T-Code: 3-Erasure Longest Lowest-Density MDS Codes

Sheng Lin, Gang Wang, *Member, IEEE,* Douglas S. Stones, Xiaoguang Liu and Jing Liu

***Abstract*—In this paper, we study longest lowest-density MDS codes, a simple kind of multi-erasure array code with optimal redundancy and minimum update penalty. We prove some basic structure properties for longest lowest-density MDS codes. We define a "perfect" property for near-resolvable block designs (NRBs) and establish a bijection between 3-erasure longest lowest-density MDS codes (T-Codes) and perfect NRB$(3k+1,3,2)$s. We present a class of NRB$(3k+1,3,2)$s, and prove that it produces a family of T-Codes. This family is infinite assuming Artin's Conjecture. We also test some other NRBs and find some T-Code instances outside of this family.**

***Index Terms*—3-erasure correcting codes, parity array codes, near-resolvable design, perfect one-factorization.**

I. Introduction

In the last two decades, along with the fast progress of large-scale data storage systems, especially large-scale networked storage systems, multi-erasure coding techniques have attracted increasing attention. An *m-erasure code* for a storage system is a scheme that encodes the content on n data disks into m check disks so that the system is resilient to any m disk failures [1]. Erasure codes have been used for many applications, such as traditional disk arrays, data grids, peer-to-peer applications, digital fountains, etc [1]. Unfortunately, there is no consensus on the best coding technique for general $n, m > 1$.

Reed-Solomon (RS) *codes* [2] are a well-known example of a multi-erasure code. They are also the only known maximum distance separable (MDS) codes for arbitrary n and m. MDS codes have optimal storage efficiency. On the other hand, the computational complexity of using MDS codes poses a significant problem. To relieve this problem, optimized algorithms have been developed for operations over a Galois field [3].

Binary linear codes [4] are linear codes that are inherently XOR-based. They have minimal computational complexity, hence allowing for more efficient encoding and decoding algorithms. Binary linear codes are divided into overlapping parity groups, i.e., each data disk participates in multiple parity groups (in our case m parity groups) in order to recover from m erasures. However, poor storage efficiency is a drawback to binary linear codes.

disk0	disk1	disk2	disk3	disk4	disk5	disk6
P_1	P_2	P_3	P_4	P_5	P_6	$D_{1,6}$
$D_{3,6}$	$D_{1,3}$	$D_{2,4}$	$D_{3,5}$	$D_{4,6}$	$D_{1,4}$	$D_{2,5}$
$D_{4,5}$	$D_{5,6}$	$D_{1,5}$	$D_{2,6}$	$D_{1,2}$	$D_{2,3}$	$D_{3,4}$

Fig. 1. A 7-disk B-Code.

In this paper, we consider *parity array codes*, which divide each disk into *stripe units* (or *strips* [5]) all of the same size. A collection of n stripe units from distinct disks, each having the same offset, is called a *stripe* [5]. Individual stripes are a self-contained m-erasure correcting unit. The disk layout is just the cyclic repetition of a stripe, so we will typically focus only on a single stripe when designing a code. Each stripe unit is further divided into *packets* of a fixed size [5]. Each packet either stores data or parity information, which will be called a *data packet* or *parity packet*, respectively. Each packet is organized into overlapping *parity groups*.

We say that a parity group i *appears* in a packet (data or parity) if that packet has a non-trivial intersection with i. We say that i *appears* in a disk if it appears in some packet in that disk. If i does not appear in a disk, then it is called a *hole* of the disk.

The purpose of parity array codes is to combine the most favorable attributes of RS codes and binary linear codes. Parity array codes possess the XOR-based architecture, while packing the data and parity information into fewer disks.

In Fig. 1 we give an example of a 7-disk parity array code, in which we decompose a given packet into one of the following.

1. Six parity packets P_i in which parity group i appears.
2. Fifteen data packets $D_{i,j}$ in which parity groups i and j appear.

For reasons of cost, performance and simplicity, we focus on m-erasure parity array codes that satisfy the following properties:

1. They are inherently XOR-based, allowing for efficient practical computation.
2. They are MDS codes, minimizing redundancy and therefore minimizing storage space costs.
3. They have minimum update penalty in terms of both the number of disk operations and the computational complexity. Therefore, each data packet should participate in exactly m parity groups.
4. They have a minimum number of packets for given parameters n and m. We will show that an m-erasure code with n disks is composed of either $N = n(n-1)/m$

Manuscript received 15 January 2009; revised 01 August 2009. This work was supported in part by the National High Technology Research and Development Program of China (2008AA01Z401), NSFC of China (60903028), RFDP of China (20070055054), and Science and Technology Development Plan of Tianjin (08JCYBJC13000).

Sheng Lin, Gang Wang, Xiaoguang Liu and Jing Liu are with Nankai-Baidu Joint Lab, College of Information Technical Science, Nankai University, 94 Weijin Road, Tianjin 300071, China (e-mail: shshsh.0510@gmail.com, wgzwp@163.com, liuxg74@yahoo.com.cn, jingliu@nankai.edu.cn).

Douglas S. Stones is with the School of Mathematical Sciences, Monash University, VIC 3800 Australia (e-mail: douglas.stones@sci.monash.edu.au).

Digital Object Identifier 10.1109/JSAC.2010.1002xx.

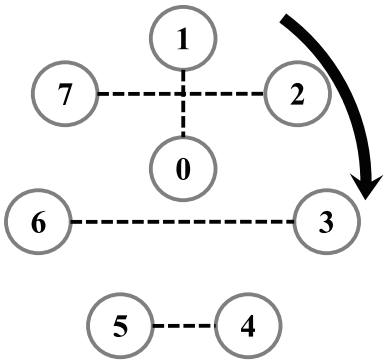

Fig. 2. The starter for GK_8.

or $N = n^2/m$ packets if the previous three properties are satisfied.

We call these properties the *performance properties*. Following the terminology of [6], we call a code lowest-density if it satisfies the first three properties. However, our definition does not confine the codes to be $\mathbb{F}_2$-linear, unlike [6], although this is not an undesirable property. We call a code satisfying all four of the performance properties *longest lowest-density*. Despite the tautology, we will write "lowest-density MDS codes" and "longest lowest-density MDS codes" to emphasize that they are a special type of MDS code.

For example, EVENODD [7], RDP [8] and their multi-erasure extensions [9], [10] satisfy the first two performance properties, X-Code [11] satisfies the first three performance properties and, in the next section, we will show that B-Code [12] satisfies all the four performance properties. Therefore, a B-Code is a 2-erasure longest lowest-density MDS code. Fig. 1 gives an example of a 7-disk B-Code. A *horizontal* code contains only disks that consist entirely of data or entirely of parity information [13]. On the other hand, the disks in a *vertical* code have some data and parity packets stored within a single disk [13]. EVENODD, RDP and their multi-erasure extensions are examples of horizontal codes. B-Code and X-Code are examples of vertical codes.

Some 2-erasure horizontal codes, such as EVENODD and RDP have been generalized to multiple-erasure successfully. However it is difficult to generalize 2-erasure vertical codes such as B-Code. Using combinatorial means, we will construct 3-erasure longest lowest-density MDS codes from B-Codes, which we call T-Codes. We also describe how to construct a family of T-Codes, that is likely to be infinite, using a combinatorial method. This family of T-Code is in fact equivalent to the codes presented in [14] although we discovered it independently of [14]. However, our contribution lies in a new combinatorial construction method which might provide some helpful insights to further research into finding longest lowest-density MDS codes with even larger distances.

Due to the XOR-based architecture of parity array codes, every bit within a packet may be treated independently. Thus, for our purposes, packet size is not important.

01	02	03	04	05	06	07
27	13	24	35	46	57	16
36	47	15	26	37	14	25
45	56	67	17	12	23	34

Fig. 3. The perfect one-factorization GK_8.

II. From B-Codes To T-Codes

A. B-Code Review

An n-disk B-Code, for odd n, is identified with an $(n-1)/2 \times n$ array, as described in [12]. We give an example of this array for $n = 7$ in Fig. 1. One disk contains only data packets and every other disk contains exactly one parity packet and $(n-1)/2-1$ data packets. Disks that only contain data packets are called *pure data* disks. Deleting the pure data disk in Fig. 1 produces a 2-erasure longest lowest-density MDS code with $n-1$ disks.

Xu et al. [12] established a bijection between B-Codes and so-called perfect one-factorizations (P1Fs) of complete graphs. A *one-factor* of a graph G is a 1-regular spanning subgraph of G, i.e., a subgraph such that each vertex is adjacent to precisely one other vertex. A *one-factorization* of G is a decomposition of G into one-factors. A one-factorization is called *perfect* if the union any pair of distinct one-factors is a Hamiltonian cycle. A complete graph, denoted K_n, is a graph on n vertices such that every pair of vertices are adjacent. Given a P1F of K_{2n}, it is possible to construct a B-Code with $2n-1$ disks. A B-Code with $2n-2$ disk can then be found by deleting the pure data disk, as mentioned above. We will later explicitly describe the construction of a B-Code with $2n-1$ disks from a P1F.

There is a conjecture (see e.g. [15]) that every complete graph K_{2n} on $2n$ vertices admits a P1F. Should this conjecture be true, it would imply that B-Codes exist for any number of disks. Several infinite families and individual constructions of P1Fs of complete graphs K_{2n} have been identified [16], [17]. Therefore, in many cases we can construct B-Codes from P1Fs.

We will now describe a simple construction of a P1F $\{F_1, F_2, \ldots, F_7\}$ of K_8 with vertices labeled $0, 1, \ldots, 7$. We choose the one-factor $F_1 = \{01, 27, 36, 45\}$, which we will call the *starter*. We generate the remaining one-factors by the cyclic automorphism $\alpha = (0)(1234567)$, i.e., we define $F_{\alpha^k(1)} = \{\alpha^k(i)\alpha^k(j) : ij \in F_1\}$ for all $1 \leq k \leq 6$. Fig. 2 depicts the starter in this construction; the remaining one-factors are formed by rotating Fig. 2 about the vertex 0. The one-factorization formed in this way is tabulated in Fig. 3, with each column identifying a one-factor. It is straightforward to verify that this one-factorization is indeed a P1F. In fact, this method produces a P1F of K_{2n} if and only if $2n-1$ is prime. This family of P1Fs is named GK_{2n} [17].

We note that GK_{2n} could be tabulated in numerous ways. For example, in Fig. 3 if (a) we permute the columns or (b) we permute the cells within a column, we will still have a table that describes the perfect one-factorization GK_{2n}. However, the choice of table will be unimportant for our purposes.

Given a P1F of K_{2n}, we can construct a B-Code with $2n-1$ disks in the following way. Each one-factor F of the P1F corresponds to a unique disk d. To construct d from F, we

P_1	P_2	P_3	P_4	P_5	P_6	~~$\emptyset\overline{7}$~~
~~$2\overline{7}$~~	$D_{1,3}$	$D_{2,4}$	$D_{3,5}$	$D_{4,6}$	~~$5\overline{7}$~~	$D_{1,6}$
$D_{3,6}$	~~$4\overline{7}$~~	$D_{1,5}$	$D_{2,6}$	~~$3\overline{7}$~~	$D_{1,4}$	$D_{2,5}$
$D_{4,5}$	$D_{5,6}$	~~$6\overline{7}$~~	~~$1\overline{7}$~~	$D_{1,2}$	$D_{2,3}$	$D_{3,4}$

Fig. 4. The process of constructing a B-Code from GK_8.

delete from F the edge that has $2n-1$ as an endpoint. We then replace the edge $0j \in F$ (if we haven't deleted it), by the parity packet P_j. Finally, the remaining edges $ij \in F$ are replaced by the data packet $D_{i,j}$. This process is illustrated in Fig. 4 and gives rise to the B-Code in Fig. 1. Xu et al. [12] have shown that the codes produced in this way are 2-erasure.

B. Structure Properties of Longest Lowest-Density MDS Codes

Xu et al. [12] proved that B-Codes satisfy the first three performance properties. We will now show that they also satisfy the last performance property. We will also show that a B-Code is composed of either $N = n(n-1)/m$ or $N = n^2/m$ packets. Upon inspection of Fig. 4, we can see that the layout of this particular B-Code satisfies:

I. There is at most one pure data disk.
II. The packets from the same disk do not share parity groups.

In general, we call these properties the *structure properties*. We will show that not only B-Codes, but also all longest lowest-density MDS codes satisfy these properties.

Theorem 1: Longest lowest-density MDS codes satisfy structure property II.

Proof: Suppose that an m-erasure longest lowest-density MDS code $\mathcal{C}$ is composed of n disks. Let p denote the number of parity groups. Let N denote the total number of packets, including both data packets and parity packets, over all stripe units in every disk. Let $k = N/n$, i.e., k is the total number of packets in any one disk. Since $\mathcal{C}$ is a MDS code, the redundancy is

$$R = p/N = m/n \tag{1}$$

and so $k = p/m$.

From $\mathcal{C}$ we construct a $p \times N$ $(0,1)$-matrix H called the *parity check matrix* [4]. Each column of H corresponds to a packet in $\mathcal{C}$ and each row corresponds to a parity group. A cell (i,j) contains 1 if that parity group appears in the corresponding packet, otherwise it contains 0. We partition H into n disjoint $p \times k$ submatrices, $\{M_i\}_{1 \le i \le n}$ such that each submatrix M_i is formed by the k columns of H corresponding to the packets within the i-th disk. Moreover, if $S \subseteq \{1, 2, \ldots, n\}$, we let H_S be the submatrix of H formed by the columns corresponding to the packets within the i-th disk for all $i \in S$. A codeword X is similarly partitioned into n disjoint segments $Z = \{X_i\}_{1 \le i \le n}$, such that each segment is formed by the k elements (bits) of the packets within the i-th disk. Again, if $S \subseteq \{1, 2, \ldots, n\}$, we similarly define X_S to be the concatenation of the segments X_i with $i \in S$. Codewords X satify

$$HX = \vec{0} \tag{2}$$

where $\vec{0}$ is the $p \times 1$ zero vector. The process of decoding is equivalent to solving (2). We observe the following four equivalent statements [4].

	P_1	$D_{3,6}$	$D_{4,5}$	P_2	$D_{1,3}$	$D_{5,6}$
1	1	0	0	0	1	0
2	0	0	0	1	0	0
3	0	1	0	0	1	0
4	0	0	1	0	0	0
5	0	0	1	0	0	1
6	0	1	0	0	0	1

Fig. 5. $H_{\{0,1\}}$ for the 7-disk B-Code in Fig. 1.

1) $\mathcal{C}$ is an m-erasure correcting code.
2) If we know the value of every segment $\{X_i\}_{i \in S}$ for some $S \subseteq \{1, 2, \ldots, n\}$ with $|S| \ge n - m$, i.e., if no more than m erasures occur, then it is possible to find a unique solution to the system of Equations (2).
3) For any subset $S \subseteq \{1, 2, \ldots, n\}$ with $|S| = m$, the columns of the set of matrices $\{M_i\}_{i \in S}$ are linearly independent over GF(2). Equivalently, any non-empty XOR sum of columns of H_S for any $S \subseteq \{1, 2, \ldots, n\}$ is non-zero.
4) For any $S \subseteq \{1, 2, \ldots, n\}$ with $|S| = m$, the matrix H_S has full rank.

Fig. 5 displays the submatrix $H_{\{0,1\}}$, corresponding to disk0 and disk1 in the B-Code shown in Fig. 1.

Suppose that a disk d is composed of a_d data packets and b_d parity packets. Let x_d be the total number of parity groups that appear in d. Then $x_d \le ma_d + b_d \le m(p/m) = p$. There are $p = mk$ parity groups and k packets in each disk. Therefore, we have $x_d = p$ if and only if $b_d = 0$, $a_d = k = p/m$ and each packet within the disk is disjoint.

If there is a parity group that does not appear in d, which occurs if $b > 0$, then the submatrix M_i contains a row of zeroes. For example, the parity group 2 does not appear in disk0 in the B-Code shown in Fig. 1, thus the second row in its submatrix M_0 is a row of zeroes. Recall that, if a parity group does not appear in a disk, we call it a hole. We define the total number of holes in $\mathcal{C}$ is the sum of the number of holes of each disk.

Claim: The total number of holes in $\mathcal{C}$ is at most $p(m-1)$. Otherwise, for some parity group i, there are at least m holes, by the pigeonhole principle. Therefore, we can find $S \subseteq \{1, 2, \ldots, n\}$ of cardinality m for which each M_i in Y has i as a hole. However, in this case, row i of H_S consists entirely of zeroes, implying it does not have full rank, contradiction condition 4).

Claim: The total number of holes in $\mathcal{C}$ is at least $p(m-1)$. We know $x_d \le ma_d + b_d = m(a_d + b_d) - b_d(m-1)$. Therefore the total number of holes in $\mathcal{C}$ is $\sum_d (p - x_d) \ge np - mN + p(m-1) = p(m-1)$ by (1).

To review, we showed that there are exactly $p(m-1)$ holes in $\mathcal{C}$. They all arise due to the existence of a parity packet within a disk. Therefore, the packets from the same disk do not share parity groups. □

We will now use Theorem 1 to deduce that longest lowest-density MDS codes also satisfy structure property I.

Corollary 2: Longest lowest-density MDS codes $\mathcal{C}$ satisfy structure property I.

Proof: Theorem 1 implies that $\mathcal{C}$ satisfies structure property II. Therefore if a disk d is a pure data disk, then every

disk0	disk1	disk2	disk3	disk4	disk5	disk6
$D_{1,3,9}$	P_1	P_2	P_3	P_4	P_5	P_6
$D_{2,5,6}$	$D_{2,4,10}$	$D_{3,5,11}$	$D_{4,6,12}$	$D_{6,9,10}$	$D_{1,6,8}$	$D_{2,7,9}$
$D_{4,10,12}$	$D_{3,6,7}$	$D_{4,7,8}$	$D_{5,8,9}$	$D_{1,3,8}$	$D_{7,10,11}$	$D_{8,11,12}$
$D_{7,8,11}$	$D_{8,9,12}$	$D_{1,6,12}$	$D_{1,10,11}$	$D_{2,11,12}$	$D_{2,4,9}$	$D_{3,5,10}$

disk7	disk8	disk9	disk10	disk11	disk12
P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
$D_{3,8,10}$	$D_{4,9,11}$	$D_{5,10,12}$	$D_{2,3,12}$	$D_{1,7,12}$	$D_{1,4,5}$
$D_{4,6,11}$	$D_{5,7,12}$	$D_{1,2,11}$	$D_{1,7,9}$	$D_{2,8,10}$	$D_{3,9,11}$
$D_{1,2,5}$	$D_{2,3,6}$	$D_{3,4,7}$	$D_{4,5,8}$	$D_{5,6,9}$	$D_{6,7,10}$

Fig. 6. A 13-disk T-Code.

P_∞	P_1	P_2	P_3	P_4	P_5	P_6
$D_{1,3,9}$	$D_{2,4,10}$	$D_{3,5,11}$	$D_{4,6,12}$	$D_{6,9,10}$	$D_{1,6,8}$	$D_{2,7,9}$
$D_{2,5,6}$	$D_{3,6,7}$	$D_{4,7,8}$	$D_{5,8,9}$	$D_{1,3,8}$	$D_{7,10,11}$	$D_{8,11,12}$
$D_{4,10,12}$	$D_{8,9,12}$	$D_{1,6,12}$	$D_{1,10,11}$	$D_{2,11,12}$	$D_{2,4,9}$	$D_{3,5,10}$
$D_{7,8,11}$	$D_{5,11,\infty}$	$D_{9,10,\infty}$	$D_{2,7,\infty}$	$D_{5,7,\infty}$	$D_{3,12,\infty}$	$D_{1,4,\infty}$

P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
$D_{3,8,10}$	$D_{4,9,11}$	$D_{5,10,12}$	$D_{2,3,12}$	$D_{1,7,12}$	$D_{1,4,5}$
$D_{4,6,11}$	$D_{5,7,12}$	$D_{1,2,11}$	$D_{1,7,9}$	$D_{2,8,10}$	$D_{3,9,11}$
$D_{1,2,5}$	$D_{2,3,6}$	$D_{3,4,7}$	$D_{4,5,8}$	$D_{5,6,9}$	$D_{6,7,10}$
$D_{9,12,\infty}$	$D_{1,10,\infty}$	$D_{6,8,\infty}$	$D_{6,11,\infty}$	$D_{3,4,\infty}$	$D_{2,8,\infty}$

Fig. 7. T-Code→NRB.

parity group appears in d. Suppose, seeking a contradiction, that there are two pure data disks in $\mathcal{C}$. If we erase both of those disks, then that data cannot be uniquely recovered. Hence the code $\mathcal{C}$ is not an m-erasure correcting code. In fact, $\mathcal{C}$ is not even a 2-erasure correcting code. We conclude that $\mathcal{C}$ contains no more than one pure data disk. □

Since we seek minimum number of packets, each disk contains at most one parity packet. Hence p is either $n-1$ or n. From Equation (1), we can deduce that the total number of packets in $\mathcal{C}$ is either $N = n(n-1)/m$ or $N = n^2/m$.

The result in this section agree with [6]. However, our derivation is based on a combinatorial method, thus not requiring that the codes are $\mathbb{F}_2$-linear. We have identified some structural properties of longest lowest-density MDS codes. Moreover, we can now derive some necessary conditions for the existence of a longest lowest-density MDS code. The remainder of this paper is devoted to constructing an infinite family of T-Codes.

III. T-Codes and NRBs

We call a longest lowest-density MDS code a *T-Code* if it is a 3-erasure code. In this section, following the methodology used in [12], we will construct a bijection between T-Codes and a certain type of so-called perfect near-resolvable block designs. Since each disk contains $k = p/m$ packets and $m = 3$ now, p must be divisible by 3 and therefore $p = 3k$ and $n = p = 3k$ or $n = p+1 = 3k+1$. We will consider the case when $n = 3k+1$.

Fig. 6 displays a 13-disk T-Code. We observe that (a) it satisfies all the structure properties, (b) packets from the same disk do not share parity groups, (c) there is one pure data disk d and (d) every disk other than d contains a parity packet and has precisely two holes. We introduce a new parity group ∞ into the code in the following way. To each disk d that contains a parity packet, we add a new data packet consisting of the parity group ∞ and the two parity groups originally missing from d. To the unique pure data disk, we instead add P_∞. This completes the construction, which we will call the *complement transformation*, denoted $\mathcal{T}$. In Fig. 7 we give the complement transformation $\mathcal{T}$ of the T-Code in Fig. 6.

We see that each pair of distinct parity groups, i and j, simultaneously appear in exactly two data packets in $\mathcal{T}$. Taking the structure properties into account, this system is a so-called near-resolvable design [17] with parameters $(13, 3, 2)$.

A *balanced incomplete block design*, abbreviated $\mathrm{BIBD}(v, m, \lambda)$, is a pair (V, B), where V is a set of cardinality v and B is a collection of subsets of V each of cardinality m such that $|B| = b$ and (a) every element $x \in V$ appears in exactly r blocks and (b) every pair of distinct elements $x, y \in V$ appears simultaneously in λ blocks. Every set in B is called a *block*. It is necessary that the $r = \lambda(v-1)/(m-1)$ and $b = vr/m$, so these parameters are uniquely determined by v, m and λ.

A *near-resolvable (block) design*, abbreviated $\mathrm{NRB}(v, m, \lambda)$, is a balanced incomplete block design $\mathrm{BIBD}(v, m, \lambda)$ with the additional property that B can be partitioned into *near parallel classes*, i.e., there exists a partition Q of B, such that (a) each part $q \in Q$ has cardinality $|q| = m$, (b) each element $x \in V$ appears in at most one part $q \in Q$ and (c) for each $x \in V$ there exists a unique $q \in Q$ with $x \notin Q$.

We can construct an $\mathrm{NRB}(v, 3, 2)$ B from the complement transformation $\mathcal{T}$ of a T-Code $\mathcal{C}$ in the following way. The parity groups form V and each data packet t identifies a block in B corresponding to the parity groups that appear in t. The partition Q is formed by the set of data packets in each disk. The missing element in each $q \in Q$ corresponds to the parity group of the parity packet.

We can reverse this construction by performing the following steps on each near parallel class $q \in Q$ to construct a disk.

1) We convert the missing element into the corresponding parity packet.
2) We convert each block into a data packet – the three elements in the block designate the parity groups which appear in the data packet.

We then pick a parity group e and delete every packet (both data and parity) in which e appears. We call this the *elimination transformation*. This raises a question: is the code constructed from an $\mathrm{NRB}(v, 3, 2)$ necessarily a 3-erasure code? Actually, it turns out not to be the case, therefore we will need to introduce a stronger property.

Let V be a set. Suppose G is a collection of subsets (blocks) of V, and $S = \cup G$, we say that G is a *cover* of S and that S is the *base set* of G, denoted by $S = \textsc{Base}(G)$. We say $||G|| := |S|$ is the *size* of the cover G. If each element of S appears in an even number of blocks in G, then we call G an *even cover* of S. Define $\mathrm{MC}(G) = \min_{G'}(||G'||)$ where the minimum is taken over all even covers $G' \subseteq G$ of $\textsc{Base}(G)$. We call $\mathrm{MC}(G)$ the *minimum even cover size*. If G does not possess a subset G' that is a even cover of S, then we use the convention $\mathrm{MC}(G) = \infty$.

Given a block g of an $\mathrm{NRB}(3k+1, 3, 2)$, it is contained in a unique near-parallel class $q \in Q$. If $x \in V$ and $x \notin \cup q$ then $P := q \cup \{\{x\}\}$ is a partition of V into k parts of cardinality 3 (the original blocks of q) and one part of cardinality 1 (i.e. $\{x\}$). We call P the *complement block* of g. We call an $\mathrm{NRB}(3k+1, 3, 2)$ *perfect*, if for any three complement

blocks B_1, B_2 and B_3, we have $\text{MC}(B_1 \cup B_2 \cup B_3) = |V| = 3k+1$. That is, every three complement blocks B_1, B_2 and B_3 in a perfect $\text{NRB}(3k+1,3,2)$ are such that every even cover G' of $B_1 \cup B_2 \cup B_3$ must have size $||G'|| = |V|$. Observe that the union $B_1 \cup B_2$ of any pair of complement blocks of an $\text{NRB}(3k+1,3,2)$ is an even cover of V, thus $\text{MC}(B) \leq |V| = 3k+1$. Therefore, a perfect $\text{NRB}(3k+1,3,2)$ maximizes the minimum even cover size of every $B_1 \cup B_2 \cup B_3$. We will now identify a bijection between T-Codes and perfect $\text{NRB}(3k+1,3,2)$.

We can associate any subset $G \subseteq B$ of an $\text{NRB}(3k+1,3,2)$ with a $(0,1)$-matrix H_G in the following way. Each row corresponds to an element $x \in V$ and each column corresponds to a block $g \in G$. The cell in row $x \in V$ and column $g \in G$ contains 1 if $x \in g$, otherwise it contains 0. The matrix H_G is a submatrix of parity check matrix H of the code produced by the elimination transformation on B.

Lemma 3: $\text{MC}(G) = \infty$ for every $G \subseteq B$ if and only if H_G has full rank.

Proof: If $\text{MC}(G) < \infty$, then there exists a non-empty subset $G' \subseteq G$ that is an even cover. Therefore the XOR sum of the columns of $H_{G'}$ is the zero vector, implying that $H_{G'}$ and hence H_G does not have full rank. Conversely, if H_G does not have full rank, then there exists some cover $G' \subseteq G$, for which the XOR sum of the columns of $H_{G'}$ is the zero vector. Moreover, G' must be an even cover, otherwise for some $x \in \text{BASE}(G')$, the row of $H_{G'}$ corresponding to x contains an odd number of 1 entries, giving a contradiction. □

Lemma 4: Performing complement transformation on a T-Code $\mathcal{C}$ with $n = 3k+1$ disks produces a system $\mathcal{T}$ in which each pair of distinct parity groups, i and j, appears in precisely two pairs of data packets of $\mathcal{T}$.

Proof: Observe that $\mathcal{T}$ contains $n(n-1)/3$ data packets. Precisely three parity groups appear in each data packet. There are $\binom{n}{2} = n(n-1)/2$ pairs of distinct parity groups. Therefore, each pair of distinct parity groups simultaneously appears in two data packets on average. We will now show that each pair of distinct parity groups simultaneously appears in at most two data packets. Suppose, seeking a contradiction, that there exists a pair of distinct symbols i and j that simultaneously appear in three data packets.

Case I: $i = \infty$ or $j = \infty$. Without loss of generality, suppose that $i = \infty$. Since j must be a hole in $\mathcal{C}$ and each hole appears exactly twice in $\mathcal{C}$, this pair (∞, j) appears exactly twice in $\mathcal{T}$.

Case II: $i \neq \infty$ and $j \neq \infty$. If both i and j appear in three data packets, then structure property II implies that these data packets appear in distinct disks in $\mathcal{C}$. Let S index those disks and so H_S is a square $(0,1)$-matrix (since $m = 3$) with identical row i and row j, by structure property II. Therefore, the rows are not linearly independent and H_S does not have full rank, giving a contradiction. □

Theorem 5: There is a bijection between T-Codes with $n = 3k+1$ disks and perfect $\text{NRB}(3k+1,3,2)$.

Proof: We have shown that an $\text{NRB}(3k+1,3,2)$ B can be converted into a code $\mathcal{C}$ satisfying the structure properties by the elimination transformation. By definition, if B is a perfect $\text{NRB}(3k+1,3,2)$, then any three complement blocks B_1, B_2 and B_3 in B have $\text{MC}(B_1 \cup B_2 \cup B_3) = |V| = 3k+1$. During the eliminiation transformation we have deleted the set containing e from each of B_1, B_2 and B_3 to form B_1^*, B_2^* and B_3^*, respectively. Therefore, if G' is an even cover and $G' \subseteq B_1^* \cup B_2^* \cup B_3^*$, then $|\text{BASE}(G')| \leq |V \setminus \{e\}| = |V| - 1$. Moreover, $G' \subseteq B_1 \cup B_2 \cup B_3$, contradicting that B is perfect. Hence $\text{MC}(B_1^* \cup B_2^* \cup B_3^*) = \infty$ and Lemma 3 implies that $\mathcal{C}$ is a T-Code.

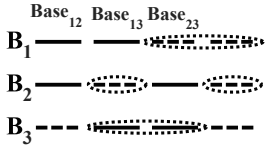

Fig. 8. Illustration of an even cover excluding ∞, marked by ovals.

Conversely, according to Lemma 4, performing complement transformation on a T-Code $\mathcal{C}$ with $3k+1$ disks yields an $\text{NRB}(3k+1,3,2)$ B. If B is not perfect, there must be three complement blocks B_1, B_2 and B_3 with $\text{MC}(B_1 \cup B_2 \cup B_3) < |V| = 3k+1$. So there is an even cover $R \subseteq B_1 \cup B_2 \cup B_3$ of size less than $|V|$. Since R is an even cover, any element that is in some set in R is in exactly two sets in R. If ∞ is not in some set in R, then R is fully contained in $\mathcal{C}$. Thus some columns selected form $\mathcal{C}$'s parity check matrix are linear dependent, implying that $\mathcal{C}$ is not a 3-erasure code, giving a contradiction. Thus ∞ is in exactly two sets in R. Without loss of generality, assume ∞ appears in B_1 and B_2. Let

$$B_1^R = R \cap B_1 \qquad B_2^R = R \cap B_2 \qquad B_3^R = R \cap B_3$$

Since each element is in exactly two sets in R and B_1, B_2 and B_3 are mutually disjoint, we have

$$\text{BASE}(B_1^R) \cap \text{BASE}(B_2^R) \cap \text{BASE}(B_3^R) = \emptyset.$$

Thus

$$\left(B_1 \setminus B_1^R\right) \cup \left(B_2 \setminus B_2^R\right) \cup B_3^R$$

is an even cover excluding ∞. This is depicted in Fig. 8, where $\text{BASE}_{ij} = \text{BASE}(B_i^R) \cap \text{BASE}(B_j^R)$ and, by assumption, BASE_{12} contains ∞. Solid lines denote B_i^R and the even cover excluding ∞ is marked by dotted ovals. This contradicts the assumption that $\mathcal{C}$ is a 3-erasure code. Thus B is perfect. □

In Fig. 6 we gave an example of a T-Code for $n = 13$. In next section, we will show that T-Code exists for many other values of n.

IV. Constructing T-Codes

We will now identify a method of constructing T-Codes from a specific class of $\text{NRB}(v,3,2)$, as constructed in [17]. We will show that this kind of NRB is perfect when both $p = 3k+1$ is a prime and 2 is a primitive root modulo p. Artin's Conjecture [18] asserts that for any a, that is not a square or -1, there exists an infinite number of primes with primitive root a. In particular, Artin's Conjecture implies that there exists infinitely many primes with primitive root 2. However, we additionally require that $p \equiv 1 \pmod 3$. The first few primes of this form are 13, 19, 37, 61, 67, 139, etc.

Since 2 is a primitive root modulo p, the non-zero elements of $\text{GF}(p)$ form a cyclic group $\text{GF}(p)^*$ under multiplication

modulo p generated by 2. Since 2 is a primitive root modulo p, we know that $2^{3k} \equiv 1 \pmod{p}$. We define the near parallel classes by

$$\Big\{\{2^i + j, 2^{i+k} + j, 2^{i+2k} + j\} | 0 \le i \le k-1\Big\}_{0 \le j \le p-1}$$

working modulo p. For example, when $p = 13$ and $k = 4$, we construct the near parallel classes

$$\begin{aligned} j = 0 \quad & \big\{\{1,3,9\},\{2,5,6\},\{4,10,12\},\{7,8,11\}\big\} \\ j = 1 \quad & \big\{\{2,4,10\},\{3,6,7\},\{0,5,11\},\{8,9,12\}\big\} \\ j = 2 \quad & \big\{\{3,5,11\},\{4,7,8\},\{1,6,12\},\{0,9,10\}\big\} \\ j = 3 \quad & \big\{\{4,6,12\},\{5,8,9\},\{0,2,7\},\{1,10,11\}\big\} \end{aligned}$$

and so on, up to $j = p-1$. The $\mathrm{NRB}(p,3,2)$, denoted $\mathcal{B}_p$, is the union of the parallel classes, which we will call a 2-NRB, since it is generated by a primitive root 2. We know that $\mathcal{B}_p$ can be converted into a code $\mathcal{C}$ by the elimination transformation. We will now show that $\mathcal{C}$, constructed from $\mathcal{B}_p$, is a T-Code, so Theorem 5 verifies that $\mathcal{B}_p$ is indeed an perfect $\mathrm{NRB}(v,3,2)$. Note that, in the elimination transformation, the blocks containing 0 are deleted. It is straightforward to show that $\mathcal{C}$ satisfies the performance properties. Therefore, it is sufficient to prove that $\mathcal{C}$ is a 3-erasure code.

For this, we will identify $\mathcal{C}$ with a polynomial of the quotient ring R of polynomials in $\mathrm{GF}(2)[x]$ modulo the ideal generated by

$$f_p(x) := 1 + x + x^2 + \cdots + x^{p-1}.$$

Observe that

$$x^p - 1 = (x-1)(1 + x + x^2 + \cdots + x^{p-1}) = (x-1)f_p(x).$$

Therefore $x^p = 1$ in R. An arbitrary polynomial $g \in R$ can be written in the form

$$g(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{p-1}x^{p-1}, \tag{3}$$

where $a_0, a_1, \ldots, a_{p-1} \in \mathrm{GF}(2)$. Observe that $x^t g(x)$, for any integer t, is the polynomial formed by cyclicly shifting coefficients $a_0, a_1, \ldots, a_{p-1}$ in (3) of $g(x)$ to the right by t steps.

For any $0 \le i \le k-1$, let

$$g_i(x) = x^{2^i} + x^{2^{i+k}} + x^{2^{i+2k}}.$$

Let $\mathcal{V} = \mathrm{GF}(2)^{k+1}$ be the direct product of $k+1$ GF(2)'s. For arbitrary $v \in \mathcal{V}$ we can write $v = (a_0, a_1, \ldots, a_k)$ for some $a_0, a_1, \ldots, a_k \in \mathrm{GF}(2)$. We define $h_v(x) \in R$ to be the polynomial

$$h_v(x) = a_0 + \sum_{i=0}^{k-1} a_{i+1} g_i(x)$$

for all $v \in \mathcal{V}$.

For all $0 \le j \le p-1$, we let B_j denote the parallel class in which j is not an element of any of the blocks. We assign an total order to the blocks in each near parallel class $B_j \in \mathcal{B}_p$ so that we can consider the i-th block of B_j, where $0 \le i \le k-1$. Given a near parallel class $B_j \in \mathcal{B}_p$ which produces the j-th disk of $\mathcal{C}$, we construct a vector $v_j = (a_0, a_1, \ldots, a_k) \in \mathcal{V}$ by making the following assignments.

- a_0 is the value of the parity packet P_j if P_j appears in disk j and $a_0 = 0$ otherwise.
- a_{i+1} is the value of the packet defined by the i-th block of B_j if the data packet appears in disk j and $a_{i+1} = 0$ otherwise, for each $0 \le i \le k-1$.

Hence the j-th disk of $\mathcal{C}$ gives rise to the polynomial $x^j h_{v_j}(x) \in R$. Observe that x^i corresponds to the i-th parity group of $\mathcal{C}$, and the coefficient of x^i is the value of the packet in which the i-th parity group appears. Therefore adding the coefficients of x^i from different disks over $\mathrm{GF}(2)$ is equivalent to taking the XOR sum of the packets that intersect with the i-th parity group. So we have

$$\sum_{j=0}^{p-1} x^j h_{v_j}(x) = 0. \tag{4}$$

Lemma 6: For all $0 \le i \le k-1$, $g_i(x) = g_i(x)^{2^k} = g_i(x)^{2^{2k}}$.

Proof: To begin, observe that, for any a, b, c,

$$\begin{aligned} (x^a + x^b + x^c)^2 &= x^{2a} + x^{2b} + x^{2c} + 2(x^{a+b} + x^{a+c} + x^{b+c}) \\ &= x^{2a} + x^{2b} + x^{2c}, \end{aligned}$$

since coefficients are in $\mathrm{GF}(2)$. By repated application we find that

$$(x^a + x^b + x^c)^{2^k} = x^{2^k a} + x^{2^k b} + x^{2^k c}.$$

In our case $a = 2^i$, $b = 2^{i+k}$ and $c = 2^{i+2k}$. Hence

$$\begin{aligned} g_i(x)^{2^k} &= x^{2^k 2^i} + x^{2^k 2^{i+k}} + x^{2^k 2^{i+2k}} \\ &= x^{2^{i+k}} + x^{2^{i+2k}} + \left(x^{2^{3k}}\right)^{2^i} \\ &= x^{2^{i+k}} + x^{2^{i+2k}} + x^{2^i} \\ &= g_i(x) \end{aligned}$$

since $2^{3k} \equiv 1 \pmod{p}$ and $x^p = 1$ in R. The second equality follows from the first. □

Theorem 7: Equation (4) admits a solution whichever three disks of $\mathcal{C}$ fail.

Proof: Without loss of generality, suppose that the 0-th disk, the a-th disk and the b-th disk fail, $1 \le a < b \le p-1$. Let $H(x)$ be the sum of the polynomials corresponding to the surviving disks. Equation (4) can be rewritten as

$$H(x) = h_{v_0}(x) + x^a h_{v_a}(x) + x^b h_{v_b}(x). \tag{5}$$

Lemma 6 implies that

$$\begin{cases} H(x) & = h_{v_0}(x) + x^a h_{v_a}(x) + x^b h_{v_b}(x) \\ H(x)^{2^k} & = h_{v_0}(x) + x^{a2^k} h_{v_a}(x) + x^{b2^k} h_{v_b}(x) \\ H(x)^{2^{2k}} & = h_{v_0}(x) + x^{a2^{2k}} h_{v_a}(x) + x^{b2^{2k}} h_{v_b}(x). \end{cases} \tag{6}$$

Let

$$\nu = \begin{pmatrix} 1 & x^a & x^b \\ 1 & x^{a2^k} & x^{b2^k} \\ 1 & x^{a2^{2k}} & x^{b2^{2k}} \end{pmatrix}.$$

The system of equations (6) admits a solution if and only if $\det(\nu) \neq 0$.

$$\det(\nu) = x^a x^b \begin{vmatrix} 1 & 1 & 1 \\ 1 & x^{a(2^k-1)} & x^{b(2^k-1)} \\ 1 & x^{a(2^{2k}-1)} & x^{b(2^{2k}-1)} \end{vmatrix}$$

Let $c = a(2^k - 1)$ and $d = b(2^k - 1)$. Since $a \neq b$ and p is a prime, $c \not\equiv d \pmod p$.

$$\begin{aligned}
\det(\nu) &= x^a x^b \begin{vmatrix} 1 & 1 & 1 \\ 1 & x^c & x^d \\ 1 & x^{c(2^k+1)} & x^{d(2^k+1)} \end{vmatrix} \\
&= x^a x^b \begin{vmatrix} 1 & 1 & 1 \\ 0 & x^c - 1 & x^d - 1 \\ 0 & x^{c(2^k+1)} - 1 & x^{d(2^k+1)} - 1 \end{vmatrix} \\
&= x^a x^b \Big(\big(x^c - 1\big)\big((x^d)^{2^k+1} - 1\big) \\
&\qquad - \big(x^d - 1\big)\big((x^c)^{2^k+1} - 1\big)\Big) \\
&= x^a x^b (x^c - 1)(x^d - 1)\Big(\sum_{i=0}^{2^k} x^{di} - \sum_{i=0}^{2^k} x^{ci} \Big)
\end{aligned}$$

Since the two monomials and the two binomials are all reversible [9], it follows that $\det(\nu)$ is identically 0 if and only if

$$\sum_{i=0}^{2^k} x^{di} = \sum_{i=0}^{2^k} x^{ci}. \tag{7}$$

For each $0 \leq s \leq p-1$ define $\ell_s \in R$ to be the polynomial $\ell_s(x) = \sum_{i=0}^{2^k} x^{si}$. By (7), $\det(\nu) = 0$ if and only if $\ell_c = \ell_d$ for some $c \not\equiv d \pmod p$ and $c, d \not\equiv 0 \pmod p$. Suppose $d \equiv c + r \pmod p$ where $r \not\equiv 0 \pmod p$. Then $\ell_c = \ell_{c+r} = \ell_{c+2r} = \ell_{c+3r}$ and so on. Since $r \not\equiv 0 \pmod p$ and p is prime, we eventually find that $\ell_c = \ell_1 = \ell_0 = 1$ since there are an odd number of terms in (7). However, by (3), if $\ell_1 = \ell_0 = 1$, then the number of terms in ℓ_1 is $2^k + 1 \equiv 1 \pmod p$, giving a contradiction.

At first sight, $v_j \mapsto x^j h_{v_j}(x)$ is not an injection. For example, if $v_j = (0, 0, \ldots, 0) \in \mathcal{V}$ and $v'_j = (1, 1, \ldots, 1) \in \mathcal{V}$, then $x^j h_{v_j}(x) + x^j h_{v'_j}(x) = 1 + x + x^2 + \cdots + x^{p-1} = f_p(x)$. However, in the process of constructing the j-th disk d of $\mathcal{C}$ from $\mathcal{B}_p$, if $j \neq 0$ then we have deleted some block $(0, u, v)$ and inserted the j-th parity packet P_j. Hence, the coefficients of x^{u+j} and x^{v+j} in $x^j h_{v_j}$ are both zero, which implies that x^{u+j} and x^{v+j} should not appear. For $j = 0$, while no block has been deleted, there is no parity packet in this disk, so x^0 should not appear. Therefore for the $v \in \mathcal{V}$ that we are interested in, this situation cannot arise. □

Theorem 7 shows that the codes $\mathcal{C}$, formed from $\mathcal{B}_p$ using the elimination transformation, are indeed 3-erasure codes, i.e. they are T-Codes.

To review, in this section we presented a combinatorial construction of a family of T-Codes that is likely to be infinite. However, the methodology presented is unable to resolve every case of $n \equiv 1 \pmod 3$. For other cases, inspecting all possible layouts, even with the aid of a computer, is not feasible. A more promising way is to check some special layouts that are likely to be T-Codes.

We have proved that 2-NRBs are perfect NRBs. An intuitive idea is to test NRBs based on Galois fields GF(p) that have a primitive root 3, 5, and so on. We have tested all prime of the form $p = 6c + 1$ less than 1000 and the results are tabulated in TABLE I. We were unable to find T-Codes for all primes p of this form, however, we have succeeded in resolving many cases. For 4 disks, there is a simple construction of a T-Code – four-way mirroring. For $p = 7$ we performed an exhaustive search to prove that no T-Code exists with 7 disks. Therefore perfect NRBs and T-Code do not exist for all n of the form $3k + 1$. For $n = 10$, we were unable to exhaustively search every possible layout.

TABLE I
T-CODE SEARCHED BY COMPUTER.

primitive root	prime numbers $p = 6c + 1 < 1000$	
	T-Code found	not found
2	13, 19, 37, 61, 67,139, 163, 181, 211, 349, 373, 379, 421, 523, 541, 547, 613, 619, 661, 709, 787, 829, 853, 859, 877, 883, 907	
3	31, 43, 79, 127, 199, 223, 283, 331, 463, 487, 571, 607, 631, 691, 739, 751, 811, 823	7
5	97, 103, 157, 193, 277, 307, 397, 433, 577, 673, 727, 937, 967	73
7	229, 241, 499, 601, 733, 919, 991, 997	151
11	109, 367, 643, 769	
13	457	
17	313, 439	
19		337
29	409	
43	271	

V. CONCLUSION

In this paper, we studied longest lowest-density MDS codes, a simple class of multi-erasure parity array codes with good performance. We proved some structure properties of this kind of codes. We defined a property, which we called "perfect", for near-resolvable block designs (NRBs) and found a bijection between T-Codes, i.e., 3-erasure longest lowest-density MDS codes, and perfect NRB$(3k + 1, 3, 2)$. We also presented a combinatorial construction of a family of T-Codes that is likely to be infinite. This family of T-Codes is constructed from NRBs based on prime fields that have a primitive root 2. We proved that this kind of NRB is perfect, verifying that the codes we produced are indeed T-Codes. We also tested some other NRBs and found some T-Codes outside of this family. An important avenue for future exploration would be to find more T-Codes, either in families or individual instances, by both theoretic and computational means. It would also be of value to develop efficient encoding and decoding algorithms for these codes.

ACKNOWLEDGMENT

We would like to thank the referees for valuable feedback.

REFERENCES

[1] J. S. Plank, "Erasure Codes for Storage Applications," Tutorial Slides, presented at *the 4th Usenix Conference on File and Storage Technologies, FAST 2005*, San Francisco, CA, USA, December 2005.

[2] ——, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-Like Systems," *Softw., Pract. Exper.*, vol. 27, no. 9, pp. 995–1012, 1997.

[3] J. S. Plank and L. Xu, "Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Network Storage Applications," in *Fifth IEEE International Symposium on Network Computing and Applications, NCA 2006*, Cambridge, Massachusetts, USA, July 2006, pp. 173–180.

[4] L. Hellerstein, G. A. Gibson, R. M. Karp, R. H. Katz, and D. A. Patterson, "Coding Techniques for Handling Failures in Large Disk Arrays," *Algorithmica*, vol. 12, no. 2/3, pp. 182–208, 1994.

[5] J. Luo, L. Xu, and J. S. Plank, "An Efficient XOR-Scheduling Algorithm for Erasure Codes Encoding," in *2009 International Conference on Dependable Systems and Networks (DSN 2009)*, Lisbon, Portugal, July 2009.
[6] M. Blaum and R. M. Roth, "On Lowest Density MDS Codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 46–59, 1999.
[7] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures," *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 192–202, 1995.
[8] P. F. Corbett, R. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-Diagonal Parity for Double Disk Failure Correction," in *Proceedings of the FAST '04 Conference on File and Storage Technologies*, San Francisco, California, USA, dec 2004, pp. 1–14.
[9] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 529–542, 1996.
[10] M. Blaum, "A Family of MDS Array Codes with Minimal Number of Encoding Operations," in *2006 IEEE International Symposium on Information Theory*, Washington, USA, July 2006, pp. 2784–2788.
[11] L. Xu and J. Bruck, "X-Code: MDS Array Codes with Optimal Encoding," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 272–276, 1999.
[12] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner, "Low-density MDS Codes and Factors of Complete Graphs," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1817–1836, 1999.
[13] J. L. Hafner, "HoVer Erasure Codes For Disk Arrays," in *2006 International Conference on Dependable Systems and Networks (DSN 2006)*, Philadelphia, Pennsylvania, USA, June 2006, pp. 217–226.
[14] E. Louidor and R. M. Roth, "Lowest Density MDS Codes Over Extension Alphabets," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3186–3197, 2006.
[15] I. M. Wanless, "Perfect Factorisations of Complete Bipartite Graphs and Latin Squares without Proper Subrectangles," *Electron. J. Combin*, vol. 6, no. R9, 1999.
[16] D. Bryant, B. Maenhaut, and I. M. Wanless, "New families of atomic Latin squares and perfect one-factorisations," *J. Comb. Theory Ser. A*, vol. 113, no. 4, pp. 608–624, 2006.
[17] C. J. Colbourn and J. H. Dintz, *Handbook of Combinatorial Designs (Second Edition)*. Boca Raton, Florida, USA: CRC Press, 2007.
[18] Artin's conjecture. [Online]. Available: http://en.wikipedia.org/wiki/Artin's_conjecture_on_primitive_roots

Sheng Lin received the B.Sc. degree in traffic management engineering from Chinese People's Public Security University, Beijing, China, in 1996, and the M.Sc. degree in computer science from Nankai University, Tianjin, China, in 2003. He is currently working toward the Ph.D. degree in computer science at Nankai University. His research interests include erasure-correcting codes and combinational algorithms.

Gang Wang received the B.Sc., the M.Sc. and the Ph.D. degrees in computer science from Nankai University, Tianjin, China, in 1996, 1999 and 2002, respectively. He is currently an associate professor at the College of Information Technical Science, Nankai University. Gang's research interests include storage systems and parallel computing.

Douglas S. Stones received his B.Sc. in mathematics from Monash University, Melbourne, Australia, in 2005. He is currently a Ph.D. candidate in combinatorial mathematics, also at Monash University, under the primary supervision of Ian M. Wanless. In his thesis he finds new congruences satisfied by the number of Latin squares. It employs a range of techniques from combinatorics, number theory and group theory and also gives new results on other important aspects of Latin squares, for example subsquares and autotopisms. He is expected to submit his thesis late 2009.

Xiaoguang Liu received the B.Sc. degree, M.Sc. degree and Ph.D. degree in computer science from Nankai University, Tianjin, China, in 1996, 1999 and 2002 respectively. He is currently an associate professor in computer science at Nankai University, Tianjin, China. His research interests include parallel computing and storage system.

Jing Liu received his M.Sc. degree in computer science from Nankai University in 1982. Since then, he worked at Nankai University. He is currently a professor and Ph.D supervisor in computer science at Nankai University, Tianjin, China. His current research interests include algorithm design and parallel computing.