Private Proximity Retrieval Codes

Yiwei Zhang, Eitan Yaakobi, *Senior Member, IEEE*, and Tuvi Etzion, *Fellow, IEEE*

***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. In this paper, this research is initiated and protocols that offer trade-offs between privacy, computational complexity, and storage are studied. In particular, we assume that each server stores a copy of the database and study the required minimum number of servers by our protocol which provides a given privacy level. Each server receives a query in the protocol and the set of queries forms a code. The main focus in the paper is dedicated to studying the family of codes generated by the set of queries. These codes will be shown to satisfy a specific covering property and will be called *private proximity retrieval intersection covering codes*. In particular, since the query every server receives is a codeword, the goal is to minimize the number of codewords in such a code which is the minimum number of servers required by the protocol. These codes are closely related to a family of codes known as *covering designs*. We introduce several lower bounds on the sizes of such codes as well as several constructions. This work focuses on the case when the records are binary vectors together with the Hamming distance. Other metrics such as the Johnson metric are also investigated.**

***Index Terms*— Private information retrieval, covering designs, proximity searching, private proximity retrieval codes.**

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.

A. Private Information Retrieval, Private Computation, and Proximity Searching

Private information retrieval (PIR) is one of the well-known studied problems that provide privacy to user's requests. This problem was introduced by Chor, Goldreich, Kushilevitz, and Sudan in [9]. 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 [3]–[7], [10], [39]–[42]. The classic PIR model of [9] views the database as a collection of bits (or records) and assumes that the user wishes to retrieve the ith record without revealing any information about the index i. A naive solution is to download the entire database, and in fact this solution cannot be improved upon if the database is stored on a single server [9].

In order to achieve sublinear communication complexity it was proposed in [9] to *replicate the database* on several servers that do not communicate with each other. One of the disadvantages of this PIR model is that every record consists of a single bit, while in practice a record is much longer. As a consequence of this model's drawback, this problem has received recently significant attention from an information-theoretic perspective, wherein the database consists of large records (the number of bits in each file is much larger than the number of files) and the goal is to minimize the number of bits that are downloaded from the servers [6], [7]. This reformulation of the problem introduces the *rate* of a PIR scheme to be the ratio between the size of the file and the total number of downloaded bits from all servers [33]. Since then, extensions of this model for several more setups have been rigorously studied; see e.g. [1], [2], [15], [20], [33], [34], [36], [38] and references therein.

PIR protocols have been rigorously studied mostly for the very basic IR problem which assumes that 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.

Manuscript received April 27, 2020; revised March 24, 2021; accepted July 18, 2021. Date of publication August 3, 2021; date of current version October 20, 2021. This work was supported in part by the Technion Hiroshi Fujiwara Cyber Security Research Center and in part by the Israel Cyber Directorate. The work of Y. Zhang was supported in part by the Israel Science Foundation (ISF) under Grant 1817/18, in part by the Israel Binational Science Foundation (BSF)-NSF under Grant 2016692, in part by the Technion Fellowship, and in part by the National Natural Science Foundation of China under Grant 12001323. The work of E. Yaakobi was supported in part by the Israel Science Foundation (ISF) under Grant 1817/18. The work of T. Etzion was supported in part by the BSF-NSF under Grant 2016692. An earlier version of this paper was presented at the 2019 IEEE International Symposium on Information Theory (ISIT 2019) [12]. *(Corresponding author: Tuvi Etzion.)*

Yiwei Zhang is with the Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education and the School of Cyber Science and Technology, Shandong University, Qingdao, Shandong 266237, China, and also with the Department of Computer Science, Technion–Israel Institute of Technology, Haifa 3200003, Israel (e-mail: ywzhang@sdu.edu.cn).

Eitan Yaakobi and Tuvi Etzion are with the Department of Computer Science, Technion–Israel Institute of Technology, Haifa 3200003, Israel (e-mail: yaakobi@cs.technion.ac.il; etzion@cs.technion.ac.il).

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

Digital Object Identifier 10.1109/TIT.2021.3102177

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 [25], [29], [35] and for polynomial functions in [18], [31]. Many open questions about private computation remain, especially for non-linear functions, one of which is the topic of this paper. However, we note that in the setup studied in the paper we are mostly concerned with the indices of the records stored in the database rather than the content of the records themselves.

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 [8], [22]. 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.

B. Private Proximity Retrieval

In our setup of the problem we assume the user has a record and is ***only*** interested in knowing the identities, i.e., the ***indices***, of the records in the database that are close to his record. The measure of closeness is determined according to some distance measure. This setup can fit for example to the case when the records stored in the database are attributes of different users. Given the user's attributes, 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 locations, 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 [27], [28], [30]. 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.

Assume an M-file database $\mathcal{X} = \{x_1, \ldots, x_M\} \subseteq \mathcal{V}$ is stored on N non-colluding servers and the user has a record $x \in \mathcal{V}$. Let $B(x, r)$ be the ball of radius r centered at x according to some distance measure d on $\mathcal{V}$. The user seeks to know the identity of every record in $B(x, r) \cap \mathcal{X}$, i.e., the index 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. Thus, it is required that this computation is invoked while minimizing the amount of information the servers learn about x.

A *private proximity retrieval* (*PPR*) scheme consists of three steps. First, the user generates, with some degree of randomness, N queries which are sent to the N servers. Based upon these queries and the database $\mathcal{X}$, each server responds back with an answer to the user, and finally the user calculates the set $I(x, r)$. The user's privacy at each server is defined to be the fraction of information bits of x which are kept private from the server. The schemes we study in this paper have a common structure. The query sent to the nth server is given by $q_n = (y_n, \rho_n)$, where y_n is an element of $\mathcal{V}$ and ρ_n is a distance parameter, and the server's answer is the set of indices given by $A_n = I(y_n, \rho_n)$. Finally, the user computes the set $I(x, r) = \bigcap_{n \in [N]} A_n = \bigcap_{n \in [N]} I(y_n, \rho_n)$. The main goal of our work is studying the trade-offs between the search radius r, the privacy at each server, and the number of servers. In general, one can achieve more privacy at the expense of a smaller search radius or the use of more servers. Finally, we note that albeit taking the intersection of all response sets of the indices seems like a naive solution, we find this approach to be the basic and straightforward one that should be well studied under the scope of this problem. Clearly, other decision algorithms, which take for example the number of times each index appears in the sets, can also be of interest, however this is out of the scope of this paper.

Remark 1: In the protocols studied in this paper the servers respond *only* with the *indices* of the records that satisfy their queries, and *not* the records themselves. Once the user receives all indices he can then easily determine the required ones, i.e., the set $I(x, r)$, by simply intersecting all sets of indices. Note that this solution is appropriate in case the user is only interested to know whether the database contains records that are close to his record and/or the indices of these records. Moreover, in case the user is interested to receive the records themselves, he can apply any of the classical private information retrieval protocols in order to retrieve each one of the records which their indices were sent back by all servers. Since the size of the indices is significantly smaller than the records size, the amount of information that is downloaded from the servers is negligible. Motivated by this fundamental difference with respect to all classical PIR protocols that require to download the records, we observe that studying the so-called *download rate* is no longer the figure of merit of interest under this setup. Furthermore, the upload cost in all of the proposed schemes is always the same since every server receives only one record and a radius which in size is again negligible. Hence, this paper focuses on the more important and practical aspect of studying the trade-off between the number of servers and the user privacy of his record.

Remark 2: It is worth noting that private proximity retrieval can be viewed as another instance of the general private computation problem, in which $I(x, r)$ is the function to be computed on $\mathcal{X}$. Here the user wishes to hide the identity of this function amongst all functions of the form $I(y, r)$ for any $y \in \mathcal{V}$. However, the techniques developed in this paper are

designed to suit specifically to the problem of PPR and, albeit possible, we do not necessarily see them generalize to other private computation problems.

While the study of PPR schemes can be investigated for arbitrary space $\mathcal{V}$ and distance measure d on $\mathcal{V}$, the primary focus of this paper is on the case $\mathcal{V} = \mathbb{F}_2^L$, where d is the Hamming distance on $\mathbb{F}_2^L$. Then, the tradeoff between the number of servers and the privacy can be translated to the problem of finding the smallest subset C of $\mathcal{W}_s \triangleq \{y \in \mathbb{F}_2^L \mid w(y) = s\}$ such that

$$B(\mathbf{0}, r) = \bigcap_{c \in C} B(c, r+s),$$

where s is a design parameter of the PPR scheme which determines its privacy and $w(y)$ is the Hamming weight of y. Such a subset C will be called an (L, s, r) *Private Proximity Retrieval Intersection Covering (PPRIC) code*, and the minimum size of C will be denoted by $N(L, s, r)$. Our primary interest in the paper is to determine exactly the value of $N(L, s, r)$ for a large range of the parameters L, s, r by providing tight lower and upper bounds on this value. An important connection in studying the value of $N(L, s, r)$ is derived by showing that PPRIC codes are closely related to a well-studied problem in design theory, called *covering design*. Namely, an (n, k, t) covering design, $n \geqslant k \geqslant t > 0$, is a collection of k-subsets (called *blocks*) of an n-set such that each t-subset of the n-set is contained in at least one block of the collection. The minimum size of an (n, k, t) covering design is denoted by $c(n, k, t)$. We will show that PPRIC codes are also covering codes which have some additional constraints. Thus, the existence of a PPRIC code implies a corresponding covering design, but the opposite direction does not always hold. Therefore, all lower bounds on covering designs hold also for PPRIC codes and yet in many cases the constructions of covering designs are still valid for PPRIC codes. For the remaining parameters we show how to close on the gap by the results that can be derived by covering designs and present tighter lower bounds and specific code constructions. In particular, our main new results concerning the minimum size of (L, s, r) PPRIC codes are summarized as follows.

1) If $L \geqslant (r+3)s$, then $N(L, s, r) = r + 3$. A related interesting result is derived for related codes with the Johnson distance.
2) If $L \geqslant 2s + r + 1$, then $N(L, s, r) \geqslant c(L, L - s, r + 2)$.
3) Three constructions for PPRIC codes which are similar to the ones for covering designs. Some constructions are identical, but the required different proofs (from the ones for covering designs) that the constructed codes are indeed PPRIC codes, are given.
4) Proving that some lower bounds on PPRIC codes are larger than the related ones on covering designs. Specifically, if $\frac{17}{8} \leqslant \frac{L}{s} < \frac{9}{4}$ then $N(L, s, 0) = 6$ and if $\frac{L}{s} < \frac{17}{8}$ then $N(L, s, 0) > 6$.

C. Paper Organization

The rest of this paper is organized as follows. In Section II, the PPR problem is formally defined and a specific PPR scheme is presented that will be studied throughout the paper. Basic properties of this PPR scheme are presented when the records are binary vectors with the Hamming distance and it is studied how to achieve a high level of privacy. However, this solution requires a significantly large number of servers. This leads us to the main problem studied in the paper in which the minimum number of servers that provide a given privacy level is investigated. Lower bounds on this value are studied in Section III. Upper bounds by specific constructions are given in Section IV. The analysis of these bounds is given in Section V. In Section VI, we generalize our results beyond the Hamming metric to distance measures based on distance regular graphs such as the Johnson graph. Finally, Section VII concludes the paper with some more observations and directions for future research.

II. Private Proximity Retrieval Schemes and Basic Properties

A. Basic Definitions

Assume a database $\mathcal{X}$ consisting of M records is stored on N servers, where each server stores the whole database. Assume further that a user has a record x and wants to know which records in the database are similar to his record, that is, within some distance r with respect to a given metric. The user wants to keep the contents of his record as private as possible, i.e., reduce the amount of information which is revealed to the servers about his record.

Let $\mathcal{V}$ be a finite set and let d be a metric on $\mathcal{V}$, $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geqslant 0}$. For $x \in \mathcal{V}$, we define by $B(x, r)$ its ball of radius r, that is, the set $B(x, r) \triangleq \{y \in \mathcal{V} \mid d(x, y) \leqslant r\}$. Let X be a random variable on $\mathcal{V}$ and let $\mathcal{X} = (x_1, \ldots, x_M)$ consist of M independent and identically distributed (i.i.d. in short) samples of $\mathcal{X}$. We refer to $\mathcal{X}$ as the *database*, and assume a copy of $\mathcal{X}$ is stored on each of the N servers. A user has his own independent sample x of X, and wishes to exactly compute the set

$$I(x, r) \triangleq \{m \in \{1, \ldots, M\} \mid x_m \in B(x, r)\}.$$

In other words, the user wishes to answer the question "Is $x_m \in B(x, r)$?" for all x_m in the database. We refer to r as the ***search radius*** of the user.

We denote by $H(\cdot)$ the binary entropy function, given by $H(p) = -p \log_2 p - (1-p) \log_2 (1-p)$ for $0 \leqslant p \leqslant 1$. 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 the 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 $[n]$.

Definition 1: Let $\mathcal{X} = (x_1, \ldots, x_M)$ be a database of M records sampled i.i.d. from a random variable X on a metric space $\mathcal{V}$. Suppose that $\mathcal{X}$ is replicated on N non-communicating servers, and that a user has his own record $x \in \mathcal{V}$ and a desired search radius $r > 0$. A **private**

proximity retrieval (**PPR**) scheme for this setup 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 exactly the set $I(x, r)$, given X, x, r, the set of queries $q_1, \ldots, q_N$, and the corresponding answers $A_1, \ldots, A_N$.

Definition 2: For a given PPR scheme, the user's **privacy** at a server $n \in [N]$ is defined to be

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

The user's privacy at server $n \in [N]$ measures the fraction of bits of x which is kept private from server n. We note that an alternative definition for the privacy can use the mutual information $I(X; q_n) = H(X) - H(X|q_n) = (1 - P_n)H(X)$ between the user's record and the query sent to the server. While we see these two definitions equivalent, we chose the one presented in Definition 2 in order to reflect the *fraction* of the bits of the user's record that remains private. Note that our definitions do not demand that a PPR scheme has perfect privacy, and a major theme of this work will be the trade-off between the privacy P_n, the search radius r, and the number of servers N required by the PPR schemes. Lastly we comment here that a more general definition of this problem, which takes also into account the distortion of computing the set $I(x, r)$, has been presented in a research which has motivated the current paper (see [12]).

Remark 3: The assumption under the model studied in the paper is that all records stored in the database as well as the user's record are i.i.d. . We follow this setup for the simplicity of the model and also to be consistent with the models studied in PIR problems which carry the same assumption on the records stored in the database and the requested record by the user. Extending this model for other cases in which this assumption no longer holds and specifically when the user's record depends on the records stored in the database is an interesting problem by itself which is out of the scope of this work and is thus left for future research.

B. A Simple Scheme for 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 whose distribution is the uniform distribution over $\mathbb{F}_2^L$, 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 $d(u, v)$ and the Hamming weight of u is denoted by $w(u)$. 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., $\mathcal{W}_s = \{y \in \mathbb{F}_2^L \mid w(y) = s\}$. The following proposition will motivate our approach.

Proposition 3: Given $r, s, L \geqslant 0$ such that $r < L$ and $x \in \mathbb{F}_2^L$, we have that

$$B(x, r) = \bigcap_{z \in \mathcal{W}_s} B(x + z, r + s) \tag{1}$$

if and only if $r + 2s + 1 \leqslant L$.

Proof: Assume first that $r + 2s + 1 \leqslant L$. By translation invariance of the Hamming distance, we can assume that $x = \mathbf{0}$. Let $y \in B(\mathbf{0}, r)$, i.e., y is a vector whose Hamming weight is at most r. Clearly, $y \in B(z, r + s)$ for all $z \in \mathcal{W}_s$, since $d(z, y) \leqslant w(z) + w(y) \leqslant s + r$. Hence, $B(\mathbf{0}, r) \subseteq \bigcap_{z \in \mathcal{W}_s} B(z, r + s)$.

Now let $w(y) > r$, and without loss of generality we assume that the support of y is exactly the first $w(y)$ coordinates. Let $z \in \mathcal{W}_s$ be the vector with support on the last s coordinates. We see that

$$\begin{aligned} d(y, z) &= w(y) + w(z) - 2|\,\mathrm{supp}(y) \cap \mathrm{supp}(z)| \\ &= w(y) + s - 2\max\{0, w(y) + s - L\}. \end{aligned}$$

Now if $w(y) \leqslant L - s$, *i.e.*, the vectors y and z share no support, it holds that $d(y, z) = w(y) + w(z) > r + s$ and hence $y \notin B(z, r + s)$. On the other hand, if $y > L - s$ it holds that $d(y, z) = w(y) + w(z) - 2(w(y) + s - L) = 2L - w(y) - s \geqslant L - s > r + s$, where the last equation follows from $L > r + 2s$. Hence, $y \notin B(z, r + s)$.

For the other direction, let $B(x, r) = \bigcap_{z \in \mathcal{W}_s} B(x + z, r + s)$ and assume for contradiction that $r + 2s + 1 > L$. We will exhibit a vector y which is in $\bigcap_{z \in \mathcal{W}_s} B(x + z, r + s)$ but not in $B(x, r)$, which will lead to a contradiction. Given x, consider the all-ones vector $\mathbf{1}$ and let $y = x + \mathbf{1}$. Clearly $y \notin B(x, r)$ since $d(x, y) = L > r$. One can easily verify that $d(x + z, y) = d(z, \mathbf{1}) = L - s \leqslant r + s$ and hence y is contained in $B(x + z, r + s)$ for all $z \in \mathcal{W}_s$. Thus, $r + 2s + 1 \leqslant L$. ■

Proposition 3 suggests that to construct a PPR scheme for retrieving $I(x, r)$, we should apply a PPR scheme wherein the user randomly picks a different translation vector $z \in \mathcal{W}_s$ for each of $N = |\mathcal{W}_s| = \binom{L}{s}$ servers, and sends each server the vector $x + z$. The queries are of the form $(x + z, r + s)$ for all $z \in \mathcal{W}_s$, and are transmitted to a set of servers indexed by the elements of $\mathcal{W}_s$. Thus, the server corresponding to an element $z \in \mathcal{W}_s$ will respond with the indices $I(x + z, r + s)$. By Proposition 3, we have

$$I(x, r) = \bigcap_{z \in \mathcal{W}_s} I(x + z, r + s), \tag{2}$$

and hence the user can compute $I(x, r)$ exactly. The next example demonstrates the ideas of the proposed scheme.

Example 1: Assume that $M = 8, L = 8, r = 1, s = 2$, the user record is the zero vector $\mathbf{0} \in \mathbb{F}_2^8$, and the database is $\mathcal{X} = (x_1 = 10000000, x_2 = 01110001, x_3 = 00100000, x_4 = 00100100, x_5 = 00010000, x_6 = 00011101, x_7 = 01101010, x_8 = 00000011)$. Hence, the user is interested in computing the set

$$I(\mathbf{0}, 1) = \{m \in \{1, \ldots, 8\} \mid x_m \in B(\mathbf{0}, 1)\} = \{1, 3, 5\}.$$

Following the scheme proposed by Proposition 3, we use the set $\mathcal{W}_2 = \{y \in \mathbb{F}_2^8 \mid w(y) = 2\}$ and $|\mathcal{W}_2| = 28$ servers.

Each server is assigned with a query of the form $(z, 1+2=3)$ where $z \in \mathcal{W}_2$ and transmits the set of indices in the set $I(z,3)$. One can verify that $B(\mathbf{0},1) = \bigcap_{z\in\mathcal{W}_2} B(z,3)$ and thus $I(\mathbf{0},1) = \bigcap_{z\in\mathcal{W}_2} I(z,3)$. The privacy at each server is $\frac{\log_2\binom{8}{2}}{8} \approx 0.6$, since every server only knows that the distance between its query and the user's record is exactly two.

However, the number of servers N for such a scheme is unacceptably large. With the intention of reducing the number of servers but retaining the general scheme outline, we introduce the following family of PPR schemes.

C. A General PPR Scheme and Some Basic Properties

While the PPR scheme described in the previous subsection 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 the equation of Proposition 3, but for which $|Z| \ll |\mathcal{W}_s|$. We formalize the family of PPR schemes we will consider in the following definition.

Definition 4: Let $\mathcal{X} = (x_1, \ldots, x_M)$ be a database of M records sampled i.i.d. from a random variable X on a metric space $\mathcal{V}$. Suppose that $\mathcal{X}$ is replicated on N non-communicating servers, and that a user has his own private record $x \in \mathcal{V}$, which is chosen uniformly at random over $\mathcal{V}$. The user also has a desired search radius $r > 0$, another integer parameter $s > 0$, and a set of *query vectors* $Z \subseteq \mathcal{W}_s$ of size N, all of which are public knowledge.

The PPR scheme $PPR(r,s,Z)$ associated with this setup 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
$$\begin{aligned} A_n &= I(x + z_n, r + s) \\ &= \{m \in [M] \mid d(x + z_n, x_m) \leqslant r + s\}. \end{aligned}$$
3) The algorithm $\mathcal{R}$ computes
$$I(x,r) = \bigcap_{n\in[N]} A_n = \bigcap_{n\in[N]} I(x + z_n, r + s).$$

Proposition 5: 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 $L \to \infty$.

Proof: The coordinate permutation used in the algorithm $\mathcal{Q}$ guarantees that any single server observes an element $x+z$ which is chosen uniformly at random amongst all vectors of $\mathbb{F}_2^L$ which are at distance s from x, i.e., the set $x + \mathcal{W}_s$. Therefore $H(X|q_n) = \log_2|\mathcal{W}_s|$. Since x is a sample of the uniform distribution on $\mathbb{F}_2^L$, we have $H(X) = L$ and the result holds. In case $\sigma = s/L$, computing the limit as $L \to \infty$ is straightforward and may be done using Stirling's approximation. ■

Let $P = P_n$ for the PPR scheme $PPR(r,s,Z)$, which by Proposition 5 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 $H(\sigma) \approx 1$. 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.

From the definition of the PPR scheme $PPR(r,s,Z)$, we see that it is possible to successfully compute the desired set of indices $I(x,r)$ if the set Z satisfies $I(x,r) = \bigcap_{z\in Z} I(x+z, r+s)$. Furthermore, the scheme uses $N = |Z|$ servers and obtains an asymptotic privacy level of $H(\sigma)$ where $\sigma = s/L$. Hence, for given values of L, s, r, the goal is to find a set Z of minimum size, which translates to the number of servers required by the PPR scheme $PPR(r,s,Z)$. For the rest of the paper we study this value which will be denoted by $N(L,s,r)$. That is, $N(L,s,r)$ is the minimum size of a set Z such that for all $x \in \mathcal{V}$, it holds that $I(x,r) = \bigcap_{z\in Z} I(x+z, r+s)$. The main ideas of this scheme are demonstrated in the next example.

Example 2: In this example, we continue Example 1 and note that instead of using the set $\mathcal{W}_2$, one can use the set $Z = \{11000000, 00110000, 00001100, 00000011\}$ and still receive that $B(\mathbf{0},1) = \bigcap_{z\in Z} B(z,3)$. Hence, instead of using 28 servers as in Example 1, it is possible to use only four servers and still successfully compute the set $I(\mathbf{0},1) = \bigcap_{z\in Z} I(z,3)$, as required by the user.

Note that the privacy level is the same as in Example 1, since the user is actually sending four queries of weight two with disjoint supports, rather than sending Z itself. Therefore, from the perspective of each server, there are still 28 possibilities for the user's record, and not four possibilities.

III. Lower Bounds on the Size of PPR Codes

In this section, we start in analyzing the quantity $N(L,s,r)$, which is the minimum number of servers required for the PPR schemes introduced in Definition 4. The set of query words related to these servers form a code for PPR schemes. This section starts by giving a formal definition and an appropriate name for these codes and continue with lower bounds on the size of such codes. Upper bounds will be provided in Section IV and analysis on the lower and upper bounds will be discussed in Section V.

Proposition 3 has motivated the definitions of PPR schemes and the quantity $N(L,s,r)$ in Section II. We note that the righthand side of (1) reflects the elements which are covered by intersection of balls centered at words whose distance from the record of the user is s. This is the motivation for the following definition. For given L, s, and r, such that $L \geqslant r + 2s + 1$, let C be a subset of $\mathcal{W}_s$ such that
$$B(\mathbf{0},r) = \bigcap_{c\in C} B(c, r+s). \tag{3}$$

Such a set C will be called an (L, s, r) *Private Proximity Retrieval Intersection Covering code*, or an (L, s, r) *PPRIC code* in short. Let $N(L, s, r)$ be the minimum number of codewords in such an (L, s, r) PPRIC code C. We continue and discuss bounds on the sizes of such codes and some properties related to their structure. Our main interest is in bounds for which r is fixed, $L \to \infty$, and $\sigma = \frac{s}{L}$ is fixed.

An important concept that will be used in our lower bounds and constructions for PPRIC codes is a word which has a nonempty intersection with the support of each codeword of a given PPRIC code. Given a PPRIC code C, a word $v \in \mathbb{F}_2^L$ is called an *intersection PPR word* related to C, if $\text{supp}\ (v) \cap \text{supp}\ (c) \neq \varnothing$ for each $c \in C$. Such a word v is called a *minimal intersection PPR word* (an *MIPPR word* in short) if there is no other intersection PPR word u such that $\text{supp}\ (u) \subset \text{supp}\ (v)$.

Lemma 6: Let r, s, and L be nonnegative integers, such that $L \geqslant 2s + r + 1$, and let $C \subseteq \mathcal{W}_s$ be an (L, s, r) PPRIC code. If v is an MIPPR word, then $wt(v) \geqslant r + 3$.

Proof: Assume to the contrary that v is an MIPPR word with respect to C and $wt(v) \leqslant r + 2$, and let $y \in \mathbb{F}_2^L$ be any word of weight $r + 2$ such that $\text{supp}\ (v) \subseteq \text{supp}(y)$. Since $\text{supp}\ (v) \cap \text{supp}(c) \neq \varnothing$ for all $c \in C$ and $\text{supp}\ (v) \subseteq \text{supp}(y)$, if follows that $|\text{supp}(y) \cap \text{supp}(c)| \geqslant 1$ for all $c \in C$. Hence, for any $c \in C$, we have that

$$\begin{aligned} d(y, c) &= wt(y) + wt(c) - 2|\text{supp}(y) \cap \text{supp}(c)| \\ &= r + 2 + s - 2|\text{supp}(y) \cap \text{supp}(c)| \leqslant r + s \end{aligned}$$

and therefore, $y \in \bigcap_{c \in C} B(c, r + s)$, a contradiction to (3) since $wt(y) > r$. ∎

The number of codewords in a PPRIC code C is always an upper bound on the weight of the related MIPPR word and hence

Corollary 7: If r, s, and L are nonnegative integers, such that $L \geqslant 2s + r + 1$, then $N(L, s, r) \geqslant r + 3$.

By Lemma 6, the weight of an MIPPR word is at least $r+3$. This motivates constructions and lower bounds on the sizes of (L, s, r) PPRIC codes for which the weight of each MIPPR word is exactly $r+3$. Lemma 6 also implies that we can obtain bounds on $N(L, s, r)$ by considering words which intersect the support of every $c \in C$ (where C is a PPRIC code which attains the value of $N(L, s, r)$) in at least one coordinate. This approach was used to prove the following theorem presented in [12] which is proved by repeatedly applying Lemma 6.

Theorem 8: If r, s, and L are nonnegative integers, such that $L \geqslant 2s + r + 1$, then

$$N(L, s, r) \geqslant \max_{k=0,\ldots,r+1} \left\{ \left\lceil \frac{r + 3 - k}{(1 - \sigma)^k} \right\rceil \right\}, \tag{4}$$

where $\sigma = s/L$.

One can easily verify that the values in (4) are increasing with k, as long as $k \leqslant \lfloor r + 4 - 1/\sigma \rfloor$. Thus, $N(L, s, r) \geqslant \left\lceil \frac{r+3-k}{(1-\sigma)^k} \right\rceil$, where $\sigma = s/L$ and $k = \max\{0, \lfloor r + 4 - 1/\sigma \rfloor\}$.

Note that Theorem 8 implies Corollary 7. In the sequel we obtain a lower bound which also implies Theorem 8. The new lower bound is based on design theory, and specifically on covering designs, and we will present first the required definitions for this lower bound. These definitions will be also used in our constructions given in the next subsection. We will show that our PPRIC codes are covering codes (more commonly known as covering designs or just coverings) which have some additional constraints.

An (n, k, t) *covering design*, $n \geqslant k \geqslant t > 0$, is a collection of k-subsets (called *blocks*) of an n-set, w.l.o.g. $[n]$, (whose elements are called *points*), such that each t-subset of the n-set is contained in at least one block of the collection. Let $c(n, k, t)$ denote the minimum number of blocks in such an (n, k, t) covering design. Finally, the *complement* $\bar{S}$ of the design S is defined by

$$\bar{S} \triangleq \{[n] \setminus B \ : \ B \in S\}.$$

Similarly for a binary code C of length n, the *complement* $\bar{C}$ is the code formed from the binary complements of all the codewords in C. In other words, the codewords in $\bar{C}$ are exactly all the words whose set of supports is

$$\{[n] \setminus \text{supp}\ (c) \ : \ c \in C\}.$$

Covering designs were extensively studied mainly from a combinatorial point of view [24]. They have also found many applications, especially in various problems related to coding theory, e.g. [14]. Bounds on the sizes of covering designs, i.e. on $c(n, k, t)$, are very important in our discussion as a consequence of the connection between PPRIC codes and covering designs given in the next result.

Lemma 9: The set of supports in the complement of an (L, s, r) PPRIC code is an $(L, L - s, r + 2)$ covering design.

Proof: Let C be an (L, s, r) PPRIC code. Define $S \triangleq \{\text{supp}(c) \ : \ c \in C\}$ and $\bar{S}$ its complement design. By Lemma 6, for an (L, s, r) PPRIC code all MIPPR words have at least weight $r + 3$. It implies that there are no $r + 2$ coordinates which contain an element from the support of each codeword. In other words, for each subset $P \subset [L]$ of size $r+2$ there exists a codeword $c \in C$ such that $P \cap \text{supp}\ (c) = \varnothing$. It follows that for each subset $P \subset [L]$ of size $r + 2$ there exists a block $B \in \bar{S}$ such that $P \subset B$. Moreover, in $\bar{S}$ each block has size $L - s$. Thus, $\bar{S}$ forms an $(L, L - s, r + 2)$ covering design. ∎

An (L, s, r) PPRIC code C is the complement of an $(L, L - s, r + 2)$ covering design. Therefore, for every word v of weight $r + 1$ or $r + 2$, there exists a codeword $c \in C$ such that $\text{supp}(c) \cap \text{supp}(v) = \varnothing$. However, a PPRIC code has some more requirements. Equality (3) suggests that, for each word v of weight $r+2\gamma-1$ or $r+2\gamma$, $\gamma \geqslant 1$, there exists a codeword $c \in C$ such that $|\text{supp}(c) \cap \text{supp}(v)| < \gamma$. This implies that $d(v, c) = wt(v) + s - 2|\text{supp}(c) \cap \text{supp}(v)| \geqslant r + 2\gamma - 1 + s - 2(\gamma - 1) > r + s$ as expected.

These properties of PPRIC codes imply a method to verify whether a code C is indeed an (L, s, r) PPRIC code. To show that C is a PPRIC code, we just need to prove that any word of length L which intersects each codeword of C in at least γ coordinates should have weight at least $r + 2\gamma + 1$. Equivalently, to show that C is not a PPRIC code, we just need to find one word of weight $r+2\gamma$ which intersects each codeword

of C in at least γ coordinates, and therefore has distance at most $r+s$ from all the codewords of C.

Remark 4: Lemma 9 (see also Corollary 50) implies that given an (L,s,r) PPRIC code C, for each word v of length L and weight $L-r-2$, there exists a codeword $c \in C$ such that $\mathrm{supp}(c) \subset \mathrm{supp}(v)$. For each word v of length L and weight $L-r-2-2\gamma$, $\gamma \geqslant 0$, there exists a codeword $c \in C$ such that $|\mathrm{supp}(c) \cap \mathrm{supp}(v)| \geqslant s-\gamma$. This is exactly what is required for data compression as explained in [14], where some bounds including asymptotic bounds in this direction are given.

Lemma 9 also implies the following important lower bound.

Corollary 10: If r, s, and L are nonnegative integers, such that $L \geqslant 2s+r+1$, then $N(L,s,r) \geqslant c(L,L-s,r+2)$. (or in the notation of covering designs, $N(n,n-k,t-2) \geqslant c(n,k,t)$.)

Corollary 10 implies several lower bounds on $N(L,s,r)$. In fact, most of our lower bounds are based on the lower bounds on the size of the related covering designs, but there are some parameters for which $N(L,s,r) > c(L,L-s,r+2)$ as we will see in the sequel. The first lower bound is the basic covering bound implied directly from Corollary 10.

Lemma 11: If r, s, and L are nonnegative integers, such that $L \geqslant 2s+r+1$, then

$$N(L,s,r) \geqslant \frac{\binom{L}{r+2}}{\binom{L-s}{r+2}} .$$

The next result is well known as the Schönheim's bound [32].

Lemma 12: If $n > k > t > 0$, then

$$c(n,k,t) \geqslant \left\lceil \frac{n}{k} c(n-1,k-1,t-1) \right\rceil .$$

Lemma 12 can be applied iteratively to obtain several results.

Corollary 13: If $n > k > t > 0$, then

$$c(n,k,t) \geqslant \left\lceil \frac{n}{k} \left\lceil \frac{n-1}{k-1} \cdots \left\lceil \frac{n-\ell+1}{k-\ell+1} c(n-\ell,k-\ell,t-\ell) \right\rceil \cdots \right\rceil \right\rceil .$$

Corollary 14: If $n > k > t > 0$, then

$$c(n,k,t) \geqslant \left\lceil \frac{n}{k} \left\lceil \frac{n-1}{k-1} \cdots \left\lceil \frac{n-t+1}{k-t+1} \right\rceil \cdots \right\rceil \right\rceil .$$

Corollary 15: If r, s, and L are nonnegative integers, such that $L \geqslant 2s+r+1$, then

$$N(L,s,r) \geqslant \left\lceil \frac{L}{L-s} \left\lceil \frac{L-1}{L-s-1} \cdots \left\lceil \frac{L-\ell+1}{L-s-\ell+1} c(L-\ell,L-s-\ell,r+2-\ell) \right\rceil \cdots \right\rceil \right\rceil .$$

Lemma 11 can be derived from Corollary 14 as similarly Corollary 15 implies Lemma 11. Corollary 14 also implies that $c(n,k,t) \geqslant t+1$ (since $\left\lceil \frac{n-t+1}{k-t+1} \right\rceil \geqslant 2$ and $\left\lceil \frac{n-i}{k-i} M \right\rceil \geqslant M+1$ for any positive integer M and $0 \leqslant i \leqslant k-1$) and by applying Corollary 15 we obtain Theorem 8. By Corollary 15 and since $s = \sigma L$ it follows that

$$N(L,s,r) \geqslant \left\lceil \frac{L}{L(1-\sigma)} \cdots \left\lceil \frac{L-\ell+1}{L(1-\sigma)-\ell+1} c(L-\ell,L-s-\ell,r+2-\ell) \right\rceil \cdots \right\rceil ,$$

and since $c(n,k,t) \geqslant t+1$ we have that

$$\begin{aligned} N(L,s,r) &\geqslant \frac{L}{L(1-\sigma)} \cdot \frac{L-1}{L(1-\sigma)-1} \cdots \frac{L-\ell+1}{L(1-\sigma)-\ell+1} (r+3-\ell) \\ &\geqslant \left(\frac{L}{L(1-\sigma)} \right)^{\ell} (r+3-\ell), \end{aligned}$$

and hence Theorem 8 follows.

Usually, (n,k,t) covering designs were considered for small k and t, i.e. $n \geqslant 2k$. In our context we need small t and large k, i.e. small $n-k$. These types of designs were considered for a given m and t, and pairs (n,k) such that $m = c(n,k,t)$. There has been extensive research in this direction, which is exactly what is required for our bounds on PPRIC codes. The main difficulty as was pointed in [24] is to establish the correct lower bound. This usually has required a very long and complicated proof. Mills [23] and Todorov [37] found the maximum ratio $\frac{n}{k}$ on the pairs (n,k) such that $m = c(n,k,t)$. The first value was given by Mills [23, Theorem 2.2 and Theorem 2.3]. This is quoted as in Todorov [37, Theorem 2].

Theorem 16: Let $n > k > t > 0$ and assume that $m = c(n,k,t)$. If m and t are positive integers such that $t < m \leqslant \frac{3(t+1)}{2}$, then

$$\frac{n}{k} \leqslant \frac{3t+3-m}{3t+1-m} .$$

Therefore, if $\frac{n}{k} \leqslant \frac{t+1}{t}$ then $c(n,k,t) \geqslant t+1$; If $t+1 < m \leqslant \frac{3(t+1)}{2}$ and $\frac{3t+4-m}{3t+2-m} < \frac{n}{k} \leqslant \frac{3t+3-m}{3t+1-m}$, then $c(n,k,t) \geqslant m$.

Clearly, we have to translate the results from the pair of parameters (n,k) to the pair (L,s) which is performed by the simple equation $\frac{n}{k} = \frac{L}{L-s}$. This implies that $\frac{k}{n} = \frac{L-s}{L} = 1 - \frac{s}{L}$ and $\frac{s}{L} = 1 - \frac{k}{n}$.

Corollary 17: Let r, s, and L be nonnegative integers, such that $L \geqslant 2s+r+1$. If m and r are nonnegative integers such that $r+2 < m \leqslant \frac{3(r+3)}{2}$, then

$$\frac{L}{s} \geqslant \frac{3r+9-m}{2} .$$

If $\frac{L}{s} \geqslant r+3$ then $N(L,s,r) \geqslant r+3$; If $r+3 < m \leqslant \frac{3(r+3)}{2}$ and $\frac{3r+9-m}{2} \leqslant \frac{L}{s} < \frac{3r+10-m}{2}$, then $N(L,s,r) \geqslant m$.

Theorem 16 was extended by Todorov [37] as follows.

Theorem 18: If $n > k > t > 0$ and $m = c(n,k,t)$, then for every integer $\ell \geqslant 1$ we have the following results.

1) If $m = 3\ell+1$ and $t = 2\ell-1$, then $\frac{n}{k} \leqslant \frac{9\ell-1}{9\ell-7}$.
2) If $m = 3\ell+2$ and $t = 2\ell$, then $\frac{n}{k} \leqslant \frac{6\ell+3}{6\ell-1}$.
3) If $m = 3\ell+3$ and $t = 2\ell$, then $\frac{n}{k} \leqslant \frac{3\ell+1}{3\ell-1}$.

Theorem 16 and Theorem 18 lead to the following consequences.

Corollary 19: If $n > k > t > 0$ and $m = c(n,k,t)$, then for every integer $\ell \geqslant 1$ we have the following results.

1) If t is odd and $\frac{3t+3}{3t-1} < n/k \leqslant \frac{9t+7}{9t-5}$, then $c(n,k,t) \geqslant \frac{3t+5}{2}$.

2) If t is even and $\frac{3t+4}{3t} < n/k \leqslant \frac{3t+3}{3t-1}$, then $c(n,k,t) \geqslant \frac{3t+4}{2}$.
3) If t is even and $\frac{3t+3}{3t-1} < n/k \leqslant \frac{3t+2}{3t-2}$, then $c(n,k,t) \geqslant \frac{3t+6}{2}$.

Corollary 20: Let r, s, and L be nonnegative integers, such that $L \geqslant 2s+r+1$.

1) If r is odd and $\frac{9r+25}{12} \leqslant L/s < \frac{3r+9}{4}$, then we have $N(L,s,r) \geqslant \frac{3r+11}{2}$.
2) If r is even and $\frac{3r+9}{4} \leqslant L/s < \frac{3r+10}{4}$, then we have $N(L,s,r) \geqslant \frac{3r+10}{2}$.
3) If r is even and $\frac{3r+8}{4} \leqslant L/s < \frac{3r+9}{4}$, then we have $N(L,s,r) \geqslant \frac{3r+12}{2}$.

So far we have listed several lower bounds of $N(L,s,r)$ implied by lower bounds on $c(n,k,t)$. In the sequel it will be shown that equality holds for almost all the bounds in Corollaries 17 and 20, by presenting optimal covering designs whose complement are PPRIC codes. The covering designs will be constructed in Section IV, where the analysis and the exact parameters for which they are attained are summarized in Section V. However, there is one exception for Corollary 20(3) when $r=0$, which is the next result, demonstrating the difference between the lower bounds on $c(n,k,t)$ and $N(L,s,r)$. The proof is quite tedious and to simplify some parts we present some auxiliary claims throughout the proof. The complete proof is presented in Appendix A.

Theorem 21: If L and s are integers such that $L/s = 2+\epsilon$, where $\epsilon < \frac{1}{8}$, then $N(L,s,0) > 6$.

Theorem 21 is important as it presents a set of parameters for which the smallest PPRIC code has larger size than the smallest related covering design. This will be further discussed in Section V.

IV. Upper Bounds on the Size of PPRIC Codes

In this subsection we present a few constructions for PPRIC codes which imply upper bounds on $N(L,s,r)$. Some of these bounds can be derived from constructions of covering designs in [17], [23], where the covering designs can be proved to be also PPRIC codes. But, these constructions of covering designs are not always PPRIC codes, and for many parameters no constructions are given. Other bounds known from the general constructions on covering designs are given in [16]. Analysis of our construction, and especially with comparison of related covering designs, will be given in Section V. By Proposition 3 we know that for an (L,s,r) PPRIC code, $L \geqslant 2s+r+1$. There are only trivial requirements on the parameters of an (n,k,t) covering design. By Corollary 10 we have that $N(L,s,r) \geqslant c(L, L-s, r+2)$ and hence for the related $(L, L-s, r+2)$ covering design we only have $L \geqslant L-s \geqslant r+2$, i.e. $L \geqslant s+r+2$. The first result implies that if $L \geqslant (r+3)s$, then the optimal (L,s,r) PPRIC code is derived in a trivial way.

Theorem 22: If r, s, and L are nonnegative integers such that $L \geqslant (r+3)s$, then $N(L,s,r) \leqslant r+3$.

Proof: Let $L \geqslant (r+3)s$ and C be a code which consists of $r+3$ codewords with $r+3$ disjoint support sets. In view of the definition for an (L,s,r) PPRIC code, we only have to show that for any vector v of weight $r+2\gamma-1$ or $r+2\gamma$, $\gamma > 0$, there exists a codeword $c \in C$ such that $d(v,c) > r+s$. Let v be such a vector. If $|\operatorname{supp}(c) \cap \operatorname{supp}(v)| \geqslant \gamma$ for each $c \in C$, then since the codewords of C have disjoint support sets it follows that $wt(v) \geqslant (r+3)\gamma > r+2\gamma$, a contradiction. Hence, there exists at least one codeword $c \in C$ such that $|\operatorname{supp}(c) \cap \operatorname{supp}(v)| < \gamma$. For this codeword c we have $d(v,c) = s + wt(v) - 2|\operatorname{supp}(c) \cap \operatorname{supp}(v)| > r+s$ and therefore C is an (L,s,r) PPRIC code and the proof is completed. ■

In view of Theorem 22 and Corollary 7 (also Theorem 8 with $k=0$) we have the following corollary.

Corollary 23: If r, s, and L are nonnegative integers such that $L \geqslant (r+3)s$, then $N(L,s,r) = r+3$.

As a consequence of Corollary 23, henceforth we only consider values of L for which $2s+r+1 \leqslant L < (r+3)s$. It appears that many of the constructions of optimal covering designs or almost optimal, are also constructions for optimal PPRIC codes or almost optimal, respectively. But, for some range of parameters there are constructions of covering designs which are certainly not PPRIC and also constructions of optimal covering designs which are not PPRIC codes. These will be discussed in Section V. Upper bounds on $c(n,k,t)$ are not translated immediately into analog bounds on $N(n, n-k, t-2)$, but most upper bounds (constructions) on the size of covering designs have analog upper bounds (constructions) for PPRIC codes. These bounds on $c(n,k,t)$ for which a related bound on $N(n, n-k, t-2)$ can be obtained, will be discussed in the next few paragraphs. These bounds can be found for example in [17].

Lemma 24: If $n > k > t > 0$ then $c(n+1, k+1, t) \leqslant c(n,k,t)$.

Corollary 25: If $n > k > t > 0$ and $\alpha > 0$ are integers, then $c(\alpha n, \alpha k, t) \leqslant c(n,k,t)$.

The analog of Lemma 24 is the following lemma which can be easily verified.

Lemma 26: If L, s, and r, are nonnegative integer such that $L \geqslant 2s+r+1$, then $N(L+1,s,r) \leqslant N(L,s,r)$.

Corollary 25 also has an analog bound on $N(L,s,r)$ as follows.

Lemma 27: If L, s, r, and α, are nonnegative integer such that $L \geqslant 2s+r+1$, then $N(\alpha L, \alpha s, r) \leqslant N(L,s,r)$.

In contrary to Corollary 25 which is an immediate consequence from Lemma 24, Lemma 27 is not a consequence of Lemma 26 and a proof has to be provided. We omit the proof which is very similar (and relatively simpler) to proofs of results given in the sequel.

We continue with a construction for PPRIC codes which is general and all our other constructions for PPRIC codes are variants of this construction. To this end, we will need the following definition. For a given $\ell \geqslant 0$ a *design of Type* ℓ is a collection of blocks of size s from a set Q having the following property. For any subset $P \subseteq Q$, intersecting with each block in a subset whose size is at least t, we have that $|P| \geqslant \ell + t$. In other words, for a subset $R \subseteq Q$ of size less than $\ell+t$, there exists at least one block whose intersection with R is smaller than t.

Construction 1: Let $\ell_1, \ldots, \ell_p$, $p \geqslant 2$, be nonnegative integers and let $r = \sum_{i=1}^{p} \ell_i + p - 3$. Choose p designs of

Types $\ell_1, \ldots, \ell_p$ on p subsets of disjoint coordinates of $[L]$ to form a code C.

Theorem 28: The code C produced in Construction 1 is an (L, s, r) PPRIC code, where $r = \sum_{i=1}^{p} \ell_i + p - 3$.

Proof: Let S_i, $1 \leqslant i \leqslant p$, be the related design of Type ℓ_i. It is sufficient to show that for each word v of weight $r + 2t - 1$ or $r + 2t$, where $t \geqslant 1$, there exists a codeword $c \in C$, such that $d(v, c) > r + s$. Since $r + 2t = \sum_{i=1}^{p} \ell_i + p + 2t - 3 \leqslant \sum_{i=1}^{p} (\ell_i + t) - 1$, it follows that there exists one i for which $|\operatorname{supp}(v) \cap S_i| < \ell_i + t$. Hence, at least for one of codeword (block) $c \in S_i$ we have $|\operatorname{supp}(v) \cap \operatorname{supp}(c)| < t$ and hence

$$d(c, v) \geqslant s + r + 2t - 1 - 2(t - 1) > r + s$$

which completes the proof. ■

Next, we consider how to construct the best PPRIC code by using Construction 1. For this purpose, an important upper bound on $c(n, k, t)$ is the following theorem of Morley and van Rees [26] generalized in [17]. Covering designs obtained by the related methods are referred in that paper and also in [16] as a special case of a dynamic programming construction.

Theorem 29: If $n_1 > n_1 - s > t_1 > 0$ and $n_2 > n_2 - s > t_2 > 0$ are integers, then

$$\begin{aligned} &c(n_1 + n_2, n_1 + n_2 - s, t_1 + t_2 + 1) \\ &\leqslant c(n_1, n_1 - s, t_1) + c(n_2, n_2 - s, t_2). \end{aligned}$$

Construction 1 implies a related result for PPRIC codes.

Theorem 30: If $L_1 > L_1 - s > t_1 > 0$ and $L_2 > L_2 - s > t_2 > 0$ are integers, then

$$N(L_1 + L_2, s, t_1 + t_2 - 1) \leqslant c(L_1, L_1 - s, t_1) + c(L_2, L_2 - s, t_2).$$

To apply Construction 1 we have to consider constructions for designs of Type ℓ. For this we will define the concept of a superset. Let S be a collection of α-subsets (called *blocks*) of an n-set, where α divides s and $\bar{S}$ is an $(n, n - \alpha, t)$ covering design. An S-*superset* is a set of $\frac{ns}{\alpha}$ elements partitioned into n pairwise disjoint subsets called *grain-sets*, where each grain-set has size $\frac{s}{\alpha}$. Each S-superset will contribute exactly $|S|$ codewords to the related PPRIC code, which will be a union of such S-supersets. Each codeword is defined by the support of α distinct grain-sets related to the $|S|$ blocks of S. Clearly, each support of such a codeword has weight s. The following lemma is required to apply Construction 1.

Lemma 31: If $\bar{S}$ is an $(n, n - \alpha, \ell)$ covering design, then the related S-superset (obtained from S), where each grain-set has size $\frac{s}{\alpha}$, is a design of Type ℓ.

Proof: In the S-superset there are n grain-sets each one of size $\frac{s}{\alpha}$ with labels $\{1, 2, \ldots, n\}$. Let Q be the union of all these grain-sets. For any block $B \in S$ whose size is α, we form a codeword of size s from the grain-sets related to B.

Assume the contrary that the S-superset is not a design of Type ℓ, i.e. that we have a subset $Q' \subseteq Q$ of size at most $\ell + t - 1$, which intersects each block from the S-superset in a subset whose size is at least t. We distinguish now between two cases:

Case 1: Q' contains elements from at least ℓ distinct grain-sets. Let $P \subseteq Q'$ be a subset of size ℓ with elements from distinct ℓ grain-sets. P intersects ℓ grain-sets with labels $\{i_1, \ldots, i_\ell\} \subset \{1, 2, \ldots, n\}$. By the definition of an $(n, n - \alpha, \ell)$ covering design, $\{i_1, \ldots, i_\ell\}$ is contained in some $(n - \alpha)$-subset $\bar{B} \in \bar{S}$. Therefore the block $B \in S$ is disjoint from the grain-sets with labels $\{i_1, \ldots, i_\ell\}$, and hence $|B \cap Q'| \leqslant t - 1$, a contradiction.

Case 2: Q' contains elements from exactly ℓ' distinct grain-sets, where $\ell' \leqslant \ell$, with labels $\{i_1, \ldots, i_{\ell'}\} \subset \{1, 2, \ldots, n\}$. By the definition of an $(n, n - \alpha, \ell)$ covering design, $\{i_1, \ldots, i_{\ell'}\}$ is also contained in some $(n - \alpha)$-subset $\bar{B} \in \bar{S}$. Therefore the block $B \in S$ is disjoint from the grain-sets with labels $\{i_1, \ldots, i_{\ell'}\}$, and hence $B \cap Q' = \varnothing$, a contradiction.

Therefore, we have proved that any subset of Q, which intersects each block from the S-superset in a subset whose size is at least t, should have size at least $\ell + t$. Thus, an S-superset is a design of Type ℓ. ■

For the following two constructions, which are special cases of Construction 1, let $(k, 1)$-superset denote an S-superset, where $\bar{S}$ is the trivial $(k + 1, 1, 1)$ covering design which consists of $k+1$ blocks, each one of size one. This is a design of Type 1.

Construction 2 (A construction for odd r): Let $k \geqslant 1$ and t, $0 \leqslant t \leqslant \frac{r+1}{2}$, be an integer such that

$$\frac{L}{s} \geqslant \left(\frac{r+3}{2} - t\right) \frac{k+1}{k} + t\frac{k+2}{k+1} = \frac{(r+3)(k+1)}{2k} - \frac{t}{k(k+1)}.$$

Let $\{A_i : 1 \leqslant i \leqslant t\}$ be a family of t $(k+1, 1)$-supersets and $\{A_i : t + 1 \leqslant i \leqslant \frac{r+3}{2}\}$ be a family of $\frac{r+3}{2} - t$ $(k, 1)$-supersets, where the $\frac{r+3}{2}$ supersets are pairwise disjoint. From each $(k + 1, 1)$-superset we form $k + 2$ codewords of weight s and from each $(k, 1)$-superset we form $k + 1$ codewords of weight s.

Theorem 32: Construction 2 produces an (L, s, r) PPRIC code with

$$\left(\frac{r+3}{2} - t\right)(k+1) + t(k+2) = \frac{(r+3)(k+1)}{2} + t$$

codewords.

Proof: First, note that since

$$L \geqslant \left(\frac{r+3}{2} - t\right) \frac{k+1}{k} s + t\frac{k+2}{k+1} s,$$

it follows that it is possible to choose $\frac{r+3}{2}$ pairwise disjoint supersets as required. The size of the code is readily derived by its definition. In view of (3) we only have to show that for any vector v of weight $r + 2\gamma - 1$ or $r + 2\gamma$, $\gamma > 0$, there exists a codeword $c \in C$ such that $d(v, c) > r + s$. Assume to the contrary, i.e., there exists such a vector v for which $v \in \cap_{c \in C} B(c, r + s)$, i.e. $d(v, c) \leqslant r + s$ for each $c \in C$. Since $wt(c) = s$ for each $c \in C$ and $d(v, c) \leqslant s + r$, it follows that $|\operatorname{supp}(v) \cap \operatorname{supp}(c)| \geqslant \gamma$ for each $c \in C$. For any $(\ell, 1)$-superset S, $\ell \in \{k, k + 1\}$, $\operatorname{supp}(v)$ must have a non-empty intersection with at least one of its $\ell + 1$ grain-sets. The other ℓ grain-sets of S form the support set of a codeword c and hence intersects $\operatorname{supp}(v)$ in at least γ elements. Therefore, $|\operatorname{supp}(v) \cap S| \geqslant \gamma + 1$ and since there are $\frac{r+3}{2}$ supersets, it follows that this argument holds to all these supersets and

hence we have

$$r + 2\gamma \geqslant wt(v) \geqslant \frac{r+3}{2}(\gamma + 1),$$

which is not possible when $\gamma > 0$ and r is an odd positive integer. ■

Construction 3 (A construction for even r):

Let $k \geqslant 1$ and t, $0 \leqslant t \leqslant \frac{r}{2}$, be an integer such that

$$\frac{L}{s} \geqslant \left(\frac{r+2}{2} - t\right)\frac{k+1}{k} + t\frac{k+2}{k+1} + 1 = \frac{(r+2)(k+1)}{2k} - \frac{t}{k(k+1)} + 1.$$

Let $\{A_i : 1 \leqslant i \leqslant t\}$ be a family of t $(k+1,1)$-supersets, $\{A_i : t+1 \leqslant i \leqslant \frac{r+2}{2}\}$ be a family of $\frac{r+2}{2} - t$ $(k,1)$-supersets, where the $\frac{r+2}{2}$ supersets are pairwise disjoint. From each $(k+1,1)$-superset we form $k+2$ codewords of weight s and from each $(k,1)$-superset we form $k+1$ codewords of weight s. To these codewords we add the last codeword of weight s whose support is disjoint to the supports of all the other codewords (which can be considered as a design of Type 0 itself).

Similarly to the proof of Theorem 32 one can prove the following theorem.

Theorem 33: Construction 3 produces an (L, s, r) PPRIC code with

$$\left(\frac{r+2}{2} - t\right)(k+1) + t(k+2) + 1 = \frac{(r+2)(k+1)}{2} + t + 1$$

codewords.

The disadvantage of Construction 3 is that s elements of $[L]$ are occupied by just a single codeword and it requires just one element from a related MIPPR word with $r+3$ elements. The other supersets occupy either $s + \frac{s}{k}$ or $s + \frac{s}{k+1}$ elements of $[L]$ for two elements from the related MIPPR word. Consider this single codeword and an arbitrary superset. Together they contribute three elements to an MIPPR word. Replacing them by another type of superset, which still contributes three elements to an MIPPR word, but occupies less elements of $[L]$, may improve our upper bounds. Moreover, using other types of supersets and their combinations will yield better upper bounds and upper bounds for more parameters.

Finally, for many parameters we might find a smaller upper bound on $N(L, s, r)$ than the one obtained via our constructions. One example is when $r = 0$ and $L/s \geqslant \frac{17}{8}$. We have constructed the following $(L, s, 0)$ PPRIC code $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ of size six implying that $N(L, s, 0) = 6$. We define the following six sets of intersecting pairs, triples, and quadruples (note, that codewords are represented by s-subsets).

$A_1 \triangleq c_1 \cap c_2 \cap c_3 \cap c_4$, $A_2 \triangleq c_1 \cap c_2 \cap c_5$, $A_3 \triangleq c_3 \cap c_4 \cap c_5$, $A_4 \triangleq c_1 \cap c_3 \cap c_6$, $A_5 \triangleq c_2 \cap c_3 \cap c_6$, and $A_6 \triangleq c_4 \cap c_6$. Furthermore, let $|A_1| = s/8$, $|A_2| = 5s/8$, $|A_3| = 3s/8$, $|A_4| = s/4$, $|A_5| = s/4$, and $|A_6| = s/2$. The code C consists of the following six codewords:

$$\begin{aligned} c_1 &= 11111100011000000,\\ c_2 &= 11111100000110000,\\ c_3 &= 10000011111110000,\\ c_4 &= 10000011100001111,\\ c_5 &= 01111111100000000,\\ c_6 &= 00000000011111111. \end{aligned}$$

Theorem 34: The code C is an $(L, s, 0)$ PPRIC code of size six.

Proof: Let v be a word which intersects each codeword in C in at least t coordinates, $t \geqslant 0$. For each i, $1 \leqslant i \leqslant 6$, let a_i be the size of the intersection of A_i and v. Since v intersects each codeword of C in at least t coordinates, it follows that the following six inequalities must be satisfied.

$$\text{for } c_1: \ a_1 + a_2 + a_4 \geqslant t, \tag{5}$$

$$\text{for } c_2: \ a_1 + a_2 + a_5 \geqslant t, \tag{6}$$

$$\text{for } c_3: \ a_1 + a_3 + a_4 + a_5 \geqslant t, \tag{7}$$

$$\text{for } c_4: \ a_1 + a_3 + a_6 \geqslant t, \tag{8}$$

$$\text{for } c_5: \ a_2 + a_3 \geqslant t, \tag{9}$$

$$\text{for } c_6: \ a_4 + a_5 + a_6 \geqslant t. \tag{10}$$

Assume now that v has weight at most $2t$, i.e.

$$a_1 + a_2 + a_3 + a_4 + a_5 + a_6 \leqslant 2t. \tag{11}$$

Equation (11), equation (9), and equation (10) imply that $a_1 = 0$. Equation (11), $a_1 = 0$, equation (6), and equation (8) imply that $a_4 = 0$. Equation (11), $a_4 = 0$, equation (5), and equation (10) imply that $a_3 = 0$. Equation (11), $a_1 = 0$, $a_4 = 0$, equation (5), and equation (7) imply that $a_6 = 0$, contradicting (8). Therefore, $wt(v) \geqslant 2t + 1$ which implies that C is an $(L, s, 0)$ PPRIC code. ■

Theorem 21 and Theorem 34 imply the following result

Corollary 35: If $\frac{17}{8} \leqslant \frac{L}{s} < \frac{9}{4}$ then $N(L, s, 0) = 6$ and if $\frac{L}{s} < \frac{17}{8}$ then $N(L, s, 0) > 6$.

The $(L, s, 0)$ PPRIC code $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ should motivate further research to find more constructions for (L, s, r) PPRIC codes with $r > 0$. The related covering design presented in [23] has size six, but it is not a PPRIC code. This will be further discussed in Section V.

V. Analysis of the Lower and Upper Bounds

In this subsection we present a short analysis for the lower and upper bounds on $N(L, s, r)$ which were obtained in the previous subsections. We are interested to know the gap between the upper and the lower bound on $N(L, s, r)$ and in particular when these bounds coincide, i.e. the exact value of $N(L, s, r)$ is known. We would like to know the range of parameters for which the bounds in Corollary 10 is attained, i.e. when the (L, s, r) PPRIC code of the minimum size has the same size as the minimum size of an $(L, L - s, r + 2)$ covering design. A followup question is for which range the size of the minimum PPRIC code is strictly larger? Is there a region in which any $(L, L - s, r + 2)$ covering design is also an (L, s, r) PPRIC code?

By Corollary 23 for $L \geqslant (r+3)s$ we have that $N(L,s,r) = r+3$. The results of Mills and Todorov [23], [37] on covering designs provide several lower bounds for $N(L,s,r)$ for certain regions of $\frac{L}{s}$, while our constructions meet some of these bounds. To be more specific:

- The lower bound of Corollary 17 for odd r is met by applying Construction 2 and Theorem 32 with $k = 1$ and arbitrary t;
- The lower bound of Corollary 17 for even r is met by applying Construction 3 and Theorem 33 with $k = 1$ and arbitrary t;
- The lower bound of Corollary 20(1) is met by applying Construction 2 and Theorem 32 with $k = 2$ and $t = 1$;
- The lower bound of Corollary 20(2) is met by applying Construction 1 with $r/2$ $(2,1)$-supersets, $r > 0$, and the design of Type 2 with 5 blocks coming from the $(9,5,2)$ covering design with the blocks

$$\{1,2,3,4,5\},\{1,2,3,4,6\},\{1,2,7,8,9\},\\ \{3,4,7,8,9\},\{5,6,7,8,9\}.$$

- The lower bound of Corollary 20(3) for $r > 0$ is met by applying Construction 1 with $r/2$ $(2,1)$-supersets and the design of Type 2 with 6 blocks coming from the $(4,2,2)$ covering design with the blocks

$$\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}.$$

To summarize, we have the following exact bounds on $N(L,s,r)$.

Theorem 36: Let r, s, L, be nonnegative integers such that $L \geqslant 2s+r+1$.

1) If $\frac{L}{s} \geqslant r+3$ then $N(L,s,r) = r+3$.

2) If $r+3 < m \leqslant \frac{3(r+3)}{2}$ and $\frac{3r+9-m}{2} \leqslant \frac{L}{s} < \frac{3r+10-m}{2}$, then $N(L,s,r) = m$.

3) If r is odd and $\frac{9r+25}{12} \leqslant \frac{L}{s} < \frac{3r+9}{4}$, then $N(L,s,r) = \frac{3r+11}{2}$.

4) If r is even and $\frac{3r+9}{4} \leqslant \frac{L}{s} < \frac{3r+10}{4}$, then $N(L,s,r) = \frac{3r+10}{2}$.

5) If $r > 0$ is even and $\frac{3r+8}{4} \leqslant \frac{L}{s} < \frac{3r+9}{4}$, then $N(L,s,r) = \frac{3r+12}{2}$.

6) If $r = 0$ and $\frac{17}{8} \leqslant \frac{L}{s} < \frac{9}{4}$, then $N(L,s,0) = 6$.

Given any other set of parameters L, s, and r, what is the best strategy to form the smallest possible (L,s,r) PPRIC code? Clearly, this is a difficult question, for which also the answer for the related covering designs is not known. Construction 1 is our general construction and it can be applied recursively using Theorem 30. But, this application is not unique, so dynamic programming (as mentioned in [17]) should be used to obtain the best bound. Note, that since our main interest is for an (L,s,r) PPRIC for fixed r and $\frac{L}{s}$ (note that $\sigma = \frac{s}{L}$), where $L \to \infty$, it follows that our construction will be based on supersets.

Example 3: Suppose that $r = 11$ and $\frac{L}{s} = 7$. We choose $L = 56$ and $s = 8$ and write $56 = L = L_1 + L_2 = 28 + 28$, i.e. $L_1 = L_2 = 28$. Since $c(28,20,6) \leqslant 25$ [16] it follows that $N(L,s,11) \leqslant 50$ for this case. Note that other covering designs can be used in this case. In general using (n,k,t) covering designs with larger n would be better. For this case $c(21,15,6) \leqslant 27$ yields an $(L,s,11)$ PPRIC code of size 54, and $c(35,25,6) \leqslant 27$ yields an $(L,s,11)$ PPRIC code of size 54 implying that for small covering designs we sometimes have to be careful in our choice of code on which we apply Construction 1.

Is the value of $N(L,s,r)$ always equal to the one of $c(L,L-s,r+2)$? The answer is definitely no. For example, by Theorem 21 and Corollary 35, for $r = 0$, $L/s = 2+\epsilon$ where $\epsilon < \frac{1}{8}$, we have $N(L,s,r = 0) > 6$, while $c(L,L-s,r+2 = 2) = 6$ [23]. Moreover, for any given nonnegative integers L, s, r, such that $L = 2s+r+1$, there exists an $(L,L-s,r+2)$ covering design S, where $\bar{S}$ is not an (L,s,r) PPRIC code. Let $\mathcal{A}$ and $\mathcal{B}$ be two disjoint subsets of size $2s$ and $r+1$, respectively. $\mathcal{A} \cup \mathcal{B}$ will be the set of points for the design. Let R be any $(2s,s,r+2)$ covering design on the points of $\mathcal{A}$. The set of blocks of the design is defined by

$$S \triangleq \{T \cup \mathcal{B} \ : \ T \in R\}.$$

Clearly, the size of a block in S is $s+r+1 = L-s$, the number of blocks in S is the same as the number of blocks in R, and S is a $(2s+r+1, s+r+1, r+2)$ covering design. Moreover, the codewords in $\bar{S}$ have weight s. Let v be the vector of length $2s+r+1$ for which $\mathrm{supp}(v) = \mathcal{A}$. It is readily verified that the weight of v is $2s$ and since $s > r$, it follows that $2s \geqslant r+1$. But, $d(v,x) = s \leqslant r+s$ for each $x \in \bar{S}$, contradicting (3) and hence $\bar{S}$ is not an (L,s,r) PPRIC code.

A related interesting question is the value of $N(2s+r+1,s,r)$ for each r and $s > r$. It is easy to verify that for r odd

$$N(2s+r+1,s,r) \leqslant 2\binom{s+\frac{r+1}{2}}{s}.$$

For r even

$$N(2s+r+1,s,r) \leqslant \binom{s+\frac{r}{2}}{s} + \binom{s+\frac{r+2}{2}}{s}.$$

The construction for these bounds is by a partition of $[2s+r+1]$ into two subsets S_1 and S_2 of size which differ by at most one, and taking from S_1 and S_2 all the s-subsets.

As indicated, some of the exact values for $c(L,k,t)$ given in [23], [37] are also exact values for $N(L,L-k,t-2)$. The construction of the related covering designs yields also PPRIC codes. The constructions given in [23], [37] are special cases (or variants) of Construction 2 and Construction 3. For any given r and $\sigma = \frac{s}{L}$, there exists an integer $m_0(r,\sigma)$ such that for all $m \leqslant m_0$, $N(L,s,r) = c(L,L-s,r+2) = m$, for large enough L. Finding $m_0(r,\sigma)$ as well as proving more relations between PPRIC codes and covering designs are interesting problems for future research. The gap between the size of the optimal PPRIC code and the related covering design is a very intriguing question for future research as well.

VI. Generalizations to Other Metrics

In our exposition so far we have considered the database to consist of binary words of length L and the distance measure which was taken between the words was the Hamming metric. It appears that our framework of the PPR scheme and the PPRIC codes is not restricted for binary words and to the

Hamming scheme. In this section we consider the PPR scheme in other metrics. Some of the basic foundations of our setup are true to other metrics as well. The result of Proposition 3 (as well as many other results in this work) can be generalized for other metrics on other spaces. In other words, the database $\mathcal{X}$ need not to be a set of binary words of length L with the Hamming distance taken as the distance measure. The generalization of the PPRIC codes to other metrics yields interesting covering codes in other metrics which were not examined before. In this section we consider a general framework for some of these spaces with their related distance measures.

Let $\mathcal{V}$ be a space with a distance measure d. By abuse of definition we call such a pair, a $(\mathcal{V}, d)$ scheme. The largest distance L in the scheme is called the *diameter* of the scheme. A scheme will be called *symmetric* if

- for any two elements $u_1, u_2 \in \mathcal{V}$ such that $d(u_1, u_2) = k$, there exists a vertex $v \in \mathcal{V}$ such that $d(u_1, v) = L$ and $d(u_2, v) = L - k$;
- the size of a ball with radius r around a word $x \in \mathcal{V}$ does not depend on x.

We will be interested only in symmetric schemes, some of which are defined in the next few paragraphs.

The *Hamming scheme* $H_q(L)$ consists of all words of length L over an alphabet of size q. The *Hamming distance* $d_H(x, y)$ between two words x and y of length L over an alphabet with q letters is the number of coordinates in which they differ, i.e., $d_H(x, y) \triangleq |\{i : x_i \neq y_i\}|$.

The *Johnson scheme* $J(n, L)$ consists of all L-subsets of an n-set (equivalent to binary words on length n and weight L). By abuse of notation we won't distinguish between the representation by words or the one by L-subsets. The one which will be used will be understood from the context. The *Johnson distance* $d_J(x, y)$ between two L-subsets x and y is half of the related Hamming distance, i.e., $d_J(x, y) \triangleq d_H(x, y)/2$ which also implies that that $d_J(x, y) = |\operatorname{supp}(x) \setminus \operatorname{supp}(y)|$. The Johnson schemes $J(n, L)$ and $J(n, n-L)$ are isomorphic and hence w.l.o.g. we always assume that $2L \leqslant n$.

The Hamming scheme and the Johnson scheme (as well as some other schemes) are symmetric (the proof can be found in [21] as all these scheme are association schemes). There are other interesting and important schemes in coding theory, but we restrict ourselves to the previous metrics and ignore the others. For more information on other metrics and schemes the interested reader is referred to [21].

Our PPR schemes can be adapted to any symmetric scheme. We will leave this claim as an exercise to the reader and only refer in the sequel to the random permutation used in the scheme. For example, in the Hamming scheme over an arbitrary alphabet this permutation is exactly as in the binary Hamming scheme.

In our exposition there are three parameters — the radius r for the record of the user, the radius s from the proximity queries, and the diameter L of the scheme. For an element $x \in \mathcal{V}$, let $\mathcal{W}_s^x$ denote the set of words in $\mathcal{V}$ which are at distance s from x, i.e. $\mathcal{W}_s^x \triangleq \{z : d(z, x) = s\}$.

Given $x \in \mathcal{V}$, we want to express the ball $B(x, r)$ as an intersection of some other balls. The first step towards this goal is proved in the next result which generalizes one direction of Proposition 3.

Proposition 37: If r, s, L are nonnegative integers such that $L \geqslant r + 2s + 1$ and x is an element in a symmetric scheme $\mathcal{V}$ whose diameter is L, then

$$B(x, r) = \bigcap_{z \in \mathcal{W}_s^x} B(z, r + s) \ . \tag{12}$$

The proof of Proposition 37 is presented in Appendix B. Note, that the other side given in Proposition 3 for the binary Hamming scheme is not true for all the other related schemes. It is not difficult to verify this claim and it is also left for the interested reader.

As for the binary Hamming scheme, a PPRIC code C is a subset $C \subseteq \mathcal{W}_s^x$ such that $B(x, r) = \bigcap_{c \in C} B(c, r + s)$. We continue to examine PPRIC codes in symmetric schemes. For L, r, s, let us denote by $N_d(L, s, r)$ the minimum size of a PPRIC code with these parameters in a scheme with metric d. Some of our analysis for the binary Hamming scheme can be generalized for the Hamming scheme over an alphabet with q letters.

Theorem 38: If r, s, L are nonnegative integers such that $L \geqslant r + 2s + 1$, then $N_H(L, s, r) \leqslant N(L, s, r)$.

Proof: Let C be a binary (L, s, r) PPRIC code whose size is $N(L, s, r)$. We claim that C is also a PPRIC code over an alphabet of size $q > 2$. To complete the proof it is sufficient to show that for each word y of length L, over an alphabet with size q, whose weight is $r + t$, $t \geqslant 1$, there exists a codeword $c \in C$ such that $d_H(y, c) > r + s$. Let y be such a word and y' be the binary word obtained from y by replacing all the nonzero elements in y with *ones*. Since C is a binary PPRIC code, it follows that there exists a codeword $c \in C$ such that $d(y', c) > r + s$. Hence we have

$$d_H(y, c) \geqslant d(y', c) > r + s,$$

and the proof of the claim is completed. ■

Our next step is to generalize the concept of an MIPPR word for a PPRIC code over an arbitrary alphabet in the Hamming scheme. Given a PPRIC code C, over an alphabet of size q, a word v of length L is called an *intersection PPR word* related to C, if for each $c \in C$ there exists a coordinate i such that $v_i = c_i \neq 0$. Such a word v is called an MIPPR word if there is no intersection PPR word u such that $\operatorname{supp}(u) \subset \operatorname{supp}(v)$.

Lemma 39: Let r, s, and L be integers, such that $L \geqslant 2s + r + 1$, and let $C \subseteq \mathcal{W}_s$ be a PPRIC code over an alphabet with q letters. If v is an MIPPR word, then $wt(v) \geqslant r + 3$.

Proof: Assume to the contrary that v is an MIPPR word with respect to C and $wt(v) \leqslant r + 2$, and let y be any vector of weight $r + 2$ such that $v_i = y_i$ if $v_i \neq 0$. Since v in an MIPPR word, the definition of y implies that for all $c \in C$, y has at least one nonzero entry i such that $y_i = c_i$. Hence, for any $c \in C$, we have that

$$d(y, c) \leqslant wt(y) + wt(c) - 2 = r + 2 + s - 2 \leqslant r + s$$

and therefore, $y \in \bigcap_{c \in C} B(c, r + s)$, a contradiction to the assumption that C is a PPRIC code since $wt(y) > r$. ■

Corollary 40: For any admissible L, s, r, and an alphabet with at least two symbols, $N_H(L,s,r) \geqslant r+3$.

Combining Theorem 38, Corollary 23, and Corollary 40 implies

Corollary 41: If $L \geqslant s(r+3)$ in the Hamming scheme $H_q(L)$, then $N_H(L,s,r) = r+3$.

Corollary 41 implies that in the extreme case when $L \geqslant s(r+3)$ the size of optimal PPRIC codes is the same for all the Hamming schemes. This is not the case for all admissible values of L, s, and r. One can verify that

Theorem 42: For all admissible values of L, s, and r, if q is large enough then $N_H(L,s,r) = r+3$ over an alphabet with q letters.

Next, we discuss the Johnson scheme. We note again that the PPR scheme is defined exactly as in the Hamming scheme, with one exception. The PPRIC code, formed by the queries, is related to a binary word of length n and weight L. Clearly, all such words are isomorphic and hence w.l.o.g. we can take x to be the word with *ones* in the first L coordinates, and it is akin to the all-zero word in the Hamming scheme. The set $\mathcal{W}_s^x$ which replaces $\mathcal{W}_s$, consists of all words of length n with $L-s$ *ones* in the first L coordinates and s *ones* in the remaining $n-L$ coordinates. Finally, the random permutation (used for privacy) on the L coordinates in the Hamming scheme is replaced by two random permutations, one on the first L coordinates and a second one on the last $n-L$ coordinates. We continue to consider bounds on $N_J(L,s,r)$ and to establish a result which is analog to Corollary 23 and to Corollary 41.

Theorem 43: If $L \geqslant s(2r+3)$ in the Johnson scheme $J(n,L)$, then $N_J(L,s,r) \leqslant 2r+3$.

Theorem 44: If $L \geqslant s(2r+3)$ in the Johnson scheme $J(n,L)$, then $N_J(L,s,r) \geqslant 2r+3$.

The proofs of Theorem 43 and Theorem 44 are presented in Appendix B.

Corollary 45: If $L \geqslant s(2r+3)$ in the Johnson scheme $J(n,L)$, then $N_J(L,s,r) = 2r+3$.

PPRIC codes have their own combinatorial interest and not just as the minimum number of servers in the PPR schemes. For the Johnson schemes, an (n,L,k,t) *covering code* C is a collection of binary words of length n and weight L with $L-k$ *ones* in the first L coordinates, such that for each vector v of length n and weight L with $L-t$ *ones* in the first coordinates, there exists exactly one codeword $c \in C$ such that $d_J(c,v) = k-t$. These types of codes (designs) were considered for various problems, e.g. [11]. Bounds on their size were not considered before, but they can be easily derived from (n,k,t) covering design.

A *Steiner system* $S(t,k,n)$ is a collection of k-subsets (called *blocks*) of an n-set, such that each t-subset of the n-set is contained in exactly one block of the collection. Clearly, a Steiner system $S(t,k,n)$ is a covering design $C(t,k,n)$.

One can easily verify the following result.

Theorem 46: If C_1 is an (L,k,t) covering design and C_2 is an $(n-L,k,t)$ covering design, then $C \triangleq \bar{C}_1 \times C_2$ is an (n,L,k,t) covering code C in the Johnson scheme. Moreover, if C_1 is a Steiner system $S(t,k,L)$ and C_2 is a Steiner system $S(t,k,n-L)$, then C is a covering code of minimum size.

VII. Conclusion and Future Research

This paper studies a new family of protocols, called private proximity retrieval, which provide a certain form of private computation. Under this paradigm, the user has some file and is interested to retrieve the indices of all files in the database which are close to his file. As opposed to many existing privacy protocols, here perfect privacy cannot be achieved and the main goal of the paper is to determine the tradeoff between the user's privacy and the number of servers in the system. The construction of these protocols is mostly based on a new family of codes, called PPRIC codes, which have their own interest as covering codes. In the binary Hamming scheme they are constant weight codes whose balls with radius ℓ intersect in exactly a ball of radius r. These covering codes form a generalization for covering designs. Our exposition raises many problems for future research, some of which were mentioned alongside our discussion. These problems are of interest from both coding theory and combinatorics points of view. A few more directions are given in the next few paragraphs.

In general MIPPR words of (L,s,r) PPRIC codes do not have to be of weight $r+3$. For example, let C be the (L,s,r) PPRIC code C which contains all the $\binom{L}{s}$ words of length L and weight s. The support of any word v of weight $L-s$, have an empty intersection with the support of its binary complement $\bar{v}$, which has weight s and hence $\bar{v} \in C$. Therefore, any MIPPR word related to the PPRIC code C has weight $L-s+1$. Nevertheless, we have the following two conjectures (a strong one and a slightly weaker one) which are intriguing.

Conjecture 47: Any MIPPR word of a minimal (L,s,r) PPRIC code has weight $r+3$.

Conjecture 48: Any MIPPR word of minimum weight, in a minimal (L,s,r) PPRIC code, has weight $r+3$.

Clearly, the correctness of the claim in Conjecture 47 implies the claim in Conjecture 48. We believe that MIPPR words can throw light on the general structure of the related PPRIC code and hence we are interested in the results on MIPPR words.

Theorem 46 implies a simple way to construct optimal (n,L,k,t) covering codes. But, generally a product construction for two covering designs in the Hamming scheme does not yield an optimal covering code in the Johnson scheme. The simplest example for this claim can be viewed from the minimum covering design of size $r+3$ implied by Corollary 23 for the Hamming scheme, and the minimum covering code of size $2r+3$ implied by Corollary 45 for the Johnson scheme. It is clear that this leaves a lot of interesting questions concerning PPRIC codes, MIPPR words, and covering codes in the Johnson scheme. These questions are left for future research.

Another direction is to consider these problems for other metrics which are important in coding theory, e.g. the Grassmann metric. The *Grassmann scheme* $\mathcal{G}_q(n,L)$ consists of all L-dimensional 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 $d_G(x,y) \triangleq L - \dim(x \cap y)$. The Grassmann schemes $\mathcal{G}_q(n,L)$ and $\mathcal{G}_q(n,n-L)$ are isomorphic and hence

w.l.o.g. we can always assume that $2L \leqslant n$. Codes with this metric have found application mainly in error-correction for random network coding [13], [19].

Finally, an (n, k, t) *Turán design*, $n \geqslant k \geqslant t > 0$, is a collection S of t-subsets (called *blocks*) of an n-set, such that each k-subset of the n-set contains at least one block of S. There is a simple connection between covering designs and Turán designs implied by the complement of the design.

Theorem 49: S is an (n, k, t) covering design if and only if $\bar{S}$ is an $(n, n-t, n-k)$ Turán design.

Corollary 50: The supports of an (L, s, r) PPRIC code form an $(L, L-r-2, s)$ Turán design.

(n, k, t) Turán designs were considered only for small values of k and t, i.e. $n \geqslant 2k$. Hence, most of the results on Turán designs will not help in our exposition on PPRIC codes. Those which can help were translated into covering designs by Theorem 49. Hence, Turán designs were not considered in our exposition. But, since translation of covering codes into PPRIC code is also through complements of codewords, it might be that the techniques used for bounds on the sizes of Turán designs will be very useful for bounds on PPRIC codes.

Appendix A
Proof of Theorem 21

Assume that for $r = 0$ and $L/s = 2 + \epsilon$, where $\epsilon < \frac{1}{8}$, there exists an (L, s, r) PPRIC code with only 6 codewords A, B, C, D, E, F. If some five codewords have a nonempty intersection then obviously we can find two coordinates intersecting all codewords, a contradiction to Lemma 6. We distinguish now between four cases depending on the number of four codewords with a nonempty intersection. The distinction is between three such quadruples of intersection codewords (called later *intersecting quadruples* and similarly we have *intersecting triples*), exactly two such quadruples, exactly one such quadruple, or no such quadruple of codewords. Note that if there exist two such quadruples $\{A, B, C, D\}$ and $\{A, B, E, F\}$ then there exists an MIPPR word with weight two contradicting Lemma 6. Therefore, any two intersecting quadruples should have exactly three codewords in common. W.l.o.g. two of the quadruples are $\{A, B, C, D\}$ and $\{A, B, C, E\}$, a third quadruple can be w.l.o.g. either $\{A, B, C, F\}$ or $\{A, B, D, E\}$. It is easy to verify that in both cases there is no other quadruple which has three codewords in common with the other three quadruples. This implies that the four cases which follow are comprehensive. We continue with a lemma which will be used throughout the proof. A *type of intersecting triples* will be characterize by the codewords which participate in the nonempty intersection. Similarly *type of intersecting quadruples* is characterized.

Lemma 51: There do not exist $2t$ types of intersecting triples, where each codeword is contained in exactly t types of intersecting triples.

Proof: Let v be a word of weight $2t$ which intersects each type in one coordinate. The weight of v is $2t$ and it intersects each codeword in t coordinates. Hence, for $X \in \{A, B, C, D, E, F\}$ we have $d(v, X) = 2t + s - 2t = s$, a contradiction to (3). ∎

Now, we continue with the proof of Theorem 21 and distinguish between four cases.

Case 1: There are exactly three such intersecting quadruples.

W.l.o.g. the first two quadruples are $\{A, B, C, D\}$ and $\{A, B, C, E\}$, i.e. $A \cap B \cap C \cap D \neq \varnothing$ and $A \cap B \cap C \cap E \neq \varnothing$. W.l.o.g., a third quadruple can be either $\{A, B, C, F\}$ or $\{A, B, D, E\}$.

1) If $A \cap B \cap C \cap F \neq \varnothing$, then D, E, F should be pairwise disjoint (otherwise there exists an MIPPR word with weight two, contradicting Lemma 6). But, this implies that $|D \cup E \cup F| \geqslant 3s > L$, a contradiction.
2) If $A \cap B \cap D \cap E \neq \varnothing$, then to avoid an MIPPR word with weight two F should be disjoint from $C \cup D \cup E$ and hence $[L]$ is a union of three pairwise disjoint sets, F, $C \cap D \cap E$ and R. If $|C \cap D \cap E| = \beta s$, then it implies that

$$\begin{aligned} 3s &= |C| + |D| + |E| \\ &\leqslant 3|C \cap D \cap E| + 2|[L] \setminus (F \cup (C \cap D \cap E))| \\ &= 3\beta s + 2(1 + \epsilon - \beta)s = (2 + 2\epsilon + \beta)s, \end{aligned}$$

and hence $1 - 2\epsilon \leqslant \beta$, i.e. $|C \cap D \cap E| \geqslant (1 - 2\epsilon)s$. Therefore, we have

$$\begin{aligned} |R| &= |[L] \setminus (F \cup (C \cap D \cap E))| \\ &\leqslant (2 + \epsilon)s - s - (1 - 2\epsilon)s = 3\epsilon s. \end{aligned}$$

Clearly to avoid more intersecting quadruples, A is disjoint to $C \cap D \cap E$ and also B is disjoint to $C \cap D \cap E$. This implies that $|A \cap R| \leqslant 3\epsilon s$ and $|A \cap F| \geqslant (1 - 3\epsilon)s$ and similarly $|B \cap F| \geqslant (1 - 3\epsilon)s$ and hence $|A \cap F| + |B \cap F| \geqslant (2 - 6\epsilon)s > s = |F|$ and therefore $A \cap B \cap F \neq \varnothing$. Since also $C \cap D \cap E \neq \varnothing$, it follows that there exists an MIPPR word with weight two, contradicting Lemma 6.

Case 2: There are exactly two such quadruples of codewords.

W.l.o.g. the two quadruples are $\{A, B, C, D\}$ and $\{A, B, C, E\}$. To avoid an MIPPR word with weight two, F has to be disjoint from $D \cup E$. Let $|D \cap E| = \beta s$ and since F is disjoint from $D \cup E$, it follows that $D \cup E \subseteq [L] \setminus F$. This implies that

$$\begin{aligned} 2s = |D| + |E| &\leqslant 2|D \cap E| + |([L] \setminus (F \cup (D \cap E))| \\ &= 2\beta s + (1 + \epsilon - \beta)s = (1 + \epsilon + \beta)s, \end{aligned}$$

and hence $1 - \epsilon \leqslant \beta$, i.e. $|D \cap E| \geqslant (1 - \epsilon)s > 0$. Therefore, we have

$$|[L] \setminus (F \cup (D \cap E))| \leqslant (2 + \epsilon)s - s - (1 - \epsilon)s = 2\epsilon s.$$

Let $[L]$ be the union of three pairwise disjoint sets, F, $D \cap E$ and R. Assume first that $X \cap Y \cap F = \varnothing$, where $\{X, Y\} \subset \{A, B, C\}$. It implies that

$$\begin{aligned} &|A| + |B| + |C| \leqslant |A \cap F| + |B \cap F| + |C \cap F| + |A \cap D \cap E| \\ &+ |B \cap D \cap E| + |C \cap D \cap E| + |A \cap R| + |B \cap R| + |C \cap R| \\ &\leqslant |F| + |D \cap E| + 3|R| = (2 + \epsilon)s + 2 \cdot 2\epsilon s = 2s + 5\epsilon s < 3s, \end{aligned}$$

a contradiction. Therefore, w.l.o.g. we assume that $A \cap B \cap F \neq \varnothing$. This implies that $C \cap D \cap E = \varnothing$ to avoid an MIPPR word with weight two and hence $|C \cap F| = |C| - |C \cap D \cap E| - |C \cap R| \geqslant s - 0 - 2\epsilon s = (1-2\epsilon)s$. We continue and distinguish between two subcases.

Case 2.1: If $A \cap C \cap F \neq \varnothing$, then the same arguments imply that $|B \cap F| \geqslant (1-2\epsilon)s$. Hence, $|B \cap F| + |C \cap F| \geqslant (2-4\epsilon)s > s$ which is only possible if $B \cap C \cap F \neq \varnothing$. Therefore, the same arguments also imply that $|A \cap F| \geqslant (1-2\epsilon)s$. But now,

$$|A \cap F| + |B \cap F| + |C \cap F| \geqslant (3-6\epsilon)s > 2s,$$

which implies that $A \cap B \cap C \cap F \neq \varnothing$, contradicting the assumption that we have exactly two distinct intersecting quadruples.

Case 2.2: If $C \cap F$ is disjoint from $A \cup B$, then recall that $|C \cap F| \geqslant (1-2\epsilon)s$ and it implies that $|A \cap F| \leqslant 2\epsilon s$ and $|B \cap F| \leqslant 2\epsilon s$. Hence,

$$\begin{aligned}&|A \cap ([L] \setminus F)| + |B \cap ([L] \setminus F)| + |D \cap ([L] \setminus F)| + |E \cap ([L] \setminus F)| \\ &\geqslant (1-2\epsilon)s + (1-2\epsilon)s + s + s = 4s - 4\epsilon s > 3(1+\epsilon)s.\end{aligned}$$

Since $|[L] \setminus F| = (1+\epsilon)s$ it follows that there exists one coordinate in $[L] \setminus F$ which is contained in the four codewords A, B, D, E, i.e. $A \cap B \cap D \cap E \neq \varnothing$, a contradiction to the fact that there are exactly two intersecting quadruples.

Case 3: There is exactly one such intersecting quadruple of codewords. This case is more complicated and hence it will be considered separately in Lemma 52 which follows.

Case 4: There are no such intersecting quadruples of codewords.

Assume that there are exactly βs coordinates which are contained in three codewords. Hence, $6s = |A| + |B| + |C| + |D| + |E| + |F| \leqslant 3\beta s + 2(2+\epsilon-\beta)s$ since $(2+\epsilon-\beta)s$ coordinates are contained in at most two codewords. It implies that $\beta \geqslant 2 - 2\epsilon$. We continue and distinguish between two cases:

Case 4.1: Any two codewords are contained in some intersecting triple.

The *types* of candidates for these intersecting triples are as follows (the notation $\cap$ is omitted for simplicity).

ABC	ABD	ABE	ABF	ACD	ACE	ACF
ADE	ADF	AEF	DEF	CEF	CDF	CDE
BEF	BDF	BDE	BCF	BCE	BCD	

To avoid an MIPPR word with weight two there are no two triples from the same column in the code. Consider now the type of intersecting triples which are contained in the code. W.l.o.g we may assume that A is a codeword which appears the most number of times in distinct types of intersecting triples (other might appear the same number of times). There are βs coordinates which are contained in three codewords, where $\beta s \geqslant (2-2\epsilon)s > s$, and since $|A| = s$, it follows that not all the types contain A. Hence, w.l.o.g we may assume the existence of the intersecting triple $D \cap E \cap F$ in the code. Since B and C are involved in an intersecting triple and to avoid an MIPPR word with weight two it cannot be $A \cap B \cap C$, it follows that w.l.o.g we may assume the existence of the intersecting triple $B \cap C \cap F$. The updated table of candidates (where the red color (with underbar) is for a triple in the code and strike-through line is for a triple not in the code) is as follows.

~~ABC~~	ABD	ABE	ABF	ACD	ACE	ACF
~~ADE~~	ADF	AEF	$\underline{DEF}$	CEF	CDF	CDE
BEF	BDF	BDE	$\underline{BCF}$	BCE	BCD	

Since the number of existing types of triples containing A in the code is no less than that of the types containing F, we should have at least two types of triples in the code containing A but not containing F. The possible types are in the set$\{A \cap B \cap D,\ A \cap B \cap E,\ A \cap C \cap D,\ A \cap C \cap E\}$. If the types of intersecting triples $A \cap B \cap D$ and $A \cap C \cap E$ are in the code (or if $A \cap B \cap E$ and $A \cap C \cap D$ are in the code), then these two triples, together with $D \cap E \cap F$ and $B \cap C \cap F$ form a contradiction to Lemma 51. Hence, w.l.o.g the types $A \cap B \cap D$ and $A \cap B \cap E$ are in the code. The table of types in the code is updated as follows.

~~ABC~~	$\underline{ABD}$	$\underline{ABE}$	ABF	~~ACD~~	~~ACE~~	ACF
~~ADE~~	ADF	AEF	$\underline{DEF}$	~~CEF~~	~~CDF~~	CDE
BEF	BDF	BDE	$\underline{BCF}$	BCE	BCD	

Recall that each pair of codewords is part of a type which is contained in the code. Consider the pair $\{A, C\}$, one can readily verify that $A \cap C \cap F$ is the only possible type which remained in the table to be in the code. The three types $A \cap C \cap F$, $D \cap E \cap F$ and $A \cap B \cap E$ forbid the appearance of the type $B \cap C \cap D$ in the code by Lemma 51. Finally, there is a type containing $C \cap D$ in the code and the only remaining choice is $C \cap D \cap E$. The table for the types of triples is as follows.

~~ABC~~	$\underline{ABD}$	$\underline{ABE}$	ABF	~~ACD~~	~~ACE~~	$\underline{ACF}$
~~ADE~~	ADF	AEF	$\underline{DEF}$	~~CEF~~	~~CDF~~	$\underline{CDE}$
BEF	BDF	BDE	$\underline{BCF}$	BCE	~~BCD~~	

These six intersecting triples form a contradiction to Lemma 51.

Case 4.2: There is a pair of codewords, say E and F which are not contained together in any intersecting triple. This case is more complicated and hence it will be considered separately in Lemma 53 which follows.

Thus, all possible cases were analyzed and it was shown that in the given range there is no PPRIC code with six codewords and hence $N(L, s, 0) > 6$. This completes the proof of Theorem 21.

Lemma 52: If L and s are integers such that $L/s = 2 + \epsilon$, where $\epsilon < \frac{1}{8}$, then there is no $(L, s, 0)$ PPRIC code with six codewords A, B, C, D, E, F and exactly one intersecting quadruple.

Proof: W.l.o.g, the intersecting quadruple is $A \cap B \cap C \cap D$, which implies that E and F are disjoint, and hence $|A \cap B \cap C \cap D| = \pi s \leqslant \epsilon s$. Each one of the $2s$ coordinates of $E \cup F$ can support at most three codewords. Those which support exactly three codewords will be called *saturated*. Assume $E \cup F$ contains βs saturated coordinates and therefore each of the other $(2-\beta)s$ coordinates are contained in at most two codewords and hence they are unsaturated coordinates. $|A| + |B| + |C| + |D| + |E| + |F|$ is equal to $6s$ and it is also

at most $4\epsilon s+3\beta s+2(2-\beta)s$. Therefore, $6 \leqslant 4\epsilon+3\beta+2(2-\beta)$ and since $\epsilon < 1/8$, it follows that $\beta \geqslant 2-4\epsilon > \frac{3s}{2}$. We continue to show, by using other counting arguments, that the number of saturated coordinates in the code is smaller than $(2-4\epsilon)s$, which is a contradiction.

For the saturated coordinates we only have to consider intersecting triples which have the form $\{X, Y, Z\}$, where $X, Y \in \{A, B, C, D\}$ and $Z \in \{E, F\}$. There are twelve such possible intersecting triples, but to avoid an MIPPR word with weight two if $\{X_1, Y_1, Z_1\}$ is an intersecting triple, it follows that $\{X_2, Y_2, Z_2\}$, where $\{X_1, Y_1, Z_1\} \cap \{X_2, Y_2, Z_2\} = \varnothing$, is not an intersecting triple, and hence at most six such intersecting triples can be contained in the code. W.l.o.g. assume that $A \cap B \cap E$ is such an intersecting triple with the largest size and that $|A \cap B \cap E| = \gamma s$. Since there are only six possible intersecting triples and their total size is greater than $\frac{3s}{2}$, it follows that $6\gamma s > \frac{3s}{2}$ and hence $\gamma > \frac{1}{4}$.

If $C \cap D \cap E = \varnothing$, then we consider the distribution of C and D in the code. First note that since $|A \cap B \cap E| > \frac{1}{4}s$, it follows that $|E \cap C| + |E \cap D| < \frac{3s}{4}$. Moreover, to avoid an MIPPR word with weight two $C \cap D \cap F = \varnothing$ which implies that $|F \cap C| + |F \cap D| \leqslant s$. Therefore, the distribution of C and D on $[L]$ partitioned by E, F, and R is

$$2s = |C| + |D| \leqslant |E \cap C| + |E \cap D| + |F \cap C| + |F \cap D| + |R \cap C| + |R \cap D| < \frac{3s}{4} + s + 2\epsilon s < 2s,$$

a contradiction. Hence, the code has the intersecting triples $A \cap B \cap E$ and $C \cap D \cap E$ which implies that $A \cap B \cap F = \varnothing$ and $C \cap D \cap F = \varnothing$.

Clearly, there is at least one intersecting triple which contains F and w.l.o.g. we assume that $A \cap C \cap F \neq \varnothing$. Assume further that $A \cap C \cap F$ is the only intersecting triple which contains F. Since there are more than $\frac{3s}{2}$ saturated coordinates and at most s of them contain E, it follows that $|A \cap C \cap F| > \frac{s}{2}$. But, $|A \cap B \cap E| \geqslant |A \cap C \cap F|$ which implies that $|A| \geqslant |A \cap B \cap E| + |A \cap C \cap F| > s$, a contradiction. Hence, there is at least one more intersecting triple which contains F. We already have the intersecting triples $A \cap B \cap E$, $C \cap D \cap E$, and $A \cap C \cap F$, and by Lemma 51 $B \cap D \cap F$ cannot be an intersecting triple. Hence, w.l.o.g. a second intersecting triple which contains F is $B \cap C \cap F$.

Since $|A \cap B \cap E| = \gamma s$, it follows that $|D \cap E| \leqslant (1-\gamma)s$ and since $|A \cap B \cap C \cap D| \leqslant \epsilon s$ it follows that $|D \cap (E \cup F)| \geqslant (1-\epsilon)s$, which implies that $|D \cap F| \geqslant (\gamma - \epsilon)s$.

By Lemma 51 the code cannot contain the intersecting triples $A \cap D \cap F$, $B \cap D \cap F$, and since also $C \cap D \cap F$ is not an intersecting triple, it follows that all coordinates in $D \cap F$ are not saturated. Recall, that the number of saturated coordinates is $\beta s > \frac{3s}{2}$ and hence $|D \cap F| < s/2$, which implies that $(\gamma - \epsilon)s < s/2$ and hence $\gamma < 5/8$. Hence, there are more than $\frac{11s}{8}$ coordinates in $E \cup F$ which are not in $A \cap B \cap E$.

To summarize, we have four intersecting triples, $A \cap B \cap E$ for which we proved that $|A \cap B \cap E| < \frac{5s}{8}$, $C \cap D \cap E$, $A \cap C \cap F$, $B \cap C \cap F$; six intersecting triples were excluded from the code, and the two intersecting triples $A \cap C \cap E$ and $B \cap C \cap E$ may or may not be contained in the code. Note, that the last five intersecting triples contain C. We call the coordinates in $E \cup F$ which support the codeword C, *C-coordinates*. All the coordinates in $E \cup F$ which are not in $A \cap B \cap E$ are C-coordinates or unsaturated coordinates. Clearly, the number of C-coordinates is at most $(1-\pi)s$. The number of appearances of A, B, C, and D, in the coordinates of $E \cup F$ is at least $4s - 4\pi s - 3(\epsilon - \pi)s = (4 - 3\epsilon - \pi)s$ and hence the number of saturated coordinates is at least $(2-3\epsilon-\pi)s$ (since $|E \cup F| = 2s$). This implies that the number of coordinates (inside $E \cup F$) which are unsaturated or C-coordinates is at most $(3\epsilon + \pi)s + (1-\pi)s < \frac{11s}{8}$. But, these are exactly the coordinates in $E \cup F$ which are not in $A \cap B \cap E$. There are more than $\frac{11s}{8}$ such coordinates, a contradiction.

Thus, the lemma and as a consequence the related case in Theorem 21 are proved. ■

Lemma 53: Assume that L and s are integers such that $L/s = 2+\epsilon$, where $\epsilon < \frac{1}{8}$, and assume that there is no $(L, s, 0)$ PPRIC code which has six codewords A, B, C, D, E, F with no intersecting quadruple. Such a code where $\{E, F\}$ is not contained in any intersecting triple, does not exist.

Proof: We define the *saturated* coordinates to be the ones on which there are intersecting triples. There are a total of $(2+\epsilon)s$ coordinates and hence at least $6s - 2(2+\epsilon)s > \frac{7s}{4}$ saturated coordinates which implies that there are at most $(2+\epsilon)s - \frac{7s}{4} < \frac{3s}{8}$ unsaturated coordinates.

Since there is no intersecting triple of the form $X \cap E \cap F$, it follows that there are sixteen possible intersecting triples as follows

$$\begin{array}{cccccccc} ABE & ACE & ADE & BCE & BDE & CDE & ABC & ABD \\ ABF & ACF & ADF & BCF & BDF & CDF & ACD & BCD \end{array}$$

Claim: If $|X \cap Y \cap Z| = \lambda s$, then $\lambda \leqslant \frac{1}{2} + \epsilon$.

To verify the claim, we note that if $\{X, Y, Z\} \cup \{X', Y', Z'\} = \{A, B, C, D, E, F\}$, then X', Y', Z' are distributed only on $(2+\epsilon-\lambda)s$ coordinates. If $3s > 2(2+\epsilon-\lambda)s$ then $\{X', Y', Z'\}$ is an intersecting triple, contradicting Lemma 6. Hence, $3s \leqslant 2(2+\epsilon-\lambda)s$ implying that $\lambda \leqslant \frac{1}{2}+\epsilon$.

Assume that there are no two intersecting triples of the form $X \cap Y \cap P$ and $Z \cap W \cap P$, where $P \in \{E, F\}$, $\{X, Y, Z, W\} = \{A, B, C, D\}$. Then w.l.o.g there are only two possibilities:

1) the intersecting triples are chosen from

$$\{ABE, ACE, ADE, ABF, ACF, ADF, ABC, ABD, ACD, BCD\}$$

such that nine triples contain A and only $B \cap C \cap D$ does not contain A. By the Claim $|B \cap C \cap D| \leqslant (\frac{1}{2}+\epsilon)s$, and all the intersecting triples which contain A have size at most s, and hence the total size of these triples is at most $(1 + \frac{1}{2} + \epsilon)s$ which is smaller than $\frac{7s}{4}$, a contradiction.

2) the intersecting triples are chosen from

$$\{ABE, ACE, BCE, ABF, ACF, BCF, ABC, ABD, ACD, BCD\}.$$

Each triple has at least one pair from the set $\{A, B, C\}$. Since $|A| + |B| + |C| = 3s$ and each intersecting triple have a pair from these three codewords, it follows that the total size of these triples is at most $\frac{3s}{2}$ which is smaller than $\frac{7s}{4}$, a contradiction.

Hence, w.l.o.g. there are at least two such intersecting triples, say $A \cap B \cap E$ and $C \cap D \cap E$. We can also assume w.l.o.g. that $A \cap C \cap F$ is an intersecting triple. If there is no other intersecting triples which contain F, then by the claim the number of saturated coordinates which support F is at most $\frac{1}{2} + \epsilon$, and hence the total number of unsaturated coordinates supporting F is at least $\frac{1}{2} - \epsilon$ which is larger than $\frac{3s}{8}$, a contradiction.

Hence, there is another intersecting triple which contains F. By Lemma 51, $B \cap D \cap F$ is not an intersecting triple and hence w.l.o.g. $B \cap C \cap F$ is an intersecting triple. To summarize, we have the intersecting triples $\{ABE, CDE, ACF, BCF\}$. Now, any triple from the set $\{ACE, BCE, ABC, ABD, ACD, BCD\}$ can be also an intersecting triple. All the other triples cannot be intersecting triples.

Let $|C \cap D \cap E| = \gamma s$; note that every other intersecting triple has two codeword from the set $\{A, B, C\}$. Hence, the number of saturated coordinates is at most $(\gamma + \frac{3-\gamma}{2})s$. Since the total number of saturated coordinates is at least $(2-2\epsilon)s$ it follows that $(\gamma + \frac{3-\gamma}{2})s \geqslant (2-2\epsilon)s$, i.e. $\gamma > 1-4\epsilon$. All the coordinates in F which are not contained in $A \cap C \cap F$ or $B \cap C \cap F$ are unsaturated. The total number of unsaturated coordinates is at most $3\epsilon s$ and hence $|A \cap C \cap F| + |B \cap C \cap F| \geqslant (1-3\epsilon)s$. Therefore, we have that the total number of coordinates which support C is at least

$$|C \cap D \cap E| + |A \cap C \cap F| + |B \cap C \cap F| \geqslant (2-7\epsilon)s > s,$$

a contradiction.

Thus, the lemma and as a consequence the related case in Theorem 21 are proved. ■

Appendix B
Proofs for Results of Section VI

A. Proof of Proposition 37

Proof: Assume that $L \geqslant r + 2s + 1$ and let $y \in B(x, r)$, i.e., $d(x, y) \leqslant r$. Clearly for all $z \in \mathcal{W}_s^x$, by the triangle inequality $d(z, y) \leqslant d(z, x) + d(x, y) \leqslant s + r$, which implies that $y \in B(z, r + s)$. Hence,

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

To prove that the inclusion is actually an equality, let $y \in \mathcal{V}$ be such that $d(x, y) > r$. We will show that y is not contained in $\bigcap_{z \in \mathcal{W}_s^x} B(z, r + s)$. This will imply that if y is contained in $\bigcap_{z \in \mathcal{W}_s^x} B(z, r + s)$ then $d(x, y) \leqslant r$, i.e.

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

Assume first, that $r < d(x, y) \leqslant r + s$. Since the scheme is symmetric, $d(x, y) = r + i$, $0 < i \leqslant s$, and the diameter of the scheme is $L \geqslant r + 2s + 1$, it follows that there exists an element $z \in \mathcal{W}_s^x$ such that $d(y, z) = r + i + s > r + s$. Thus, $y \notin B(z, r + s)$.

Assume now, that $r + s < d(x, y) \leqslant r + 2s$. Since the scheme is symmetric, $d(x, y) = r + s + i$, $0 < i \leqslant s$, and the diameter of the scheme is $L \geqslant r + 2s + 1$, it follows that there exists an element $z' \in \mathcal{W}_{s-i+1}^x$ such that $d(y, z') = r + 2s + 1$. For any $z \in \mathcal{W}_s^x$ such that $d(z', z) = i - 1$ we have $d(z, y) \geqslant d(y, z') - (i-1) = r + 2s + 1 - (i-1) = r + 2s - i + 2 > r + s$. Thus, $y \notin B(z, r + s)$.

Finally, assume that $r + 2s < d(x, y)$, i.e., $d(x, y) = r + 2s + i$, $0 < i \leqslant s$. Hence, for all $z \in \mathcal{W}_s^x$ we have that $d(z, y) \geqslant r + 2s + i - s > r + s$ and therefore, $y \notin B(z, r + s)$.

This implies that $y \notin \bigcap_{z \in \mathcal{W}_s^x} B(z, r + s)$ and hence

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

■

B. Proofs of Theorem 43 and Theorem 44

The first lemma will be used in the proofs of Theorem 43 and Theorem 44.

Lemma 54: If x, y, z are three words in $J(n, L)$ such that $z = x \setminus \sigma \cup \beta$, where $\sigma \subset x$ and $\beta \cap x = \varnothing$, then

$$d_J(z, y) = |x \setminus y| + |\beta \setminus y| - |\sigma \setminus y| \ .$$

Proof: The claim is derived from the following sequence of equalities using the fact that x and β are disjoint and σ is contained in x:

$$\begin{aligned} d_J(z, y) &= |z \setminus y| = |(x \setminus \sigma \cup \beta) \setminus y| \\ &= |(x \cup \beta) \setminus (y \cup \sigma)| = |x \setminus y| + |\beta \setminus y| - |\sigma \setminus y| \ . \end{aligned}$$

■

1) Proof of Theorem 43:

Proof: Let $x \in \mathcal{V}$ and let $C = \{v_1, v_2, \ldots, v_{2r+3}\} \subseteq \mathcal{W}_s^x$ be a code with the following properties:

(p.1) $d_J(x, v_i) = s$ for each $1 \leqslant i \leqslant 2r + 3$.

(p.2) $d_J(v_i, v_j) = 2s$ for each $1 \leqslant i < j \leqslant 2r + 3$

Such a code C is easily constructed. Each v_i differs from x in s of the first L coordinates and s of the last $n - L$ coordinates. Any two distinct v_i's differ from x in s different coordinates. This implies that $L \geqslant s(2r + 3)$ as given and $n - L \geqslant s(2r + 3)$ which is also true since $n - L \geqslant L$. Next, we claim that

$$B(x, r) = \bigcap_{i=1}^{2r+3} B(v_i, r + s). \tag{13}$$

It is readily verified that $B(x, r) \subseteq \bigcap_{c \in C} B(c, r + s)$. Assume now that $y \in \mathcal{V}$ and $d_J(x, y) \geqslant r + 1$. To complete the proof it is sufficient to prove that $y \notin \bigcap_{c \in C} B(c, r + s)$.

Property (p.1) implies that $|x \setminus v_i| = s$ and $|v_i \cap ([n] \setminus x)| = s$. Property (p.2) implies that for $1 \leqslant i < j \leqslant 2r + 3$ we have that $|(x \setminus v_i) \cup (x \setminus v_j)| = 2s$ and $|(v_i \cap ([n] \setminus x)) \cup (v_j \cap ([n] \setminus x))| = 2s$.

Let $a_i \stackrel{\text{def}}{=} |y \cap (x \setminus v_i)|$ be the number of elements shared by x and y, which are not contained in v_i. Let $b_i \stackrel{\text{def}}{=} |y \cap v_i \cap ([n] \setminus x)|$ be the number of elements shared by y and v_i which are outside x. Now, we have

$$\begin{aligned} \sum_{i=1}^{2r+3} b_i &= \sum_{i=1}^{2r+3} |y \cap v_i \cap ([n] \setminus x)| \\ &= |y \cap \bigcup_{i=1}^{2r+3} (v_i \cap ([n] \setminus x))| \leqslant d_J(x, y), \end{aligned}$$

and similarly

$$\sum_{i=1}^{2r+3} a_i = \sum_{i=1}^{2r+3} |y \cap (x \setminus v_i)|$$
$$= |y \cap \bigcup_{i=1}^{2r+3} (x \setminus v_i)| \geqslant (2r+3)s - d_J(x,y) \ .$$

For each i, $v_i = x \setminus (x \setminus v_i) \cup (v_i \cap ([n] \setminus x))$, where $x \setminus v_i \subseteq x$ and $(v_i \cap ([n] \setminus x)) \cap x = \varnothing$, and hence by Lemma 54 we have

$$d_J(v_i, y) = |v_i \setminus y| = |x \setminus y| + |(v_i \cap ([n] \setminus x)) \setminus y| - |(x \setminus v_i) \setminus y|$$

which is equal to

$$= d_J(x,y) + s - b_i - (s - a_i) = d_J(x,y) + a_i - b_i \ .$$

Therefore,

$$\sum_{i=1}^{2r+3} d_J(y, v_i) = d_J(x,y)(2r+3) + \sum_{i=1}^{2r+3} a_i - \sum_{i=1}^{2r+3} b_i$$
$$\geqslant (d_J(x,y) + s)(2r+3) - 2d_J(x,y)$$
$$= (2r+1)d_J(x,y) + s(2r+3)$$
$$\geqslant (2r+1)(r+1) + s(2r+3) > (r+s)(2r+3),$$

and hence there exists some $v_i \in C$ such that $d_J(y, v_i) > r + s$.

Thus, $y \notin B(v_i, r+s)$ and the equality

$$B(x,r) = \bigcap_{i=1}^{2r+3} B(v_i, r+s)$$

follows. ■

2) Proof of Theorem 44:

Proof: Assume to the contrary that $N_J(L,r,s) \leqslant 2r+2$, i.e., there exists a PPRIC code $C = \{v_1, v_2, \ldots, v_{2r+2}\} \subseteq \mathcal{W}_s^x$ which implies that $B(x,r) = \bigcap_{c \in C} B(c, r+s)$.

Let $u_i = v_i \setminus x$ and $t_i = x \setminus v_i$ for $1 \leqslant i \leqslant 2r+2$, i.e., $v_i = (x \setminus t_i) \cup u_i$. For each $1 \leqslant i \leqslant r+1$ let $a_i \in t_i$ be any arbitrary element in $\mathrm{supp}\ (x)$ which is contained in $\mathrm{supp}\ (v_i)$. For each $r+2 \leqslant i \leqslant 2r+2$ let $b_i \in u_i$ be an arbitrary element in $\mathrm{supp}\ (v_i)$ which is not contained in $\mathrm{supp}(x)$.

Let σ be a subset of x with size $r+1$ which contains $\{a_i \ : \ 1 \leqslant i \leqslant r+1\}$ (note that some of the a_i's might be the same). Similarly, let β be a subset of $[n] \setminus x$ with size $r+1$ which contains $\{b_i \ : \ r+2 \leqslant i \leqslant 2r+2\}$.

Define the word $y = x \setminus \sigma \cup \beta$. Since $x \in J(n,L)$, $|\sigma| = |\beta| = r+1$, $\sigma \subset x$, and $\beta \cap x = \varnothing$, it follows that $y \in J(n,L)$ and $d_J(x,y) = r+1$ and hence $y \notin B(x,r)$. Next, we compute a lower bound on the distance between y and each codeword of C. Since $v_i = (x \setminus t_i) \cup u_i$, where $t_i \subseteq x$ and $u_i \cap x = \varnothing$, it follows that Lemma 54 can be applied to obtain

$$d_J(v_i, y) = |x \setminus y| + |u_i \setminus y| - |t_i \setminus y| \ .$$

We distinguish between two cases:

Case 1: If $1 \leqslant i \leqslant r+1$, then $|x \setminus y| = d_J(x,y) = r+1$, $|u_i| = s$ and hence $|u_i \setminus y| \leqslant s$. Furthermore, since $a_i \in t_i$ and $a_i \notin y$, it follows that $|t_i \setminus y| \geqslant 1$ and therefore,

$$d_J(v_i, y) = |x \setminus y| + |u_i \setminus y| - |t_i \setminus y| \leqslant r+1+s-1 = r+s.$$

Case 2: If $r+2 \leqslant i \leqslant 2r+2$, then $|x \setminus y| = d_J(x,y) = r+1$, $|u_i| = s$ and since $b_i \in u_i \cap y$, it follows that $|u_i \setminus y| \leqslant s-1$. Furthermore, $|t_i \setminus y| \geqslant 0$ and therefore,

$$d_J(v_i, y) = |x \setminus y| + |u_i \setminus y| - |t_i \setminus y| \leqslant r+1+s-1 = r+s.$$

Thus, by Cases 1 and 2 we have that for each $1 \leqslant i \leqslant 2r+2$, $d(v_i, y) \leqslant r+s$, i.e. $y \in \bigcap_{c \in C} B(c, r+s)$, a contradiction to $B(x,r) = \bigcap_{c \in C} B(c, r+s)$ since $y \notin B(x,r)$. Thus, $N_J(L,r,s) \geqslant 2r+3$. ■

ACKNOWLEDGMENT

The authors are deeply indebted to Oliver Gnilke and David Karpuk for many helpful discussions and for their collaboration and contribution to the results of this work which were presented in [12]. They are also thankful for the three anonymous reviewers for their helpful comments in improving the presentation of the paper.

REFERENCES

[1] 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.

[2] K. Banawan and S. Ulukus, "The capacity of private information retrieval from Byzantine and colluding databases," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1206–1219, Feb. 2019.

[3] A. Beimel, Y. Ishai, and E. Kushilevitz, "General constructions for information-theoretic private information retrieval," *J. Comput. Syst. Sci.*, vol. 71, no. 2, pp. 213–247, 2005.

[4] A. Beimel, Y. Ishai, E. Kushilevitz, and J. F. Raymond, "Breaking the $O(n^{1/(2k-1)})$ barrier for information theoretic private information retrieval," in *Proc. 43rd Symp. Found. Comput. Sci.*, Vancouver, BC, USA, 2002, pp. 261–270.

[5] A. Beimel, Y. Ishai, and T. Malkin, "Reducing the servers computation in private information retrieval: PIR with preprocessing," in *Proc. 20th Annu. Int. Cryptol. Conf.*, in Lecture Notes in Computer Science, vol. 1880. Santa Barbara, CA, USA: Springer, 2000, pp. 55–73.

[6] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," 2014, *arXiv:1410.5489*. [Online]. Available: http://arxiv.org/abs/1410.5489

[7] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Nov. 2015, pp. 2842–2846.

[8] E. Chávez, K. Figueroa, and G. Navarro, "Effective proximity retrieval by ordering permutations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1647–1658, Sep. 2008.

[9] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.

[10] Z. Dvir and S. Gopi, "2-server PIR with subpolynomial communication," *J. ACM*, vol. 63, no. 4, Nov. 2016, Art. no. 39.

[11] T. Etzion, "On the nonexistence of perfect codes in the Johnson scheme," *SIAM J. Discrete Math.*, vol. 9, no. 2, pp. 201–209, May 1996.

[12] T. Etzion, O. W. Gnilke, D. Karpuk, E. Yaakobi, and Y. Zhang, "Private proximity retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 2119–2123.

[13] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1165–1173, Feb. 2011.

[14] T. Etzion, V. Wei, and Z. Zhang, "Bounds on the sizes of constant weight covering codes," *Des., Codes Cryptogr.*, vol. 5, no. 3, pp. 217–239, May 1995.

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

[16] D. M. Gordon. [Online]. Available: https://www.dmgordon.org/cover

[17] D. M. Gordon, O. Patashnik, and G. Kuperberg, "New constructions for covering designs," *J. Combinat. Des.*, vol. 3, no. 4, pp. 269–284, 1995.

[18] D. Karpuk, "Private computation of systematically encoded data with colluding servers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 2018, pp. 2112–2116.

[19] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.
[20] S. Kumar, H.-Y. Lin, E. Rosnes, and A. G. I. Amat, "Achieving maximum distance separable private information retrieval capacity with linear codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4243–4273, Jul. 2019.
[21] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, May 1978.
[22] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020.
[23] W. H. Mills, "Covering designs I: Coverings by a small number of subsets," *ARS Combin.*, vol. 8, pp. 199–315, 1979.
[24] W. H. Mills and R. C. Mullin, "Coverings and packings," in *Contemporary Design Theory: A Collection of Surveys*, J. H. Dinitz and D. R. Stinson, Eds. New York, NY, USA: Wiley, 1992, pp. 371–399.
[25] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *Proc. Iran Workshop Commun. Inf. Theory (IWCIT)*, Apr. 2018, pp. 1–6.
[26] M. Morley and G. H. van Rees, "Lottery schemes and covers," *Utilitas Math.*, vol. 79, pp. 159–165, 1990.
[27] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2011, pp. 1–17.
[28] J. D. Nielsen, J. I. Pagter, and M. B. Stausholm, "Location privacy via actively secure private proximity testing," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops*, Lugano, Switzerland, Mar. 2012, pp. 381–386.
[29] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Capacity of private linear computation for coded databases," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Oct. 2018, pp. 813–820.
[30] C. Patsakis, P. Kotzanikolaou, and M. Bouroche, "Private proximity testing on steroids: An NTRU-based protocol," in *Security and Trust Management* (Lecture Notes in Computer Science), vol. 9331. Springer, 2015, pp. 172–184.
[31] N. Raviv and D. A. Karpuk, "Private polynomial computation from Lagrange encoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, vol. 15, Jul. 2019, pp. 553–563.
[32] J. Schönheim, "On coverings," *Pacific J. Math.*, vol. 14, no. 4, pp. 1405–1411, 1964.
[33] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.
[34] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2361–2370, Apr. 2018.
[35] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.
[36] 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.
[37] D. T. Todorov, "On some covering designs," *J. Combinat. Theory A*, vol. 39, no. 1, pp. 83–101, May 1985.
[38] Q. Wang and M. Skoglund, "Linear symmetric private information retrieval for MDS coded distributed storage with colluding servers," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2017, pp. 71–75.
[39] W. Gasarch, "A survey on private information retrieval," *Bull. EATCS*, vol. 82, pp. 72–107, Feb. 2004.
[40] D. Woodruff and S. Yekhanin, "A geometric approach to information-theoretic private information retrieval," in *Proc. 20th Annu. IEEE Conf. Comput. Complex. (CCC)*, Jun. 2005, pp. 275–284.
[41] S. Yekhanin, "Private information retrieval," *Commun. ACM*, vol. 53, no. 4, pp. 68–73, Apr. 2010.
[42] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *J. ACM*, vol. 55, no. 1, pp. 1–16, Feb. 2008.

Yiwei Zhang received the B.A. and Ph.D. degrees in mathematics from Zhejiang University, Hangzhou, Zhejiang, China, in 2011 and 2016, respectively. From 2016 to 2017, he was a Post-Doctoral Researcher with the School of Mathematical Sciences, Capital Normal University, Beijing, China. From 2017 to 2019, he was a Post-Doctoral Researcher with the Department of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel. He is currently a Professor with the School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China. He is also with the Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University. His research interests include coding theory, extremal combinatorics, and their interactions.

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

Tuvi Etzion (Fellow, IEEE) was born in Tel Aviv, Israel, in 1956. He received the B.A., M.Sc., and D.Sc. degrees from the Technion–Israel Institute of Technology, Haifa, Israel, in 1980, 1982, and 1984, respectively.

In 1984, he held a position with the Department of Computer Science, Technion, where he is currently the Bernard Elkin Chair in computer science. From 1985 to 1987, he was a Visiting Research Professor with the Department of Electrical Engineering Systems, University of Southern California, Los Angeles. During the Summer of 1990 and 1991, he was visiting Bellcore, Morristown, New Jersey. From 1994 to 1996, he was a Visiting Research Fellow with the Computer Science Department, Royal Holloway University of London, Egham, England. He also had several visits to the Coordinated Science Laboratory, University of Illinois in Urbana–Champaign from 1995 to 1998, two visits to HP Bristol during the Summer of 1996 and 2000, a few visits to the Department of Electrical Engineering, University of California at San Diego, from 2000 to 2017, several visits to the Mathematics Department, Royal Holloway University of London, from 2007 to 2017, a few visits to the School of Physical and Mathematical Science (SPMS), Nanyang Technological University, and to the Department of Industrial Systems Engineering and Management the National University of Singapore, Singapore, from 2016 to 2019, and a few visits to Jiaotong University, Beijing, from 2017 to 2019. His research interests include applications of discrete mathematics to problems in computer science and information theory, coding theory, coding for storage, and combinatorial designs.

Dr. Etzion was an Associate Editor of Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2006 till 2009. From 2004 to 2009, he was an Editor of the *Journal of Combinatorial Designs*. Since 2011, he has been an Editor of *Designs, Codes, and Cryptography* and an Editor for *Advances of Mathematics in Communications* since 2013. He has been an Editor-in-Chief for the *Journal of Combinatorial Theory, Series A,* since 2021.