieval

Tuvi Etzion,* **Oliver W. Gnilke**,† **David Karpuk**,‡ **Eitan Yaakobi**,* and **Yiwei Zhang***

*Department of Computer Science, Technion — Israel Institute of Technology, Haifa, 3200009 Israel
†Department of Mathematical Sciences, Aalborg Universitet, Denmark
‡Departamento de Matemáticas, Universidad de los Andes, Bogotá, Colombia
{etzion,yaakobi,ywzhang}@cs.technion.ac.il, owg@math.aau.dk,
da.karpuk@uniandes.edu.co

***Abstract*—A *private proximity retrieval* (*PPR*) scheme is a protocol which allows a user to retrieve the identities of all records in a database that are within some distance r from the user's record x. The user's *privacy* at each server is given by the fraction of the record x that is kept private. The *distortion* of a PPR scheme measures how accurately the user can calculate the identities of the desired files. We assume that each server stores a copy of the database. This paper studies protocols that offer trade-offs between perfect privacy and low computational complexity and storage.**

In this paper, this study is initiated. The work focuses on the case when the records are binary vectors together with the Hamming distance. In particular, for a given privacy level, we investigate the minimum number of servers that guarantee a prescribed distortion value. The collusions of pairs of servers as well as other distance measures are investigated.

I. Introduction

The growing amount of available information, which mostly resides in the web and on the cloud, has made *information retrieval* (*IR*) one of the more important computing tasks. In fact, web search has become the standard source of information search and in general, IR refers to information access in order to obtain data, in any form, from any available information resources. However, this form of communication also poses a risk to the user privacy, since the servers can monitor the user's requests in order to deduce important information on the user and his interests. Therefore, an important aspect of IR is hiding the information the user is searching for.

Private information retrieval (PIR) is one of the well-known problems that provide privacy to user's requests. This problem was introduced by Chor et al. in [5]. PIR protocols make it possible to retrieve a data item from a database without disclosing any information about the identity of the item being retrieved. This problem has attracted considerable attention since its inception, see e.g. [6], [22]. The classic PIR model of [5] views the database as a collection of bits and assumes that the user wishes to retrieve the ith bit without revealing any information about the index i. This problem has received recently significant attention from an information-theoretic perspective, wherein the database consists of large records and the goal is to minimize the number of bits that are downloaded from the servers [2]. Since then, extensions of this model for several more setups have been rigorously studied; see e.g. [1], [7], [9], [18]–[20] and references therein.

Authors appear in alphabetical order. Research supported in part by the Israel Science Foundation (ISF) grant No. 1817/18 and the NSF-BSF grant No. 2016692. Y. Zhang was also supported in part by a Technion Fellowship. This research of T. Etzion, E. Yaakobi, and Y. Zhang was also partially supported by the Technion Hiroshi Fujiwara cyber security research center and the Israel cyber directorate. Part of this work was carried out while Oliver Gnilke was with the Department of Mathematics and Systems Analysis, Aalto University, Finland.

PIR protocols have been rigorously studied mostly for the basic IR problem when the user knows which data records are stored in the database (but not their content) and simply asks for the content of one of them. However, the user may be interested in other forms of information from the database rather than just asking for a particular record. The recently-introduced problem of *private computation* is a generalization of PIR which allows a user to compute an arbitrary function of a database, without revealing the identity of the function to the database. This problem has been studied for linear functions in [12], [15], [18] and for polynomial functions in [8], [17]. Many open questions about private computation remain, especially for non-linear functions, one of which is the topic of this paper.

Another important IR problem is that of *proximity searching*. One example of proximity searching is the *K-nearest neighbor search* (*K-NNS*), where the goal is to find the K elements from the database that minimize the distance to a given query [3], [11]. Proximity searching has several applications, among them are classification, searching for similar objects in multimedia databases, searching for similar documents in information retrieval, similar biological sequences in computational biology, and more. While proximity searching has been well-studied in the literature, to the best of our knowledge, there are no existing solutions which offer both proximity searching and privacy simultaneously. The assumption in proximity searching algorithms is that there is only a single server which stores the database. However, modern data storage systems are stored across several servers in a distributed manner, and thus we can take advantage of this setup in order to provide privacy for proximity retrieval.

In our setup of the problem we assume the user has a record and is interested in knowing the identities of the records in the database that are close to his record, according to some distance measure. This setup can fit to the case when the records stored in the database are attributes of different users. Given the user's attribute, he is only interested to know the users which have similar attributes to his according to some distance measure between the attributes. For example, a record may be a user's location and the database may consist of the locations of agents. In this context, the user seeks to determine the identity of the agents nearest to him, without necessarily knowing their exact location, while minimizing the amount of information that is exposed to the servers about his location. This example is related to the *private proximity testing* problem in which two users seek to determine whether they are close to each other, without revealing any information about each other's location [13], [14], [16]. Yet another example assumes that each record is a file (song, movie, DNA sequence etc.), and the user is interested in determining the records which are similar to his. There

are several more related problems to private proximity retrieval such as private keyword search [10], [21] and private search [4].

A. Setup

Let $\mathcal{V}$ be a finite set, assume an M-file database $\mathcal{X} = \{x_1, \ldots, x_M\} \subseteq \mathcal{V}$, where the x_i are i.i.d. samples of a random variable X, is stored on N different non-colluding servers. Let d be a metric on $\mathcal{V}$, $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geqslant 0}$. For $v \in \mathcal{V}$, $B(v, r)$ is its ball of radius r, that is, the set $B(v, r) \triangleq \{y \in \mathcal{V} \mid d(v, y) \leqslant r\}$. The user has a record $x \in \mathcal{V}$ chosen from X and seeks to know the identity of every record in $B(x, r) \cap \mathcal{X}$, *i.e.*, the identity of every record in the database similar to his, which is given by the set $I(x, r) \triangleq \{m \in \{1, \ldots, M\} |\ x_m \in B(x, r)\}$.

As opposed to the classical PIR problem, our solutions do not provide full privacy, but only partial privacy which will depend on the accuracy of calculating the set $I(x, r)$.

Definition 1. *Given the above setup, a **private proximity retrieval (PPR)** scheme consists of three algorithms.*

1) *A randomized algorithm $\mathcal{Q}$ that forms N queries, $q_1, \ldots, q_N$, depending on X, M, the user's record x, and the search radius r.*
2) *An algorithm $\mathcal{A}$ that calculates answers A_n, for $n \in [N]$, given a query q_n generated by the algorithm $\mathcal{Q}$, and the database $\mathcal{X}$.*
3) *An algorithm $\mathcal{R}$, that calculates an approximation $\widehat{I}(x, r)$ of the set $I(x, r)$, given X, x, r, the set of queries $q_1, \ldots, q_N$, and the corresponding answers $A_1, \ldots, A_N$.*

The user's privacy at server n is defined, roughly speaking, to be the uncertainty in X after the server sees the query q_n, relative to the entropy of X. That is,

$$P_n = \frac{H(X|q_n)}{H(X)}.$$

While the goal in our schemes is to accurately calculate the set $I(x, r)$, i.e., $\widehat{I}(x, r) = I(x, r)$, it may happen that it is only possible to approximately calculate the set, *i.e.*, $I(x, r) \subsetneq \widehat{I}(x, r)$. The distortion of a PPR scheme is the smallest integer ϵ s.t. $\widehat{I}(x, r) \subseteq I(x + \epsilon, r)$, in other words, $\widehat{I}(x, r)$ contains an index of at least one element of distance $r + \epsilon$ from x (where we now assume X to be uniform on $\mathcal{V}$ or at least all elements of $\mathcal{V}$ to have a positive probability).

Remark 1. It is possible to define a lower and an upper distortion ϵ_1, ϵ_2, via $I(x, r - \epsilon_1) \subseteq \widehat{I}(x, r) \subseteq I(x, r + \epsilon_2)$ where ϵ_1 and ϵ_2 are minimal. We reinterpret the approximation $\widehat{I}(x, r)$ as a solution to the search around x with radius $r' := r - \epsilon_1$. Then

$$I(x, r') \subseteq \widehat{I}(x, r') \subseteq I(x, r' + \epsilon_1 + \epsilon_2),$$

and it is therefore sufficient to study schemes that have lower distortion 0 and upper distortion $\epsilon = \epsilon_1 + \epsilon_2$.

We denote by $H(\cdot)$ the binary entropy function. For a discrete random variable Y supported on a set $\mathcal{Y}$, we define its entropy by $H(Y) = \mathbb{E}_Y \left[\log_2 \left(\frac{1}{Pr(y)}\right)\right] = -\sum_{y \in \mathcal{Y}} Pr(y) \log_2(Pr(y))$. It will always be clear from context whether the function $H(\cdot)$ is taking a scalar or a random variable as its argument. For a positive integer n, the set $\{1, 2, \ldots, n\}$ is denoted by $[n]$.

II. Basic Properties and Constructions of Private Proximity Retrieval Schemes

A. A Simple Error-Free Scheme for the Hamming Space

The main part of the current work concentrates on the case of $\mathcal{V} = \mathbb{F}_2^L$, d is the Hamming distance on $\mathbb{F}_2^L$, and the random variable X equal to the uniform distribution on $\mathbb{F}_2^L$, and this is the case to which we now specify unless stated explicitly otherwise. For two vectors u and v of the same length, the Hamming distance between u and v is denoted by $d_H(u, v)$ and the Hamming weight of u is denoted by $w_H(u)$. The support of u will be denoted by $\mathrm{supp}(u)$. For $s \geqslant 0$, let $\mathcal{W}_s$ be the set of all vectors of weight exactly s in $\mathbb{F}_2^L$, i.e., $\mathcal{W}_s = \{y \in \mathbb{F}_2^L \mid w_H(y) = s\}$. The following lemma will motivate our first example.

Lemma 2. *For all r, s, L such that $r + 2s + 1 \leqslant L$, and $x \in \mathbb{F}_2^L$, it holds that $B(x, r) = \bigcap_{z \in \mathcal{W}_s} B(x + z, r + s)$, and hence $I(x, r) = \bigcap_{z \in \mathcal{W}_s} I(x + z, r + s)$.*

Example 1. Lemma 2 suggests a PPR scheme for retrieving $I(x, r)$. Assign each vector $z \in \mathcal{W}_s$ to one server at random and send the query $q_z = (x + z, r + s)$. The server computes the set $A_z = I(x + z, r + s)$, and sends it as its response back to the user. Finally, the user computes the intersection of all responses and therefore the set $I(x, r)$, by Lemma 2. Hence this scheme has distortion $\epsilon = 0$. We discuss the privacy at each server in more generality in the next section. □

B. A General PPR Scheme and Some Basic Properties

While the PPR scheme described in Example 1 using the set $\mathcal{W}_s$ has the advantage of being easy to describe, the number of servers is $\binom{L}{s}$, which even for small values of L is unreasonable. Our general strategy for improving on this construction will be to consider subsets $Z \subseteq \mathcal{W}_s$ which satisfy, or approximately satisfy, the equation of Lemma 2, but for which $|Z| \ll |\mathcal{W}_s|$. We formalize the studied family of PPR schemes in the following definition.

Definition 3. *Given a search radius r, a parameter s, and a set of* query vectors *$Z \subseteq \mathcal{W}_s$ of size N, the PPR scheme $PPR(r, s, Z)$ is defined to consist of the following algorithms:*

1) *The algorithm $\mathcal{Q}$ applies a uniform random permutation of $[L]$ to the coordinates of all vectors $z_n \in Z$, and then sends the query $q_n = (x + z_n, r + s)$ to server $n = 1, \ldots, N$, where x is the user's record.*
2) *The algorithm $\mathcal{A}$ computes*

$$A_n = I(x + z_n, r + s) = \{m \in [M] \mid d(x + z_n, x_m) \leqslant r + s\}.$$

3) *The algorithm $\mathcal{R}$ computes*

$$\hat{I}(x, r) = \bigcap_{n \in [N]} A_n = \bigcap_{z_n \in Z} I(x + z_n, r + s).$$

*The matrix $\mathbf{M} \in \mathbb{F}_2^{N \times L}$ whose rows are the vectors $z \in Z$ is referred to as the **query matrix**.*

Proposition 4. *For the PPR scheme $PPR(r, s, Z)$, the privacy at any server $n \in [N]$ satisfies $P_n = \frac{\log_2 \binom{L}{s}}{L}$. In particular, if $\sigma = s/L$ is constant with respect to L, then $P_n \to H(\sigma)$ as $L \to \infty$.*

Let $P = P_n$ for the PPR scheme $PPR(r, s, Z)$, which by Proposition 4 is independent of n. It is not hard to show that

$P < H(\sigma)$, thus approximating $P \approx H(\sigma)$ for large L slightly overestimates the privacy level. Nevertheless, to maximize privacy, one wants σ to be as close as possible to $1/2$, so that $H(\sigma) \approx 1$. Using the scheme constructions which we will outline in the next few sections, we can obtain $H(\sigma) > 1 - \delta$ for any $\delta > 0$ and search radius $r = 0$. On the other hand, it will be shown in the sequel that it is impossible to attain privacy of exactly 1 for any PPR scheme of the form $PPR(r, s, Z)$.

Apart from its role in the above brief analysis of the privacy of $PPR(r, s, Z)$, the random coordinate permutation used by the algorithm $\mathcal{Q}$ is largely immaterial to the analysis of the scheme, and in what follows we will largely ignore it. To begin our analysis of the family $PPR(r, s, Z)$ of PPR schemes, we state several elementary but useful lemmas.

In case $s \geqslant (L-r)/2$ we have the following negative result on the distortion value.

Lemma 5. *If $s \geqslant (L-r)/2$, then $\epsilon = L - r$ for any choice of query vectors $Z \subseteq \mathcal{W}_s$.*

Proof: Let $\mathbf{1}$ be the all ones vector, and $z \in \mathcal{W}_s$. We see that $d(x + \mathbf{1}, x + z) = d(\mathbf{1}, z) = L - s \leqslant r + s$. Hence $x + \mathbf{1} \in \bigcap_{z \in \mathcal{W}_s} B(z, r + s)$ and the distortion for any $PPR(r, s, Z)$ scheme is maximal, i.e., *$\epsilon = L - r$.* ∎

By Lemma 5, the PPR scheme $PPR(r, s, Z)$ has non-trivial distortion only when $s \leqslant (L - r - 1)/2$, or equivalently, $\sigma \leqslant \frac{1}{2} - \frac{r+1}{2L}$. Hence $H(\sigma) < 1$ for any such scheme as long as we insist on non-trivial distortion, that is, we will always fall short of perfect privacy. On the other hand, by Proposition 4, we need s to be a sizable fraction of L to achieve a good privacy level. For the rest of the paper, we assume that $s \leqslant (L - r - 1)/2$. We will repeatedly use the following two lemmas to compute the distortion of a $PPR(r, s, Z)$ scheme.

Lemma 6. *For $y, z \in \mathbb{F}_2^L$ and $r \geqslant 0$, $d_H(z, y) \leqslant w_H(z) + r$ if and only if $\frac{w_H(y)-r}{2} \leqslant |\operatorname{supp}(y) \cap \operatorname{supp}(z)|$.*

Lemma 7. *For any set $Z \subseteq \mathcal{W}_s$ the distortion ϵ is even.*

Proof: Suppose there exists a vector $y \in \bigcap_{z \in Z} B(z, r + s)$ with weight $w_H(y) = r + \delta$ where δ is a positive odd integer. By Lemma 6, we have $\delta/2 \leqslant |\operatorname{supp}(y) \cap \operatorname{supp}(z)|$ for all $z \in Z$, and since the right-hand side is an integer, it follows that $(\delta + 1)/2 \leqslant |\operatorname{supp}(y) \cap \operatorname{supp}(z)|$. Now consider a vector y' of weight $w_H(y') = r + \delta + 1$ given by adding a single 1 to y in any coordinate not in $\operatorname{supp}(y)$. We have

$$\frac{w_H(y')-r}{2} = \frac{\delta+1}{2} \leqslant |\operatorname{supp}(y) \cap \operatorname{supp}(z)| \leqslant |\operatorname{supp}(y') \cap \operatorname{supp}(z)|$$

for all $z \in Z$, which by Lemma 6 shows that $y' \in \bigcap_{z \in Z} B(z, r + s)$. Thus there exists a vector $y' \in \bigcap_{z \in Z} B(z, r + s)$ with weight $r + \delta + 1$, and hence ϵ is even. ∎

C. Lower Bounds on the Number of Servers

Every set $Z \subseteq \mathcal{W}_s$ gives us a PPR scheme $PPR(r, s, Z)$ and according to Proposition 4 the privacy at each server of this scheme is $P = \frac{\log_2 \binom{L}{s}}{L}$. Thus, for a given distortion value ϵ we try to find a set Z of minimal size that guarantees this distortion level. This motivates us to study the following design problem.

Definition 8. *Let L, s, r, and ϵ be positive integers such that $\epsilon, s, r \leqslant L$. Let $N(L, s, r, \epsilon)$ be the minimal value of N such that there exists a set of query vectors $Z \subseteq \mathcal{W}_s$ of size N satisfying*

$$B(\mathbf{0}, r) \subseteq \bigcap_{z \in Z} B(z, r + s) \subseteq B(\mathbf{0}, r + \epsilon).$$

Clearly $N(L, s, r, \epsilon) \leqslant \binom{L}{s}$ for $s \leqslant (L - r - 1)/2$, since by Lemma 2 taking $Z = \mathcal{W}_s$ satisfies the above with the left-hand inclusion an equality. Our goal in this section is to prove lower bounds on the value of $N(L, s, r, \epsilon)$.

Lemma 9. *Suppose that $|Z| = 2$, so $Z = \{z_1, z_2\} \subseteq \mathcal{W}_s$. Then the distortion of $PPR(r, s, Z)$ satisfies $2s \leqslant \epsilon$. In particular if $s \geqslant \epsilon/2 + 1$ then $N(L, s, r, \epsilon) \geqslant 3$.*

Proof: Let $y \in \mathbb{F}_2^L$ be a vector of Hamming weight $2s + r$ such that $\operatorname{supp}(z_1) \cup \operatorname{supp}(z_2) \subseteq \operatorname{supp}(y)$. Then, $d_H(y, z_1) = d_H(y, z_2) = s + r$, and hence $y \in B(z_1, r + s) \cap B(z_2, r + s)$. Therefore $\epsilon \geqslant 2s$ and the result follows. ∎

Lemma 10. *Let $Z \subseteq \mathcal{W}_s$ be any set of query vectors of size $|Z| = N$ such that $PPR(r, s, Z)$ has distortion ϵ. Suppose that there exists a subset $S \subseteq [L]$ such that $|\operatorname{supp}(z) \cap S| \geqslant \epsilon/2 + 1$ for all $z \in Z$. Then, it holds that $|S| > r + \epsilon + 2$.*

Proof: Assume to the contrary that $|S| \leqslant r + \epsilon + 2$, and let y is a vector with $w_H(y) = r + \epsilon + 2$ such that $S \subseteq \operatorname{supp}(y)$. By construction, we have $|\operatorname{supp}(y) \cap \operatorname{supp}(z)| \geqslant \epsilon/2 + 1$ for all $z \in Z$. Thus, by Lemma 6 we have $y \in \bigcap_{z \in Z} B(z, r + s)$. But this contradicts the assumption that Z has distortion ϵ. ∎

Thus effective lower bounds on $N(L, s, r, \epsilon)$ will come from constructing small sets S which intersect the support of every $z \in Z$ in at least $\epsilon/2 + 1$ coordinates. The following theorem does exactly that by recursively finding columns of large weight in the query matrix $\mathbf{M}$ and repeatedly applying Lemma 10.

Theorem 11. *Suppose that $s \geqslant \epsilon/2 + 1$. Then the quantity $N(L, s, r, \epsilon)$ is lower bounded by*

$$N(L, s, r, \epsilon) \geqslant \max_{k=0,\ldots,r+\epsilon+1} \left\{ \left\lfloor \frac{r + 2 + \epsilon - k}{\epsilon/2 + (1 - \sigma)^k} \right\rfloor + 1 \right\}.$$

III. Upper Bounds on the Number of Servers

In this section we derive upper bounds on the value of $N(L, s, r, \epsilon)$ by explicitly constructing query matrices. Our first construction uses matrices of fixed row and column weight and achieves the following result.

Theorem 12.

1) *If $\sigma \leqslant \frac{1}{r+3}$ then $N(L, r, s, \epsilon) = \left\lfloor \frac{r}{\epsilon/2+1} \right\rfloor + 3$.*
2) *If $\sigma < \frac{\epsilon/2+1}{r+\epsilon+2} \triangleq \tau$, then $N(L, s, r, \epsilon) \leqslant \left\lfloor \frac{1}{\tau} \left\lceil \left(\frac{1}{\sigma} - \frac{1}{\tau}\right)^{-1} \right\rceil \right\rfloor + 1$.*

For the rest of this section we focus on the case $\epsilon = 0$ and for shorthand, we denote the value of $N(L, s, r, 0)$ by $N(L, s, r)$. In order to accomplish this task, a new family of codes, called *anti-covering codes*, is presented which will be used as a building block on the construction of sets Z that achieve zero distortion.

Given L, s, r we say that a length-L code $\mathcal{C}$ is an $(r, s)_L$ *anti-covering code* if

1) $\mathcal{C} \subseteq \mathcal{W}_s$,
2) for any length-L vector v such that $w_H(v) > r$, there exists a codeword $z \in \mathcal{C}$ such that $d_H(v, z) > r + s$.

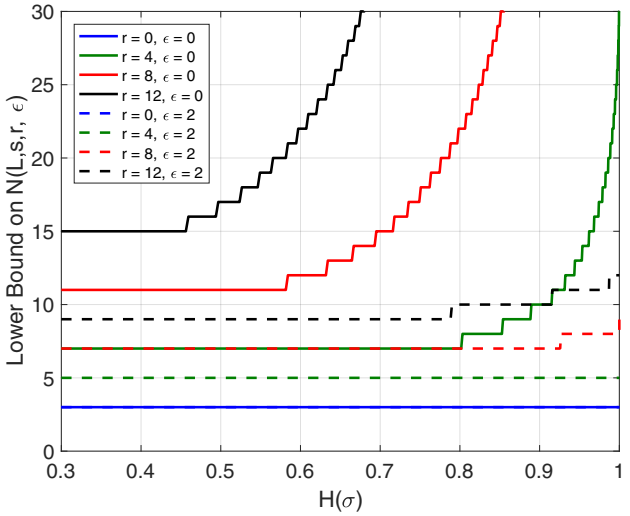

Fig. 1: The lower bound on $N(L,s,r,\epsilon)$ from Theorem 11 for $r = 0,4,8,12$ and $\epsilon = 0,2$ as a function of the asymptotic privacy level $H(\sigma)$, where $\sigma = s/L$.

For all L,s,r we denote by $\mathrm{C}^A(L,s,r)$ the minimum size of any $(r,s)_L$ anti-covering code. We see that for all L,s,r, it holds that $N(L,s,r) = \mathrm{C}^A(L,s,r)$, and based upon this relation we present our main construction of anti-covering codes in the next theorem. This construction provides an upper bound on the value of $N(L,s,r)$.

Theorem 13. *For all $2 \leqslant C \leqslant r+2$ and $\frac{L}{r+3} \leqslant s \leqslant \frac{L-r-3+C}{C}$ such that s is a multiple of $A = \left\lceil \frac{Cs\lceil \frac{r+3-C}{C} \rceil}{L-Cs} \right\rceil$ it holds that*

$$N(L,s,r) \leqslant C\binom{A+\lceil \frac{r+3-C}{C} \rceil}{A} = C\binom{\left\lceil \frac{Cs\lceil \frac{r+3-C}{C} \rceil}{L-Cs} \right\rceil + \lceil \frac{r+3-C}{C} \rceil}{\left\lceil \frac{Cs\lceil \frac{r+3-C}{C} \rceil}{L-Cs} \right\rceil}.$$

Proof Sketch: We sketch the proof for the case when $C = 2$ and r is odd. In this case it holds that $A = \lceil \frac{s(r+1)}{L-2s} \rceil$ and we assume that s is a multiple of A. First we generate $2A+r+1$ disjoint subsets of the set $[L]$, each of size $\frac{s}{A}$ (note that $(2A+r+1)\cdot\frac{s}{A} \leqslant L$). We denote these sets by $S_1, S_2, \ldots, S_{2A+r+1}$, and they are further partitioned into two families of sets $\mathcal{S}_1 = \{S_1, \ldots, S_{A+\frac{r+1}{2}}\}, \mathcal{S}_2 = \{S_{A+\frac{r+1}{2}+1}, \ldots, S_{2A+r+1}\}$. Next we define a set of N vectors $z_1, z_2, \ldots, z_N$ as follows. For every $1 \leqslant i_1 < i_2 < \cdots < i_A \leqslant A + \frac{r+1}{2}$, we let $z_{i_1,i_2,\ldots,i_A}$ be a vector whose support set is $S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_A}$. Similarly, for every $A + \frac{r+1}{2} + 1 \leqslant i_1 < i_2 < \cdots < i_A \leqslant 2A+r+1$ we let $z_{i_1,i_2,\ldots,i_A}$ be a vector whose support set is $S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_A}$. Hence, the number of vectors is $N = 2\binom{A+\frac{r+1}{2}}{A}$. The rest of the proof follows by proving that $\mathcal{C} = \{z_1, z_2, \ldots, z_N\}$ is an $(r,s)_L$ anti-covering code. The first property clearly holds and the second one is verified by proving that for any vector v of Hamming weight greater than r there exists $z \in \mathcal{C}$ such that $d_H(v,z) > r+s$. ∎

While the condition in Theorem 13 requires that s is a multiple A, for L large enough and $\sigma = s/L$ this condition is negligible so it is possible to conclude with the following corollary.

Corollary 14. *For all constant $\sigma = s/L < 0.5$, it holds that*

$$N(L,\sigma L,r) \leqslant \min_{2\leqslant C<1/\sigma} \left\{ C\binom{\left\lceil \frac{C\sigma}{1-C\sigma}\left\lceil \frac{r+3-C}{C}\right\rceil \right\rceil + \left\lceil \frac{r+3-C}{C}\right\rceil}{\left\lceil \frac{C\sigma}{1-C\sigma}\left\lceil \frac{r+3-C}{C}\right\rceil \right\rceil} \right\},$$

and for r large enough

$$N(L, s=\sigma L, r) \lesssim \min_{2\leqslant C<1/\sigma} \left\{ C\binom{\frac{1}{1-C\sigma}\cdot\frac{r}{C}}{\frac{C\sigma}{1-C\sigma}\cdot\frac{r}{C}} \right\}.$$

IV. Server Collusions

Previous sections assumed that the servers do not communicate with each other in an attempt to deduce more information about the user's vector of interest x. In this section, we analyze the privacy loss from pairs of servers colluding to determine the record x, and construct a scheme which is particularly resistant to pairwise server collusion.

In general, we let $T \subseteq [N]$ be a subset of servers of size t. The privacy with respect to $T = \{j_1, \ldots, j_t\}$ is given by

$$P_T = \frac{H(X|q_{j_1}, \ldots, q_{j_t})}{H(X)} = \frac{H(X|q_{j_1}, \ldots, q_{j_t})}{L}.$$

In this section, we study the case of two-server collusions. Remember that in our PPR scheme $PPR(r,s,Z)$, when a server receives a request z, it is possible to deduce that the user's record x is any vector which is of distance s from z, that is, any vector in $S(x+z,r) = \{y \in \mathbb{F}_2^L \mid d_H(x+z,y) = r\}$. The next lemma determines the privacy level based upon the size of the intersection of the support sets of their queries.

Lemma 15. *Let $(x+z_1, r+s)$ and $(x+z_2, r+s)$ be queries received by two colluding servers $T = \{j_1, j_2\}$, where $w_H(z_1) = w_H(z_2) = s$ and $|\mathrm{supp}(z_1) \cap \mathrm{supp}(z_1)| = t$. The privacy at these servers is $P_T = \frac{\log\binom{2(s-t)}{s-t}\binom{L-2(s-t)}{t}}{L}$. In particular, if $\sigma = s/L$, $\beta = t/s$ and $\gamma = 1-\beta$ are constants with respect to L, then $P_T \to 2\sigma\gamma + (1-2\sigma\gamma)H((\sigma\beta)/(1-2\sigma\gamma))$ as $L \to \infty$.*

Given a fixed number of servers N and a column weight $c < N$ we build the set Z as a matrix containing all vectors of length N and weight c as columns. We see that $L = \binom{N}{c}$ and $s = \binom{N-1}{c-1}$. Hence $\sigma = c/N$ and any two rows of Z share a support of size $t = \binom{N-2}{c-2}$ leading to $\beta = (c-1)/(N-1)$. Applying Lemma 15 shows that when L approaches infinity, the privacy when two servers collude is

$$P = 2\frac{c(N-c)}{N(N-1)} + \left(1 - 2\frac{c(N-c)}{N(N-1)}\right) H\left(\frac{c(c-1)}{N(N-1)-2c(N-c)}\right).$$

V. Generalization to other Metrics

The definitions for a private private proximity retrieval scheme given in Sections I are general for any type of database and any metric defined on it. Fortunately, other results can be generalized to other metrics and in particular to metrics which are based on *distance regular graphs* (DRG metrics in short). A metric is called a DRG metric if for any $v_1, v_2 \in \mathcal{V}$ such that $d(v_1,v_2) = k$, the number of vertices $v \in \mathcal{V}$ such that $d(u_1,v) = i$ and $d(u_2,v) = j$ is a constant c_{ijk} independent of the choice of u_1 and u_2. The *diameter* of the metric, denoted by L, is the maximum distance between two elements in $\mathcal{V}$. Furthermore, let r be the proximity's radius and s the search's radius parameters for the scheme. An immediate result from this definition is the following result which will be used in the sequel. It holds for DRG metric as well as other important metrics like the L_1 metric.

Lemma 16. *If $d(u_1, u_2) = k$ for some $u_1, u_2 \in \mathcal{V}$, then there exists a vertex $v \in V$ such that $d(u_1, v) = L$ and $d(u_2, v) = L - k$.*

Lemma 16 does not hold for all metrics and it is essential for generalizing the proof of Lemma 2 which motivated our approach. For an element $x \in \mathcal{V}$, let $\mathcal{W}_s^x$ denote the set of words in $\mathcal{V}$ of distance s from x, i.e. $\mathcal{W}_s^x \triangleq \{z : d(z, x) = s\}$.

Lemma 17. *For all r, s, L such that $L \geqslant r + 2s + 1$ and $x \in \mathcal{V}$, we have*

$$B(x, r) = \bigcap_{z \in \mathcal{W}_s^x} B(z, r + s).$$

There are many DRG metrics, but the most important and interesting ones, which can be used for the private proximity scheme, are the Hamming scheme (over any finite field), the Johnson scheme, and the Grassmann scheme.

The *Johnson scheme* $J(n, L)$ consists of all L-subsets of an n-set. The *Johnson distance* $d_J(x, y)$ between two L-subsets x and y is defined by, $d_J(x, y) \triangleq |x \setminus y|$.

The *Grassmann scheme* $\mathcal{G}_q(n, L)$ consists of all L-subspaces of an n-space over $\mathbb{F}_q$. The *Grassmann distance* $d_G(x, y)$ between two L-subspaces x and y is defined by, $d_G(x, y) \triangleq L - \dim(x \cap y)$.

For L, r, s, let us denote by $N_d(L, r, s)$ the minimum size of a set $\mathcal{D} \subseteq \mathcal{W}_s^x$ such that $B(x, r) = \bigcap_{z \in \mathcal{D}} B(z, r + s)$, in a scheme with distance measure d.

Lemma 18. *If $L \geqslant s(r+3)$ and d is the Hamming distance, then $N_d(L, r, s) = r + 3$.*

Lemma 19. *If $L \geqslant s(2r + 3)$ and d is the Johnson distance or the Grassmann distance, then $N_d(L, r, s) = 2r + 3$.*

VI. Comparison with Parallel Work

In this section we compare our scheme with both a trivial scheme, analogous to the trivial scheme in private information retrieval of downloading the entire database, and the scheme of Chen et al. [4].

We compare the three strategies with respect to the achieved privacy level P, the required storage, the upload cost, and the download cost. For the present scheme, we only consider the case where $\epsilon = 0$, since the other two schemes assume this condition.

In the trivial scheme, the user uses a single server which is storing the entire database $\mathcal{X}$. The user simply retrieves the indices of the files in $B(z, r) \cap \mathcal{X}$ for *every* $z \in \mathbb{F}_2^L$. In other words, the user sends 2^L queries to a single server, each of which is a bit string of length L.

The private search scheme of [4] applies not only for balls in Hamming space, but any set of subsets of a given set. In this scheme, every server stores 2^L vectors of length M. The PIR scheme requires the user to upload a binary vector of length 2^L to each server, and each server responds by transmitting a vector of length M. For the analysis done in [4] we must have $M \gg 2^L$. That is, the database contains an enormous amount of identical files.

We compare these two schemes with our scheme in Table I. In summary, our scheme loses to that of [4] in terms of privacy and download rate, but saves in terms of storage and upload cost. The download cost, that is, the total amount of downloaded data for both schemes, is approximately equal. Note lastly that if L is especially small, then the user is likely better off simply using the trivial scheme.

	Trivial Scheme	[4]	this work
Privacy P	1	1	$\approx H(s/L)$
Storage	ML	$NM2^L$	NML
Upload Cost	$L2^L$	$N2^L$	NL
Download Cost	$\log_2(M) \cdot \lvert\mathcal{X}\rvert \sum_{i=0}^{r} \binom{L}{i}$	NM	$N \log_2(M) \cdot A_z$

TABLE I: A comparison of the basic performance metrics of three Private Search schemes, where $A_z = \sum_{z \in \mathbb{F}_2^L} |B(x+z, r+s) \cap \mathcal{X}|$.

References

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

[2] T. H. Chan, S. Ho, and H. Yamamoto, "Private information retrieval for coded storage," arXiv preprint arXiv:1410.5489 (2014).

[3] E. Chavez, K. Figueroa, and G. Navarro, "Effective proximity retrieval by ordering permutations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1647–1658, 2008.

[4] Z. Chen, Z. Wang, and S. Jafar, "The asymptotic capacity of private search", *Proc. IEEE Int. Symp. on Inform. Theory*, pp. 2122–2126, Vail, CO, 2018.

[5] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, 45, 1998. Earlier version in FOCS 95.

[6] Z. Dvir, and S. Gopi, "2-server pir with sub-polynomial communication," arXiv:1407.6692v1, Jul. 2014.

[7] R. Freij-Hollanti, O.W. Gnilke, C.Hollanti, and D.A. Karpuk, "Private information retrieval from coded databases with colluding servers," arXiv:1611.02062v3, Aug. 2017.

[8] D. Karpuk, "Private computation of systematically encoded data with colluding servers," *Proc. IEEE Int. Symp. on Inform. Theory*, pp. 2122–2126, Vail, CO, 2018.

[9] S. Kumar, H.-Y. Lin, E. Rosnes, A. Graell i Amat, "Achieving maximum distance separable private information retrieval capacity with linear codes," arXiv:1712.03898v3, Aug. 2018.

[10] M. Li, S. Yu, N. Cao and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," *2011 31st Int. Conf. on Distributed Computing Systems*, pp. 383–392, Minneapolis, MN, 2011.

[11] Y.A. Malkov and D.A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," arXiv:1603.09320, Mar. 2016.

[12] M. Mirmohseni and M.A. Maddah-Ali, "Private function retrieval," arXiv:1711.04677v2, Nov. 2017.

[13] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing," *Proc. Network and Distributed System Security Symp.*, 2011.

[14] J.D. Nielsen, J.I. Pagter and M.B. Stausholm, "Location privacy via actively secure private proximity testing," *IEEE Int. Conf. on Pervasive Computing and Comm. Workshops*, pp. 381–386, Lugano, 2012.

[15] S.A. Obead, H.-Y. Lin, E. Rosnes and J. Kliewer, "Capacity of private linear computation for coded databases," arXiv:1810.04230, Oct. 2018.

[16] C. Patsakis, P. Kotzanikolaou, and M. Bouroche, "Private proximity testing on steroids: AnNTRU-based protocol," *Security and Trust Management*, pp. 172–184, 2015.

[17] N. Raviv and D. Karpuk, "Private polynomial computation from Lagrange encoding," arXiv:1812.04142, Dec. 2018.

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

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

[20] Q. Wang and M. Skoglund, "Linear symmetric private information retrieval for MDS coded distributed storage with colluding servers," arXiv:1708.05673, Aug. 2017.

[21] Y. Yang, H. Lu and J. Weng, "Multi-user private keyword search for cloud computing," *IEEE Third Int. Conf. on Cloud Computing Technology and Science*, pp. 264–271, Athens, 2011.

[22] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *Joural ACM*, vol. 55, no. 1, pp. 1–16, 2008.