Nearly Optimal Constructions of PIR and Batch Codes

Hilal Asi
Technion - Israel Institute of Technology
Haifa 32000, Israel
shelal@cs.technion.ac.il

Eitan Yaakobi
Technion - Israel Institute of Technology
Haifa 32000, Israel
yaakobi@cs.technion.ac.il

***Abstract*—In this work we study two families of codes with availability, namely *private information retrieval (PIR) codes* and *batch codes*. While the former requires that every information symbol has k mutually disjoint recovering sets, the latter asks this property for every multiset request of k information symbols. The main problem under this paradigm is to minimize the number of redundancy symbols. We denote this value by $r_P(n,k), r_B(n,k)$, for PIR, batch codes, respectively, where n is the number of information symbols. Previous results showed that for any constant k, $r_P(n,k) = \Theta(\sqrt{n})$ and $r_B(n,k) = \mathcal{O}(\sqrt{n}\log(n))$. In this work we study the asymptotic behavior of these codes for non-constant k and specifically for $k = \Theta(n^\epsilon)$. We also study the largest value of k such that the rate of the codes approaches 1, and show that for all $\epsilon < 1$, $r_P(n, n^\epsilon) = o(n)$, while for batch codes, this property holds for all $\epsilon < 0.5$.**

I. Introduction

In this paper we study two families of codes with availability for distributed storage. The first family of codes, called *private information retrieval* (*PIR*) *Codes*, requires that every information symbol has some k mutually disjoint recovering sets. These codes were studied recently in [2] due to their applicability for private information retrieval in a coded storage system. They are also very similar to *one-step majority-logic decodable codes* that were studied a while ago by Massey [7] and later by Lin and others [5] and were prompted by applications of error-correction with low-complexity.

The second family of codes, which is a generalization of the first one, was first proposed in the last decade by Ishai et al. under the framework of *batch codes* [3]. These codes were originally motivated by different applications such as load-balancing in storage and cryptographic protocols. Here it is required that every multiset request of k symbols can be recovered by k mutually disjoint recovering sets.

Formally, we denote a k-PIR code by $[N,n,k]^P$ to be a coding scheme which encodes n information bits to N bits such that each information bit has k mutually disjoint recovering sets. Similarly, a k-batch code will be denoted by $[N,n,k]^B$ and the requirement of mutually disjoint recovering sets is imposed for every multiset request of size k. The main figure of merit when studying PIR and batch codes is the value of N, given n and k. Thus, we denote by $P(n,k), B(n,k)$ the minimum value of N for which an $[N,n,k]^P, [N,n,k]^B$ code exists, respectively.

Since it is known that for all fixed k, $\lim_{n\to\infty} B_q(n,k)/n = \lim_{n\to\infty} P_q(n,k)/n = 1$, [3], we evaluate these codes by their redundancy and define $r_B(n,k) \triangleq B(n,k) - n, r_P(n,k) \triangleq P(n,k) - n$. One of the problems we study in the paper studies the largest value of k (as a function of n) for which one can still have $r_P(n,k) = o(n)$ and $r_B(n,k) = o(n)$, so the rate of the codes approaches 1. We show that for PIR codes this holds for $k = \Theta(n^\epsilon)$, for all $\epsilon < 1$, while for batch codes for all $\epsilon < 1/2$. Since $r_P(n,k), r_B(n,k) \geqslant k$, the result for PIR codes is indeed optimal. Furthermore, in order to have a better understanding of the asymptotic behavior of the redundancy, we study the values $r_P(n,k)$ and $r_B(n,k)$ when $k = \Theta(n^\epsilon)$.

The results we achieve in the paper are based on two constructions. The first one uses *multiplicity codes* which generalized Reed Muller codes and were first presented by Kopparty et al. in [4]. These codes were also used for the construction of *locally decodable codes* [11]. The second construction we use is based on the subcube construction from [3]. This basic construction can be used to construct both PIR and batch codes. While the idea in the works in [2], [3] was to use multidimensional cubes in order to achieve large values of k, here we take a different approach and position the information bits in a two dimensional array and then form multiple parity sets by taking different diagonals in the array.

The rest of the paper is organized as follows. In Section II, we formally define the codes studied in this paper and review previous results. In Section III, we review multiplicity codes. Then, in Section IV we show how to use multiplicity codes to construct PIR codes, and in Section V we carry the same task for batch codes. Then, in Section VI, we present our array construction and its results for PIR codes and batch codes. Due to the lack of space some proofs in the paper are omitted.

II. Definitions and Preliminaries

Let $\mathbb{F}_q$ denote the field of size q, where q is a prime power. A linear code of length N and dimension n over $\mathbb{F}_q$ will be denoted by $[N,n]_q$. For binary codes we will remove the notation of the field. The set $[n]$ denotes the set of integers $\{1,2,\ldots,n\}$.

In this work we focus on two families of codes, namely *private information retrieval* (*PIR*) codes that were defined recently in [2] and *batch codes* that were first studied by Ishai et al. in [3]. Formally, these codes are defined as follows.

Definition 1. *Let $\mathcal{C}$ be an $[N,n]_q$ linear code over the field $\mathbb{F}_q$.*

1) *The code $\mathcal{C}$ will be called a k-**PIR code**, and will be denoted by $[N,n,k]_q^P$, if for every information symbol $x_i, i \in [n]$, there exist k mutually disjoint sets $R_{i,0},\ldots,R_{i,k-1} \subseteq [N]$ such that for all $j \in [k]$, x_i is a function of the symbols in $R_{i,j}$.*
2) *The code $\mathcal{C}$ will be called a k-**batch code**, and will be denoted by $[N,n,k]_q^B$, if for every multiset request of symbols $\{i_0,i_1,\ldots,i_{k-1}\}$, there exist k mutually disjoint sets $R_{i_0},R_{i_1},\ldots,R_{i_{k-1}} \subseteq [N]$ such that for all $j \in [k]$, x_{i_j} is a function of the symbols in R_{i_j}.*

We slightly modified here the definition of batch codes. In their conventional definition, n symbols are encoded into some m tuples of strings, called buckets, such that each batch (i.e. request) of k information symbols can be decoded by reading at most some t symbols from each bucket. In case each bucket can store a single symbol, these codes are called *primitive batch codes*, which is the setup we study here and for simplicity call them batch codes. In this work we study the binary and non-binary cases of PIR and batch codes.

978-1-5090-4096-4/17/$31.00 ©2017 IEEE

The main problem in studying PIR and batch codes is to minimize the length N given the values of n and k. We denote by $P_q(n,k), B_q(n,k)$ the value of the smallest N such that there exists an $[N,n,k]_q^P, [N,n,k]_q^B$ code, respectively. Since every batch code is also a PIR code with the same parameters we get that $B_q(n,k) \geqslant P_q(n,k)$. For the binary case, we will remove q from these and subsequent notations.

In [3], it was shown using the subcube construction that for any fixed k there exists an asymptotically optimal construction of $[N,n,k]_q^B$ batch code, and hence

$$\lim_{n\to\infty} B_q(n,k)/n = \lim_{n\to\infty} P_q(n,k)/n = 1.$$

Therefore, it is important to study how fast the rate of these codes converges to one, and so the redundancy of PIR and batch codes is studied. We define $r_B(n,k)_q$ to be the value $r_B(n,k)_q \triangleq B(n,k)_q - n$ and similarly, $r_P(n,k)_q \triangleq P(n,k)_q - n$.

In [2], it was shown that for any fixed $k \geqslant 3$ there exists an $[N,n,k]$ PIR code where $N = n + \mathcal{O}(\sqrt{n})$, so $r_P(n,3) = \mathcal{O}(\sqrt{n})$ and in [8] it was proved that $r_P(n,3) = \Theta(\sqrt{n})$, by providing a lower bound on the redundancy of 3-PIR codes. These results assure also that for any fixed k, $r_P(n,k) = \Theta(\sqrt{n})$ and also implied that for any fixed k, $r_B(n,k) = \Omega(\sqrt{n})$. In [10], it was proved that for $k = 3,4$, $r_B(n,k) = \Theta(\sqrt{n})$, and for any fixed $k \geqslant 5$, $r_B(n,k) = \mathcal{O}(\sqrt{n}\log(n))$. In this paper, we will mostly study the values of $r_P(n,k)$ and $r_B(n,k)$, when k is a function of n, for example $k = \Theta(n^\epsilon)$. One of the problems we will also investigate is finding the largest ϵ for which $r_P\big(n, k = \Theta(n^\epsilon)\big) = o(n)$, and similarly for batch codes.

There are several more constructions of PIR and batch codes, which we summarize below.

1) $r_B(n, n^{1/3}) \leqslant n$, [9].
2) $r_B(n, n^\epsilon) \leqslant n^{7/8}$ for $7/32 \leqslant \epsilon \leqslant 1/4$, [9].
3) $r_B(n, n^\epsilon) \leqslant n^{4\epsilon}$ for $1/5 < \epsilon \leqslant 7/32$, [9].
4) $B(n,n) \leqslant 2n^{1.5}$, [1].
5) $r_P(n, \sqrt{n}) = O(n^{(\log 3)/2})$, [5].
6) $r_P(n, n^\epsilon) = O(n^{0.5+\epsilon})$, [5].

III. Mutliplicity Codes

In this section we review the construction of *multiplicity codes*. This family of codes was first presented by Kopparty et al. in [4] as a generalization of Reed Muller codes by calculating the derivatives of polynomials. We follow the definitions of these codes as were presented in [4] and first start with the definition of the Hasse derivative.

For a field $\mathbb{F}$, let $\mathbb{F}[x_1,\ldots,x_s] = \mathbb{F}[\boldsymbol{x}]$ be the ring of polynomials in the variables $x_1,\ldots,x_s$ with coefficients in $\mathbb{F}$. For a vector $\boldsymbol{i} = (i_1,\ldots,i_s)$ of non-negative integers, its weight $wt(\boldsymbol{i})$ is $\sum_{j=1}^s i_j$, and let $\boldsymbol{x}^{\boldsymbol{i}}$ denote the monomial $\prod_{j=1}^s x_j^{i_j}$. The total degree of this monomial equals $wt(\boldsymbol{i})$. For $P(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$, let the degree of $P(\boldsymbol{x})$, $\deg(P)$, be the maximum total degree over all monomials in $P(\boldsymbol{x})$.

Definition 2. *For a polynomial $P(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ and a non-negative vector $\boldsymbol{i}$, the $\boldsymbol{i}$-th **Hasse derivative** of $P(\boldsymbol{x})$, denoted by $P^{(\boldsymbol{i})}(\boldsymbol{x})$, is the coefficient of $\boldsymbol{z}^{\boldsymbol{i}}$ in the polynomial $P'(\boldsymbol{x},\boldsymbol{z}) = P(\boldsymbol{x}+\boldsymbol{z}) \in \mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$.*

Definition 3. *Let m,d,s be nonnegative integers and let q be a prime power. Let $\Sigma = \mathbb{F}_q^{|\{\boldsymbol{i}:wt(\boldsymbol{i})<m\}|} = \mathbb{F}_q^{\binom{s+m-1}{s}}$. For a polynomial $P(x_1,\ldots,x_s) \in \mathbb{F}_q[x_1,\ldots,x_s]$, we define the order m evaluation of P at $\boldsymbol{w} \in \mathbb{F}_q^s$, denoted by $P^{(<m)}(\boldsymbol{w})$, to be the vector*

$$P^{(<m)}(\boldsymbol{w}) = \big(P^{(\boldsymbol{i})}(\boldsymbol{w})\big)_{\boldsymbol{i}:wt(\boldsymbol{i})<m} \in \Sigma.$$

*The **multiplicity code** $\mathcal{C}(m,d,s,q)$ of order m evaluations of degree d polynomials in s variables is defined as follows. The code is over Σ, has length q^s, and its coordinates are indexed by elements in $\mathbb{F}_q^s$. For each polynomial $P(\boldsymbol{x}) \in \mathbb{F}_q[x_1,\ldots,x_s]$ with $\deg(P) \leqslant d$, there is a codeword in $\mathcal{C}$ given by: $Enc_{m,d,s,q}(P) = \big(P^{(<m)}(\boldsymbol{w})\big)_{\boldsymbol{w}\in\mathbb{F}_q^s} \in (\Sigma)^{q^s}$. That is,*

$$\mathcal{C}(m,d,s,q) = \{\big(P^{(<m)}(\boldsymbol{w})\big)_{\boldsymbol{w}\in\mathbb{F}_q^s} \in \Sigma^{q^s} : P \in \mathbb{F}_q[\boldsymbol{x}], \deg(P) \leqslant d\}.$$

The following lemma was proved in [4], Lemma 9.

Lemma 4. *The multiplicity code $\mathcal{C}(m,d,s,q)$ has relative distance at least $\delta = 1 - \frac{d}{mq}$ and rate $\binom{d+s}{s}/\binom{s+m-1}{s}q^s$.*

Lastly, we note that since the multiplicity code $\mathcal{C}(m,d,s,q)$ is a linear code it can also be a systematic code and thus for the rest of the paper we assume these codes to be systematic; for more details see Lemma 2.3 in [11]. For the rest of the paper and unless stated otherwise, we assume that m,d,s,q are positive integers.

IV. PIR Codes from multiplicity codes

In [4], multiplicity codes were used to construct *locally decodable codes* in order to retrieve the value of a single symbol with high probability, given that at most a fixed fraction of the codeword's symbol has errors [11]. Since we are not concerned with errors, we modify the recovering procedure so that each information symbol has a large number of disjoint recovering sets. For this end, we establish several properties on interpolation sets of polynomials which will help us later to construct the recovering sets, and thus PIR and batch codes.

Lemma 5. *Let $P(\boldsymbol{x}) \in \mathbb{F}_q[x_1,\ldots,x_s]$ be an homogeneous polynomial[1] such that $\deg(P) = d$. Let $A_1,\ldots,A_{s-1}$ be subsets of $\mathbb{F}_q$ such that $|A_i| = d+1$. Then the set $A = A_1 \times \cdots \times A_{s-1} \times \{1\}$ is an interpolation set[2] of $P(\boldsymbol{x})$, where $1 \in \mathbb{F}_q$ is the unitary element of the field.*

The following definition will be used in the construction of recovering sets for multiplicity codes.

Definition 6. *Let $\mathbb{F}_q$ be a field, and $S_1, S_2 \subseteq \mathbb{F}_q^s$ where s is a positive integer. We say that the sets S_1 and S_2 are **disjoint under multiplication** if for every $x \in S_1$ and $\alpha \in \mathbb{F}_q \setminus \{0\}$ it holds that $\alpha x \notin S_2$.*

Lemma 7. *Let $P(\boldsymbol{x}) \in \mathbb{F}_q[x_1,\ldots,x_s]$ be an homogeneous polynomial such that $\deg(P) = d$. Then there exists $\lfloor \frac{q}{d+1} \rfloor^{s-1}$ interpolation sets of $P(\boldsymbol{x})$, each of size $(d+1)^{s-1}$, which are mutually disjoint under multiplication.*

Now we are in a good position to present the recovering procedure for multiplicity codes. First, we show a general structure of the recovering sets, and then we argue that many disjoint sets can be constructed this way.

Theorem 8. *Let m,d,s,q be such that $d/m < q-1$, and $\mathcal{C} = \mathcal{C}(m,d,s,q)$ is the multiplicity code of length q^s over $\mathbb{F}_q^{\binom{s+m-1}{s}}$. Let $A \subseteq \mathbb{F}_q^s$ be an interpolation set for homogeneous polynomials of degree at most $m-1$. Then, for*

[1] We say that $P(\boldsymbol{x}) \in \mathbb{F}_q[\boldsymbol{x}]$ is homogeneous if all the monomials of $P(\boldsymbol{x})$ have the same total degree.

[2] For $P(\boldsymbol{x}) \in \mathbb{F}_q[x_1,\ldots,x_s]$ and $R \subseteq \mathbb{F}_q^s$, we say that R is an interpolation set of $P(\boldsymbol{x})$ if for every polynomial $Q(\boldsymbol{x})$ such that $P(\boldsymbol{x}) = Q(\boldsymbol{x})$ for every $\boldsymbol{x} \in R$, it holds that $P(\boldsymbol{x}) = Q(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathbb{F}_q^s$.

every $\boldsymbol{y} = (y_{\boldsymbol{w}})_{\boldsymbol{w}\in\mathbb{F}_q^s} \in \mathcal{C}$, *and for any* $\boldsymbol{w}_0 \in \mathbb{F}_q^s$, *the set of coordinates indexed by the set*

$$R = \{\boldsymbol{w}_0\} + \mathbb{F}_q A \triangleq \{\boldsymbol{w}_0 + \lambda \boldsymbol{v} : \boldsymbol{v} \in A, \lambda \in \mathbb{F}_q \setminus \{0\}\}$$

is a recovering set for the symbol $y_{\boldsymbol{w}_0}$.

Proof: The proof follows similar ideas to the one from [4]. Recall that every codeword $\boldsymbol{y} = (y_{\boldsymbol{w}})_{\boldsymbol{w}\in\mathbb{F}_q^s} \in \mathcal{C}$ corresponds to a polynomial $P(\boldsymbol{x}) \in \mathbb{F}_q[\boldsymbol{x}]$, of degree at most d, where for all $\boldsymbol{w} \in \mathbb{F}_q^s$, $y_{\boldsymbol{w}} = P^{(<m)}(\boldsymbol{w})$. Every vector $\boldsymbol{v}$ in the interpolation set A is called a *direction* and will correspond to a line containing $\boldsymbol{w}_0$ in the direction $\boldsymbol{v}$. Reading the order m evaluations of the polynomial $P(\boldsymbol{x})$ at these lines will enable us to recover the value of $P^{(<m)}(\boldsymbol{w}_0)$. This procedure consists of two steps, described as follows.

Step 1: For every direction $\boldsymbol{v} \in A$, define the following univariate polynomial $p_{\boldsymbol{v}}(\lambda) = P(\boldsymbol{w}_0 + \lambda \boldsymbol{v}) \stackrel{\text{def}}{=} \sum_{j=0}^{d} c_{\boldsymbol{v},j}\lambda^j \in \mathbb{F}_q[x]$. Since the values and the derivatives of $P(\boldsymbol{w}_0 + \lambda \boldsymbol{v})$ for all $\lambda \in \mathbb{F}_q \setminus \{0\}$ are known, and $\deg(p_{\boldsymbol{v}}) \leqslant d$, one can prove, as in [4], that $p_{\boldsymbol{v}}(\lambda)$ is unique, and thus can be recovered.

Step 2: From Step 1, one can get that

$$p_{\boldsymbol{v}}(\lambda) = \sum_{\boldsymbol{i}} P^{(\boldsymbol{i})}(\boldsymbol{w}_0)\boldsymbol{v}^{\boldsymbol{i}}\lambda^{wt(\boldsymbol{i})} = \sum_{j=0}^{d} c_{\boldsymbol{v},j}\lambda^j,$$

and therefore for $0 \leqslant j \leqslant d$, $\sum_{\boldsymbol{i}:wt(\boldsymbol{i})=j} P^{(\boldsymbol{i})}(\boldsymbol{w}_0)\boldsymbol{v}^{\boldsymbol{i}} = c_{\boldsymbol{v},j}$. Considering only the first m of these $d+1$ equations, we get that $u_{\boldsymbol{i}} = P^{(\boldsymbol{i})}(\boldsymbol{w}_0)$ is a solution for the equations system

$$\sum_{\boldsymbol{i}:wt(\boldsymbol{i})=j} u_{\boldsymbol{i}}\boldsymbol{v}^{\boldsymbol{i}} = c_{\boldsymbol{v},j}, \quad 0 \leqslant j < m \leqslant d. \tag{1}$$

Now we prove that the equations system (1) has a unique solution. Indeed, if we denote $Q_j(\boldsymbol{x}) = \sum_{\boldsymbol{i}:wt(\boldsymbol{i})=j} u_{\boldsymbol{i}}\boldsymbol{x}^{\boldsymbol{i}} \in \mathbb{F}_q[x_1,\ldots,x_s]$ where $0 \leqslant j < m$, we get that the equations in (1) are equivalent to $Q_j(\boldsymbol{v}) = c_{\boldsymbol{v},j}$ for every $\boldsymbol{v} \in A$. But since for every j we know that Q_j is an homogeneous polynomial of degree j, and A is an interpolation set for homogeneous polynomials of degree at most $m-1$, we get that the polynomial $Q_j(\boldsymbol{x})$ is unique. Therefore, we can recover the value of $P^{(<m)}(\boldsymbol{w}_0)$ by solving the equations system (1). ■

The next theorem shows how to construct PIR codes from Multiplicity Codes.

Theorem 9. *For all* m,d,s,q *such that* $\frac{d}{m} < q-1$, *the code* $\mathcal{C}(m,d,s,q)$ *is a k-PIR code* $[q^s,n,k]_Q^P$, *where* $n = \frac{\binom{d+s}{s}}{\binom{s+m-1}{s}}, k = \lfloor \frac{q}{m} \rfloor^{s-1}$, *and* $Q = q^{\binom{s+m-1}{s}}$.

Proof: According to Theorem 8, every interpolation set A for homogeneous polynomials of degree $m-1$ defines a recovering set, which consists of the lines containing $\boldsymbol{w}_0$ in the directions of $\boldsymbol{v}$ for all $\boldsymbol{v} \in A$. Therefore, in order to get disjoint recovering sets, all we need to do is to pick different lines. According to Lemma 7, there are $\lfloor \frac{q}{m} \rfloor^{s-1}$ interpolation sets for homogeneous polynomials of degree $m-1$ which are mutually disjoint under multiplication. This means that each line cannot appear in two sets, thus the recovering sets defined by these interpolation sets are disjoint. ■

The next theorem summarizes the results in this section.

Theorem 10. *For every positive integer* $s \geqslant 2$, $0 < \alpha < 1$, *and* n *sufficiently large, there exists a k-PIR code* $[N,n,k]_Q^P$, *over* $\mathbb{F}_Q$ *with redundancy* $r = N - n$ *such that*

$$k = \Theta(n^{(1-\frac{1}{s})(1-\alpha)}), Q = n^{\Theta(n^{\alpha})}, r = \mathcal{O}(n^{1-\frac{\alpha}{s}}).$$

In particular, for $0 \leqslant \epsilon < 1$, *it holds that* $r_P(n, k = \Theta(n^{\epsilon})) = \mathcal{O}(n^{\delta(\epsilon)})$, *where* $\delta(\epsilon) = \min_{s:s>\frac{1}{1-\epsilon}}\{\delta_s(\epsilon)\}$, *and* $\delta_s(\epsilon) = 1 - \frac{1}{s} + \frac{\epsilon}{s-1}$. *For a given value of* ϵ, *the value* s^* *that minimizes* $\delta(\epsilon)$ *is* $s^* = \lfloor \frac{2}{1-\epsilon} \rfloor$.

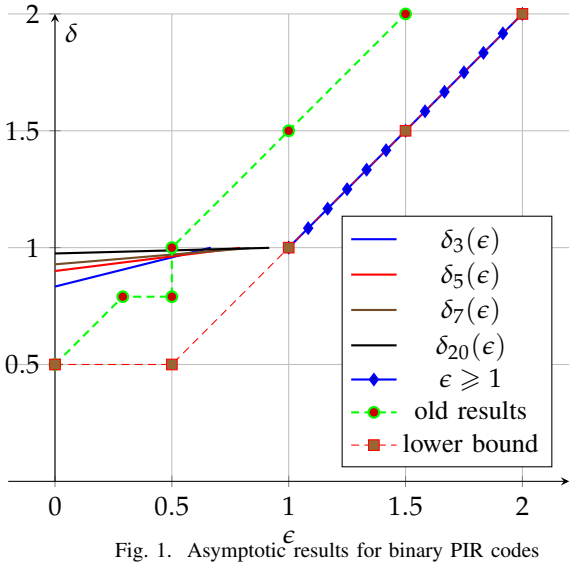

Fig. 1. Asymptotic results for binary PIR codes

Now we use our last result in order to construct binary k-PIR codes. The main idea is to convert every symbol of the field $\mathbb{F}_Q$ to $\log(Q)$ binary symbols. We say that $f(n) = \Omega(n^{a^-})$ is for all $\tau > 0$, $f(n) = \Omega(n^{a-\tau})$. Similarly we define $f(n) = \mathcal{O}(n^{a^+})$ if for all $\tau > 0$, $f(n) = \mathcal{O}(n^{a+\tau})$.

Theorem 11. *For every positive integer* $2 \leqslant s$, $0 < \alpha < 1$, *and* n *sufficiently large, there exists a binary k-PIR code* $[N,n,k]^P$, *with redundancy* $r = N - n$ *such that*

$$k = \Theta\big(\big(\frac{n}{\log(n)}\big)^{(1-\frac{1}{s})\frac{1-\alpha}{1+\alpha}}\big), r = \mathcal{O}\big(n^{1-\frac{\alpha}{s(1+\alpha)}}(\log(n))^{\frac{\alpha}{s(1+\alpha)}}\big).$$

In particular, for $0 \leqslant \epsilon < 1$, $r_P\big(n, k = \Omega(n^{\epsilon^-})\big) = \mathcal{O}\big(n^{\delta(\epsilon)^+}\big)$, *where* $\delta(\epsilon) = \min_{s:s>\frac{1}{1-\epsilon}}\{\delta_s(\epsilon)\}$ *and* $\delta_s(\epsilon) = 1 - \frac{s(1-\epsilon)-1}{2s(s-1)}$, *and* $r_P\big(n, k = \Theta(n^{\epsilon})\big) = o(n)$.

The analysis so far dealt with constructing k-PIR when $k = \Theta(n^{\epsilon})$ and $0 \leqslant \epsilon < 1$. Now we show how to use these results to construct k-PIR codes for $\epsilon \geqslant 1$. The idea is to concatenate a sufficient copies of k'-PIR codes, when $k' = \Omega(n^{1^-})$ such that each bit will have k recovering sets.

Theorem 12. *For all* $\epsilon \geqslant 1$ *and* n *sufficiently large, there exists a binary k-PIR code* $[N,n,k]_2^P$, *such that* $k = \Theta(n^{\epsilon})$ *and* $N = \mathcal{O}(n^{\epsilon^+})$.

The length achieved by the PIR construction in Theorem 12 is nearly optimal. Recall that the length of k-PIR codes is $\Omega(k)$ since every non-trivial recovering set must contain at least one redundancy bit. Fig. 1 summarizes the results of binary PIR codes we achieved in this section together with the previous results. We plot the curves $\delta_s(\epsilon)$ for $s = 3,5,9,20$ from Theorem 11 as well as the results for $\epsilon \geqslant 1$ from Theorem 12. The lower bound on the redundancy is given by $\min\{k, \sqrt{n}\}$.

V. Batch Codes from Multiplicity Codes

It turns out that multiplicity codes can be also an excellent tool to construct batch codes. Unlike the PIR case, recovering different entries in the codeword will cause intersection in the corresponding lines, and thus intersecting recovering sets. In order to overcome this obstacle, we reduce the degree d of the polynomials such that a fewer number of points is needed

from every line. This will allow different lines to avoid points which are used by other lines. That way, every recovering set can "drop out" points which are used by other sets, resulting in disjoint recovering sets.

Lemma 13. *For all m, s, q, d, k such that $d \leqslant m(q - km^{s-1} - 2)$ and $k \leqslant \lfloor \frac{q}{m} \rfloor^{s-1}$, the code $\mathcal{C}(m, d, s, q)$ is a k-batch code $[q^s, n, k]_Q^B$, where $n = \frac{\binom{d+s}{s}}{\binom{s+m-1}{s}}$ and $Q = q^{\binom{s+m-1}{s}}$.*

Proof: The claim regarding the code dimension and field size can be proven similarly to PIR codes. Now we prove that every multiset request of size k can be recovered. As we saw in the recovering procedure for PIR codes, every recovering set contains m^{s-1} different lines. Since different lines can intersect on at most one point, and there are k recovering sets, it suffices to prove that Step 1 in the recovering procedure can be completed even when km^{s-1} points on the line are not used. But since the minimum distance of $\mathcal{C}(m, d, s = 1, q)$ equals $q - \frac{d}{m} > km^{s-1} + 1$, it can be shown in a very similar way to PIR codes, that the polynomial $p_v(\lambda)$ in Step 1 can be uniquely recovered, and thus also Step 2 can be completed. ∎

Unlike the PIR case, it turns out that only the value $s = 2$ is useful for batch codes, thus getting the following theorem.

Theorem 14. *For every $0 < \alpha < 0.5$ and n sufficiently large, there exists a k-batch code $[N, n, k]_Q^B$ over $\mathbb{F}_Q$ with redundancy $r = N - n$ such that*

$$k = \Theta(n^{0.5-\alpha}), r = \mathcal{O}(n^{1-\frac{\alpha}{2}}), Q = n^{\Theta(n^{\alpha})}.$$

In particular, for $0 < \epsilon < 0.5$, it holds that $r_B\big(n, k = \Theta(n^{\epsilon})\big) = \mathcal{O}\big(n^{\delta(\epsilon)}\big)$, where $\delta(\epsilon) = \frac{3}{4} + \frac{\epsilon}{2}$.

As in the PIR case, the last result can be extended for binary batch codes.

Theorem 15. *For every $0 < \alpha < 0.5$ and n sufficiently large, there exists a binary k-batch code $[N, n, k]^B$ with redundancy $r = N - n$ such that*

$$k = \Theta\big((n/\log(n))^{0.5-\alpha}\big), r = \mathcal{O}\big(n^{1-\frac{\alpha}{3}}(\log(n))^{\frac{\alpha}{3}}\big).$$

In particular, for $0 < \epsilon < 0.5$, it holds that $r_B\big(n, k = \Omega(n^{\epsilon^-})\big) = \mathcal{O}\big(n^{\delta(\epsilon)^+}\big)$, where $\delta(\epsilon) = \frac{5}{6} + \frac{\epsilon}{3}$, and $r_B\big(n, k = \Theta(n^{\epsilon})\big) = o(n)$. For $\epsilon \geqslant 0.5$ there exists a binary k-batch code $[N, n, k]_2^B$ of dimension n such that $k = \Theta(n^{\epsilon})$ and $N = \mathcal{O}(n^{0.5+\epsilon^+})$.

VI. Array Construction for PIR and Batch Codes

Our point of departure for this section is the subcube construction from [3] which was also used in [2] to construct PIR codes. The idea of this construction is to position the information bits in a two-dimensional array, and add a simple parity bit for each row and each column. Our approach here is to extend this construction by considering also diagonals with different slopes. As there are many different slopes, this can greatly increase the number of recovering sets. However, we will have to guarantee that using the diagonals will still result with disjoint recovering sets. By a slight abuse of notation, in this section we let the set $[n]$ denote the set of integers $\{0, 1, \ldots, n-1\}$. We use the notation $\langle x \rangle_m$ to denote the value of $(x \bmod m)$.

Definition 16. *Let A be an $r \times p$ array, with indices $(i, j) \in [r] \times [p]$. For $s \in [p]$ we define the following set of sets $P_s(r, p) = \{D_{s,0}, D_{s,1}, \ldots, D_{s,p-1}\}$, where for $t \in [p]$,*

$$D_{s,t} = \{(0, t), (1, \langle t + s \rangle_p), \ldots, (r-1, \langle t + (r-1)s \rangle_p)\}$$

The idea behind Definition 16 is to fix a slope $s \in [p]$ and then define p diagonal sets which are determined by the starting point on the first row and the slope. We use these sets in order to construct array codes, where every diagonal determines a parity bit for the bits on this diagonal.

Construction 1 (Array Construction) *Let r, p, n be positive integers such that $n = rp$, and $S \subseteq [p]$ a subset of size k. We define the encoder $E_{r,p,S}$, as a mapping $E_{r,p,S} : \{0,1\}^n \to \{0,1\}^{k \cdot p}$ as follows. We denote $S = \{s_0, s_1, \ldots, s_{k-1}\}$ where $0 \leqslant s_0 < s_1 < \cdots < s_{k-1} \leqslant p - 1$.The input vector $\boldsymbol{x} \in \{0,1\}^n$ is represented as an $r \times p$ array, that is $\boldsymbol{x} = (x_{i,j})_{(i,j) \in [r] \times [p]}$ and is encoded to the following kp redundancy bits $\rho_{\ell,t}$, for $\ell \in [k]$, and $t \in [p]$,*

$$\rho_{\ell,t} = \sum_{(i,j) \in D_{s_\ell, t}} x_{i,j}.$$

Let $E_{r,p,S}(\boldsymbol{x}) = (\rho_{0,0}, \ldots, \rho_{0,p-1}, \ldots, \rho_{k-1,0}, \ldots, \rho_{k-1,p-1})$, and the code $\mathcal{C}(r, p, S)$ is defined to be

$$\mathcal{C}(r, p, S) = \{(\boldsymbol{x}, E_{r,p,S}(\boldsymbol{x})) \,:\, \boldsymbol{x} \in \{0,1\}^n\}.$$

We first list several useful properties.

Lemma 17. *For all r, p, and $s \in [p]$ the set $P_s(r, p)$ is a partition of $[r] \times [p]$.*

Lemma 18. *For all $r \leqslant p$ and $S \subseteq [p]$. If p is prime, then for all $s_1 \neq s_2 \in S$ and $t_1, t_2 \in [p]$, $|D_{s_1,t_1} \cap D_{s_2,t_2}| \leqslant 1$.*

We only state here the result of this construction for PIR codes, as we focus here mainly on batch codes.

Theorem 19. *Let $n = p^2$, where p is a prime number, and $k \leqslant \sqrt{n}$. The code $\mathcal{C}(r = p, p, S = [k])$ is a k-PIR code with redundancy $k\sqrt{n}$. In particular, for all $k \leqslant \sqrt{n}$, $r_P(n, k) = \mathcal{O}(k\sqrt{n})$.*

For batch codes, this construction can result with good batch codes as well as batch codes with restricted size for the recovering sets [13]. Formally, a k-PIR code, k-batch code, in which the size of each recovering set is at most r will be called an *(r, k)-PIR code*, *(r, k)-batch code*, respectively.

The idea here is to choose the set S in a way that for every bit, each of its recovering sets intersects with at most one recovering set of any other bit. This property for constructing batch codes from PIR codes was proved in [9] and is stated below.

Lemma 20. *Let $\mathcal{C}$ be an (r, k)-PIR code. Assume that for every distinct indices $i, j \in [n]$, it holds that each recovering set of the ith bit intersects with at most one recovering set of the jth bit. Then, the code $\mathcal{C}$ is an (r, k)-batch code.*

The main challenge is to find sets S that will generate recovering sets which satisfy the condition in Lemma 20. For that, we use the following definition.

Definition 21. *Let r be a positive integer, and S be a set of non-negative integers. We say that the set S does not contain an **r-weighted arithmetic progression modulo** p if there do* not *exist $s_1, s_2, s_3 \in S$ and $0 < x, y < r - 1$, where $x + y < r$, such that $xs_1 + ys_2 = (x + y)s_3 \bmod p$.*

Given this definition, we prove the following theorem.

Theorem 22. *Let $r \leqslant p$ and $S \subseteq [p]$, $|S| = k$. If p is prime, and S does not contain an r-weighted arithmetic progression modulo p, then the code $\mathcal{C} = \mathcal{C}(r, p, S)$ is an (r, k)-batch code of dimension rp.*

Proof: Assume that $S = \{s_0, s_1, , \ldots, s_{k-1}\}$. One can verify using Lemma 17 and 18 that for every $(i,j) \in [r] \times [p]$ the following sets

$$R_\ell^{(i,j)} = \{\rho_{\ell,t_\ell}\} \cup \{x_{i',j'} \ : \ (i',j') \in D_{s_\ell,t_\ell} \setminus \{(i,j)\}\},$$

for $\ell \in [k]$ are k mutually disjoint recovering sets for $x_{i,j}$, where $t_\ell \in [p]$ is chosen such that $(i,j) \in D_{s_\ell,t_\ell}$. We denote $D(R_\ell^{(i,j)}) = D_{s_\ell,t_\ell}$. Thus $\mathcal{C}$ is (r,k)-PIR, and it remains to prove that $\mathcal{C}$ satisfies the condition of Lemma 20. Assume in the contrary that there exist two bits $(i,j),(i',j') \in [r] \times [p]$ such that (i,j) has a recovering set $R_{\ell_1}^{(i,j)}$ that intersects with two recovering sets $R_{\ell'_1}^{(i',j')}, R_{\ell'_2}^{(i',j')}$ of (i',j'). Assume $b_1 \in R_{\ell_1}^{(i,j)} \cap R_{\ell'_1}^{(i',j')}$ and $b_2 \in R_{\ell_1}^{(i,j)} \cap R_{\ell'_2}^{(i',j')}$ where b_1, b_2 are codeword entries. It can be verified that b_1, b_2 don't correspond to parity bits. Therefore, we denote $b_1 = x_{i_1,j_1}$, $b_2 = x_{i_2,j_2}$, for $(i_1,j_1),(i_2,j_2) \in [r] \times [p]$. Denote $D(R_{\ell_1}^{(i,j)}) = D_{s'_1,t_1}, D(R_{\ell'_1}^{(i',j')}) = D_{s'_2,t_2}, D(R_{\ell'_2}^{(i',j')}) = D_{s'_3,t_3}$ for $s'_1, s'_2, s'_3 \in S$ and $t_1, t_2, t_3 \in [p]$. Thus we get that $(i_1,j_1),(i_2,j_2) \in D_{s'_1,t_1}, (i_1,j_1),(i',j') \in D_{s'_2,t_2}$, and $(i_2,j_2),(i',j') \in D_{s'_3,t_3}$. From Lemma 17 and 18 we deduce that $s'_1 \neq s'_2 \neq s'_3$ and $i' \neq i_1 \neq i_2$. Assume w.l.o.g $i_1 < i_2 < i'$. It follows that:

$$\begin{aligned} j_1 &= \langle t_1 + i_1 s'_1 \rangle_p, \quad j_2 = \langle t_1 + i_2 s'_1 \rangle_p \\ j_1 &= \langle t_2 + i_1 s'_2 \rangle_p, \quad j' = \langle t_2 + i' s'_2 \rangle_p \\ j_2 &= \langle t_3 + i_2 s'_3 \rangle_p, \quad j' = \langle t_3 + i' s'_3 \rangle_p \end{aligned}$$

This implies that $\langle (i_2 - i_1)s'_1 + (i' - i_2)s'_3 \rangle_p = \langle (i' - i_1)s'_2 \rangle_p$, which is a contradiction since S does not contain an r-weighted arithmetic progression modulo p. ∎

In order to complete the construction of batch codes, we are left with the problem of finding large sets S which satisfy the condition in Theorem 22. That is, given r and p, our goal is to find the largest such a set S. A simple greedy algorithm can give the following result.

Theorem 23. *Let r, p be positive integers, such that p is prime. Then there exists a set S with no r-weighted arithmetic progression modulo p of size at least k, where k is the largest integer such that $p > 2k^2r^2$.*

The following theorem follows from these observations.

Theorem 24. *For every r, k, let $n = rp$, where p is the smallest prime number such that $2k^2r^2 < p$. Then, there exists an (r,k)-batch code of dimension n and rate $\frac{r}{r+k}$. In particular, the redundancy of the code equals kp.*

According to Theorem 24 we are now at a point to construct k-batch codes with good redundancy.

Corollary 25. *For any n and k such that $k = o(\sqrt{n})$, there exists a k-batch code of dimension n and redundancy $\mathcal{O}(n^{\frac{2}{3}}k^{\frac{5}{3}})$. In particular, for $0 < \epsilon < 1/2$, $r_B(n, n^\epsilon) = \mathcal{O}(n^{2/3+5\epsilon/3})$.*

Proof: For n and k, let us choose $r = \lceil n^{\frac{1}{3}}/k^{\frac{2}{3}} \rceil$, and p is the smallest prime number such that $2k^2r^2 < p$. Then, according to Theorem 24, there exists an (r,k)-batch code of dimension $pr > n$ and redundancy kp. That is, the redundancy satisfies $kp = \Theta(k^3r^2) = \Theta(n^{\frac{2}{3}}k^{\frac{5}{3}})$. The second statement in the corollary is established for $k = n^\epsilon$ in the last equation. ∎

Let us denote $r_B(k = n^\epsilon) = \mathcal{O}(n^\delta)$. In Fig. 2 we plot the results on the asymptotic behavior of the redundancy of batch codes. These plots are received from Corollary 25 in this section and Theorem 15 from Section V. Note that the array construction improves the redundancy only for $\epsilon < 0.0755$.

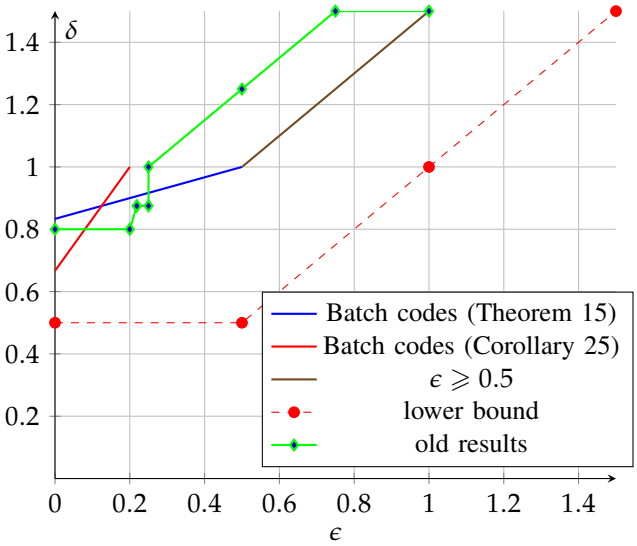

Fig. 2. Asymptotic results for binary batch codes

Lastly, we report on two more results that can be derived using the Array Construction. Note that the second result improves upon the one from [10], which states that $r_B(n,5) = \mathcal{O}(\sqrt{n}\log(n))$.

Theorem 26. *For every $0 < \alpha < 1$, $k = \mathcal{O}(n^\alpha)$, and fixed $r \geqslant 3$, there exists an (r,k)-batch code with rate $\frac{r}{r+k}$.*

Theorem 27. *Let $n = p^2$ where p is a prime number. The code $\mathcal{C}$, that extends $\mathcal{C}(r = p, p, S = [5])$ by adding a global parity bit, is a 5-batch code with redundancy $5p + 1 = \Theta(\sqrt{n})$, and therefore $r_B(n,5) = \Theta(\sqrt{n})$.*

References

[1] S. Buzaglo, E. Yaakobi, Y. Cassuto, and P. H. Siegel, "Consecutive switch codes," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 660–664, Barcelona, Spain, July. 2016.

[2] A. Fazeli, A. Vardy, and E. Yaakobi, "PIR with low storage overhead: Coding instead of replication," arXiv:1505.06241, May 2015.

[3] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," *Proc. of the 36-sixth Annual ACM Symposium on Theory of Computing*, pp. 262–271, Chicago, ACM Press, 2004.

[4] S. Kopparty, S. Saraf, and S. Yekhanin, "High-rate codes with sublinear-time decoding," in *Proc. of the Forty-third Annual ACM Symposium on Theory of Computing (STOC)*, pp. 167–176, New York, NY, 2011.

[5] S. Lin and D. J. Costello, *Error Control Coding*, Prentice Hall, 2004.

[6] H. Lipmaa and V. Skachek, "Linear batch codes," *Coding Theory and Applications, CIM Series*, vol. 3. pp. 245–253, 2015.

[7] J.L. Massey, *Threshold Decoding*, MIT Press, 1963.

[8] S. Rao and A. Vardy, "Lower bound on the redundancy of PIR codes," arxiv:1605.01869v1, May 2016.

[9] A. S. Rawat, Z. Song, A. G. Dimakis, and A. Gál, "Batch codes through dense graphs without short cycles," *IEEE Trans. Inform. Theory*, vol. 62, pp. 1592–1604, Apr. 2016.

[10] A. Vardy and E. Yaakobi, "Constructions of batch codes with near-optimal redundancy," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 1197–1201, Barcelona, Spain, July. 2016.

[11] S. Yekhanin, "Locally decodable codes," *Foundations and Trends in Theoretical Computer Science*, vol. 6, no. 3, pp.139–255, 2012.

[12] Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck, "Codes for network switches," *Proc. IEEE Int. Symp. on Inf. Theory*, pp. 1057–1061, Istanbul, Turkey, Jul. 2013.

[13] H. Zhang and V. Skachek, "Bounds for batch codes with restricted query size," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 1192–1196, Barcelona, Spain, July. 2016.