Combinatorial Constructions of Multi-Erasure-Correcting Codes with Independent Parity Symbols for Storage Systems*

Gang Wang, Sheng Lin, Xiaoguang Liu, Guangjun Xie, Jing Liu
Dept. of Computer, College of Information Technical Science,
Nankai University, 300071, Tianjin, China
wgzwp@163.com

Abstract

In this paper, we present a new class of t-erasure horizontal codes with independent parity symbols based on Column-Hamiltonian Latin Squares (CHLS). We call the codes PIHLatin (Parity Independent Horizontal Latin) codes. We prove the necessary and sufficient condition of the existence of PIHLatin codes for t=2. For t≥3, we prove some necessary conditions of the existence of PIHLatin codes. We also prove the bijection between 2-erasure PIHLatin-like codes and CHLSs and prove the mapping from t-erasure PIHLatin-like codes to t-1 mutually orthogonal CHLSs for t>2. The performance analysis shows that PIHLatin codes are superior to other multi-erasure array codes in flexibility and variety. Moreover, PIHLatin codes are suitable for both traditional disk arrays and distributed storage systems.

1. Introduction

In recent years, storage technology has changed enormously since network technology was added into. The forms of storage applications tend to vary [1], but almost all of applications use erasure codes to provide high reliability and performance. Moreover, capacity of storage devices improves quickly, size of storage systems becomes larger and larger, more and more storage systems are constructed via network, and so on, these factors are unfavorable to reliability, therefore promoted multi-erasure-correcting codes. In this paper, we develop a new class of multi-erasure-correcting codes with high flexibility and good variety.

The rest of the paper is organized as follow. In Section 2, we introduce previous works on storage erasure codes briefly. Knowledge on perfect one-factorizations and Latin squares is introduced in Section 3. In Section 4, we describe the design of PIHLatin codes, and give some existence conclusions and performance analysis. Finally, Section 5 summarizes this study.

2. Related works

An erasure codes is a scheme which encodes n data disks onto m coding disks so that the entire system can be recovered from any $\leq t$ erasures. There are two trivial but efficient schemes - mirroring and parity. But unfortunately, for t, n, $m > 1$, there are no consensual best coding schemes.

We can divide the known multi-erasure codes into several categories. In a category by themselves are the Reed-Solomon codes [2], which are the only known MDS codes for arbitrary t, m and n. The main drawback of RS codes is the requirement on finite field arithmetic. Although optimized algorithms have been developed [3], computational complex is still a serious problem for software implementation.

Second are binary linear codes which are studied in detail by Gibson et al [4]. Linear codes use only XOR operations, multiple erasures resilience is ensured by partitioning data symbols into some overlapped parity groups. Bad redundancy is the inherent defect of linear codes. Recently, researchers consider use LDPC codes - essentially irregular linear codes in storage applications [5, 6]. LDPC codes focus on "average fault tolerance" instead of "threshold fault tolerance", this idea leads to good redundancy.

Another category is so-called array codes, which arrange data units (data symbols) into a 2D array, and organize stripes along several directions. EVENODD [7] (and its generalization [8]) is the first MDS array code, perhaps also the important one - almost all the array codes developed since then can be regarded as

* This paper is partly supported by NSF of China (90612001), Science and Technology Development Plan of Tianjin, (043185111-14), Nankai university R&D innovation fund and ISC.

the varieties of EVENODD code and its generalization, such as X-Code [9], RDP code [10], STAR code [11], and so on. Certainly, they are distinguished at several aspects: fault tolerance; parity directions; horizontal vs. vertical - whether parities are stored separately; parity independent or dependent - whether parity units participate in other parity groups or not. Poor flexibility is a common problem - not suitable for any n and m, generally prime size is needed. Bad variety is another problem - generally only one structure is available for a given size.

B-CODE [12] is an interesting 2-erasure vertical code which is based on perfect one-factorizations (P1Fs) of complete graphs. Because "P1F numbers" are far denser than prime numbers (graph theorists conjecture that every complete graph with an even number of vertices K_{2N} has a P1F [13]) and a P1F produces two B-CODE instances, B-CODE has good flexibility. B-CODE also has good variety because of good variety of P1Fs.

Most of above codes pursue (near) optimization on all the metrics. But some metrics essentially conflict. For example, MDS horizontal codes with independent parity symbols inevitably have not optimal updating complexity [8, 14]. And MDS array codes naturally have large parity group size [14] which induces poor performance in distributed storage systems. WEAVER codes [17] are vertical codes with up to 50% storage efficiency! But just the high redundancy brings about small parity group size and good localization. The construction of WEAVER codes is time consuming.

Aiming at the flexibility and variety weakness of the known multi-erasure codes, we developed a new class of EVENODD-like multi-erasure horizontal codes based on Latin squares. These codes have good flexibility and variety. The "standard codes" are MDS codes which are suitable for traditional disk arrays, and "shortening code" are non-MDS codes which are suitable for distributed storage applications. With the help of research results of Latin squares, the construction of the codes needn't mass computation.

3. P1Fs, Latin squares and erasure codes

Many researchers have mentioned using graphs to represent linear codes [4, 15]. Simply, a 2-erasure linear code can be described by a simple graph - let vertices denote parity units and edges denote data units. Then a 2-erasure array code can be regarded as a partition of a graph, if it is considered as a data layout of a 2-erasure linear code (multiple units (symbols) per disk instead of one per disk). According to this idea, we got the following theorem [15]:

Theorem 1. A data layout is 2-erasure iff the union of any pair of subgraphs of its corresponding graph partition doesn't contain the following two types of structures:

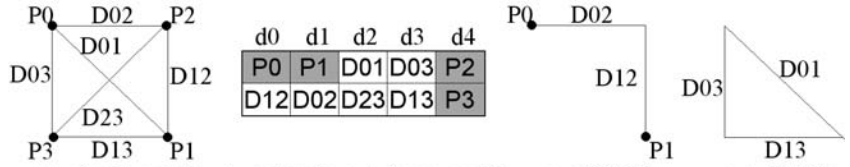
1) A path with its two endpoints. (In simple graph representation, edges and vertices are separate entities, thus containing an edge (a data unit) doesn't mean containing its two endpoints (two parity units)). We call this kind of unrecoverable erasures CPUS (Closed Parity Units Subsets).
2) A cycle. We call this kind of unrecoverable erasures CDUS (Closed Data Units Subsets).

Figure 1 shows the simple graph representation of a full-2 code, a data layout of it, a CPUS (corresponds to disk 0 and disk1) and a CDUS (corresponds to disk 2, disk 3). Almost all 2-erasure array codes can be interpreted by Theorem 1. Based on this theorem, we got some properties of optimal redundancy vertical codes based on full2-codes [16], and found that B-CODE conform to these properties. We also developed a high flexible 2-erasure horizontal codes BG-HEDP based on P1Fs of complete bipartite graphs [14].

We now introduce some basic concepts of P1F and Latin squares [13, 18]. A factor of a graph $G=(V, E)$ is a spanning subgraph of G and a one-factor of G is a one-regular spanning subgraph of G. A factorization of G is a set of factors of G $\{F_1, F_2, \ldots, F_k\}$, which are pairwise edge disjoint - no two have a common edge - whose union is G. A one-factorization of G is a factorization of G consisting of only one-factors. If for any distinct pair F_i, F_j of factors, $F_i \cup F_j$ induces a Hamiltonian cycle, the 1-factorization is called perfect 1-factorization.

For $k \le n$, a *k*n Latin rectangle* is a *k*n* matrix of entries chosen from some set of symbols of cardinality n (generally $Z_n=\{1, 2, \ldots, n\}$ is used), so that no symbol is duplicated within any row or any column. We use $L(k, n)$ for the set of *k*n* Latin rectangles. Elements of $L(n, n)$ are called *Latin squares* of order n. The symbol in row r, column c of a Latin rectangle R is denoted by R_{rc}. A Latin square of order n can be described by a set of n^2 triples of the form (*row*, *column*, *symbol*). For each Latin square there are six *conjugate* squares (itself, transpose, row inverse and their three compositions). Two squares are *isotopic* if one can be obtained from the other by rearranging the rows, rearranging the columns and renaming the symbols. The set of all squares isotopic to a given square forms an *isotopy class*. The closure of an isotopy class under conjugacy yields a *main class*.

Each row r of a Latin rectangle R is the image of some permutation σ_r of Z_n, namely $R_{ri}=\sigma_r(i)$. Moreover, each pair of rows (r; s) defines a permutation by $\sigma_{r,s}=\sigma_r\sigma_s^{-1}$. Naturally $\sigma_{r,s}=\sigma_{s,r}^{-1}$. $\sigma_{r,s}$ describes a shuttle



a. simple graph representation of the full-2 code with 10 disks **b.** a data layout (array code) based on the full-2 code **c.** a CPUS **c.** a CDUS

Figure 1. **Data layout and graph representation**

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix}$$

Figure 2. **A CHLS of order 5**

between row r and row s which begins from any R_{sc}, then go to R_{rc} and then go to $R_{sc'}=R_{rc}$, repeats this process until go back to R_{sc}. If $\sigma_{r,s}$ consists of a single cycle for each pair of rows (r, s) in a Latin square L, we say L is *row-hamiltonian*. Similar concepts can be defined in terms of *column* and *symbol*. In this paper, we concern column-hamiltonian Latin squares, CHLS for short. Figure 2 shows a CHLS of order 5, and $\sigma_{2,3}$ of it. Obviously, it is a Cayley Table of order 5 - C_5.

Combinatorics theorists found that there is a close relationship between Latin squares and P1Fs [13] - there is a CHLS of order n iff $K_{n,n}$ has a P1F; if K_{n+1} has a P1F, then so does $K_{n,n}$. The converse of the second result is not true, thus erasure codes based on P1Fs of $K_{n,n}$ perhaps have better variety than codes based on P1Fs of K_{n+1} (B-CODE vs. BG-HEDP). Given a CHLS L of order n, we can simply transform it into a P1F of $K_{n,n} = (U, V, E)$: first, let $U=\{<0, i> \mid 1\le i\le n\}$ and $V=\{<1, i> \mid 1\le i\le n\}$; second, add edge $(<0, i>, <1, k>)$ to the jth factor of the P1F, for all $(i, j, k) \in L$. Apparently, the reverse method can transform a P1F of $K_{n,n}$ into a CHLS of order n.

4. Construction of horizontal Latin codes with independent parity symbols

4.1. Constructions of 2-erasure horizontal Latin Codes with independent parity symbols

Given a CHLS L of order n, we can construct an EVENODD-like 2-erasure horizontal code with n+2 disks:

1) Delete one row of L, then L becomes a $(n\text{-}1)*n$ Latin rectangle R.
2) Construct the jth data column (data disk) of the code through column j of R, and construct the ith data unit of the jth disk via symbol (i, j, k) of R.
3) Let the ith row of R and the ith unit of the first check disk compose the ith "horizontal parity group".
4) Let the data units which correspond to symbol "i" and the ith unit of the second check disk compose the ith "symbol parity group".

Obviously, symbol parity disk is one unit higher than other disks. We may, of course, eliminate this unbalance like EVENODD does. But the unbalanced structure is not a disaster. Storage space can be used sufficiently by repeating the code (period) in a round robin fashion. On the other hand, unbalance leads to better updating complexity. Formula 1 shows the encoding equation:

$$\begin{aligned} p_{i,1} &= \bigoplus_{j=1}^{n} d_{i,j} && 1 \le i \le n-1 \\ p_{i,2} &= \bigoplus_{(j,k,i)\in L} d_{j,k} && 1 \le i \le n \end{aligned} \qquad (1)$$

where $p_{i,1}$ and $p_{i,2}$ denote the ith horizontal parity unit and the ith symbol parity unit respectively, and $d_{i,j}$ denotes the ith unit of the jth data disk. Figure 3 shows a horizontal Latin code instance. We use $\text{PIHLatin}_{p,t}^{n,h}$ to denote the set of t-erasure horizontal Latin codes which are with independent parity symbols, based on CHLSs of order p, and have n ($n\le p$) data disks and h ($h\le p$-1) data units per data disk.

Data Disks					Check Disks	
11	12	13	14	15	1	1
22	23	24	25	21	2	2
33	34	35	31	32	3	3
44	45	41	42	43	4	4
						5

Figure 3. **A $\text{PIHLatin}_{5,2}^{5,4}$ based on C_5**

Theorem 2. The codes that belong to $\text{PIHLatin}_{n,2}^{n,n-1}$ are 2-erasure codes.

Proof: Suppose that C is an element of $\text{PIHLatin}_{n,2}^{n,n-1}$ and it's based on a CHLS L. Suppose that $F=\{F_1, F_2, \ldots, F_n\}$ is the P1F of $K_{n,n}=(U, V, E)$ converted from L, and F_i corresponds to the ith column of L for $1\le i\le n$.

Thus, the ith data column of C is constructed by deleting the edge $(<0, n>, <1, L_{n,i})>)$ from F_i, and each data unit corresponds to one remaining edge. The ith horizontal parity unit corresponds to the vertex $<0, i>$ and the ith symbol parity unit corresponds to the vertex $<1, i>$. Now we prove that any pair of disks contains no CPUS and CDUS.

1) Two check disks. The pair contains only vertices, so no CPUS or CDUS.

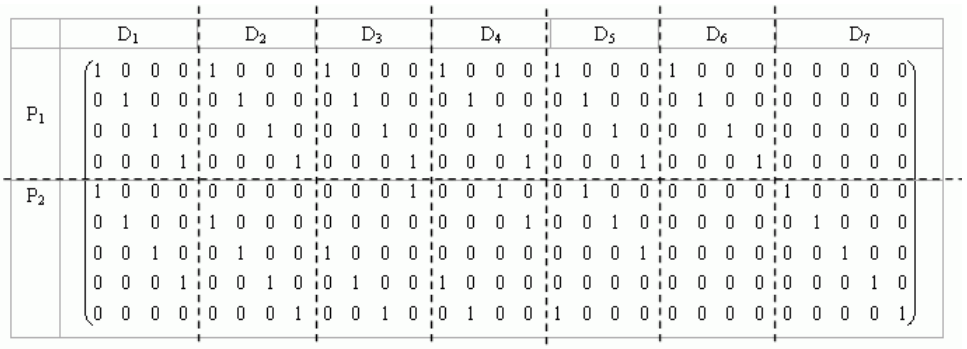


Figure 4. **The parity check matrix of the PIHLatin$_{5,2}^{5,4}$ showed in Figure 3**

2) One data disk and one check disk. The pair contains n-1 nonadjacent edges, each with just one endpoint. If the check disk is symbol parity disk, the union also contains an isolated vertex. However, no CPUS or CDUS.
3) Data disk i and j. The union contains all the edges that belong to $F_i \cup F_j \setminus \{(<0, n>, <1, L_{n,i}>), (<0, n>, <1, L_{n,j}>)\}$. Because $F_i \cup F_j$ induces a Hamiltonian cycle of $K_{n,n}$, the union compose a path of length $2n$ - 2 of $K_{n-1,n}$ - neither a CPUS nor a CDUS. □

Apparently, PIHLatin$_{n,2}^{n,n-1}$ is equivalent to BG-HEDP, but the construction via CHLS is more intuitive, especially for $t > 2$.

Each binary linear code can be described via a $m*(n+m)$ parity check matrix, $H = [P \mid I]$, where I is a $m*m$ matrix, and P is a $m*n$ matrix [4]. Each column of P represents a data disk, each column of I represents a check disk and each row of H represents a parity group. $H(i, j)$=1 means that the ith parity group includes the disk j, and 0 means excluding. It is well known that any parity check matrix H of a t-erasure linear code, has two equivalent properties [4]: (1) H will allow any t' erasures to be corrected and (2) any set of t' columns selected from H will be linearly independent considered as vectors over GF[2]. Note that a set of t' binary vectors is linearly independent over GF[2] iff the vector sum, modulo 2, of those columns, or any nonempty subset of those columns, is not equal to the zero vector.

The parity check matrix H of a PIHLatin$_{p,t}^{n,h}$ code is slightly different. It's a $(p*t-1)*(n*h+p*t-1)$ matrix, and each column represents not a disk, but a unit. We can divide H into $n+t$ horizontal zones D_1-D_{n+t} and t vertical zones P_1-P_t which correspond to $n+t$ disks and t check disks respectively. We use c_{ij} denotes the corresponding column of the ith unit of disk j. Figure 4 shows the parity check matrix of the PIHLatin code showed in Figure 3. The necessary and sufficient condition of t fault tolerance also needs to be modified as "any t horizontal zones are linearly independent".

PIHLatin$_{n,2}^{n,n-1}$ based on C_n is obviously equivalent to EVENODD when n is prime. Namely, EVENODD is a special case of PIHLatin$_{n,2}^{n,n-1}$. In fact, we have the following conclusion.

Theorem 3. There is a bijection between PIHLatin-like (EVENODD-like) horizontal codes and CHLSs.

Proof: Theorem 2 shows the mapping from CHLSs to PIHLatin codes, so now we only need to prove the reverse mapping.

First, any pair of parities on the same disk can't share data units. Otherwise, the pair of parity units and one of their common data units compose a CPUS.

Second, any pair of data units on the same disk can't participate in the same parity group. Otherwise, WLOG, suppose that d_{ij} and d_{kj} participate in the ath parity group of the first check disk and the bth and cth parity groups of the second check disk respectively. Then $d_{i,j}$, $d_{k,j}$, $p_{b,2}$ and $p_{c,2}$ compose a CPUS, whereas they come from two disks! So we can convert the first check disk into "horizontal parity disk" by rearranging the data units on each data disk. We call the second check disk "symbol parity disk".

Third, any pair of data units at the same row can't belong to the same symbol parity group. Otherwise, the two data units participate in two same parity groups (one horizontal and one symbol), then they can't be recovered if lose.

Thus, if we use a triple of form (*row*, *disk*, *symbol parity*) to represent each data unit, then the code can be represented by $(n-1)*n$ triples which can compose a Latin rectangle R. We can transform R into a Latin square L by adding $L_{nj}=Z_n-\{R_{ij} \mid 1\le i\le n-1\}$ for $1\le j\le n$.

We can transform R into a factorization F"of $K_{n-1,n}$ using the method described in section 2. Because the code is 2-erasure, any pair of one-factors of F" induces just a path with length $2n$-2 of $K_{n-1,n}$. It is easy to see that F" become a P1F F of $K_{n,n}$ when the edge (<0, n>, <1, $Z_n-\{R_{ij} \mid 1\le i\le n-1\}$>) is added to the jth factor for $1\le j\le n$. And it is easy to see that L is the corresponding Latin square of F, thus L is a CHLS. □

$$\begin{pmatrix} 1 & 3 & 5 & 2 & 4 \\ 2 & 4 & 1 & 3 & 5 \\ 3 & 5 & 2 & 4 & 1 \\ 4 & 1 & 3 & 5 & 2 \\ 5 & 2 & 4 & 1 & 3 \end{pmatrix}$$

Data Disks					Check Disks		
111	123	135	142	154	1	1	1
222	234	241	253	215	2	2	2
333	345	352	314	321	3	3	3
444	451	413	425	432	4	4	4
						5	5

Figure 5. **A CHLS and a PIHLatin$_{5,3}^{5,4}$ based on it and C_5**

Proof of theorem 2 and theorem 3 is for unbalanced structure, proof for EVENODD-like balanced structure is similar. Theorem 3 tells us that, CHLSs cover all EVENODD-like 2-erasure horizontal codes. Therefore, following combinatorics theorists, maybe we can comprehend this kind of codes completely!

4.2. Constructions of t-erasure horizontal Latin Codes with independent parity symbols for t>2

We generalized construction method of 2-erasure PIHLatin codes to t-erasure cases for t>2. Extra t-2 CHLSs are used, and the ith ($2\le i\le t$-1) CHLS designates the parity groups on i+1th check disk. Figure 5 shows a 3-erasure PIHLatin code. What kind of CHLSs can produce t-erasure PIHLatin codes?

Lemma 4. Any pair of data units of a PIHLatin$_{n,t}^{n,n-1}$ participates at most one common parity group.

Proof: WLOG, suppose that there is a pair of data units $d_{i,j}$, $d_{i',j'}$ of a PIHLatin$_{n,t}^{n,n-1}$ C participates two common parity groups $p_{n+1,k}$ and $p_{n+2,k'}$.

So $s=c_{i,j}+c_{i',j'}$ contains no 1s in P_1 and P_2, thus there are at most t-2 non-zero zones in P_3-P_t and each zone has exactly two 1s. So we can select at most (t-2) columns from D_{n+3}-D_{n+t} so that the sum of them and s is a zero vector, while they come from at most t disks. Thus C is not 2-erasure code, conflict! □

Based on Lemma 4, we got the following theorems.

Theorem 5. If a PIHLatin$_{n,t}^{n,n-1}$ code C is constructed via a Latin squares set S=$\{L_1, \ldots, L_{t-1}\}$, then any pair of elements of S is an *orthogonal* pair, namely, S is a set of *MOLS* (Mutually Orthogonal Latin Squares).

Proof: Two Latin squares L and L' of the same order are orthogonal if $L_{ab}=L_{cd}$ and $L'_{ab}=L'_{cd}$, implies $a=c$ and $b=d$ [19]. A set of Latin squares $L'_1, \ldots, L'_m$ is mutually orthogonal, if for every $1\le i<j\le m$, L'_i and L'_j are orthogonal.

Suppose there is a pair of CHLSs L_i and L_j ($1\le i<j\le t$-1) are not orthogonal. Then, there exist four integers $1\le a, b, c, d\le n$, $(L_i)_{ab}=(L_i)_{cd}$ and $(L_j)_{ab}=(L_j)_{cd}$. Thus, the two units $d_{a,b}$ and $d_{c,d}$ of C participate in at least two common symbol parity groups - $(L_i)_{ab}$ and $(L_j)_{ab}$. According to Lemma 4, C is not t-erasure, conflict! □

Theorem 6. If a PIHLatin$_{n,t}^{n,n-1}$ code C is constructed via a Latin square set S=$\{L_1, \ldots, L_{t-1}\}$, then each subset S' of S of cardinality m ($1\le m\le t$-1) constructs a PIHLatin$_{n,m+1}^{n,n-1}$.

Proof: We use reduction to absurdity. WLOG, suppose that the code C' constructed via S'=$\{L_1, \ldots, L_m\}$ ($1\le m\le t$-1) is not m+1-erasure. Then we can select a column subset COL' of parity matrix H', so that the sum of its elements is a zero vector and its elements come from at most m+1 horizontal zones (disks).

Let H denotes the parity check matrix of C, then it is a vertical extension of H'. Let's consider COL - a column subset of H and the corresponding vertical extension of COL'. Apparently, the sum of its elements has at most t-(m+1) non-zero zones. Then we can select some columns from zones $D_{n+m+2,}, \ldots, D_{n+t}$, so that the sum of them and COL is a zero vector, while these columns come from at most t disks. □

Although the existence of multi-erasure HLatin codes is not solved completely like 2-erasure HLatin codes, we still got a theorem similar to theorem 3.

Theorem 7. Given a PIHLatin$_{n,t}^{n,n-1}$-like (EVENODD generalization like) t-erasure code C, we can construct t-1 mutually orthogonal CHLSs of order n.

Proof: First, any pair of parities that belong to the same check disk can't share data units. Otherwise, a shared data unit and all its parity units come from at most t disks, and they can't be recovered if lose.

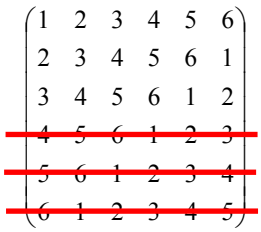
Second, any pair of data units on the same disk can't participate in the same parity group. Otherwise, the pair of data units and all their unshared parity units comes from at most t disks, and the sum of columns which correspond to them is a zero vector. So we can transform the first check disk into horizontal parity disk.

We define t-1 (n-1)*n arrays as:

$$R_l = \{(i,j,k) \mid d_{ij} \text{ participates the } k^{th} \text{ parity units on check disk } l,\ 1 \le i \le n-1, 1 \le j \le n\} \quad (2)$$

for $2\le l\le t$.

Then R_l are all Latin rectangles. First, suppose that two triples (i, j, k) and (i, j', k) belong to R_l simultaneously. This means that $d_{i,j}$ and $d_{i,j'}$ participate the ith horizontal parity and the kth parity of check disk l - conflict with lemma 4! Therefore R_l has no



Data Disks						Check Disks		
11	12	13	14	15	16	1	1	4
22	23	24	25	26	21	2	2	5
33	34	35	36	31	32	3	3	6

Figure 6. **A CHLS and a PIHLatin$_{5,3}^{5,4}$ based on it and C_5**

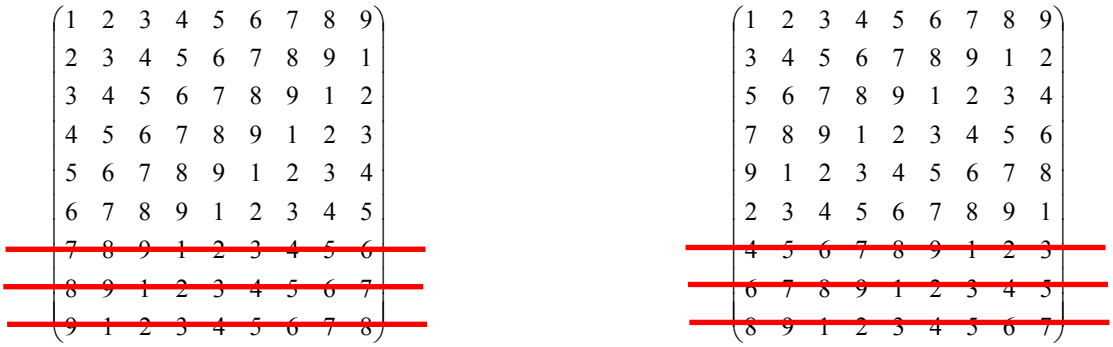


Figure 7. **Two LSs of order 9 which can produce a 3-erasure code**

duplicated symbols within any row. Second, no symbol is duplicated within any column of R_l because any pair of data units on the same disk can't participate in the same parity group. Thus R_l are all Latin rectangles.

Then, Latin squares L_l can be constructed by adding a row to each R_l. According to theorem 2 and theorem 3, if L_l is not a CHLS, then the code based on it is not 2-erasure code. This conflicts with theorem 6. Thus L_l are all CHLSs.

Finally, according to theorem 5, the set $\{L_2, \ldots, L_t\}$ is a set of mutually orthogonal CHLSs of order n. □

We only got these necessary conditions of existence of PIHLatin$_{n,t}^{n,n-1}$ for t>2 up to now. But for some special cases, the sufficient condition can be concluded easily. For example, PIHLatin$_{n,3}^{n,n-1}$ codes based on C_n and its column reverse are just STAR codes when n is prime, so they are 3-erasure. PIHLatin$_{n,t}^{n,n-1}$ codes based on C_n and its t-2 column-shift transformations are equivalent to EVENODD-generalization, so the existence conclusions of the latter [8] hold for these PIHLatin codes too. Another way to study sufficient condition of PIHLatin codes is enumerating all code instances and checking their fault tolerance. We developed a sparse matrix based algorithm which is much faster than the plain algorithm. Unfortunately, it still too slow when t=3 and n>30. But we still got some valuable results. For example, mutually orthogonal CHLSs is not a sufficient condition - because there are 2 mutually orthogonal CHLSs of order 7 induce a non 3-erasure code. In addition, Wanless has found all 35 non-isotopic pairs of MOLS of order 9 in which both squares are CHLSs [20]. We tested them and found that none of them can produce 3-erasure code. Because PIHLatin codes constructed via isotopic CHLSs have the same fault tolerance, we have the following corollary.

Corollary 8. PIHLatin$_{9,t}^{9,8}$ doesn't exist for t≥3.

4.3. Extending construction method

We can extend the construction method mentioned above to improve flexibility or decrease group size (to improve performance in distributed systems).

In order to improve flexibility, there is a general method - deleting some data disks of a PIHLatin code. Apparently, the shortened code is still a t-erasure code. This "horizontal shortening" method can be used to solve storage system extension efficiently. Initially, we can construct a horizontal shortened PIHLatin code instead of a "standard" one. When new devices are added, we just regard the disk insertion as reenter of some deleted disks. And new devices are zeroed simply to avoid parity recalculating. Certainly, the upper bound of the system size is fixed.

"Vertical shortening" - deleting some data unit rows of a standard PIHLatin code but reserving all parity units, can shrink parity groups. Apparently, shortened codes are irregular. We can produce more regular structures by "splitting" check disks. This method has another important function - constructing PIHLatin codes via non-hamiltonian Latin squares. It is easy to see from the proof of theorem 1 why we construct PIHLatin codes via Hamiltonian Latin squares. What

we really need is the single cycle of any pair of columns. We break the cycles by deleting one row, and obtain paths which lead to needed fault tolerance. So we can achieve the same object via non-hamiltonian Latin squares by deleting more rows. Figure 6 shows an example, C_6 is not a HLS, but we can construct a PIHLatin$_{6,2}^{6,3}$ by deleting last 3 rows of C_6. Figure 7 shows two LSs of order 9 which can produce a 3-erasure code by deleting last 3 rows of them. Apparently, this method improves flexibility and variety greatly!

4.4. Performance Analysis

Gibson et al present 5 metrics for erasure codes [4]: reliability, check disk overhead, update penalty, group size and extensibility. We compare the performance of PIHLatin and other array codes at these aspects.

EVENODD and RDP are MDS codes, they achieve optimal check disk overhead t/n , where n is the number of data disks and t is hamming fault tolerance. While the check disk overhead of PIHLatin is $(tn\text{-}1)/(n^2+tn-n-1)$, which is very close to the optimal value, especially for large n and t.

PIHLatin codes are superior to other horizontal codes in update penalty. PIHLatin codes achieve optimal update penalty - t, whether "symbol-unit implementation" or "column-unit implementation" is used. But other horizontal codes, such as EVENODD and RDP, must implement each column instead of each symbol as a stripe unit in order to achieve optimal update penalty. Obviously, the former is much flexible.

Per word XORs need to be done during encodes a PIHLatin$_{n,2}^{n,n\text{-}1}$ code is $2-1/n-1/(n-1)$ which is less than EVENODD's $2-1/(n-1)$ and greater than RDP's $2-2/n$. For $t > 2$, the per word encoding cost of PIHLatin$_{n,t}^{n,n\text{-}1}$ is $t+1-1/n-t/(n-1)$ which is the best one in those of all know t-erasure horizontal codes. The decoding performance is similar.

The performance of standard PIHLatin codes is comparable to EVENODD and RDP. The performance of vertically shortened PIHLatin codes is comparable to WEAVER codes. For example, deleting 2 rows of the PIHLatin$_{5,2}^{5,4}$ showed in figure 3, we get a PIHLatin$_{5,2}^{5,2}$, and the average group size decreases from 5.4 to 3.86 which may benefit the performance of distributed storage applications. The reliability is also improved - PIHLatin$_{5,2}^{5,2}$ can recover many 3 erasures. Vertical shortening can also produce PIHLatin codes between EVENODD/RDP and WEAVER codes which have moderate check disk overhead and distributed performance. Namely, PIHLatin codes cover the whole performance spectrum.

The main advantage of PIHLatin is its good flexibility. 2-erasure PIHLatin codes are suitable for all positive odd integers less than 50 and many other positive odd integers. For t>2, the flexibility of t-erasure PIHLatin codes is not lower than that of EVENODD-generalization. Certainly, EVENODD and RDP can achieve good flexibility via horizontal shortening. But this "extended flexibility" will decrease performance. For example, standard RDP has better per word encoding cost than standard PIHLatin essentially. But for 25 data disks, because 26 is not a prime, we must construct a RDP code with 28 data disks, and then delete 3 data disks. But we can construct a standard PIHLatin$_{25,2}^{25,24}$ which has 25 data disks. It is easy to calculate that the per word encoding cost of the shortened RDP code is 1.924286, and that of PIHLatin$_{25,2}^{25,24}$ is 1.918333. The difference is 0.005953 which seems trivial. But for a 10TB disk array, the real gap is 59.53G XORs! This example shows the significance of "native flexibility".

Another advantage of PIHLatin is good variety. Given a system size, most coding schemes produce a unique code structure. WEAVER codes may have several structures for a given size, but their construction need mass computation. However, for a given size, generally there are many CHLSs, thus many PIHLatin codes. For example, there are 37 main classes of CHLSs of order 9. If we use non-hamiltonian Latin squares to construct PIHLatin codes, the variety will be improved further. For example, there are 19,270,853,541 main classes of LS of order 9! Moreover, as the size increases, the variety will increase sharply.

5. Conclusion and future works

In this paper, we have presented PIHLatin codes, a new class of t-erasure parity independent horizontal codes based on column-hamiltonian Latin squares. We have proven that there is a 2-erasure PIHLatin code iff there is a CHLS. For multi-erasure case, we have proven some necessary conditions of the existence of PIHLatin codes. We also proved the bijection between 2-erasure PIHLatin-like codes and CHLSs and proved the mapping from t-erasure PIHLatin-like codes to t-1 mutually orthogonal CHLSs for t≥3. These results indicate that CHLSs cover PIHLatin-like codes, so this kind of codes can be thoroughly studied by studying CHLSs. The analysis shows that Latin codes are superior to other array codes in flexibility and variety.

Moreover, Latin codes are suitable for both traditional disk arrays and distributed storage systems.

In the future, we hope to find the algebraic representation of multi-erasure PIHLatin codes. This is the key to study the sufficient condition of the existence of multi-erasure PIHLatin codes. We also plan to go on studying this problem by algorithmic means. Another research direction is to study performance optimizing by the good variety of PIHLatin codes. Code extension algorithm and reconstruction algorithm are also important problems which need to be studied further.

Acknowledgement

Thanks Dr. Ian. M. Wanless for his kindly help on the knowledge of Latin squares!

References

[1] J. S. Plank, "Erasure Codes for Storage Applications", Tutorial of the 4th Usenix Conference on File and Storage Technologies, San Francisco, CA, Dec, 2005.

[2] J. S. Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems", Software - Practice & Experience, Vol. 27, No.9, Sep, 1997, pp.995-1012.

[3] J. S. Plank and Lihao Xu, "Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Network Storage Applications", In Proceedings of the 5th IEEE International Symposium on Network Computing and Applications, Cambridge, MA, Jul, 2006, pp.173-180.

[4] Lisa Hellerstein, Garth A. Gibson, Richard M. Karp, Randy H. Katz and David A. Patterson, "Coding techniques for handling failures in large disk arrays", Algorithmica, Vol. 12, No. 2/3, Aug,1994, pp.182-208.

[5] M. G. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman and V. Stemann, "Practical Loss-Resilient Codes", In Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, Texas, May, 1997, pp.150-159.

[6] J. S. Plank and M. G. Thomason. "A practical analysis of low-density parity-check erasure codes for wide-area storage applications", In Proceedings of the International Conference on Dependable Systems and Networks, Florence, Italy, Jun, 2004, pp.115–124,.

[7] M. Blaum, J. Brady, J. Bruck, J. Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures", IEEE Trans. on Computers, Vol. 44, No. 2, pp. Feb, 1995, 192-202.

[8] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols", IEEE Trans. on Information Theory, Vol. 42, No. 2, Mar, 1996, pp. 529-542.

[9] L. Xu and J. Bruck, "X-Code: MDS Array Codes with Optimal Encoding", IEEE Trans. on Information Theory, Vol. 45, No. 1, Jan, 1999, pp.272-276.

[10] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong and S. Sankar, "Row-Diagonal Parity for Double Disk Failure Correction", In Proceedings of the 3th USENIX Conference on File and Storage Technologies, San Francisco, CA, USA, Mar, 2004, pp.1-14.

[11] Cheng Huang, Lihao Xu, "STAR: An Efficient Coding Scheme for Correcting Triple Storage Node Failures", In Proceedings of the 4th USENIX Conference on File and Storage Technologies, San Francisco, Dec, 2005, pp.197-210.

[12] L. Xu, V. Bohossian, J. Bruck, and D.G. Wagner, "Low-Density MDS Codes and Factors of Complete Graphs", IEEE Trans. on Information Theory, Vol. 45, No. 6, Sep, 1999, pp.1817-1826.

[13] I. M. Wanless, "Perfect factorisations of complete bipartite graphs and Latin squares without proper subrectangles", Electron. J. Combin, Vol. 6, 1999, R9.

[14] Wang Gang, Dong Sha-sha, Liu Xiao-guang, Lin Sheng, Liu Jing, "Construct double-erasure-correcting Data Layout Using P1F", ACTA ELECTRONICA SINICA, Vol. 34, No. 12A, 2006, pp.2447-2450.

[15] Zhou Jie,Wang Gang, Liu Xiaoguang, Liu Jing, "The Study of Graph Decompositions and Placement of Parity and Data to Tolerate Two Failures in Disk Arrays: Conditions and Existance", Chinese Journal of Computer, Vol. 26, No. 10, Oct, 2003, pp.1379-1386.

[16] WANG Gang, LIU Xiao-guang, DONG Sha-sha, LIU Jing, "Research on Optimal Redundancy double-erasure-correcting Data Layout", Journal of Jilin University (Engineering and Technology Edition), Vol. 37, No. 03, May, 2007, pp.611-615.

[17] J. L. Hafner, "WEAVER Codes: Highly Fault Tolerant Erasure Codes for Storage Systems", In Proceedings of the 4th Usenix Conference on File and Storage Technologies, San Francisco, Dec, 2005, pp.211-224.

[18] I.M. Wanless, "Atomic Latin squares based on cyclotomic orthomorphisms", Electron. J. Combin, Vol. 12, 2005, R22.

[19] Charles. J. Colbourn, Jeffrey H. Dinitz, et al, "Handbook of Combinatorial Designs (Second Edition)" , CRC Press, 2007.

[20] Private communication between I. M. Wanless and Gang Wang, 2007.