Decoding Frequency Permutation Arrays Under Chebyshev Distance

Min-Zheng Shieh and Shi-Chun Tsai, *Member, IEEE*

***Abstract*—A frequency permutation array (FPA) of length $n = m\lambda$ and distance d is a set of permutations on a multiset over m symbols, where each symbol appears exactly λ times and the distance between any two elements in the array is at least d. FPA generalizes the notion of permutation array. In this paper, under the Chebyshev distance, we first prove lower and upper bounds on the size of FPA. Then we give several constructions of FPAs, and some of them come with efficient encoding and decoding capabilities. Moreover, we show one of our designs is locally decodable, i.e., we can decode a message bit by reading at most $\lambda + 1$ symbols, which has an interesting application to private information retrieval.**

***Index Terms*—Chebyshev distance, frequency permutation array (FPA), locally decodable code, permanent, permutation array (PA).**

I. Introduction

In 1965, Slepian [13] considered a code of length n for permutation modulation. It consists of all permutations on the multiset $\{\overbrace{\mu_1,\ldots,\mu_1}^{\lambda_1},\ldots,\overbrace{\mu_m,\ldots,\mu_m}^{\lambda_m}\}$ where $\mu_1 < \mu_2 < \cdots < \mu_m$ and $\lambda_1+\lambda_2+\cdots+\lambda_m = n$. In this paper, we consider a special case of Slepian's code. Let $\mu_1 = 1, \mu_2 = 2, \ldots, \mu_m = m$ and $\lambda_1 = \lambda_2 = \cdots = \lambda_m = \lambda$. Let S_n^λ be the set of all permutations on the multiset $\{\overbrace{1,\ldots,1}^{\lambda},\ldots,\overbrace{m,\ldots,m}^{\lambda}\}$. A *frequency permutation array* (FPA) is a subset of S_n^λ for some positive integers m, λ and $n = m\lambda$. A (λ, n, d)-FPA is a subset of S_n^λ and the distance between any pair of distinct permutations is at least d under any metric, such as Hamming distance, Chebyshev distance $d_{\max}$, etc. A permutation array (PA) is simply a special case of an FPA by choosing $\lambda = 1$. With a fixed length n, an FPA has a smaller set of symbols than a PA. Thus, codes with FPA have a better information rate than those with PA. A widely adopted approach to building PAs under Hamming distance, see for example [3], is using distance-preserving mappings or distance-increasing mappings from Z_2^k to S_n^1. Most of those encoding[1] schemes are efficient but there are only a few efficient decoding algorithms. Swart and Ferreira [14] gave a decoding algorithm for an application on power line communication. Lin *et al.* [11] proposed a couple of novel constructions with efficient encoding and decoding algorithms for PAs under $d_{\max}$. FPA was proposed by Huczynska and Mullen [5] as a generalization of PA. They gave several constructions of FPA under Hamming distance and bounds on the maximum array size.

Recently, researchers have found that PAs have applications in areas such as power line communication (e.g., [12], [18], [19] and [20]), multilevel flash memories (see [6] and [7], [15]). Similar to the application of PAs on power line communication, we can encode a message as a frequency permutation from S_n^λ. Then the message is transmitted as m-FSK[2] signals. The nature of frequency permutations provides higher information rate without losing immunity to impulse noise and permanent frequency disturbances[3] mentioned in Vinck's work [20].

For flash memory applications, different from the approach by Jiang *et al.* [6], [7], we can use FPA to provide multilevel flash memory with error correcting capabilities. For example, suppose a multilevel flash memory, where each cell has m states, which can be changed by injecting or removing charge into or from it. Overinjecting or charge leakage will alter the state as well. We can use the charge ranks of n cells to represent a permutation from S_n^λ, i.e., the cells with the lowest λ charge levels represent symbol 1, and so on. With our efficient encoding and decoding algorithms, a (λ, n, d)-FPA can be used in a flash memory system to represent information and correct errors caused by charge level fluctuation.

A locally decodable code has an extremely efficient decoding for any message bit by reading at most a fixed number of symbols from the received word. Suppose that a FPA is applied to a multilevel flash memory where the length of a codeword is nearly a block of cells (about 10^5) [2]. This feature allows us to retrieve the desired message bits from a multilevel flash without accessing the whole block. With the locally decodable property, we can raise the robustness of the code without loss of efficiency. On the other hand, locally decodable codes have been under study for years; see [16] for a survey and [21] and [4]

Manuscript received January 19, 2010; revised May 31, 2010. Date of current version October 20, 2010. This work was supported in part by the National Science Council of Taiwan under Contracts NSC-97-2221-E-009-064-MY3 and NSC-98-2221-E-009-078-MY3. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Seoul, Korea, June–July 2009.

The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30050, Taiwan (e-mail: mzhsieh@csie.nctu.edu.tw; sctsai@csie.nctu.edu.tw).

Communicated by N. Kashyap, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2010.2069253

[1] We consider only encoding and decoding schemes for error correcting codes in this paper. Some other works, such as Babaev's [1], consider encoding as computing the binary representation for some permutation, and decoding as computing the permutation represented by some binary string. The schemes of [1] are efficient, but they do not exhibit any error correcting capability.

[2] For $i \in \{1,\ldots,m\}$, we send i by a wave of some unique frequency f_i. The frequency of wave is irrelevant to the frequency parameter λ in the definition of FPA.

[3] A permanent frequency disturbance is a disturbance wave of some constant frequency. It is irrelevant to λ.

for recent progress. They are related to a cryptographic protocol called *private information retrieval* (PIR for short).

In this paper, we extend the ideas in [10] and [11] to constructing FPAs under $d_{\max}$. In Section II, we prove lower and upper bounds on the maximum cardinality of FPAs. We derive Gilbert type and sphere packing bounds by bounding size of balls under Chebyshev distance in Section II-A. Moreover, we give several constructions to obtain some lower bounds; see Section II-B. In Section III, we show a pair of efficient encoding and decoding algorithms for FPAs constructed in a simple manner. In addition, we show that they are locally decodable codes under $d_{\max}$. As a consequence, the construction of FPA can also be used for constructing PIR.

Notations: Let $n = m\lambda$ throughout the paper unless stated otherwise. We use $[n]$ to represent the set $\{1, \ldots, n\}$. Let x_i be the ith entry of vector $\boldsymbol{x}$. For two k-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, let $d_{\max}(\boldsymbol{x}, \boldsymbol{y}) = \max_{i \in [k]} |x_i - y_i|$. We say two permutations $\boldsymbol{x}$ and $\boldsymbol{y}$ are d-close to each other under metric $\delta(\cdot, \cdot)$ if $\delta(\boldsymbol{x}, \boldsymbol{y}) \le d$. The identity permutation $\boldsymbol{I}_n^\lambda$ in S_n^λ is $(\overbrace{1, \ldots, 1}^{\lambda}, \ldots, \overbrace{m, \ldots, m}^{\lambda})$. For $\pi = (p_1, \ldots, p_n) \in S_n^1$ and a n-dimensional vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, we define $\pi \circ \boldsymbol{x} = (x_{p_1}, x_{p_2}, \ldots, x_{p_n})$ and $\pi^{-1} \circ \pi = \boldsymbol{I}_n^1$. For example, $(2,1,5,3,4,6) \circ (1,1,2,2,3,3) = (1,1,3,2,2,3)$ and $(2,1,5,3,4,6)^{-1} = (2,1,4,5,3,6)$.

II. Lower and Upper Bounds

Let $F_\infty(\lambda, n, d)$ be the cardinality of the maximum (λ, n, d)-FPA and $V_\infty(\lambda, n, d)$ be the number of elements in S_n^λ being d-close to the identity $\boldsymbol{I}_n^\lambda$ under $d_{\max}$. In this section, we first give a Gilbert type lower bound and a sphere packing upper bound on $F_\infty(\lambda, n, d)$ by bounding $V_\infty(\lambda, n, d)$. Then we generalize the method in [10] to obtain several constructions that yields some inequalities for bounding the cardinality of FPAs.

A. Gilbert Type and Sphere Packing Bounds

First, we show that any d-radius ball in S_n^λ under $d_{\max}$ has the same cardinality.

Claim 1: For any $\boldsymbol{x} = (x_1, \ldots, x_n) \in S_n^\lambda$, there are exactly $V_\infty(\lambda, n, d)$ $\boldsymbol{y}$'s in S_n^λ such that $d_{\max}(\boldsymbol{x}, \boldsymbol{y}) \le d$.

Proof: Since every $i \in [m]$ appears exactly λ times in $\boldsymbol{x}$, there exists a permutation $\pi \in S_n^1$ such that $\boldsymbol{x} = \pi \circ \boldsymbol{I}_n^\lambda$. As a consequence, we have that $d_{\max}(\boldsymbol{I}_n^\lambda, \boldsymbol{z}) = d_{\max}(\boldsymbol{x}, \pi \circ \boldsymbol{z})$ for any $\boldsymbol{z} \in S_n^\lambda$. Let $Z = \{\boldsymbol{z} : \boldsymbol{z} \in S_n^\lambda, d_{\max}(\boldsymbol{I}_n^\lambda, \boldsymbol{z}) \le d\}, Y = \{\pi \circ \boldsymbol{z} : \boldsymbol{z} \in Z\}$ and $\bar{Y} = S_n^\lambda \backslash Y$. For any $\boldsymbol{y} \in Y$, we have $d_{\max}(\boldsymbol{x}, \boldsymbol{y}) = d_{\max}(\boldsymbol{I}_n^\lambda, \pi^{-1} \circ \boldsymbol{y}) \le d$, since $\pi^{-1} \circ \boldsymbol{y} \in Z$. While for $\boldsymbol{y}' \in \bar{Y}, d_{\max}(\boldsymbol{x}, \boldsymbol{y}') = d_{\max}(\boldsymbol{I}_n^\lambda, \pi^{-1} \circ \boldsymbol{y}') > d$. Therefore, only $|Y| = |Z| = V_\infty(\lambda, n, d)$ permutations in S_n^λ are d-close to $\boldsymbol{x}$. ■

Theorem 1:

$$\frac{|S_n^\lambda|}{V_\infty(\lambda, n, d-1)} \le F_\infty(\lambda, n, d) \le \frac{|S_n^\lambda|}{V_\infty\left(\lambda, n, \lfloor \frac{d-1}{2} \rfloor\right)}.$$

Proof: To prove the lower bound, we use the following algorithm to generate a (λ, n, d)-FPA with size $\ge \frac{|S_n^\lambda|}{V_\infty(\lambda, n, d-1)}$.

1) $C \leftarrow \emptyset, D \leftarrow S_n^\lambda$.
2) Add an arbitrary $\boldsymbol{x} \in D$ to C, then remove all permutations that is $(d-1)$-close to $\boldsymbol{x}$ from D.
3) If $D \ne \emptyset$ then repeat step 2, otherwise output C.

D has initially $|S_n^\lambda|$ elements and each iteration of step 2 removes at most $V_\infty(\lambda, n, d-1)$, so we conclude $|C| \ge \frac{|S_n^\lambda|}{V_\infty(\lambda, n, d-1)}$.

Now we turn to the upper bound. Consider a (λ, n, d)-FPA C^* with the maximum cardinality. Any two $\lfloor \frac{d-1}{2} \rfloor$-radius balls centered at distinct permutations in C^* do not have any common permutation, since the minimum distance is d. In other words, the $\lfloor \frac{d-1}{2} \rfloor$-radius balls centered at permutations in C^* are all disjoint. We have $|C^*| \le \frac{|S_n^\lambda|}{V_\infty(\lambda, n, \lfloor \frac{d-1}{2} \rfloor)}$. ■

It is clear that $|S_n^\lambda| = \frac{n!}{(\lambda!)^{n/\lambda}}$. It is already known that $V_\infty(1, n, d)$ equals the permanent of some special matrix [11]. In this paper, we generalize previous analysis to give asymptotic bounds for Theorem 1. The permanent of an $n \times n$ matrix $A = (a_{i,j})$ is defined as

$$\operatorname{per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i, \pi_i}.$$

Define a symmetric $n \times n$ matrix $A^{(\lambda,n,d)} = (a_{i,j}^{(\lambda,n,d)})$, where $a_{i,j}^{(\lambda,n,d)} = 1$, if $|\lceil \frac{i}{\lambda} \rceil - \lceil \frac{j}{\lambda} \rceil| \le d$; else $a_{i,j}^{(\lambda,n,d)} = 0$. Note that a permutation $(x_1, \ldots, x_n)$ is d-close to $\boldsymbol{I}_n^\lambda$ if and only if $a_{i,x_i}^{(\lambda,n,d)} = 1$ for every $i \in [n]$. Now we consider $A^{(\lambda,\lambda m,d)}$. Since the λ copies of a symbol are considered identical while computing the distance and the entries indexed from $(\ell\lambda - \lambda + 1)$ to $\ell\lambda$ of $\boldsymbol{I}_{\lambda m}^\lambda$ represent the same symbol for every $\ell \in [m]$, it implies that row $(\ell\lambda - \lambda + 1)$ through row $\ell\lambda$ of $A^{(\lambda,\lambda m,d)}$ are identical and so are columns indexed from $(\ell\lambda - \lambda + 1)$ to $\ell\lambda$ for every $\ell \in [m]$. Thus, we have $A^{(\lambda,\lambda m,d)} = A^{(1,m,d)} \otimes \mathbf{1}_\lambda$ where $\otimes$ is the operator of tensor product and $\mathbf{1}_\lambda$ is a $\lambda \times \lambda$ matrix with all entries equal to 1. For example, take $\lambda = 2, m = 5$, and $d = 2$

$$A^{(1,5,2)} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \mathbf{1}_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$A^{(2,10,2)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Let $r_i^{(1,m,d)}$ be the row sum of $A^{(1,m,d)}$'s ith row. We have

$$r_i^{(1,m,d)} = \begin{cases} d + i & \text{if } i \le d, \\ 2d + 1 & \text{if } d < i \le m - d, \\ m - i + 1 + d & \text{if } i > m - d. \end{cases}$$

Then for $i \in [m]$ and $j \in [\lambda]$, the row sum of the $(i\lambda - \lambda + j)$th row of $A^{(\lambda,\lambda m,d)}$ is $\lambda r_i^{(1,m,d)}$, due to $A^{(\lambda,\lambda m,d)} = A^{(1,m,d)} \otimes \mathbf{1}_\lambda$. We first calculate $V_\infty(\lambda,n,d)$ by using $\mathrm{per}(A^{(\lambda,n,d)})$.

Lemma 1:

$$V_\infty(\lambda,n,d) = \frac{\mathrm{per}\left(A^{(\lambda,n,d)}\right)}{(\lambda!)^{n/\lambda}}.$$

Proof:

$$\begin{aligned}
&\mathrm{per}\left(A^{(\lambda,n,d)}\right)\\
&= \left|\left\{\boldsymbol{x} \in S_n^1 : \forall i, a_{i,x_i}^{(\lambda,n,d)} = 1\right\}\right|\\
&= \left|\left\{\boldsymbol{x} \in S_n^1 : \max_i \left|\left\lceil \frac{i}{\lambda}\right\rceil - \left\lceil \frac{x_i}{\lambda}\right\rceil\right| \le d\right\}\right|\\
&= (\lambda!)^{n/\lambda}\left|\left\{\boldsymbol{y} \in S_n^\lambda : \max_i \left|\left\lceil \frac{i}{\lambda}\right\rceil - y_i\right| \le d\right\}\right|\\
&= (\lambda!)^{n/\lambda}\left|\left\{\boldsymbol{y} \in S_n^\lambda : d_{\max}(\boldsymbol{I}_n^\lambda, \boldsymbol{y}) \le d\right\}\right|\\
&= (\lambda!)^{n/\lambda} V_\infty(\lambda,n,d).
\end{aligned}$$

The first equality holds since $A^{(\lambda,n,d)}$ is a $(0,1)$-matrix and by the definition of permanent. We can convert $\boldsymbol{x} \in S_n^1$ into $\boldsymbol{y} \in S_n^\lambda$ by setting $y_i = \lceil \frac{x_i}{\lambda} \rceil$, and there are exactly $(\lambda!)^{n/\lambda}$ $\boldsymbol{x}$'s in S_n^1 converted to the same $\boldsymbol{y}$. Thus, we know the third equality holds. Therefore, the lemma holds by moving $(\lambda!)^{n/\lambda}$ to the left-hand side of the equation. ■

We still need to estimate $\mathrm{per}(A^{(\lambda,n,d)})$ in order to get asymptotic bounds. Kløve [8], [9] reported some bounds and methods to approximate $\mathrm{per}(A^{(1,n,d)})$. We extend his analysis for $\mathrm{per}(A^{(\lambda,n,d)})$.

Lemma 2: $\mathrm{per}(A^{(\lambda,n,d)}) \le [(2d\lambda + \lambda)]^{\frac{n}{2d\lambda+\lambda}}$.

Proof: It is known ([17], Th. 11.5) that for $(0,1)$-matrix $A, \mathrm{per}(A) \le \prod_{i=1}^n (r_i!)^{\frac{1}{r_i}}$ where r_i is the sum of the ith row. Since the sum of any row of $A^{(\lambda,n,d)}$ is at most $2d\lambda + \lambda$, we have $\mathrm{per}(A) \le \prod_{i=1}^n [(2d\lambda + \lambda)]^{\frac{1}{2d\lambda+\lambda}} = [(2d\lambda + \lambda)]^{\frac{n}{2d\lambda+\lambda}}$ ■

We give $\mathrm{per}(A^{(\lambda,n,d)})$ a lower bound by using the van der Waerden permanent theorem [17, p. 104]: *the permanent of an $n \times n$ doubly stochastic matrix A (i.e., A has nonnegative entries, and every row sum and column sum of A is 1.) is no less than $\frac{n!}{n^n}$*. Unfortunately, $A^{(\lambda,n,d)}$ is not a doubly stochastic matrix, since the row sums and columns sums range from $d\lambda + \lambda$ to $2d\lambda + \lambda$. We estimate the lower bound via a matrix derived from $A^{(\lambda,n,d)}$ as follows.

Lemma 3: $\mathrm{per}(A^{(\lambda,n,d)}) \ge \frac{(2d\lambda+\lambda)^n}{2^{2d\lambda}} \cdot \frac{n!}{n^n}$.

Proof: Let $\tilde{A} = \frac{1}{2d\lambda+\lambda} A^{(\lambda,n,d)}$, which has the sum of any row or column bounded by 1, but is not a doubly stochastic matrix. Observe that every row sum of $\tilde{A}$ is 1 except the first $d\lambda$ and last $d\lambda$ rows. For $i \in [d]$ and $j \in [\lambda]$, both row $(i\lambda - \lambda + j)$ and row $(n - i\lambda + j)$ sum to $\frac{d+i}{2d+1}$. Now we construct an $n \times n$ matrix B from $\tilde{A}$ with each row sum equal to 1 as follows: For $i \in [d]$ and $j \in [\lambda]$, add $\frac{1}{2d\lambda+\lambda}$ to

1) the first $(d - i + 1)\lambda$ entries of row $(i\lambda - \lambda + j)$;

2) the last $(d - i + 1)\lambda$ entries of row $(n - i\lambda + j)$.

The row sums of the first $d\lambda$ and last $d\lambda$ rows of B are now $\frac{(d-i+1)\lambda}{2d\lambda+\lambda} + \frac{d+i}{2d+1} = 1$.

We turn to check the column sums of B. Since $\tilde{A}$ is symmetric and by the definition of B, we know B is symmetric as well. Thus we have that B is doubly stochastic and $\mathrm{per}(B) \ge \frac{n!}{n^n}$.

To bound $\mathrm{per}(A^{(\lambda,n,d)})$, observe that the entries of the first $d\lambda$ and last $d\lambda$ rows of B are at most $\frac{2}{2d\lambda+\lambda}$ times of the corresponding entries of $A^{(\lambda,n,d)}$, and the other rows are exactly $\frac{1}{2d\lambda+\lambda}$ times of the corresponding rows of $A^{(\lambda,n,d)}$. We have $\mathrm{per}(A^{(\lambda,n,d)}) \ge \frac{(2d\lambda+\lambda)^n}{2^{2d\lambda}}\mathrm{per}(B) \ge \frac{(2d\lambda+\lambda)^n}{2^{2d\lambda}} \frac{n!}{n^n}$. ■

Theorem 2:

$$\frac{n!}{[(2d\lambda - \lambda)]^{\frac{n}{2d\lambda-\lambda}}} \le F_\infty(\lambda,n,d) \le \frac{2^{2\lambda \cdot \lfloor \frac{d-1}{2} \rfloor} n^n}{\left(2\lambda \cdot \lfloor \frac{d-1}{2} \rfloor + \lambda\right)^n}.$$

Proof: The theorem follows by plugging Lemmas 1, 2 and 3 to Theorem 1. ■

B. Constructions and Related Bounds

In [10], Kløve *et al.* gave several constructions for permutation arrays and they obtained better bounds via observing some properties of the constructions.

1) Construction 1: We first give an explicit constructions as follows.

Definition 1: Given λ, m, and d such that d divides m. We define $C_1(\lambda,m,d) = \{(x_1,\ldots,x_{m\lambda}) \in S_{m\lambda}^\lambda : \forall i \in [m\lambda], x_i \equiv i (\mathrm{mod}\ d)\}$.

Theorem 3: If $m = ad, C_1(\lambda,m,d)$ is a $(\lambda, m\lambda, d)$-FPA of cardinality $(\frac{(a\lambda)!}{(\lambda!)^a})^d$.

Proof: Since $C_1(\lambda,m,d) \subseteq S_{m\lambda}^\lambda, C_1(\lambda,m,d)$ has frequency λ and length $m\lambda$. To show the minimum distance, consider two different elements $\boldsymbol{x}, \boldsymbol{y} \in C_1(\lambda,m,d)$ and assume their ith entries are different, i.e., $x_i \neq y_i$. Since $x_i \equiv y_i (\mathrm{mod}\ d)$, we have d divides $(x_i - y_i)$ and we have $|x_i - y_i| \ge d$. Hence the minimum distance is at least d.

Now we turn to the cardinality of $C_1(\lambda,m,d)$. For each $j \in [d]$, define a family of index sets

$$\begin{aligned}
X_j &= \{(k-1)d + j : k \in [a\lambda]\}\\
&= \{j, d+j, \ldots, (a\lambda - 1)d + j\}.
\end{aligned}$$

For any $\boldsymbol{x} \in C_1(\lambda,m,d)$ and any $i \in X_j$, we have $x_i \equiv j (\mathrm{mod}\, d)$ and x_i can be any member in $\{j, d+j, \ldots, (a-1)d + j\} = \{(k-1)d + j : k \in [a]\}$. Note that each $(k-1)d + j$ appears exactly λ times in $\boldsymbol{x}$. Thus for each $j \in [d]$, there are $\frac{(a\lambda)!}{(\lambda!)^a}$ possible combinations for $\boldsymbol{x}$ at the indices in X_j. Thus, $|C_1(\lambda,m,d)| = (\frac{(a\lambda)!}{(\lambda!)^a})^d$ and the theorem holds. ■

Theorem 3 implies the following lower bound on the cardinality of (λ,n,d)-FPAs.

Corollary 1: If $m = ad$, then $F_\infty(\lambda,n,d) \ge (\frac{(a\lambda)!}{(\lambda!)^a})^d$.

The information rate of this construction is higher than the lower bound in theorem 2 under certain parameters. The code length, information length, and the bounds are counted in bits. Since $|S_{\lambda m}^\lambda|$ and $|C_1(\lambda,m,d)|$ do not have to be powers of 2, the bit-length could be a fractional number. Table I illustrates some codes of 100-symbols. In the first three rows, we discover the information length is longer than the lower bound and it is

TABLE I
PARAMETERS FOR CONSTRUCTION C_1

λ	m	d	Code length	Info. length	Lower bound	Upper bound
10	10	5	306.9	87.5	14.8	140.0
5	20	5	386.6	167.2	110.8	220.0
2	50	5	474.8	255.4	233.1	340.2
1	100	5	524.8	305.4	319.6	436.2
1	100	10	524.8	217.9	226.1	355.4
1	100	20	524.8	138.1	130.3	257.6
1	100	50	524.8	50.0	1.4	150.9

closer to the upper bound when λ is greater. For the last four rows, we fix $\lambda = 1$. When the d is small, see row 4 and 5, C_1 even does not reach the lower bound. But for large d, C_1 still has a chance to outperform the lower bound.

2) Simple Recursive Constructions: We give several recursive constructions for FPAs. These constructions could give codes with higher information rate than construction 1; however, they might require a better initial code than C_1. The first one is by concatenation. For $\boldsymbol{x} = (x_1, \ldots, x_p)$ and $\boldsymbol{y} = (y_1, \ldots, y_q)$, let $\boldsymbol{x} \,|\, \boldsymbol{y}$ denote $(x_1, \ldots, x_p, y_1, \ldots, y_q)$.

Definition 2: Given a $(\lambda, \lambda m, d)$-FPA C_λ and a $(\rho, \rho m, d)$-FPA C_ρ. Define

$$C_\lambda * C_\rho = \{\boldsymbol{c}_\lambda \,|\, \boldsymbol{c}_\rho : \boldsymbol{c}_\lambda \in C_\lambda, \boldsymbol{c}_\rho \in C_\rho\}.$$

Theorem 4: Given a $(\lambda, \lambda m, d)$-FPA C_λ and a $(\rho, \rho m, d)$-FPA C_ρ, $C_\lambda * C_\rho$ is a $(\lambda + \rho, \lambda m + \rho m, d)$-FPA of cardinality $|C_\lambda| \cdot |C_\rho|$.

Proof: Consider $\boldsymbol{x}, \boldsymbol{x}' \in C_\lambda$ and $\boldsymbol{y}, \boldsymbol{y}' \in C_\rho$. $\boldsymbol{x}|\boldsymbol{y} \neq \boldsymbol{x}'|\boldsymbol{y}'$ if and only if $\boldsymbol{x} \neq \boldsymbol{x}'$ or $\boldsymbol{y} \neq \boldsymbol{y}'$, hence the cardinality of $C_\lambda * C_\rho$ is $|C_\lambda| \cdot |C_\rho|$. Since Chebyshev distance between $\boldsymbol{x} \,|\, \boldsymbol{y}$ and $\boldsymbol{x}' \,|\, \boldsymbol{y}'$ is $\max\{d_{\max}(\boldsymbol{x}, \boldsymbol{x}'), d_{\max}(\boldsymbol{y}, \boldsymbol{y}')\}$, we have the minimum distance of $C_\lambda * C_\rho$ is at least

$$\min_{\substack{\boldsymbol{x}, \boldsymbol{x}' \in C_\lambda \\ \boldsymbol{y}, \boldsymbol{y}' \in C_\rho \\ \boldsymbol{x}|\boldsymbol{y} \neq \boldsymbol{x}'|\boldsymbol{y}'}} \{d_{\max}(\boldsymbol{x}, \boldsymbol{x}'), d_{\max}(\boldsymbol{y}, \boldsymbol{y}')\} = d.$$

The frequency and length of $C_\lambda * C_\rho$ are trivially $\lambda + \rho$ and $\lambda m + \rho m$, respectively. We conclude $C_\lambda * C_\rho$ is a $(\lambda + \rho, \lambda m + \rho m, d)$-FPA of cardinality $|C_\lambda| \cdot |C_\rho|$. ■

Corollary 2: For integers λ, ρ, m and $d, F_\infty(\lambda + \rho, \lambda m + \rho m, d) \geq F_\infty(\lambda, \lambda m, d) F_\infty(\rho, \rho m, d)$.

The second recursive construction is by interleaving technics. For an ℓ-tuple $\boldsymbol{c}$ and integers r, j, let $\nu(\boldsymbol{c}, r, j) = r\boldsymbol{c} + (j-1) \cdot \mathbf{1}$ where $\mathbf{1} = \overbrace{(1, \ldots, 1)}^{\ell}$.

Definition 3: Given a (λ, n, d)-FPA C and a positive integer r. Define

$$C_2(C, r) = \{\nu(\boldsymbol{c}_1, r, 1) | \cdots | \nu(\boldsymbol{c}_r, r, r) : \forall i \in [r], \boldsymbol{c}_i \in C\}.$$

Theorem 5: Given C and r as above, $C_2(C, r)$ is a (λ, rn, rd)-FPA of cardinality $|C|^r$.

Corollary 3: $F_\infty(\lambda, rn, rd) \geq F_\infty(\lambda, n, d)^r$.

The following recursive constructions use extra k new symbols. We give three kinds of k-symbol extension constructions. Assume we have $\boldsymbol{x} = (x_1, \ldots, x_{\lambda m}) \in S^\lambda_{\lambda m}$ and $\boldsymbol{y} = (y_1, \ldots, y_{k\lambda}) \in [m+k]^{k\lambda}$ such that for any $i \in [m+k]$, there are at most λ entries in $\boldsymbol{y}$ equal to i. Our goal is to extend $\boldsymbol{x}$ into an element in $S^\lambda_{\lambda(m+k)}$. Observe that initially $\boldsymbol{y} \,|\, \boldsymbol{x}$ may not be a legitimate element in $S^\lambda_{\lambda(m+k)}$. However, we can reassign values to some of $\boldsymbol{x}$'s entries such that $\boldsymbol{y} \,|\, \boldsymbol{x}$ is in $S^\lambda_{\lambda(m+k)}$. To do that we define a total order for the entries in $\boldsymbol{x}$, i.e., for $i, j \in [\lambda m]$, we say x_i is larger than x_j if the value of x_i is strictly larger than x_j's, or when $x_i = x_j$ and $i < j$. Let γ_i be the number of entries in $\boldsymbol{y}$ equal to i. The extension algorithm $\phi_k(\boldsymbol{y}, \boldsymbol{x})$ operates as follows:

```
1. for $i = m + k$ downto 1 do
2.     set the largest $\lambda - \gamma_i$ unchanged entries in $\boldsymbol{x}$ to $i$
       and mark them as changed;
3. next $i$
4. return $\boldsymbol{y} \,|\, \boldsymbol{x}$;
```

For example, let $k = 2, \boldsymbol{y} = (1, 2, 3, 4)$ and $\boldsymbol{x} = (1, 2, 1, 2)$, then we have $\phi_k(\boldsymbol{y}, \boldsymbol{x}) = (1, 2, 3, 4, 1, 4, 2, 3)$.

It is easy to check that $\phi_k(\boldsymbol{y}, \boldsymbol{x})$ returns a permutation of frequency λ and length $m\lambda + k\lambda$. Since every symbol appears at most λ times in $\boldsymbol{y}$, we have $\gamma_j \leq \lambda$ for every $j \in [m+k]$, and there are $\lambda - \gamma_j + \gamma_j = \lambda$ entries equal to j in $\boldsymbol{y} \,|\, \boldsymbol{x}$ after the jth iteration. Moreover, those entries will not be changed afterwards. This shows $\phi_k(\boldsymbol{y}, \boldsymbol{x})$ transforms $\boldsymbol{y} \,|\, \boldsymbol{x}$ into a legitimate element in $S^\lambda_{\lambda(m+k)}$.

For positive integers k, t with $k \leq t$, we consider a selective function $f : [k] \rightarrow [t]$, which selects k elements in order from $[t]$, i.e., $f(i)$ is the ith smallest one among the selected k elements. For any $\pi \in S^\lambda_{k\lambda}$ and any t-tuple $\boldsymbol{s}$ with $1 \leq s_1 < \ldots < s_t \leq m + k$, define $\psi_{k,t,\boldsymbol{s}}(f, \pi) = (s_{f(\pi_1)}, \ldots, s_{f(\pi_{k\lambda})})$. For example, set $\lambda = 3, k = 3, t = 4, s_1 = 1, s_2 = 3, s_3 = 5, s_4 = 7, f(x) = x + 1$, and $\pi = (1, 2, 3, 2, 2, 3, 1, 3, 1)$, we have $\psi_{k,t,\boldsymbol{s}}(f, \pi) = (s_2, s_3, s_4, s_3, s_3, s_4, s_2, s_4, s_2) = (3, 5, 7, 5, 5, 7, 3, 7, 3)$. We define $\Psi_{k,t,\boldsymbol{s}}(C) = \{\psi_{k,t,\boldsymbol{s}}(f, \pi) : f \text{ is a selective function from } [k] \text{ to } [t] \text{ and } \pi \in C\}$ for $C \subseteq S^\lambda_{k\lambda}$. We give two constructions by extending $\boldsymbol{x}$ with $\boldsymbol{y} \in \Psi_{k,t,\boldsymbol{s}}(S^\lambda_{k\lambda})$ and with $\boldsymbol{y} \in \Psi_{k,k,\boldsymbol{s}}(C)$ where C is an FPA. Both recursive constructions inherit the minimum distance of the base FPA.

Definition 4: Given positive integers k, t with $t \geq k$, a $(\lambda, \lambda m, d)$-FPA C and a t-tuple $\boldsymbol{s}$ with $1 \leq s_1 < s_2 < \ldots < s_t \leq m + k$ and $s_{i+1} - s_i \geq d$ for $i \in [t-1]$. Define $C_3(C, k, t, \boldsymbol{s}) = \{\phi_k(\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{x} \in C, \boldsymbol{y} \in \Psi_{k,t,\boldsymbol{s}}(S^\lambda_{k\lambda})\}$.

Definition 5: Given a positive integers k, a $(\lambda, \lambda m, d)$-FPA C, a $(\lambda, \lambda k, \delta)$-FPA C' and a k-tuple $\boldsymbol{s}$ with $1 \leq s_1 < s_2 < \ldots < s_k \leq m + k$ and $s_{i+1} - s_i \geq \lceil \frac{d}{\delta} \rceil$ for $i \in [t-1]$. Define $C_4(C, C', k, \boldsymbol{s}) = \{\phi_k(\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{x} \in C, \boldsymbol{y} \in \Psi_{k,k,\boldsymbol{s}}(C')\}$.

Both $C_3(C, k, t, \boldsymbol{s})$ and $C_4(C, C', k, \boldsymbol{s})$ are $(\lambda, m\lambda + k\lambda, d)$-FPAs. To prove this, we argue that the distance between any pair of codewords is at least d when they are constructed from distinct $\boldsymbol{y}$'s or distinct $\boldsymbol{x}$'s. Therefore, we need the following lemma.

Lemma 4: For two selective functions f and g mapping from $[k]$ to $[t]$ and $\pi, \rho \in S_{k\lambda}^{\lambda}$, let $\boldsymbol{y} = \psi_{k,t,\boldsymbol{s}}(f, \pi)$ and $\boldsymbol{y}' = \psi_{k,t,\boldsymbol{s}}(g, \rho)$. We have $d_{\max}(\boldsymbol{y}, \boldsymbol{y}') \geq d$ when $f \neq g$ or $\pi \neq \rho$. Moreover, $d_{\max}(\boldsymbol{y}, \boldsymbol{y}') \geq d \cdot d_{\max}(\pi, \rho)$ when $f = g$ and $\pi \neq \rho$.

Proof: Let $F = \{f(i) : i \in [k]\}$ and $G = \{g(i) : i \in [k]\}$. If $f \neq g$, there exists i^* such that $i^* \in F \backslash G$. Therefore $f(\pi_j) = i^* \neq g(\rho_j)$ implies $y_j \neq y'_j$. We have $d_{\max}(\boldsymbol{y}, \boldsymbol{y}') \geq |y_j - y'_j| \geq d$. If $f = g$ and $\pi \neq \rho$, then there exists j^* such that $\pi_{j^*} \neq \rho_{j^*}$. Assume $|\pi_{j^*} - \rho_{j^*}| = \delta$, we have $d_{\max}(\boldsymbol{y}, \boldsymbol{y}') \geq |y_{j^*} - y'_{j^*}| \geq |s_{f(\pi_{j^*})} - s_{f(\rho_{j^*})}| \geq \delta \min_{i \in [t-1]}(s_{i+1} - s_i) \geq \delta d$. ■

Lemma 5: If for every $i \in [m+k], \boldsymbol{y}$ has either λ entries equal to i or no such entry, then for $\boldsymbol{x}, \boldsymbol{x}' \in S_{\lambda m}^{\lambda}, d_{\max}(\phi_k(\boldsymbol{y}, \boldsymbol{x}), \phi_k(\boldsymbol{y}, \boldsymbol{x}')) \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}')$.

Proof: By the assumption on $\boldsymbol{y}$, we know γ_i is either λ or 0. Thus in every iteration of $\phi_k(\boldsymbol{y}, \boldsymbol{z})$ either 0 or λ symbols in $\boldsymbol{z}$ are changed. Suppose that we run $\phi_k(\boldsymbol{y}, \boldsymbol{x})$ and $\phi_k(\boldsymbol{y}, \boldsymbol{x}')$ in parallel. According to ϕ_k, there are either λ entries or nothing changed in each iteration. Without loss of generality, let j be the index such that $x_j - x'_j = d$. Note that the algorithm changes only one kind of symbol in each iteration. Therefore x'_j must be changed at least $x_j - x'_j = d$ iterations later after x_j is changed and the magnitude is smaller than x_j's. This implies $d_{\max}(\phi_k(\boldsymbol{y}, \boldsymbol{x}), \phi_k(\boldsymbol{y}, \boldsymbol{x}')) \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}')$. ■

Now, we prove the following theorems.

Theorem 6: $C_3(C, k, t, \boldsymbol{s})$ is a $(\lambda, m\lambda + k\lambda, d)$-FPA of cardinality $\binom{t}{k}\frac{(k\lambda)!}{(\lambda!)^k}|C|$.

Proof: Consider codewords $z = \phi_k(\psi_{k,t,\boldsymbol{s}}(f, \pi), \boldsymbol{x})$ and $z' = \phi_k(\psi_{k,t,\boldsymbol{s}}(g, \rho), \boldsymbol{x}')$. If $f \neq g$ or $\pi \neq \rho$, then we have $d_{\max}(z, z') \geq d_{\max}(\psi_{k,t,\boldsymbol{s}}(f, \pi), \psi_{k,t,\boldsymbol{s}}(g, \rho)) \geq d$ by lemma 4. If $f = g$ and $\pi = \rho$, then $d_{\max}(z, z') \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}') \geq d$ by lemma 5 and C is a $(\lambda, \lambda m, d)$-FPA. Note that there are $\binom{t}{k}$ selective functions from $[k]$ to $[t]$ and $\frac{(k\lambda)!}{(\lambda!)^k}$ permutations in $S_{k\lambda}^{\lambda}$, thus the theorem holds. ■

Corollary 4: $F_\infty(\lambda, m\lambda + k\lambda, d) \geq \binom{t}{k}\frac{(k\lambda)!}{(\lambda!)^k}F_\infty(\lambda, m\lambda, d)$ for $k \leq t$ and $td < m + k$.

Theorem 7: $C_4(C, C', k, \boldsymbol{s})$ is a $(\lambda, m\lambda + k\lambda, d)$-FPA of cardinality $|C'| \cdot |C|$.

Proof: Consider codewords $z = \phi_k(\psi_{k,k,\boldsymbol{s}}(f, \pi), \boldsymbol{x})$ and $z' = \phi_k(\psi_{k,k,\boldsymbol{s}}(g, \rho), \boldsymbol{x}')$. Since there is only one selective function from $[k]$ to $[k]$, we have $f = g$. If $\pi \neq \rho$, then we have $d_{\max}(z, z') \geq \delta \cdot \frac{d}{\delta} = d$ by lemma 4 and C' is a $(\lambda, k\lambda, \delta)$-FPA. If $\pi = \rho$, then $d_{\max}(z, z') \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}') \geq d$ by lemma 5 and C is a $(\lambda, \lambda m, d)$-FPA. Hence the theorem holds. ■

Corollary 5: $F_\infty(\lambda, m\lambda + k\lambda, d) \geq F_\infty(\lambda, k\lambda, \delta) \cdot F_\infty(\lambda, m\lambda, d)$ for $k \cdot \lceil \frac{d}{\delta} \rceil < m + k$.

Finally, we provide another symbol extension construction which allows us to obtain a FPA of greater cardinality while the minimum distance is decreased by 1. Similar to the construction C_3, we relax the constraints on $\boldsymbol{y}$ by allowing some symbols appearing less than λ times in $\boldsymbol{y}$.

Lemma 6: If for every $i \in [m+k], \boldsymbol{y}$ has at most λ entries equal to i, then $d_{\max}(\phi_k(\boldsymbol{y}, \boldsymbol{x}), \phi_k(\boldsymbol{y}, \boldsymbol{x}')) \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}') - 1$, for any $\boldsymbol{x}, \boldsymbol{x}' \in S_{m\lambda}^{\lambda}$.

Proof: Consider the jth entries of $\boldsymbol{x}$ and $\boldsymbol{x}'$, and without loss of generality, let $x_j - x'_j = d > 0$. Assume x_j and x'_j is the αth and the βth smallest entry in $\boldsymbol{x}$ and $\boldsymbol{x}'$ respectively. α and β must be in the form $x_j\lambda - p$ and $x'_j\lambda - q$ for some $p, q \in \{0, \lambda - 1\}$. Since $x_j - x'_j = d$, we have $\alpha - \beta \geq d\lambda - \lambda + 1$.

Suppose that we run $\phi_k(\boldsymbol{y}, \boldsymbol{x})$ and $\phi_k(\boldsymbol{y}, \boldsymbol{x}')$ in parallel. According to ϕ_k, there are at most λ entries changed in each iteration. The iteration difference between the iterations when x'_j and x_j are respectively changed is at least $\lfloor \frac{\alpha - \beta}{\lambda} \rfloor \geq \lfloor \frac{d\lambda - \lambda + 1}{\lambda} \rfloor = d - 1$. The corresponding entries in the output have difference at least $d - 1$, and thus we conclude $d_{\max}(\phi_k(\boldsymbol{y}, \boldsymbol{x}), \phi_k(\boldsymbol{y}, \boldsymbol{x}')) \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}') - 1$. ■

Lemma 6 shows that the distance at most decreases 1 after applying $\phi_k(\boldsymbol{y}, \cdot)$. From this point of view, we can trade minimum distance for larger code size. For integer k, t and t-tuple $\boldsymbol{s}$, let $Q(k, t, \boldsymbol{s})$ be the set of vectors of length $k\lambda$ consisting of symbols in $\{s_1, \ldots, s_t\}$ such that no symbol appears more than λ times. Note that for any simple set $\boldsymbol{s}, Q(k, t, \boldsymbol{s})$ has the same cardinality, thus we define $q(k, t) = |Q(k, t, \boldsymbol{s})|$.

Definition 6: Given positive integers k, t with $t \geq k$, a $(\lambda, m\lambda, d+1)$-FPA C and a t-tuple $\boldsymbol{s}, 1 \leq s_1 < s_2 < \ldots < s_t \leq m + k$ and $s_{i+1} - s_i \geq d$ for $i \in [t-1]$. Define $C_5(C, k, t, \boldsymbol{s}) = \{\phi_k(\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{x} \in C, \boldsymbol{y} \in Q(k, t, \boldsymbol{s})\}$.

Theorem 8: For $d \geq 1, C_5(C, k, t, \boldsymbol{s})$ is a $(\lambda, m\lambda + k\lambda, d)$-FPA of cardinality $q(k, t)|C|$.

Proof: Consider codewords $\boldsymbol{z} = \phi_k(\boldsymbol{y}, \boldsymbol{x})$ and $\boldsymbol{z}' = \phi_k(\boldsymbol{y}', \boldsymbol{x}')$ where $\boldsymbol{y}, \boldsymbol{y}' \in Q(k, t, \boldsymbol{s})$ and $\boldsymbol{x}, \boldsymbol{x}'$ in C. We know that $d_{\max}(z, z') \geq d$ when $\boldsymbol{y} \neq \boldsymbol{y}'$, since for some $i \in [k\lambda]$, $|y_i - y'_i| \geq \min_{j \in [t-1]} s_{j+1} - s_j \geq d$. By lemma 6, if $\boldsymbol{y} = \boldsymbol{y}'$, then $d_{\max}(z, \boldsymbol{z}') \geq d_{\max}(\boldsymbol{x}, \boldsymbol{x}') \geq d + 1 - 1 = d > 0$. These facts imply $\boldsymbol{z} \neq \boldsymbol{z}'$ if $\boldsymbol{y} \neq \boldsymbol{y}'$ or $\boldsymbol{x} \neq \boldsymbol{x}'$. By the construction of C_5, it is clear that the cardinality is $q(k, t)|C|$. ■

Corollary 6: $F_\infty(\lambda, m\lambda + k\lambda, d) \geq q(k, t)F_\infty(\lambda, m\lambda, d+1)$ for $k \leq t$ and $td < m + k$.

Let $C^{(\lambda, d)} = \{\pi \in S_{(d+1)\lambda}^{\lambda} : \pi_1, \ldots, \pi_\lambda \in \{1, d+1\}$ and $\pi_{\lambda+1} \leq \cdots \leq \pi_{(d+1)\lambda}\}$. The code constructed by the simple encoding algorithm in the next section can also be obtained by

```
1. $C^1 = C^{(\lambda, d)}$;
2. for $i = 2$ to $k$ do
3.     $C^i = C_5(C^i, 1, 2, (1, d+i))$;
4. next
5. Output $C^k$;
```

However, the minimum distance of the output is d. In other words, the procedure above never decrease the minimum distance on line 3. This fact implies that our analysis on the minimum distance of $C_5(C, k, t, \boldsymbol{s})$ is not tight.

III. Construction With Efficient Encoding and Decoding

In this section, we give a construction with efficient encoding and decoding algorithms. The idea of this construction is based on the previous work by Lin *et al.* [11]. We generalize their algorithm for constructing FPAs. Furthermore, we give the first local decoding algorithm for FPAs under Chebyshev distance.

A. Encoding Algorithm

We give an encoding algorithm $E^{\lambda}_{n,k}$ which convert k-bit message into a permutation in S^{λ}_{n} where $n \geq k + \lambda$.

Algorithm $E^{\lambda}_{n,k}$

```
Input: $(m_1, \dots, m_k) \in Z_2^k$
Output: $(x_1, \dots, x_n) \in S_n^{\lambda}$
$\max \leftarrow n; \min \leftarrow 1;$
for $i \leftarrow 1$ to $k$ do
  if $m_i = 1$
    then $\{x_i \leftarrow \lceil \frac{\max}{\lambda} \rceil; \max \leftarrow \max - 1;\}$
    else $\{x_i \leftarrow \lceil \frac{\min}{\lambda} \rceil; \min \leftarrow \min + 1;\}$
for $i \leftarrow k + 1$ to $n$ do
  $x_i \leftarrow \lceil \frac{\min}{\lambda} \rceil; \min \leftarrow \min + 1;$
Output $(x_1, \dots, x_n)$;
```

The encoding algorithm $E^{\lambda}_{n,k}$ maps binary vectors from Z_2^k to S^{λ}_{n}. For examples, the output of $E^{2}_{10,4}(0,1,0,0)$ is $(1,5,1,2,2,3,3,4,4,5)$, and $E^{2}_{10,4}(0,1,1,1)$ outputs $(1,5,5,4,1,2,2,3,3,4)$. It is clear that $E^{\lambda}_{n,k}$ runs in $O(n)$ time while encoding any k-bit message. Next we investigate the properties of the code obtained by $E^{\lambda}_{n,k}$. Let $C^{\lambda}_{n,k}$ be the image of $E^{\lambda}_{n,k}$.

Theorem 9: $C^{\lambda}_{n,k}$ is a $(\lambda, n, \lfloor \frac{n-k}{\lambda} \rfloor)$-FPA with cardinality 2^k.

Proof: Consider two messages $\boldsymbol{p} = (p_1, \dots, p_k)$ and $\boldsymbol{q} = (q_1, \dots, q_k) \in Z_2^k$. Let $\boldsymbol{x^p}$ and $\boldsymbol{x^q}$ be the outputs of $E^{\lambda}_{n,k}$, respectively. Let r be the smallest index such that $p_r \neq q_r$. Without loss of generality, we assume $p_r = 1, q_r = 0$ and there are exactly z zeroes among $p_1, \dots, p_{r-1}$. Consequently, $x^{\boldsymbol{p}}_r$ is set to $\lceil \frac{\max}{\lambda} \rceil = \lceil \frac{n-r+1+z}{\lambda} \rceil$ and $x^{\boldsymbol{q}}_r$ is set to $\lceil \frac{\min}{\lambda} \rceil = \lceil \frac{1+z}{\lambda} \rceil$ by $E^{\lambda}_{n,k}$. The distance between $\boldsymbol{x^p}$ and $\boldsymbol{x^q}$ is

$$\begin{aligned} &\left\lceil \frac{n-r+1+z}{\lambda} \right\rceil - \left\lceil \frac{1+z}{\lambda} \right\rceil \\ &\quad > \frac{n-r+1+z}{\lambda} - \frac{1+z}{\lambda} - 1 \\ &\quad = \frac{n-r}{\lambda} - 1 \\ &\quad \geq \frac{n-k}{\lambda} - 1, \text{ since } r \leq k. \end{aligned}$$

The first inequality holds by the fact of ceiling function: $a \leq \lceil a \rceil < a + 1$, for any real number a. Note that the distance has integer value only here. If $\frac{n-k}{\lambda}$ is integer then the distance is at least $\lfloor \frac{n-k}{\lambda} \rfloor$; else it is at least $\lceil \frac{n-k}{\lambda} - 1 \rceil$, which is $\lfloor \frac{n-k}{\lambda} \rfloor$ exactly, i.e., the distance between any two codewords in $C^{\lambda}_{n,k}$ is at least $\lfloor \frac{n-k}{\lambda} \rfloor$. Since every message is encoded into a distinct codeword, we have $C^{\lambda}_{n,k} = 2^k$. ■

Since $C^{\lambda}_{n,k}$ is a $(\lambda, n, \lfloor \frac{n-k}{\lambda} \rfloor)$-FPA, we let $d = \lfloor \frac{n-k}{\lambda} \rfloor$ for convenience.

B. Unique Decoding Algorithm

Unique decoding algorithms for classic error correcting codes are usually much more complicated than their encoding algorithms. While, our proposed decoding algorithm $U^{\lambda}_{n,k}$ remains simple.

Algorithm $U^{\lambda}_{n,k}$

```
Input: $(x_1, \dots, x_n) \in S_n^{\lambda}$
Output: $(m_1, \dots, m_k) \in Z_2^k$
$\max \leftarrow n; \min \leftarrow 1;$
for $i \leftarrow 1$ to $k$ do
  if $|x_i - \lceil \frac{\max}{\lambda} \rceil| < |x_i - \lceil \frac{\min}{\lambda} \rceil|$
    then $\{m_i \leftarrow 1; \max \leftarrow \max - 1;\}$
    else $\{m_i \leftarrow 0; \min \leftarrow min + 1;\}$
Output $(m_1, \dots, m_k)$;
```

The running time of $U^{\lambda}_{n,k}$ is clearly $O(k)$, even faster than the encoding algorithm. We show its correctness as follows.

Theorem 10: Given a permutation $\boldsymbol{x} = (x_1, \dots, x_n)$ which is $\frac{d-1}{2}$-close to $E^{\lambda}_{n,k}(\boldsymbol{m})$ for some $\boldsymbol{m} \in Z_2^k$, algorithm $U^{\lambda}_{n,k}$ outputs $\boldsymbol{m}$ correctly.

Proof: By contradiction, assume $U^{\lambda}_{n,k}$ outputs $\hat{\boldsymbol{m}} = (\hat{m}_1, \dots, \hat{m}_k) \neq \boldsymbol{m}$. Let $E^{\lambda}_{n,k}(\boldsymbol{m}) = (y_1, \dots, y_n), r$ be the smallest index such that $m_r \neq \hat{m}_r$ and z be the number of zeroes in $m_1, \dots, m_{r-1}$. At the beginning of the rth iteration, $\max = n - r + 1 + z$ and $\min = 1 + z$ because for every $i < r, m_i = \hat{m}_i$. Without loss of generality, assume $1 = m_r \neq \hat{m}_r = 0$. Note that y_r is set to $\lceil \frac{\max}{\lambda} \rceil = \lceil \frac{n-r+1+z}{\lambda} \rceil$ by $E^{\lambda}_{n,k}$. While $\hat{m}_r$ is decoded to 0 by $U^{\lambda}_{n,k}$, we have $|x_r - \lceil \frac{\max}{\lambda} \rceil| \geq |x_r - \lceil \frac{\min}{\lambda} \rceil|$. Thus,

$$\begin{aligned} &d_{\max}\left(\boldsymbol{x}, E^{\lambda}_{n,k}(\boldsymbol{m})\right) \\ &\quad \geq |x_r - y_r| = \left| x_r - \left\lceil \frac{\max}{\lambda} \right\rceil \right| \\ &\quad \geq \frac{1}{2}\left(\left| x_r - \left\lceil \frac{\max}{\lambda} \right\rceil \right| + \left| x_r - \left\lceil \frac{\min}{\lambda} \right\rceil \right| \right) \\ &\quad \geq \frac{1}{2}\left(\left\lceil \frac{\max}{\lambda} \right\rceil - \left\lceil \frac{\min}{\lambda} \right\rceil \right) \\ &\quad = \frac{1}{2}\left(\left\lceil \frac{n-r+1+z}{\lambda} \right\rceil - \left\lceil \frac{1+z}{\lambda} \right\rceil \right) \geq \frac{d}{2}. \end{aligned}$$

The last inequality is true, since we know $\lceil \frac{n-r+1+z}{\lambda} \rceil - \lceil \frac{1+z}{\lambda} \rceil \geq \lfloor \frac{n-k}{\lambda} \rfloor = d$ from the proof of Theorem 9. This contradicts that $\boldsymbol{x}$ is $\frac{d-1}{2}$-close to $E^{\lambda}_{n,k}(\boldsymbol{m})$. ■

C. Local Decoding Algorithm

Next we show a local decoding algorithm $L^{\lambda}_{n,k}$, which is a probabilistic algorithm.

Algorithm $L_{n,k}^{\lambda}$

Input: $i \in [n], (x_1, \ldots, x_n) \in S_n^{\lambda}$

Output: m_i, the ith message bit

```
1. $J \leftarrow \{i+1, \ldots, n\}$;
2. do
3.   Uniformly and randomly pick $j \in J$;
4.   if $x_i > x_j$ then output 1;
5.   if $x_i < x_j$ then output 0;
6.   $J \leftarrow J - \{j\}$;
7. loop;
```

$L_{n,k}^{\lambda}$ allows us to decode ith message bit more efficiently than $U_{n,k}^{\lambda}$, but it may give a wrong result with certain probability. To illustrate this fact, we consider two permutations $\boldsymbol{x} = (1,4,1,2,2,3,3,4,5,5), \boldsymbol{y} = (1,5,1,2,2,3,3,4,4,5) = E_{10,4}^{2}(0,1,0,0)$ and the result of $L_{10,4}^{2}(2,\boldsymbol{x})$. Since $d_{\max}(\boldsymbol{x},\boldsymbol{y}) = 1 \leq \frac{\lfloor \frac{10-4}{2} \rfloor - 1}{2}, U_{10,4}^{2}(\boldsymbol{x})$ outputs $(0,1,0,0)$. $L_{10,4}^{2}(2,\boldsymbol{x})$ should output 1, but when j is picked as 9 or 10 at line $3, L_{10,4}^{2}(2,\boldsymbol{x})$ outputs 0.

We discuss its efficiency and error probability in this subsection. We prove that it reads at most $\lambda + 1$ entries of the received word in Lemma 7, hence its running time is $O(\lambda)$. It has a chance to output wrongly, but we show that the error probability is small in Theorem 11. Furthermore, $L_{n,k}^{\lambda}$ always outputs correct message bit when it was given a codeword as input, see Corollary 7.

Lemma 7: Given a permutation $\boldsymbol{x} = (x_1, \ldots, x_n) \in C_{n,k}^{\lambda}$ and an index $i \in [k], L_{n,k}^{\lambda}$ terminates within λ iterations.

Proof: By contradiction, assume $L_{n,k}^{\lambda}$ does not output before the end of the λth iteration. For $\ell \leq \lambda$, let j_ℓ be the index picked in the ℓth iteration. For every $\ell \leq \lambda$, we have $x_i = x_{j_\ell}$, otherwise $L_{n,k}^{\lambda}$ outputs at the ℓth iteration. Therefore, there are at least $\lambda + 1$ entries of $\boldsymbol{x}$ equal to x_i. It implies $\boldsymbol{x} \notin C_{n,k}^{\lambda}$, a contradiction. There is some $x_{j_\ell} \neq x_i$, and $L_{n,k}^{\lambda}$ outputs in the ℓth iteration. ■

Theorem 7: Given a permutation $\boldsymbol{x} = (x_1, \ldots, x_n)$ δ-close to a codeword $E_{n,k}^{\lambda}(\boldsymbol{m}) = (y_1, \ldots, y_n) \in C_{n,k}^{\lambda}$ for some $\boldsymbol{m}$ and an index $i \in [k], L_{n,k}^{\lambda}$ outputs m_i with probability at least $1 - \frac{2\delta+1}{d}$ at its first iteration.

Proof: Without loss of generality, we assume $m_i = 0, y_i = t$ and let u be the maximum number among $y_{i+1}, \ldots, y_n$, i.e., at the start of the ith iteration $\min = t$ and $\max = u$ while encoding. Assume there are γ numbers equal to t among $y_1, \ldots, y_{i-1}$, and there are γ' numbers equal to u among $y_{i+1}, \ldots, y_n$. According to the encoding algorithm, we have

$$\{y_{i+1}, \ldots, y_n\} = \{\overbrace{t, \ldots, t}^{\lambda-\gamma-1}, \overbrace{t+1, \ldots, t+1}^{\lambda}, \ldots, \overbrace{u, \ldots, u}^{\gamma'}\}.$$

Since $d_{\max}(\boldsymbol{x}, E_{n,k}^{\lambda}(\boldsymbol{m})) \leq \delta$, we have $|x_j - y_j| \leq \delta$ and $|x_i - y_i| \leq \delta$. The probability that $L_{n,k}^{\lambda}$ does not output m_i at the first iteration is

$$\begin{aligned} \Pr[x_i \geq x_j] &\leq \Pr[y_i + \delta \geq x_j] \\ &\leq \Pr[y_i + \delta \geq y_j - \delta] \\ &= \Pr[y_i + 2\delta \geq y_j]. \end{aligned}$$

There are at most $2\delta\lambda + \lambda - \gamma - 1$ possible y_j's less than or equal to $y_i + 2\delta$. Thus,

$$\begin{aligned} \Pr[x_i \geq x_j] &\leq \frac{(2\delta+1)\lambda - \gamma - 1}{n-i} \\ &\leq \frac{2\delta\lambda + \lambda}{d\lambda} = \frac{2\delta+1}{d}. \end{aligned}$$

Therefore, the probability that $L_{n,k}^{\lambda}$ outputs m_i correctly at the first iteration is at least $1 - \frac{2\delta+1}{d}$. ■

Corollary 7: Given a codeword $\boldsymbol{x} = E_{n,k}^{\lambda}(\boldsymbol{m})$ for some $\boldsymbol{m}$ and an index $i, L_{n,k}^{\lambda}$ outputs m_i correctly.

Proof: By Lemma 7, there exists $\ell \leq \lambda$ such that $L_{n,k}^{\lambda}$ terminates at the ℓth iteration. Let j be the index picked at the ℓth iteration, we have $x_j \neq x_i$, where $j > i$. Note that $\boldsymbol{x}$ is a codeword: $x_i < x_j$ implies $m_i = 0$ and $x_i > x_j$ implies $m_i = 1$. Hence, $L_{n,k}^{\lambda}$ outputs m_i correctly. ■

A private information retrieval (PIR) system consists of q servers. All servers know a codeword $\boldsymbol{x} = (x_1, \ldots, x_n)$ representing a message $\boldsymbol{m} = (m_1, \ldots, m_k)$, and a user wants to know one bit m_i of $\boldsymbol{m}$ via query a symbol from each server. We say a PIR has *retrievability* r if the user can obtain the message bit with probability r. Let $\mathcal{D}(s,i)$ be the distribution of entry queried from server s when the user tries to retrieve m_i. A PIR has *privacy* p if $\max_{i,j \in [k], s \in [q]} \Delta(\mathcal{D}(s,i), \mathcal{D}(s,j)) \leq p$, where $\Delta(\cdot,\cdot)$ is the statistical distance. A (q,r,p)-PIR is a q-server PIR with retrievability r and privacy p. A (q,r,p)-PIR has perfect retrievability if $r = 1$ and perfect privacy if $p = 0$.

With our FPA $C_{n,k}^{\lambda}$, we construct a $(\lambda+1, 1, p)$-PIR with perfect retrievability and privacy $p > 0$. The scheme is simple:

- For a message $\boldsymbol{m}$, we put $\boldsymbol{x} = E_{n,k}^{\lambda}(\boldsymbol{m})$ on all $\lambda+1$ servers.
- We retrieve m_i by $L_{n,k}^{\lambda}$ by querying entries from servers in a random order.

The perfect retrievability is guaranteed by Corollary 7. However, in order to retrieve m_i, x_i must be queried from some servers at certain positions $\ell > i$, and we have $rp > 0$. We leave the improvement on the privacy p as our future work.

Acknowledgment

The authors would like to thank Torleiv Kløve for his helpful comments and discussion, which inspire them to construct new FPAs.

References

[1] A. A. Babaev, "Procedures of encoding and decoding of permutations," *Cybern. Syst. Anal.*, vol. 20, no. 6, pp. 861–863, Nov. 1984.

[2] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash Memories*. Norwell, MA: Kluwer, 1999.

[3] J. C. Chang, R. J. Chen, T. Kløve, and S. C. Tsai, "Distance-preserving mappings from binary vectors to permutations," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1054–1059, Apr. 2003.

[4] K. Efremenko, "3-query locally decodable codes of subexponential length," in *Proc. ACM Symp. Theory Comput.*, 2009, pp. 39–44.

[5] S. Huczynska and G. L. Mullen, "Frequency permutation arrays," *J. Combinatorial Des.*, vol. 14, pp. 463–478, 2006.

[6] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 1731–1735.

[7] A. Jiang, M. Schwartz, and J. Bruck, "Error-correcting codes for rank modulation," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 1736–1740.

[8] T. Kløve, "Spheres of permutations under the infinity norm—Permutations with limited displacement," Dept. of Informatics, Univ. Bergen, Reports in Informatics no. 376, 2008.

[9] T. Kløve, "Generating functions for the number of permutations with limited displacement," *Electron. J. Combinatorics*, vol. 16, 2009, #R104.

[10] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Permutation arrays under the Chebyshev distance," *IEEE Trans. Inf. Theory*, vol. 56, pp. 2611–2617, Jun. 2010.

[11] T. T. Lin, S. C. Tsai, and W. G. Tzeng, "Efficient encoding and decoding with permutation arrays," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 211–214.

[12] K. W. Shum, "Permutation coding and MFSK modulation for frequency selective channel," in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 2002, vol. 5, pp. 2063–2066.

[13] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, no. 3, pp. 228–236, Mar. 1965.

[14] T. G. Swart and H. C. Ferreira, "Decoding distance-preserving permutation codes for power-line communications," presented at the IEEE AFRICON, Windhoek, Namibia, Sep. 26–28, 2007.

[15] I. Tamo and M. Schwatz, "Correcting limited-magnitude errors in the rank-modulation scheme," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2551–2560, Jun. 2010.

[16] L. Trevisan, "Some applications of coding theory in computational complexity," *Quaderni di Matematica*, vol. 13, pp. 347–424, 2004.

[17] J. H. van Lint and R. M. Wilson, *A Course in Combinatiorics*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[18] A. J. H. Vinck and J. Häring, "Coding and modulation for power-line communications," presented at the Int. Symp. Power Line Commun., Limerick, Ireland, Apr. 2000.

[19] A. J. H. Vinck, J. Häring, and T. Wadayama, "Coded M-FSK for power line communications," in *Proc. IEEE Int. Symp. Inf. Theory*, 2000, p. 137.

[20] A. J. H. Vinck, "Coded modulation for powerline communications," in *Proc. Int. J. Electron. Commun*, 2000, vol. 54, pp. 45–49.

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

Min-Zheng Shieh received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Taiwan, in 2003 and 2004, respectively. Currently, he is working toward the Ph.D. degree in the computer science program of National Chiao Tung University.

His main research interests include computational complexity, algorithms, coding theory and discrete mathematics.

Shi-Chun Tsai (M'06) received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taiwan, in 1984 and 1988, respectively; and the Ph.D. degree in computer science from the University of Chicago, IL, in 1996.

During 1993–1996, he served as Lecturer in Computer Science Department, University of Chicago. During 1996–2001, he was Associate Professor of Information Management Department, and Computer Science and Information Engineering Department, National Chi Nan University, Taiwan. He has been with the Department of Computer Science, National Chiao Tung University, Taiwan since 2001, and was promoted to full Professor in 2007. His research interests include computational complexity, algorithms, coding theory, and combinatorics.