

IBM Research Report

Partial-MDS Codes and Their Application to RAID Type of Architectures

Mario Blaum, James Lee Hafner, Steven Hetzler

IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099
USA



Research Division

Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Partial-MDS Codes and their Application to RAID Type of Architectures

Mario Blaum, James Lee Hafner and Steven Hetzler
IBM Almaden Research Center
San Jose, CA 95120

May 1, 2012

Abstract

A family of codes with a natural two-dimensional structure is presented, inspired by an application of RAID type of architectures whose units are solid state drives (SSDs). Arrays of SSDs behave differently to arrays of hard disk drives (HDDs), since hard errors in sectors are common and traditional RAID approaches (like RAID 5 or RAID 6) may be either insufficient or excessive. An efficient solution to this problem is given by the new codes presented, called partial-MDS (PMDS) codes.

Keywords: Error-correcting codes, flash storage devices, solid state drives, RAID architectures, hard errors, MDS codes, array codes, Reed-Solomon codes, Blaum-Roth codes.

1 Introduction

Consider an array of, say, n storage devices. Each storage device contains a (large) number of sectors, each sector protected by an error-correcting code (ECC) dealing with the most common errors in the media. However, it may occur that one or more of the storage devices experiences a catastrophic failure. In that case, data loss will occur if no further protection is implemented. For that reason, the architecture known as Redundant Arrays of Inexpensive Disks (RAID) was proposed [15].

The way RAID architectures work is by assigning one or more devices to parity. For instance, take n sectors in the same location in each device (we call this set of n sectors, a “stripe”): $n - 1$ sectors carry information, while the n th is the XOR of the $n - 1$ information sectors. We repeat this for each stripe of sectors in the array. Such an architecture is called a RAID 4 or a RAID 5 type of architecture. In what follows, we will call it RAID 5, the difference between RAID 5 and RAID 4 consisting on the distribution of the parity sectors,

but we do not address this issue here. A RAID 6 architecture gives protection against two catastrophic failures.

From a coding point of view, the model of failures corresponds to erasures, i.e., errors whose location is known [29][30]. It is preferable to use Maximum Distance Separable (MDS) codes for RAID 6 types of architectures: in order to correct two erasures exactly two parities are needed. There are many choices for MDS codes correcting two or more erasures: we can use Reed-Solomon (RS) codes [31][35], or array codes, like EVENODD [4], RDP [11], X-codes [43], B-codes [42], C-codes [28], Liberation codes [36], and others [5][39].

Architectures like RAID 5 and RAID 6 are efficient when the storage devices are hard disk drives (HDDs). However, when using solid state drives (SSDs) like flash, these types of architectures, by themselves, either are not efficient or they are wasteful. Arrays of SSDs pose new challenges for code design, so we will spend the rest of this section addressing some of them. Different ways to adapt RAID architectures to SSDs are being considered in recent literature. For instance, ways to enhance the performance of RAID 5 are described in [2][21]. See also [16][20], where the internal ECC and a RAID type of architecture communicate. In particular, [16] uses an adaptive method to increase the redundancy when the bit-error rate increases.

Contrary to HDDs, SSDs degrade significantly in time and as a function of the number of writes [33]. As time goes by and the number of writes increases, the likelihood of a hard error in a sector also increases. A hard error occurs mainly when the internal ECC of a sector is exceeded. In general, BCH codes [29][30] are used for the internal ECC of a sector, although many other codes (including LDPC codes with soft decoding) are possible. Moreover, in recent years some remarkable non-traditional approaches for the ECC that exploit the asymmetry of the SSD channel have been developed [3][19][22][23][24][25][26][32]. However, we do not address the internal ECC problem in this paper. The point is, a hard error corresponds to an uncorrectable error in a sector. Normally, the ECC is coupled with a Cyclic Redundancy Code (CRC), which detects the situation when the ECC miscorrects (the ECC has an inherent detection capability that may not be sufficient, hence it often needs to be reinforced by the CRC). There are several ways to implement the CRC, but we do not address them here. We will assume instead that a hard error means that the information in a sector is lost (an erased sector) and that we can detect this situation.

From the discussion above, we see that, contrary to arrays of HDDs, arrays of SSDs present a mixed failure mode: on one hand we have catastrophic SSDs failures, as in the case of HDDs. On the other hand, we also have hard errors, which in general are silent: their existence is unknown until the sectors are accessed. This situation complicates the task of a RAID type of architecture. In effect, assume that a catastrophic SSD failure occurs in a RAID 5 architecture. Each sector of the failed device is reconstructed by XORing the corresponding sectors in each stripe of the surviving devices. However, if there is a stripe that in addition has suffered a hard error, such a stripe has two sectors that have failed. Since we are using RAID 5, we cannot recover from such an event and data loss will occur.

A possible solution to the situation above is using a RAID 6 type of architecture, in which two SSDs are used for parity. Certainly, this architecture allows for the recovery of two

erased sectors in the same stripe. However, such a solution is expensive, since it requires an additional whole device to protect against hard errors. Moreover, two hard errors in a stripe, in addition to the catastrophic device failure, would still cause data loss, and such a scenario may not be unlikely, depending on the statistics of errors. We would like some solution intermediate between RAID 5 and RAID 6 allowing the handling of hard errors without the need of dedicating a whole second SSD to parity, and in addition being able to handle at least two hard errors in the same stripe, a catastrophic failure having occurred.

In order to handle this mixed environment of hard errors with catastrophic failures, we need to take into account the way information is written in SSDs, which is quite different to the way it is done in HDDs. In an SSD, a new write consists of erasing first a number of consecutive sectors and then rewriting all of them. Therefore, the short write operation in arrays of SSDs (like one sector at a time) is not an issue here: each time a new write is done, a group of, say, m sectors in each SSD is erased and then rewritten. So, the parity needs to be recomputed as part of the new write. We can assume that the array consists of $m \times n$ blocks (i.e., each block consists of m stripes), repeated one after the other. Each $m \times n$ block is an independent unit, and we will show how to compute the parity for each block. Also, each new write consists of writing a number of $m \times n$ blocks (this number may be one, depending on the application, the particular SSD used, and other factors). Our goal is to present a family of codes, that we call partial-MDS (PMDS) codes, allowing for the simultaneous correction of catastrophic failures and hard errors.

The paper is organized as follows: in Section 2, we present the theoretical framework as well as the basic definitions. In Section 3, we present our main construction. In Section 4, we study the special case in which the general construction of Section 3 extends RAID 5, and we find the general conditions for such codes to be PMDS. In Section 5, we study specific cases with parameters relevant to applications, each case analyzed in a separate subsection. In Section 6, we present an alternative construction to the one presented in Section 3, we compare the two and we study some relevant special cases of this second construction. In Section 7 we present a third construction for cases extending RAID 5. This third construction is not as powerful as the the previous ones (it cannot handle three erasures in the same stripe) but uses finite fields of smaller size, simplifying the implementation. In Section 8, we compute the probability of data loss when a catastrophic device failure has occurred under different scenarios. We conclude the paper by drawing some conclusions.

Although the results can be extended to finite fields of arbitrary characteristic, for simplicity, we consider only fields of characteristic 2.

2 Partial-MDS codes

Consider an $m \times n$ array, each entry of the array consisting of b symbols (we assume that each of the b symbols is a bit for the sake of the description, but in practice it may be a much larger symbol). Each stripe in the array is protected by r parity entries in such a way that any r erasures in the stripe will be recovered. In other words, each stripe of the array

				P
				P
				P
		P	P	P

Figure 1: A 4×5 array with $r = 1$ and $s = 2$

	H		F				F	
			F				F	
			F	H	H		F	H
			F				F	

Figure 2: 4×5 arrays with a catastrophic failure and two hard errors

constitutes and $[n, n - r, r + 1]$ MDS code. In addition, we will add s extra “global” parities. Those s extra parities may be placed in different ways in the array, but in order to simplify the description we will place them in the last stripe. Being global means that these parities affect all mn entries in the array. For instance, Figure 1 shows a 4×5 array with $r = 1$ and $s = 2$ such that the two extra global parities are placed in the last stripe.

The idea of a partial-MDS code (to be defined formally), is the following: looking at Figure 1, assume that a catastrophic failure occurs (that is, a whole column in the array has failed), and in addition, we have up to two hard errors anywhere in the array. Then we want the code to correct these failures (erasures in coding parlance). The situation is illustrated in Figure 2, where the hard errors are indicated with the letter ‘H’: the two hard errors may occur either in different stripes or in the same stripe.

A natural way of solving this problem is by using an MDS code. In our 4×5 array example, we have a total of 6 parity sectors. So, it is feasible to implement an MDS code on 20 symbols with 6 parity symbols. In other words, a $[20, 14, 7]$ MDS code (like a RS code). The problem with this approach is its complexity. The case of a 4×5 array is given for the purpose of illustration, but more typical values of m in applications are $m = 16$ and even $m = 32$. That would give 18 or 34 parity sectors. Implementing such a code, although feasible, is complex. We want that the code, in normal operation, utilizes its underlying RAID structure based on stripes, like single parity in the case of RAID 5. The extra parities are invoked in rare occasions. So, given this constraint of an horizontal code, we want to establish an optimality criterium for codes, that we will call partial-MDS (PMDS) codes. In the case of the example of RAID 5 plus two global parities, we want the code to correct up to one erasure per stripe, and in addition, two extra erasures anywhere. For example, the code of Figure 1 is PMDS if it can correct any of the situations depicted in Figure 2. Formally,

Definition 2.1 Let \mathcal{C} be a linear $[mn, m(n - r) - s]$ code over a field or ring such that when codewords are taken row-wise as $m \times n$ arrays, each row belongs in an $[n, n - r, r + 1]$ MDS code. Given (s_1, s_2, \dots, s_t) such that each $s_j \geq 1$ and $\sum_{j=1}^t s_j = s$, we say that \mathcal{C} is

$(r; s_1, s_2, \dots, s_t)$ -erasure correcting if, for any $0 \leq i_1 < i_2 < \dots < i_t \leq m - 1$, \mathcal{C} can correct up to $s_j + r$ erasures in each row i_j of an array in \mathcal{C} . We say that \mathcal{C} is an $(r; s)$ partial-MDS (PMDS) code if, for every (s_1, s_2, \dots, s_t) such that each $s_j \geq 1$ and $\sum_{j=1}^t s_j = s$, \mathcal{C} is an $(r; s_1, s_2, \dots, s_t)$ -erasure correcting code.

In the next section we give a general construction of codes by providing their $(mr + s) \times mn$ parity-check matrices. Some of these codes are going to be PMDS. In particular, we will analyze the case $r = 1$ in Section 4 due to its important practical value, since it extends RAID 5.

3 Code Construction

As stated in Section 2, our entries consist of b bits. We will assume that each entry is in a ring. The ring is defined by a polynomial $f(x)$ of degree b , i.e., the product of two elements in the ring (taken as polynomials of degree up to $b - 1$), is the remainder of dividing the product of both elements by $f(x)$ (if $f(x)$ is irreducible, the ring becomes the field $GF(2^b)$ [31]).

Let α be a root of the polynomial $f(x)$ defining the ring. We call the *exponent* of $f(x)$, denoted $e(f(x))$, the exponent of α , i.e., the minimum ℓ , $0 < \ell$, such that $\alpha^\ell = 1$. If $f(x)$ is primitive [31], $e(f(x)) = 2^b - 1$.

A special case that will be important in applications is $f(x) = M_p(x) = 1 + x + \dots + x^{p-1}$, p a prime number. In this case, $e(M_p(x)) = p$ and $f(x)$ may not be irreducible. In fact, it is not difficult to prove that $f(x)$ is irreducible if and only if 2 is primitive in $GF(p)$. So, the polynomials of degree up to $p - 2$ modulo $M_p(x)$ constitute a ring and not generally a field. This ring was used in [7] to construct the Blaum-Roth (BR) codes, and for the rest of the paper, we either assume that $f(x)$ is irreducible or that $f(x) = M_p(x)$, and p will always denote a prime number.

We present next a general construction, and then we illustrate it with some examples.

Construction 3.1 Consider the binary polynomials modulo $f(x)$, where either $f(x)$ is irreducible or $f(x) = M_p(x)$, and let $mn \leq e(f(x))$, where $e(f(x))$ is the exponent of $f(x)$. Let $\mathcal{C}(m, n, r, s; f(x))$ be the code whose $(mr + s) \times mn$ parity-check matrix is

$$\mathcal{H}(m, n, r, s) = \left(\begin{array}{c|c|c|c} H(n, r, 0, 0) & \underline{0}(n, r) & \dots & \underline{0}(n, r) \\ \underline{0}(n, r) & H(n, r, 0, r) & \dots & \underline{0}(n, r) \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}(n, r) & \underline{0}(n, r) & \dots & H(n, r, 0, (m-1)r) \\ \hline & & & H(mn, s, r, 0) \end{array} \right) \quad (1)$$

where, if $f(\alpha) = 0$, $H(n, r, i, j)$ is the $r \times n$ matrix

$$H(n, r, i, j) = \left(\begin{array}{c|c|c|c|c} \alpha^{j2^i} & \alpha^{(j+1)2^i} & \alpha^{(j+2)2^i} & \dots & \alpha^{(j+n-1)2^i} \\ \alpha^{j2^{i+1}} & \alpha^{(j+1)2^{i+1}} & \alpha^{(j+2)2^{i+1}} & \dots & \alpha^{(j+n-1)2^{i+1}} \\ \alpha^{j2^{i+2}} & \alpha^{(j+1)2^{i+2}} & \alpha^{(j+2)2^{i+2}} & \dots & \alpha^{(j+n-1)2^{i+2}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^{j2^{i+r-1}} & \alpha^{(j+1)2^{i+r-1}} & \alpha^{(j+2)2^{i+r-1}} & \dots & \alpha^{(j+n-1)2^{i+r-1}} \end{array} \right) \quad (2)$$

Let us point out that matrices $H(n, r, i, j)$ as given by (2), in which each row is the square of the previous one, were used in [12][14][37] for constructing codes for which the metric is given by the rank, in [6] for constructing codes that can be encoded on columns and decoded on rows, and in [27] for constructing the so called differential MDS codes.

Let us illustrate Construction 3.1 in the next example.

Example 3.1 Consider $m = 3$ and $n = 5$, then,

$$\mathcal{H}(3, 5, 1, 3) = \left(\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} & \alpha^{16} & \alpha^{18} & \alpha^{20} & \alpha^{22} & \alpha^{24} & \alpha^{26} & \alpha^{28} \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} & \alpha^{24} & \alpha^{28} & \alpha^{32} & \alpha^{36} & \alpha^{40} & \alpha^{44} & \alpha^{48} & \alpha^{52} & \alpha^{56} \end{array} \right)$$

$$\mathcal{H}(3, 5, 2, 2) = \left(\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ \hline 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} & \alpha^{16} & \alpha^{18} & \alpha^{20} & \alpha^{22} & \alpha^{24} & \alpha^{26} & \alpha^{28} \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} & \alpha^{24} & \alpha^{28} & \alpha^{32} & \alpha^{36} & \alpha^{40} & \alpha^{44} & \alpha^{48} & \alpha^{52} & \alpha^{56} \end{array} \right)$$

So far we have not proved that Construction 3.1 provides PMDS codes. Actually, this is not true in general. The answer depends on the particular parameters and on the polynomial $f(x)$ defining the ring or field.

We denote by $(a_{i,j})_{\substack{0 \leq i \leq m-1 \\ 0 \leq j \leq n-1}}$ the received entries from a stored array in $\mathcal{C}(m, n, r, s; f(x))$, assuming that the erased ones are equal to 0. The first step to retrieve the erased entries consists of computing the $rm+s$ syndromes. Using the parity-check matrix $\mathcal{H}(m, n, r, s; f(x))$ given by (1), the syndromes are

$$S_{ir} = \bigoplus_{j=0}^{n-1} a_{i,j} \text{ for } 0 \leq i \leq m-1 \quad (3)$$

$$S_{ir+l+1} = \bigoplus_{j=0}^{n-1} \alpha^{(ni+j)2^l} a_{i,j} \text{ for } 0 \leq i \leq m-1, 0 \leq l \leq r-2 \quad (4)$$

$$S_{mr+u} = \bigoplus_{i=0}^{m-1} \bigoplus_{j=0}^{n-1} \alpha^{(ni+j)(2^{r+u-1})} a_{i,j} \text{ for } 0 \leq u \leq s-1 \quad (5)$$

After computing the syndromes, the erasures are recovered by solving a linear system based on the parity-check matrix, provided that such a solution exists. In the next section, we study the case $r=1$ and give necessary and sufficient conditions that determine whether a $\mathcal{C}(m, n, 1, s; f(x))$ code is PMDS.

4 The case $r=1$

In this section, we assume that $r=1$, thus, the parity-check matrix $\mathcal{H}(m, n, 1, s)$ given by (1) can be written as

$$\mathcal{H}(m, n, 1, s) = (\mathcal{H}_0(m, n, 1, s), \mathcal{H}_1(m, n, 1, s), \dots, \mathcal{H}_{m-1}(m, n, 1, s)),$$

where, for $0 \leq j \leq m-1$

$$\mathcal{H}_j(m, n, 1, s) = \begin{pmatrix} \begin{matrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ j \rightarrow 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{matrix} \Bigg\} m \\ \begin{matrix} \alpha^{jn} & \alpha^{jn+1} & \dots & \alpha^{(j+1)n-1} \\ \alpha^{2jn} & \alpha^{2(jn+1)} & \dots & \alpha^{2((j+1)n-1)} \\ \alpha^{4jn} & \alpha^{4(jn+1)} & \dots & \alpha^{4((j+1)n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{2^{s-1}jn} & \alpha^{2^{s-1}(jn+1)} & \dots & \alpha^{2^{s-1}((j+1)n-1)} \end{matrix} \end{pmatrix}.$$

Assume that $\sum_{j=1}^t s_j = s$ for integers $s_j \geq 1$. According to Definition 2.1, we will characterize when $\mathcal{C}(m, n, 1, s; f(x))$ is $(r; s_1, s_2, \dots, s_t)$ -erasure correcting. We need a series of lemmas first.

Lemma 4.1 For $s \geq 1$, code $\mathcal{C}(m, n, 1, s; f(x))$ as given by Construction 3.1 is PMDS if and only if:

1. code $\mathcal{C}(m, n, 1, s - 1; f(x))$ is PMDS, and;
2. for any (s_1, s_2, \dots, s_t) such that $\sum_{j=1}^t s_j = s$, for any $0 \leq i_2 < i_3 < \dots < i_t \leq m - 1$, and for any $1 \leq j \leq t$ and $0 \leq l_{j,0} < l_{j,1} < \dots < l_{j,s_j} \leq n - 1$,

$$\gcd \left(\left(\sum_{u=1}^{s_1} (1 + x^{l_{1,u} - l_{1,0}}) \right) + \left(\sum_{j=2}^t x^{i_j n + l_{j,0} - l_{1,0}} \sum_{u=1}^{s_j} (1 + x^{l_{j,u} - l_{j,0}}) \right), f(x) \right) = 1 \quad (6)$$

Proof: Since $f(x)$ is implicit, let us denote $\mathcal{C}(m, n, 1, s; f(x))$ simply by $\mathcal{C}(m, n, 1, s)$. Consider rows $0 \leq i_1 < i_2 < \dots < i_t \leq m - 1$ such that row i_j has exactly $s_j + 1$ erasures in locations $(i_j, l_{j,0}), (i_j, l_{j,1}), \dots, (i_j, l_{j,s_j})$, for $0 \leq l_{j,0} < l_{j,1} < \dots < l_{j,s_j} \leq n - 1$ and $\sum_{j=1}^t s_j = s$. Assuming the erased entries to be equal to zero and computing the syndromes according to (3) and (5), we obtain

$$\bigoplus_{v=0}^{s_j} a_{i_j, l_{j,v}} = S_{i_j} \text{ for } 1 \leq j \leq t \quad (7)$$

$$\bigoplus_{j=1}^t \bigoplus_{v=0}^{s_j} \alpha^{2^u(i_j n + l_{j,v})} a_{i_j, l_{j,v}} = S_{m+u} \text{ for } 0 \leq u \leq s - 1 \quad (8)$$

The system given by (7) and (8) has a unique solution if and only if the $(t + s) \times (t + s)$ matrix

$$\underline{c} = \left(\underline{c}_1 \mid \underline{c}_2 \mid \dots \mid \underline{c}_t \right)$$

is invertible, where \underline{c}_j is the $(t + s) \times (s_j + 1)$ matrix

$$\underline{c}_j = \left(\begin{array}{cccc} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ j \rightarrow 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \hline \alpha^{i_j n + l_{j,0}} & \alpha^{i_j n + l_{j,1}} & \cdots & \alpha^{i_j n + l_{j,s_j}} \\ \alpha^{2(i_j n + l_{j,0})} & \alpha^{2(i_j n + l_{j,1})} & \cdots & \alpha^{2(i_j n + l_{j,s_j})} \\ \alpha^{4(i_j n + l_{j,0})} & \alpha^{4(i_j n + l_{j,1})} & \cdots & \alpha^{4(i_j n + l_{j,s_j})} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{2^{s-1}(i_j n + l_{j,0})} & \alpha^{2^{s-1}(i_j n + l_{j,1})} & \cdots & \alpha^{2^{s-1}(i_j n + l_{j,s_j})} \end{array} \right)$$

By row operations on \underline{c} , we obtain a new $(t+s) \times (t+s)$ matrix

$$\underline{c}' = \left(\underline{c}'_1 \mid \underline{c}'_2 \mid \cdots \mid \underline{c}'_t \right)$$

where \underline{c}'_j is the $(t+s) \times (s_j+1)$ matrix

$$\underline{c}'_j = \left(\begin{array}{cccc} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ j \rightarrow 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \hline 0 & \alpha^{i_j n + l_{j,0}} \left(1 \oplus \alpha^{l_{j,1} - l_{j,0}} \right) & \cdots & \alpha^{i_j n + l_{j,0}} \left(1 \oplus \alpha^{l_{j,s_j} - l_{j,0}} \right) \\ 0 & \alpha^{2(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2(l_{j,1} - l_{j,0})} \right) & \cdots & \alpha^{2(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2(l_{j,s_j} - l_{j,0})} \right) \\ 0 & \alpha^{4(i_j n + l_{j,0})} \left(1 \oplus \alpha^{4(l_{j,1} - l_{j,0})} \right) & \cdots & \alpha^{4(i_j n + l_{j,0})} \left(1 \oplus \alpha^{4(l_{j,s_j} - l_{j,0})} \right) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \alpha^{2^{s-1}(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2^{s-1}(l_{j,1} - l_{j,0})} \right) & \cdots & \alpha^{2^{s-1}(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2^{s-1}(l_{j,s_j} - l_{j,0})} \right) \end{array} \right)$$

Notice that \underline{c} is invertible if and only if \underline{c}' is invertible, if and only if the $s \times s$ matrix $\underline{c}'' = \left(\underline{c}''_1 \mid \underline{c}''_2 \mid \dots \mid \underline{c}''_t \right)$ is invertible, where

$$\underline{c}''_j = \begin{pmatrix} \alpha^{i_j n + l_{j,0}} \left(1 \oplus \alpha^{l_{j,1} - l_{j,0}} \right) & \dots & \alpha^{i_j n + l_{j,0}} \left(1 \oplus \alpha^{l_{j,s_j} - l_{j,0}} \right) \\ \alpha^{2(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2(l_{j,1} - l_{j,0})} \right) & \dots & \alpha^{2(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2(l_{j,s_j} - l_{j,0})} \right) \\ \alpha^{4(i_j n + l_{j,0})} \left(1 \oplus \alpha^{4(l_{j,1} - l_{j,0})} \right) & \dots & \alpha^{4(i_j n + l_{j,0})} \left(1 \oplus \alpha^{4(l_{j,s_j} - l_{j,0})} \right) \\ \vdots & \ddots & \vdots \\ \alpha^{2^{s-1}(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2^{s-1}(l_{j,1} - l_{j,0})} \right) & \dots & \alpha^{2^{s-1}(i_j n + l_{j,0})} \left(1 \oplus \alpha^{2^{s-1}(l_{j,s_j} - l_{j,0})} \right) \end{pmatrix}$$

Dividing each row u , $0 \leq u \leq s$, by $\alpha^{2^u(i_1 n + l_{1,0})}$, we obtain the $s \times s$ matrix

$$\hat{c} = \left(\hat{c}_1 \mid \hat{c}_2 \mid \dots \mid \hat{c}_t \right),$$

where

$$\hat{c}_1 = \begin{pmatrix} 1 \oplus \alpha^{l_{1,1} - l_{1,0}} & \dots & 1 \oplus \alpha^{l_{1,s_1} - l_{1,0}} \\ 1 \oplus \alpha^{2(l_{1,1} - l_{1,0})} & \dots & 1 \oplus \alpha^{2(l_{1,s_1} - l_{1,0})} \\ 1 \oplus \alpha^{4(l_{1,1} - l_{1,0})} & \dots & 1 \oplus \alpha^{4(l_{1,s_1} - l_{1,0})} \\ \vdots & \ddots & \vdots \\ 1 \oplus \alpha^{2^{s-1}(l_{1,1} - l_{1,0})} & \dots & 1 \oplus \alpha^{2^{s-1}(l_{1,s_1} - l_{1,0})} \end{pmatrix}$$

and, for $2 \leq j \leq t$, making $i_{j \leftarrow} i_j - i_1$, we have

$$\hat{c}_j = \begin{pmatrix} \alpha^{i_j n + l_{j,0} - l_{1,0}} \left(1 \oplus \alpha^{l_{j,1} - l_{j,0}} \right) & \dots & \alpha^{i_j n + l_{j,0} - l_{1,0}} \left(1 \oplus \alpha^{l_{j,s_j} - l_{j,0}} \right) \\ \alpha^{2(i_j n + l_{j,0} - l_{1,0})} \left(1 \oplus \alpha^{2(l_{j,1} - l_{j,0})} \right) & \dots & \alpha^{2(i_j n + l_{j,0} - l_{1,0})} \left(1 \oplus \alpha^{2(l_{j,s_j} - l_{j,0})} \right) \\ \alpha^{4(i_j n + l_{j,0} - l_{1,0})} \left(1 \oplus \alpha^{4(l_{j,1} - l_{j,0})} \right) & \dots & \alpha^{4(i_j n + l_{j,0} - l_{1,0})} \left(1 \oplus \alpha^{4(l_{j,s_j} - l_{j,0})} \right) \\ \vdots & \ddots & \vdots \\ \alpha^{2^{s-1}(i_j n + l_{j,0} - l_{1,0})} \left(1 \oplus \alpha^{2^{s-1}(l_{j,1} - l_{j,0})} \right) & \dots & \alpha^{2^{s-1}(i_j n + l_{j,0} - l_{1,0})} \left(1 \oplus \alpha^{2^{s-1}(l_{j,s_j} - l_{j,0})} \right) \end{pmatrix}$$

Therefore, the $s \times s$ matrix \hat{c} consists of a first row \underline{w}_0 followed by successive rows \underline{w}_u , where each row is the square of the previous row, i.e., $\underline{w}_u = \underline{w}_0^{2^u}$ for $1 \leq u \leq s-1$. Matrix \hat{c} is invertible if and only if its determinant is invertible. The determinant of a matrix of this type is known [6]: it is the product of the XOR of all possible subsets of elements of the first row. For example, if we have a matrix

$$\begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_1^2 & \gamma_2^2 & \gamma_3^2 \\ \gamma_1^4 & \gamma_2^4 & \gamma_3^4 \end{pmatrix},$$

then its determinant is $\gamma_1\gamma_2\gamma_3(\gamma_1 \oplus \gamma_2)(\gamma_1 \oplus \gamma_3)(\gamma_2 \oplus \gamma_3)(\gamma_1 \oplus \gamma_2 \oplus \gamma_3)$. This result is proven similarly to the one of Vandermonde matrices. For the sake of completeness, we prove it in the Appendix.

The first row of matrix \hat{c} is given by

$$\underline{w}_0 = (\underline{w}_{0,1}, \underline{w}_{0,2}, \dots, \underline{w}_{0,t}),$$

where

$$\underline{w}_{0,1} = (1 \oplus \alpha^{l_{1,1}-l_{1,0}}, 1 \oplus \alpha^{l_{1,2}-l_{1,0}}, \dots, 1 \oplus \alpha^{l_{1,s_1}-l_{1,0}})$$

and, for $2 \leq j \leq t$,

$$\underline{w}_{0,j} = (\alpha^{i_j n + l_{j,0} - l_{1,0}} (1 \oplus \alpha^{l_{j,1} - l_{j,0}}), \dots, \alpha^{i_j n + l_{j,0} - l_{1,0}} (1 \oplus \alpha^{l_{j,s_j} - l_{j,0}})). \quad (9)$$

Then, code $\mathcal{C}(m, n, 1, s)$ is PMDS if and only if the determinant $\det(\hat{c})$ is invertible, and only if the XOR of any subset of the elements of \underline{w}_0 is invertible. Since \underline{w}_0 has s elements, we may assume that if we XOR a number elements smaller than s , the result is true by induction, so assume that we take the XOR of all the s elements in \underline{w}_0 . Then, code $\mathcal{C}(m, n, 1, s)$ is PMDS if and only if code $\mathcal{C}(m, n, 1, s-1)$ is PMDS and

$$\left(\bigoplus_{u=1}^{s_1} (1 \oplus \alpha^{l_{1,u} - l_{1,0}}) \right) \oplus \left(\bigoplus_{j=2}^t \alpha^{i_j n + l_{j,0} - l_{1,0}} \bigoplus_{u=1}^{s_j} (1 \oplus \alpha^{l_{j,u} - l_{j,0}}) \right) \quad (10)$$

is invertible. But (10) is invertible if and only if (6) holds. \square

Lemma 4.2 Consider a code $\mathcal{C}(m, n, 1, s; f(x))$ and let $s_j \geq 1$ for $1 \leq j \leq t$ such that $\sum_{j=1}^t s_j = s$. For each s_j , if s_j is odd, let $s'_j = s_j$, while if s_j is even, let $s'_j = s_j - 1$ and $s'_t = \sum_{j=1}^t s'_j$. Then, $\mathcal{C}(m, n, 1, s; f(x))$ is $(1; s_1, s_2, \dots, s_t)$ -erasure correcting if and only if $\mathcal{C}(m, n, 1, s'; f(x))$ is $(1; s'_1, s'_2, \dots, s'_t)$ -erasure correcting.

Proof: Consider (10). If s_j is odd,

$$\bigoplus_{u=1}^{s_j} (1 \oplus \alpha^{l_{j,u} - l_{j,0}}) = 1 \oplus \bigoplus_{u=1}^{s_j} \alpha^{l_{j,u} - l_{j,0}}, \quad (11)$$

while if s_j is even,

$$\begin{aligned} \bigoplus_{u=1}^{s_j} (1 \oplus \alpha^{l_{j,u} - l_{j,0}}) &= \bigoplus_{u=1}^{s_j} \alpha^{l_{j,u} - l_{j,0}} \\ &= \alpha^{l_{j,1} - l_{j,0}} \left(1 \oplus \bigoplus_{u=2}^{s_j} \alpha^{l_{j,u} - l_{j,1}} \right) \end{aligned} \quad (12)$$

If s_1 is even, making $s'_1 = s_1 - 1$, according to (12) and (11), (10) becomes

$$\begin{aligned}
& \alpha^{l_{1,1}-l_{1,0}} \left(1 \oplus \bigoplus_{u=2}^{s_1} \alpha^{l_{1,u}-l_{1,1}} \right) \oplus \left(\bigoplus_{j=2}^t \alpha^{i_j n + l_{j,0} - l_{1,0}} \bigoplus_{u=1}^{s_j} \left(1 \oplus \alpha^{l_{j,u} - l_{j,0}} \right) \right) = \\
& \alpha^{l_{1,1}-l_{1,0}} \left(\left(1 \oplus \bigoplus_{u=2}^{s_1} \alpha^{l_{1,u}-l_{1,1}} \right) \oplus \left(\bigoplus_{j=2}^t \alpha^{i_j n + l_{j,0} - l_{1,1}} \bigoplus_{u=1}^{s_j} \left(1 \oplus \alpha^{l_{j,u} - l_{j,0}} \right) \right) \right) = \\
& \alpha^{l_{1,1}-l_{1,0}} \left(\bigoplus_{u=2}^{s_1} \left(1 \oplus \alpha^{l_{1,u}-l_{1,0}} \right) \oplus \left(\bigoplus_{j=2}^t \alpha^{i_j n + l_{j,0} - l_{1,1}} \bigoplus_{u=1}^{s_j} \left(1 \oplus \alpha^{l_{j,u} - l_{j,0}} \right) \right) \right) \quad (13)
\end{aligned}$$

Since $\alpha^{l_{1,1}-l_{1,0}}$ is always invertible, then, by (10) and (13), if s_1 is even, $\mathcal{C}(m, n, 1, s - 1)$ is $(1; s'_1, s_2, \dots, s_t)$ -erasure correcting if and only if $\mathcal{C}(m, n, 1, s)$ is $(1; s_1, s_2, \dots, s_t)$ -erasure correcting. Similarly, if $2 \leq v \leq t$ and s_v is even, according to (12) and (11), (10) becomes

$$\begin{aligned}
& \left(\bigoplus_{u=1}^{s_1} \left(1 \oplus \alpha^{l_{1,u}-l_{1,0}} \right) \right) \oplus \left(\alpha^{i_v n + l_{v,0} - l_{1,0}} \bigoplus_{u=1}^{s_v} \left(1 \oplus \alpha^{l_{v,u} - l_{v,0}} \right) \right) \oplus \\
& \left(\bigoplus_{\substack{j=2 \\ j \neq v}}^t \alpha^{i_j n + l_{j,0} - l_{1,0}} \bigoplus_{u=1}^{s_j} \left(1 \oplus \alpha^{l_{j,u} - l_{j,0}} \right) \right) = \\
& \left(\bigoplus_{u=1}^{s_1} \left(1 \oplus \alpha^{l_{1,u}-l_{1,0}} \right) \right) \oplus \left(\alpha^{i_v n + l_{v,1} - l_{1,0}} \left(1 \oplus \bigoplus_{u=2}^{s_v} \alpha^{l_{v,u} - l_{v,1}} \right) \right) \oplus \\
& \left(\bigoplus_{\substack{j=2 \\ j \neq v}}^t \alpha^{i_j n + l_{j,0} - l_{1,0}} \bigoplus_{u=1}^{s_j} \left(1 \oplus \alpha^{l_{j,u} - l_{j,0}} \right) \right) = \\
& \left(\bigoplus_{u=1}^{s_1} \left(1 \oplus \alpha^{l_{1,u}-l_{1,0}} \right) \right) \oplus \left(\alpha^{i_v n + l_{v,1} - l_{1,0}} \bigoplus_{u=2}^{s_v} \left(1 \oplus \alpha^{l_{v,u} - l_{v,1}} \right) \right) \oplus \\
& \left(\bigoplus_{\substack{j=2 \\ j \neq v}}^t \alpha^{i_j n + l_{j,0} - l_{1,0}} \bigoplus_{u=1}^{s_j} \left(1 \oplus \alpha^{l_{j,u} - l_{j,0}} \right) \right) \quad (14)
\end{aligned}$$

By (10) and (14), we may claim that if s_v is even for some $2 \leq v \leq t$, then, making $s'_v = s_v - 1$, $\mathcal{C}(m, n, 1, s - 1)$ is $(1; s_1, s_2, \dots, s_{v-1}, s'_v, s_{v+1}, \dots, s_t)$ -erasure correcting if and only if $\mathcal{C}(m, n, 1, s)$ is $(s_1 + 1, s_2 + 1, \dots, s_v + 1, \dots, s_t + 1)$ -erasure correcting, completing the proof. \square

Lemma 4.3 Consider a code $\mathcal{C}(m, n, 1, s; f(x))$ and let $s_j \geq 1$, each s_j an odd number for $1 \leq j \leq t$ such that $\sum_{j=1}^t s_j = s$. Then, $\mathcal{C}(m, n, 1, s; f(x))$ is $(1; s_1, s_2, \dots, s_t)$ -erasure

correcting if and only if, for any $0 \leq i_2 < i_3 < \dots < i_t \leq m - 1$ and for any $0 \leq l_{j,0} < l_{j,1} < \dots < l_{j,s_j} \leq n - 1$ for each $1 \leq j \leq t$,

$$\gcd \left(1 + \sum_{u=1}^{s_1} x^{l_{1,u}-l_{1,0}} + \sum_{j=2}^t x^{i_j n + l_{j,0} - l_{1,0}} \left(1 + \sum_{u=1}^{s_j} x^{l_{j,u}-l_{j,0}} \right), f(x) \right) = 1 \quad (15)$$

Proof: Notice that in this case, (6) becomes (15). \square

The combination of Lemmas 4.1, 4.2 and 4.3 gives the following theorem:

Theorem 4.1 For $s \geq 1$, code $\mathcal{C}(m, n, 1, s; f(x))$ as given by Construction 3.1 is PMDS if and only if:

1. code $\mathcal{C}(m, n, 1, s - 1; f(x))$ is PMDS, and;
2. for every (s_1, s_2, \dots, s_t) such that $\sum_{j=1}^t s_j = s$ and each s_j is odd, for any $0 \leq i_2 < i_3 < \dots < i_t \leq m - 1$, and for any $1 \leq j \leq t$ and $0 \leq l_{j,0} < l_{j,1} < \dots < l_{j,s_j} \leq n - 1$, condition (15) holds.

Theorem 4.1 gives us conditions to check in order to determine if a code $\mathcal{C}(m, n, 1, s; f(x))$ as given by Construction 3.1 is PMDS, but by itself it does not provide us with any family of PMDS codes. Consider the ring of polynomials modulo $M_p(x)$, such that $mn < e(M_p(x)) = p$. There are cases in which $M_p(x)$ is irreducible [7] and the ring becomes a field (equivalently, 2 is primitive in $GF(p)$). Notice that the polynomials in Theorem 4.1 have degree at most $mn - 1 < p - 1 = \deg(f(x))$. Therefore, if $M_p(x)$ is irreducible then all such polynomials are relatively prime with $M_p(x)$ and the code is PMDS. Let us state this fact as a theorem, which provides a family of PMDS codes (it is not known whether the number of irreducible polynomials $M_p(x)$ is infinite):

Theorem 4.2 Consider the code $\mathcal{C}(m, n, 1, s; M_p(x))$ given by Construction 3.1 such that $M_p(x)$ is irreducible (or equivalently, 2 is primitive in $GF(p)$). Then, $\mathcal{C}(m, n, 1, s; M_p(x))$ is PMDS.

So far we have dealt with general values of s . In the next section we examine special cases that are important in applications.

5 Special cases

We examine each case into a separate subsection.

5.1 The case $\mathcal{C}(m, n, 1, 1; f(x))$

Notice that $\mathcal{C}(m, n, 1, 1; f(x))$ is always PMDS, since by Theorem 4.1, we have to check if the binomials of type $1+x^j$ for $1 \leq j \leq n-1$ and $f(x)$ are relatively prime. This is certainly the case when $f(x)$ is irreducible, but also when $M_p(x)$ is reducible [7]. Let us state this result as a lemma:

Lemma 5.1 Code $\mathcal{C}(m, n, 1, 1; f(x))$ is always PMDS.

5.2 The case $\mathcal{C}(m, n, 1, 2; f(x))$

This case is important in applications, in particular, for arrays of SSDs. Since $\mathcal{C}(m, n, 1, 1; f(x))$ is PMDS, Theorem 4.1 gives the following theorem for the case $s=2$:

Theorem 5.1 Code $\mathcal{C}(m, n, 1, 2; f(x))$ is PMDS if and only if, for any $1 < i \leq m-1$, and for any $0 \leq l_{1,0} < l_{1,1} \leq n-1$, $0 \leq l_{2,0} < l_{2,1} \leq n-1$,

$$\gcd\left(1 + x^{l_{1,1}-l_{1,0}} + x^{in+l_{2,0}-l_{1,0}}\left(1 + x^{l_{2,1}-l_{2,0}}\right), f(x)\right) = 1 \quad (16)$$

Given the practical importance of this case, let us examine the decoding (of which the encoding is a special case) in some detail.

Consider a PMDS code $\mathcal{C}(m, n, 1, 2; f(x))$, i.e., it satisfies the conditions of Theorem 5.1. Without loss of generality, assume that we either have three erasures in the same row i_0 , or two pairs of erasures in different rows i_0 and i_1 , where $0 \leq i_0 < i_1 \leq m-1$. Consider first the case in which the three erasures occur in the same row i_0 and in entries j_0 , j_1 and j_2 of row i_0 , $0 \leq j_0 < j_1 < j_2$. Assuming initially that $a_{i_0, j_0} = a_{i_0, j_1} = a_{i_0, j_2} = 0$, using (3) and (5) ((4) is used only for $r > 1$), we compute the syndromes S_{i_0} , S_m and S_{m+1} . Using the parity-check matrix $\mathcal{H}(m, n, 1, 2)$ as given by (1), we have to solve the linear system

$$\begin{array}{rclclcl} a_{i_0, j_0} & \oplus & a_{i_0, j_1} & \oplus & a_{i_0, j_2} & = & S_{i_0} \\ \alpha^{i_0n+j_0} a_{i_0, j_0} & \oplus & \alpha^{i_0n+j_1} a_{i_0, j_1} & \oplus & \alpha^{i_0n+j_2} a_{i_0, j_2} & = & S_m \\ \alpha^{2(i_0n+j_0)} a_{i_0, j_0} & \oplus & \alpha^{2(i_0n+j_1)} a_{i_0, j_1} & \oplus & \alpha^{2(i_0n+j_2)} a_{i_0, j_2} & = & S_{m+1} \end{array}$$

The solution to this system is

$$a_{i_0, j_0} = \frac{\det \begin{pmatrix} S_{i_0} & 1 & 1 \\ S_m & \alpha^{i_0n+j_1} & \alpha^{i_0n+j_2} \\ S_{m+1} & \alpha^{2(i_0n+j_1)} & \alpha^{2(i_0n+j_2)} \end{pmatrix}}{\det \begin{pmatrix} 1 & 1 & 1 \\ \alpha^{i_0n+j_0} & \alpha^{i_0n+j_1} & \alpha^{i_0n+j_2} \\ \alpha^{2(i_0n+j_0)} & \alpha^{2(i_0n+j_1)} & \alpha^{2(i_0n+j_2)} \end{pmatrix}}$$

$$\begin{aligned}
a_{i_0, j_1} &= \frac{\det \begin{pmatrix} 1 & S_{i_0} & 1 \\ \alpha^{i_0 n + j_0} & S_m & \alpha^{i_0 n + j_2} \\ \alpha^{2(i_0 n + j_0)} & S_{m+1} & \alpha^{2(i_0 n + j_2)} \end{pmatrix}}{\det \begin{pmatrix} 1 & 1 & 1 \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & \alpha^{i_0 n + j_2} \\ \alpha^{2(i_0 n + j_0)} & \alpha^{2(i_0 n + j_1)} & \alpha^{2(i_0 n + j_2)} \end{pmatrix}} \\
a_{i_0, j_2} &= \frac{\det \begin{pmatrix} 1 & 1 & S_{i_0} \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & S_m \\ \alpha^{2(i_0 n + j_0)} & \alpha^{2(i_0 n + j_1)} & S_{m+1} \end{pmatrix}}{\det \begin{pmatrix} 1 & 1 & 1 \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & \alpha^{i_0 n + j_2} \\ \alpha^{2(i_0 n + j_0)} & \alpha^{2(i_0 n + j_1)} & \alpha^{2(i_0 n + j_2)} \end{pmatrix}}
\end{aligned}$$

Since matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & \alpha^{i_0 n + j_2} \\ \alpha^{2(i_0 n + j_0)} & \alpha^{2(i_0 n + j_1)} & \alpha^{2(i_0 n + j_2)} \end{pmatrix}$$

is a Vandermonde matrix,

$$\det \begin{pmatrix} 1 & 1 & 1 \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & \alpha^{i_0 n + j_2} \\ \alpha^{2(i_0 n + j_0)} & \alpha^{2(i_0 n + j_1)} & \alpha^{2(i_0 n + j_2)} \end{pmatrix} = \alpha^{4i_0 n + 3j_0 + j_1} (1 \oplus \alpha^{j_1 - j_0}) (1 \oplus \alpha^{j_2 - j_0}) (1 \oplus \alpha^{j_2 - j_1})$$

This determinant is easily inverted in a field, while in the ring of elements modulo $M_p(x)$, the elements $1 \oplus \alpha^{j_1 - j_0}$, $1 \oplus \alpha^{j_2 - j_0}$ and $1 \oplus \alpha^{j_2 - j_1}$ can be efficiently inverted (for details, see [7]).

The encoding is a special case of the decoding. For instance, assume that we place the two global parities in locations $(m-1, n-3)$ and $(m-1, n-2)$, as depicted in Figure 1. After computing the parities $a_{i, n-1}$ for $0 \leq i \leq m-2$ using single parity, we have to compute the parities $a_{m-1, n-3}$, $a_{m-1, n-2}$ and $a_{m-1, n-1}$ using the method above. In particular, the Vandermonde determinant becomes (making $i_0 = m-1$, $j_0 = n-3$, $j_1 = n-2$ and $j_2 = n-1$) $\alpha^{4(m+n)-15} (1 \oplus \alpha) (1 \oplus \alpha^2) (1 \oplus \alpha) = \alpha^{4(m+n)-15} (1 \oplus \alpha^4)$. So, we have to invert only $\alpha^{4(m+n)-15} (1 \oplus \alpha^4)$ for the encoding and some operations may be precalculated, making the encoding very efficient. We omit the details.

We analyze next the case of two pairs of erasures in rows i_0 and i_1 , $0 \leq i_0 < i_1 \leq m-1$, and assume that the erased entries are a_{i_0, j_0} and a_{i_0, j_1} in row i_0 , $0 \leq j_0 < j_1 \leq n-1$, and a_{i_1, ℓ_0} and a_{i_1, ℓ_1} in row i_1 , $0 \leq \ell_0 < \ell_1 \leq n-1$.

Again using the parity-check matrix $\mathcal{H}(m, n, 1, 2)$, we have to solve the linear system of 4 equations with 4 unknowns

$$\begin{array}{rccccccccc}
a_{i_0, j_0} & \oplus & & a_{i_0, j_1} & & & & & & = & S_{i_0} \\
& & & & & & a_{i_1, \ell_0} & \oplus & & a_{i_1, \ell_1} & = & S_{i_1} \\
\alpha^{i_0 n + j_0} a_{i_0, j_0} & \oplus & & \alpha^{i_0 n + j_1} a_{i_0, j_1} & \oplus & & \alpha^{i_1 n + \ell_0} a_{i_1, \ell_0} & \oplus & & \alpha^{i_1 n + \ell_1} a_{i_1, \ell_1} & = & S_m \\
\alpha^{2(i_0 n + j_0)} a_{i_0, j_0} & \oplus & & \alpha^{2(i_0 n + j_1)} a_{i_0, j_1} & \oplus & & \alpha^{2(i_1 n + \ell_0)} a_{i_1, \ell_0} & \oplus & & \alpha^{2(i_1 n + \ell_1)} a_{i_1, \ell_1} & = & S_{m+1}
\end{array}$$

where S_{i_0} and S_{i_1} are given by (3) and S_m and S_{m+1} are given by (5). In order to solve this linear system, we need to invert the determinant

$$\det \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & \alpha^{i_1 n + \ell_0} & \alpha^{i_1 n + \ell_1} \\ \alpha^{2(i_0 n + j_0)} & \alpha^{2(i_0 n + j_1)} & \alpha^{2(i_1 n + \ell_0)} & \alpha^{2(i_1 n + \ell_1)} \end{pmatrix}.$$

By row operations, we can easily see that this determinant is equal to the following determinant times a power of α :

$$\det \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 \oplus \alpha^{j_1 - j_0} & 0 & \alpha^{(i_1 - i_0)n + \ell_0 - j_0} (1 \oplus \alpha^{\ell_1 - \ell_0}) \\ 0 & 1 \oplus \alpha^{2(j_1 - j_0)} & 0 & \alpha^{2((i_1 - i_0)n + \ell_0 - j_0)} (1 \oplus \alpha^{2(\ell_1 - \ell_0)}) \end{pmatrix} = \\
\det \begin{pmatrix} 1 \oplus \alpha^{j_1 - j_0} & \alpha^{(i_1 - i_0)n + \ell_0 - j_0} (1 \oplus \alpha^{\ell_1 - \ell_0}) \\ 1 \oplus \alpha^{2(j_1 - j_0)} & \alpha^{2((i_1 - i_0)n + \ell_0 - j_0)} (1 \oplus \alpha^{2(\ell_1 - \ell_0)}) \end{pmatrix}$$

Notice that this determinant corresponds to a 2×2 Vandermonde matrix, and it equals $1 \oplus \alpha^{j_1 - j_0} \oplus \alpha^{(i_1 - i_0)n + \ell_0 - j_0} (1 \oplus \alpha^{\ell_1 - \ell_0})$ times $\alpha^{(i_1 - i_0)n + \ell_0 - j_0} (1 \oplus \alpha^{j_1 - j_0}) (1 \oplus \alpha^{\ell_1 - \ell_0})$. We have seen that the latter is easy to invert. Inverting $1 \oplus \alpha^{j_1 - j_0} \oplus \alpha^{(i_1 - i_0)n + \ell_0 - j_0} (1 \oplus \alpha^{\ell_1 - \ell_0})$, however, is not as neat as inverting binomials $1 \oplus \alpha^j$ when the size b of the symbols is a large number. Since $1 + x^{j_1 - j_0} + x^{(i_1 - i_0)n + \ell_0 - j_0} (1 + x^{\ell_1 - \ell_0})$ and $f(x)$ are relatively prime by Theorem 5.1, we can invert $1 + x^{j_1 - j_0} + x^{(i_1 - i_0)n + \ell_0 - j_0} (1 + x^{\ell_1 - \ell_0})$ modulo $f(x)$ using Euclid's algorithm. This operation may take some computational time, but it is not done very often. When it is invoked, performance has already been degraded due in general to a catastrophic failure. The emphasis here is on data recovery and not on performance, since data loss is not acceptable.

Let us now analyze some concrete PMDS codes $\mathcal{C}(m, n, 1, 2; f(x))$. Consider first finite fields $GF(2^b)$. In Table 1, we give the value b , the irreducible polynomial $f(x)$ (in octal notation), the exponent $e(f(x))$, and values m and n for which the code $\mathcal{C}(m, n, 1, 2; f(x))$ is PMDS according to Theorem 5.1. We have not checked all possible irreducible polynomials, so we are not claiming that the values of m and n are maximal in each case, but it is certainly feasible to do so. For extensive tables of irreducible polynomials, see [40].

Next, consider the case of codes $\mathcal{C}(m, n, 1, 2; M_p(x))$. Theorem 4.2 solves the case in which $M_p(x)$ is irreducible, so assume that $M_p(x)$ is not irreducible, i.e., the ring is not a field.

b	$f(x)$	$e(f(x))$	m	n	b	$f(x)$	$e(f(x))$	m	n
8	4 3 5	255	5	5	16	2 2 7 2 1 5	13107	404	6
	5 6 7	85	7	5				346	7
	4 3 3	51	10	5				303	8
9	1 0 2 1	511	20	6				269	9
	1 2 3 1	73	10	7				242	10
10	3 0 2 5	1023	21	6				164	11
			15	7				160	12
11	6 0 1 5	2047	29	6				59	16
			25	7				45	17
			22	8				53	18
			13	10				24	20
12	1 5 6 4 7	4095	67	6				19	22
			58	7				21	23
			50	8	18	24			
			24	9	17	25			
			22	10	16	26			

Table 1: Some values of b , $f(x)$, m and n for which codes $\mathcal{C}(m, n, 1, 2; f(x))$ are PMDS

This ring was considered for the BR codes [7] because it allows for efficient correction of erasures for symbols of large size without using look-up tables like in the case of finite fields. We need to check all possible cases of Theorem 5.1 for different values of m and n , $mn < p$.

The results are tabulated in Table 2, which gives the list of primes between 17 and 257 for which $M_p(x)$ is reducible (hence, 2 is not primitive in $GF(p)$), together with some values of m and n , and a statement indicating whether the code is PMDS or not. For most such primes the codes are PMDS. The only exceptions are 31, 73 and 89. The case $p=89$ is particularly interesting, since for $m=8$ and $n=11$ as well as for $m=n=9$, the codes are not PMDS. However, for $m=11$ and $n=8$, the code is PMDS, which illustrates the fact that a code being PMDS does not depend only on the polynomial $M_p(x)$ chosen, but also on m and n .

5.3 The case $\mathcal{C}(m, n, 1, 3; f(x))$

There are two ways to obtain $s=3$ as a sum of odd numbers: one is 3 itself, the other is $1+1+1$. Then, by Theorem 4.1, we have

Theorem 5.2 Code $\mathcal{C}(m, n, 1, 3; f(x))$ is PMDS if and only if code $\mathcal{C}(m, n, 1, 2; f(x))$ is PMDS, and, for $1 \leq l_1 < l_2 < l_3 \leq n-1$,

$$\gcd(1 + x^{l_1} + x^{l_2} + x^{l_3}, f(x)) = 1 \quad (17)$$

Prime	m	n	PMDS?
17	4	4	YES
23	3	7	YES
	4	5	YES
31	5	6	NO
	6	5	NO
41	5	8	YES
	6	6	YES
	8	5	YES
43	5	8	YES
	6	7	YES
47	4	11	YES
	5	9	YES
71	7	10	YES
	8	8	YES
	10	7	YES
73	6	12	NO
	7	10	NO
	8	9	NO
	9	8	NO
79	6	13	YES
	7	11	YES
	8	9	YES
89	8	11	NO
	9	9	NO
	11	8	YES
97	8	12	YES
	10	9	YES
	12	8	YES
103	9	11	YES
	10	10	YES
	11	9	YES
109	9	12	YES
	10	10	YES
	12	9	YES
113	10	11	YES
	11	10	YES
	12	9	YES

Prime	m	n	PMDS?
127	11	11	YES
	13	9	YES
137	11	12	YES
	12	11	YES
	13	10	YES
	15	9	YES
	16	8	YES
151	15	10	YES
	16	9	YES
157	12	13	YES
	13	12	YES
	14	11	YES
	15	10	YES
	16	9	YES
167	12	13	YES
	13	12	YES
	15	11	YES
	16	10	YES
191	13	14	YES
	14	13	YES
	17	11	YES
193	16	12	YES
199	14	14	YES
	16	12	YES
223	15	14	YES
	17	13	YES
229	15	15	YES
	16	14	YES
233	15	15	YES
	16	14	YES
239	15	15	YES
	16	14	YES
241	16	15	YES
251	16	15	YES
	25	10	YES
257	16	16	YES
	32	8	YES

Table 2: Values of p such that 2 is not primitive in $GF(p)$, and some codes $\mathcal{C}(m, n, 1, 2; M_p(x))$, $mn < p$.

and, for any $1 \leq i_2 < i_3 \leq m - 1$, $0 \leq l_{1,0} < l_{1,1} \leq n - 1$, $0 \leq l_{2,0} < l_{2,1} \leq n - 1$ and $0 \leq l_{3,0} < l_{3,1} \leq n - 1$,

$$\gcd\left(1 + x^{l_{1,1}-l_{1,0}} + x^{i_2n+l_{2,0}-l_{1,0}}\left(1 + x^{l_{2,1}-l_{2,0}}\right) + x^{i_3n+l_{3,0}-l_{1,0}}\left(1 + x^{l_{3,1}-l_{3,0}}\right), f(x)\right) = 1 \quad (18)$$

So, in order to check if code $\mathcal{C}(m, n, 1, 3; f(x))$ is PMDS, we start checking if code $\mathcal{C}(m, n, 1, 2; f(x))$ is PMDS, like in the cases tabulated in Tables 1 and 2. Then we have to check if the conditions (17) and (18) of Theorem 5.2 are satisfied.

For instance, the codes $\mathcal{C}(m, n, 1, 2; f(x))$ in Table 1 are PMDS, but condition (18) in Theorem 5.2 is quite restrictive and most of the entries do not correspond to codes $\mathcal{C}(m, n, 1, 3; f(x))$ that are PMDS. In Table 2, however, several of the codes $\mathcal{C}(m, n, 1, 2; M_p(x))$ that are PMDS give codes $\mathcal{C}(m, n, 1, 3; M_p(x))$ that are also PMDS. We give the results in Table 3, which shows that for the primes 17, 43, 89, 127, 151, 241 and 257, and also for 89 with $(m, n) = (11, 8)$, the codes $\mathcal{C}(m, n, 1, 3; M_p(x))$ are not PMDS, although the corresponding codes $\mathcal{C}(m, n, 1, 2; M_p(x))$ were PMDS.

5.4 The case $\mathcal{C}(m, n, 1, 4; f(x))$

As done in Subsection 5.3, we have to start writing $s = 4$ as all possible sums of odd numbers. There are three ways of doing so: $4 = 1 + 3$, $4 = 3 + 1$ and $4 = 1 + 1 + 1 + 1$. By Theorem 4.1, we have:

Theorem 5.3 Code $\mathcal{C}(m, n, 1, 4; f(x))$ as given by Construction 3.1 is PMDS if and only if code $\mathcal{C}(m, n, 1, 3; f(x))$ is PMDS, and, for any $1 \leq i \leq m - 1$, $0 \leq l_{1,0} < l_{1,1} \leq n - 1$ and $0 \leq l_{2,0} < l_{2,1} < l_{2,2} < l_{2,3} \leq n - 1$,

$$\gcd\left(1 + x^{l_{1,1}-l_{1,0}} + x^{in+l_{2,0}-l_{1,0}}\left(1 + x^{l_{2,1}-l_{2,0}} + x^{l_{2,2}-l_{2,0}} + x^{l_{2,3}-l_{2,0}}\right), f(x)\right) = 1,$$

for any $1 \leq i \leq m - 1$, $0 \leq l_{1,0} < l_{1,1} < l_{1,2} < l_{1,3} \leq n - 1$ and $0 \leq l_{2,0} < l_{2,1} \leq n - 1$,

$$\gcd\left(1 + x^{l_{1,1}-l_{1,0}} + x^{l_{1,2}-l_{1,0}} + x^{l_{1,3}-l_{1,0}} + x^{in+l_{2,0}-l_{1,0}}\left(1 + x^{l_{2,1}-l_{2,0}}\right), f(x)\right) = 1,$$

and for any $1 \leq i_2 < i_3 < i_4 \leq m - 1$, $0 \leq l_{1,0} < l_{1,1} \leq n - 1$, $0 \leq l_{2,0} < l_{2,1} \leq n - 1$, $0 \leq l_{3,0} < l_{3,1} \leq n - 1$ and $0 \leq l_{4,0} < l_{4,1} \leq n - 1$,

$$\gcd\left(1 + x^{l_{1,1}-l_{1,0}} + x^{i_2n+l_{2,0}-l_{1,0}}\left(1 + x^{l_{2,1}-l_{2,0}}\right) + x^{i_3n+l_{3,0}-l_{1,0}}\left(1 + x^{l_{3,1}-l_{3,0}}\right) + x^{i_4n+l_{4,0}-l_{1,0}}\left(1 + x^{l_{4,1}-l_{4,0}}\right), f(x)\right) = 1.$$

Consider next a restricted situation for a code $\mathcal{C}(m, n, 1, 4; f(x))$. In [27], codes were constructed that can recover from an erased column together with a row with up to two errors, or two different rows with up to one error each. From our coding point of view, a

Prime	m	n	PMDS?
17	4	4	NO
23	3	7	YES
	4	5	YES
31	5	6	NO
	6	5	NO
41	5	8	YES
	6	6	YES
	8	5	YES
43	5	8	NO
	6	7	NO
47	4	11	YES
	5	9	YES
71	7	10	YES
	8	8	YES
	10	7	YES
73	6	12	NO
	7	10	NO
	8	9	NO
	9	8	NO
79	6	13	YES
	7	11	YES
	8	9	YES
89	8	11	NO
	9	9	NO
	11	8	NO
97	8	12	YES
	10	9	YES
	12	8	YES
103	9	11	YES
	10	10	YES
	11	9	YES

Prime	m	n	PMDS?
109	9	12	YES
	10	10	YES
	12	9	YES
113	10	11	YES
	11	10	YES
	12	9	YES
127	11	11	NO
	13	9	NO
137	11	12	YES
	12	11	YES
151	15	10	NO
	16	9	NO
157	12	13	YES
	16	9	YES
167	16	10	YES
191	17	11	YES
193	16	12	YES
199	16	12	YES
223	17	13	YES
229	16	14	YES
	28	8	YES
233	23	10	YES
239	26	9	YES
241	16	15	NO
	24	10	NO
251	25	10	YES
257	16	16	NO
	32	8	NO

Table 3: Some codes $\mathcal{C}(m, n, 1, 3; M_p(x))$ such that $p \leq 257$ and $M_p(x)$ is not irreducible

$\mathcal{C}(m, n, 1, 4; f(x))$ code that is both (1;4)-erasure correcting and (1;2,2)-erasure correcting will accomplish this (these conditions are actually stronger than those in [27], since they do not require an erased column, the erasures can be anywhere in the row). For reasons of space, we don't address at this point the decoding algorithm for errors and we concentrate on the existence of such a code.

Notice that, according to Lemma 4.3, a code $\mathcal{C}(m, n, 1, 4; f(x))$ is (1;4)-erasure correcting, if and only if for any $1 \leq l_1 < l_2 < l_3 \leq n - 1$, (17) holds.

Also by Lemma 4.3, a code $\mathcal{C}(m, n, 1, 4; f(x))$ is (1;2,2)-erasure correcting, if and only if for any $1 \leq i \leq m - 1$ and $0 \leq l_{1,0} < l_{1,1} \leq n - 1$, $0 \leq l_{2,0} < l_{2,1} \leq n - 1$, (16) holds. But (16) is exactly the condition for code $\mathcal{C}(m, n, 1, 2)$ to be PMDS by Theorem 5.1. Thus, we have the following lemma:

Lemma 5.2 Code $\mathcal{C}(m, n, 1, 4; f(x))$ is both (1;4)-erasure correcting and (1;2,2)-erasure correcting if and only if code $\mathcal{C}(m, n, 1, 2; f(x))$ is PMDS and, for any $1 \leq l_1 < l_2 < l_3 \leq n - 1$, (17) holds.

We can verify in Table 2 that for the codes $\mathcal{C}(m, n, 1, 2; M_p(x))$ that are PMDS, (17) holds. Therefore, by Lemma 5.2, the corresponding codes $\mathcal{C}(m, n, 1, 4; M_p(x))$ are both (1;4)-erasure correcting and (1;2,2)-erasure correcting.

5.5 The case $\mathcal{C}(m, n, r, 1; f(x))$

So far, in this section we have considered cases in which $r = 1$. If $r = s = 1$, we have seen in Subsection 5.1 that the code is PMDS, so we examine here the case $r > 1$. Thus, assume that row i , $0 \leq i \leq m - 1$, has $r + 1$ erasures in locations $0 \leq j_0 < j_1 < \dots < j_r \leq n - 1$. The following theorem is given without proof (it is proven similarly to the previous cases by examining determinants):

Theorem 5.4 Consider code $\mathcal{C}(m, n, r, 1; f(x))$. If r is even, then $\mathcal{C}(m, n, r, 1; f(x))$ is PMDS if and only if $\mathcal{C}(m, n, r - 1, 1; f(x))$ is PMDS, while if r is odd, $\mathcal{C}(m, n, r, 1; f(x))$ is PMDS if and only if $\mathcal{C}(m, n, r - 1, 1; f(x))$ is PMDS and, for any $1 \leq l_1 < l_2 < \dots < l_r \leq n - 1$,

$$\gcd\left(1 + \sum_{u=1}^r x^{l_u}, f(x)\right) = 1 \quad (19)$$

Since $\mathcal{C}(m, n, 1, 1; f(x))$ is PMDS, by Theorem 5.4, also $\mathcal{C}(m, n, 2, 1; f(x))$ is PMDS. According to (19), $\mathcal{C}(m, n, 3, 1; f(x))$ and $\mathcal{C}(m, n, 4, 1; f(x))$ are PMDS if and only if, for any $1 \leq l_1 < l_2 < l_3 \leq n - 1$, (17) holds.

5.6 The case $\mathcal{C}(m, n, 2, 2; f(x))$

For $\mathcal{C}(m, n, 2, 2; f(x))$ to be PMDS, it has to be both (2;2)-erasure correcting and (2;1,1)-erasure correcting. As in the previous subsection, $\mathcal{C}(m, n, 2, 2; f(x))$ will be (2;2)-erasure correcting if and only if, for any $1 \leq l_1 < l_2 < l_3 \leq n-1$, (17) holds. We have also seen at the end of the previous subsection that this is equivalent to saying that code $\mathcal{C}(m, n, 3, 1; f(x))$ is PMDS. By examining the conditions under which code $\mathcal{C}(m, n, 2, 2; f(x))$ is (2;1,1)-erasure correcting, we have the following theorem (again, without proof):

Theorem 5.5 Code $\mathcal{C}(m, n, 2, 2; f(x))$ is PMDS if and only if code $\mathcal{C}(m, n, 3, 1; f(x))$ is PMDS and, for any $1 \leq i \leq m-1$, $0 \leq l_{1,0} < l_{1,1} < l_{1,2} \leq n-1$ and $0 \leq l_{2,0} < l_{2,1} < l_{2,2} \leq n-1$, if

$$g(x) = 1 + x^{l_{1,1}-l_{1,0}} + x^{l_{1,2}-l_{1,0}} + x^{2(l_{1,1}-l_{1,0})} + x^{2(l_{1,2}-l_{1,0})} + x^{(l_{1,1}-l_{1,0})(l_{1,2}-l_{1,0})} + x^{2(in+l_{2,0}-l_{1,0})} \left(1 + x^{l_{2,1}-l_{2,0}} + x^{l_{2,2}-l_{2,0}} + x^{2(l_{2,1}-l_{2,0})} + x^{2(l_{2,2}-l_{2,0})} + x^{(l_{2,1}-l_{2,0})(l_{2,2}-l_{2,0})} \right),$$

then $\gcd(g(x), f(x)) = 1$.

6 An alternative construction

In this section we present an alternative to Construction 3.1.

Construction 6.1 Consider the binary polynomials modulo $f(x)$, where either $f(x)$ is irreducible or $f(x) = M_p(x)$, and let $mn \leq e(f(x))$. Let $\mathcal{C}^{(1)}(m, n, r, s; f(x))$ be the code whose $(mr+s) \times mn$ parity-check matrix is

$$\mathcal{H}^{(1)}(m, n, r, s) = \left(\begin{array}{c|c|c|c} H^{(1)}(n, r, 0, 0) & \mathbf{0}(n, r) & \dots & \mathbf{0}(n, r) \\ \mathbf{0}(n, r) & H^{(1)}(n, r, 0, r) & \dots & \mathbf{0}(n, r) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}(n, r) & \mathbf{0}(n, r) & \dots & H^{(1)}(n, r, 0, (m-1)r) \\ \hline & & & H^{(1)}(mn, s, r, 0) \end{array} \right) \quad (20)$$

where, if $f(\alpha) = 0$, $H^{(1)}(n, r, i, j)$ is the $r \times n$ matrix

$$H^{(1)}(n, r, i, j) = \left(\begin{array}{c|c|c|c|c} \alpha^{ij} & \alpha^{i(j+1)} & \alpha^{i(j+2)} & \dots & \alpha^{i(j+n-1)} \\ \alpha^{(i+1)j} & \alpha^{(i+1)(j+1)} & \alpha^{(i+1)(j+2)} & \dots & \alpha^{(i+1)(j+n-1)} \\ \alpha^{(i+2)j} & \alpha^{(i+2)(j+1)} & \alpha^{(i+2)(j+2)} & \dots & \alpha^{(i+2)(j+n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^{(i+r-1)j} & \alpha^{(i+r-1)(j+1)} & \alpha^{(i+r-1)(j+2)} & \dots & \alpha^{(i+r-1)(j+n-1)} \end{array} \right) \quad (21)$$

and $\mathbf{0}(n, r)$ is an $r \times n$ zero matrix.

Next we illustrate Construction 6.1 with some examples.

Example 6.1 Consider $m=3$ and $n=5$, then,

$$\mathcal{H}^{(1)}(3, 5, 1, 3) = \left(\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} & \alpha^{16} & \alpha^{18} & \alpha^{20} & \alpha^{22} & \alpha^{24} & \alpha^{26} & \alpha^{28} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} & \alpha^{24} & \alpha^{27} & \alpha^{30} & \alpha^{33} & \alpha^{36} & \alpha^{39} & \alpha^{42} \end{array} \right)$$

$$\mathcal{H}^{(1)}(3, 5, 3, 1) = \left(\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^{10} & \alpha^{12} & \alpha^{14} & \alpha^{16} & \alpha^{18} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{20} & \alpha^{22} & \alpha^{24} & \alpha^{26} & \alpha^{28} \\ \hline 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} & \alpha^{24} & \alpha^{27} & \alpha^{30} & \alpha^{33} & \alpha^{36} & \alpha^{39} & \alpha^{42} \end{array} \right)$$

$$\mathcal{H}^{(1)}(3, 5, 2, 2) = \left(\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ \hline 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} & \alpha^{16} & \alpha^{18} & \alpha^{20} & \alpha^{22} & \alpha^{24} & \alpha^{26} & \alpha^{28} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} & \alpha^{24} & \alpha^{27} & \alpha^{30} & \alpha^{33} & \alpha^{36} & \alpha^{39} & \alpha^{42} \end{array} \right)$$

Notice that $\mathcal{C}(m, n, 1, 2; f(x))$ and $\mathcal{C}^{(1)}(m, n, 1, 2; f(x))$ coincide. Let us analyze in the next subsections some special cases.

6.1 The case $\mathcal{C}^{(1)}(m, n, r, 1; f(x))$

Like in Subsection 5.5, we have to examine under which conditions code $\mathcal{C}^{(1)}(m, n, r, 1; f(x))$ is $(1; r)$ -erasure correcting. Using the parity-check matrix $\mathcal{H}^{(1)}(m, n, r, 1)$ as defined by (20), $\mathcal{C}^{(1)}(m, n, r, 1; f(x))$ is $(r; 1)$ -erasure correcting if and only if, for any $0 \leq i \leq m-1$ and for any $1 \leq j_0 < j_1 < \dots < j_r$, the Vandermonde determinant

$$\det \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha^{in+j_0} & \alpha^{in+j_1} & \dots & \alpha^{in+j_r} \\ \alpha^{2(in+j_0)} & \alpha^{2(in+j_1)} & \dots & \alpha^{2(in+j_r)} \\ \alpha^{3(in+j_0)} & \alpha^{3(in+j_1)} & \dots & \alpha^{3(in+j_r)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{r(in+j_0)} & \alpha^{r(in+j_1)} & \dots & \alpha^{r(in+j_r)} \end{pmatrix} = \prod_{0 \leq t < l \leq r} (\alpha^{in+j_t} \oplus \alpha^{in+j_l})$$

is invertible. Since this is always the case, we have the following theorem:

Theorem 6.1 Code $\mathcal{C}^{(1)}(m, n, r, 1; f(x))$ is PMDS.

Comparing Theorems 5.4 and 6.1, we conclude that codes $\mathcal{C}^{(1)}(m, n, r, 1; f(x))$ are preferable to codes $\mathcal{C}(m, n, r, 1; f(x))$ for $r \geq 2$, since the former are PMDS without restrictions.

6.2 The case $\mathcal{C}^{(1)}(m, n, 1, 3; f(x))$

We give the following theorem without proof:

Theorem 6.2 Code $\mathcal{C}^{(1)}(m, n, 1, 3; f(x))$ as given by Construction 6.1 is PMDS if and only if, for any $0 \leq i_1 \neq i_2 \leq m-1$, $0 \leq l_{1,0} < l_{1,1} < l_{1,2} \leq n-1$ and $0 \leq l_{2,0} < l_{2,1} \leq n-1$, $f(\alpha) = 0$, the following matrix is invertible,

$$\begin{pmatrix} 1 \oplus \alpha^{l_{1,1}-l_{1,0}} & 1 \oplus \alpha^{l_{1,2}-l_{1,0}} & \alpha^{(i_2-i_1)n+l_{2,0}-l_{1,0}}(1 \oplus \alpha^{l_{2,1}-l_{2,0}}) \\ 1 \oplus \alpha^{2(l_{1,1}-l_{1,0})} & 1 \oplus \alpha^{2(l_{1,2}-l_{1,0})} & \alpha^{2((i_2-i_1)n+l_{2,0}-l_{1,0})}(1 \oplus \alpha^{2(l_{2,1}-l_{2,0})}) \\ 1 \oplus \alpha^{3(l_{1,1}-l_{1,0})} & 1 \oplus \alpha^{3(l_{1,2}-l_{1,0})} & \alpha^{3((i_2-i_1)n+l_{2,0}-l_{1,0})}(1 \oplus \alpha^{3(l_{2,1}-l_{2,0})}) \end{pmatrix} \quad (22)$$

and for any $1 \leq i_2 < i_3 \leq m-1$, $0 \leq l_{1,0} < l_{1,1} \leq n-1$, $0 \leq l_{2,0} < l_{2,1} \leq n-1$ and $0 \leq l_{3,0} < l_{3,1} \leq n-1$, the following matrix is invertible:

$$\begin{pmatrix} 1 \oplus \alpha^{l_{1,1}-l_{1,0}} & \alpha^{i_2n+l_{2,0}-l_{1,0}}(1 \oplus \alpha^{l_{2,1}-l_{2,0}}) & \alpha^{i_3n+l_{3,0}-l_{1,0}}(1 \oplus \alpha^{l_{3,1}-l_{3,0}}) \\ 1 \oplus \alpha^{2(l_{1,1}-l_{1,0})} & \alpha^{2(i_2n+l_{2,0}-l_{1,0})}(1 \oplus \alpha^{2(l_{2,1}-l_{2,0})}) & \alpha^{2(i_3n+l_{3,0}-l_{1,0})}(1 \oplus \alpha^{2(l_{3,1}-l_{3,0})}) \\ 1 \oplus \alpha^{3(l_{1,1}-l_{1,0})} & \alpha^{3(i_2n+l_{2,0}-l_{1,0})}(1 \oplus \alpha^{3(l_{2,1}-l_{2,0})}) & \alpha^{3(i_3n+l_{3,0}-l_{1,0})}(1 \oplus \alpha^{3(l_{3,1}-l_{3,0})}) \end{pmatrix} \quad (23)$$

Consider $f(x) = M_p(x)$. We tested all prime numbers p such that 2 is primitive in $GF(p)$ up to $p = 227$ (i.e., $f(x)$ is irreducible), and we found out that the matrices given by (22) and (23) are invertible in all instances. Thus, we have the following lemma:

Lemma 6.1 Consider the code $\mathcal{C}^{(1)}(m, n, 1, 3; M_p(x))$ given by Construction 6.1 such that $M_p(x)$ is irreducible (or equivalently, 2 is primitive in $GF(p)$). Then, for $19 \leq p \leq 227$, code $\mathcal{C}^{(1)}(m, n, 1, 3; M_p(x))$ is PMDS.

Prime	m	n	PMDS?
17	4	4	NO
23	3	7	NO
	4	5	YES
31	5	6	NO
	6	5	NO
41	5	8	NO
	6	6	YES
	8	5	YES
43	5	8	NO
	6	7	NO
47	4	11	YES
	5	9	YES
71	7	10	YES
	8	8	YES
	10	7	YES
73	6	12	NO
	7	10	NO
	8	9	NO
	9	8	NO
79	6	13	YES
	7	11	YES
	8	9	YES
89	8	11	NO
	9	9	NO
	11	8	NO
97	8	12	YES
	10	9	YES
	12	8	YES
103	9	11	YES
	10	10	YES
	11	9	YES

Prime	m	n	PMDS?
109	9	12	YES
	10	10	YES
	12	9	YES
113	10	11	NO
	11	10	NO
	12	9	NO
127	11	11	NO
	13	9	NO
137	11	12	YES
	12	11	YES
	13	10	YES
	15	9	YES
	16	8	YES
151	15	10	NO
	16	9	NO
157	12	13	YES
	13	12	YES
	16	9	YES
167	16	10	YES
191	17	11	YES
193	16	12	YES
199	16	12	YES
223	17	13	YES
229	16	14	YES
	28	8	YES
233	23	10	YES
239	26	9	YES
241	24	10	NO
251	25	10	YES
257	16	16	NO
	32	8	NO

Table 4: Some codes $\mathcal{C}^{(1)}(m, n, 1, 3; M_p(x))$ such that $p \leq 257$ and $M_p(x)$ is not irreducible

We leave as an open problem whether codes $\mathcal{C}^{(1)}(m, n, 1, 3; M_p(x))$ are PMDS when $M_p(x)$ is irreducible (this result was true for codes $\mathcal{C}(m, n, 1, 3; M_p(x))$ by Theorem 4.2). For values of p such that 2 is not primitive in $GF(p)$, some results are tabulated in Table 4 for different values of m and n . This table is very similar to Table 3.

Comparing Tables 3 and 4, we can see that for values of p , m and n for which $\mathcal{C}^{(1)}(m, n, 1, 3; M_p(x))$ is PMDS, also $\mathcal{C}(m, n, 1, 3; M_p(x))$ is PMDS. However, for $p=23$, $\mathcal{C}(3, 7, 1, 3; M_p(x))$ is PMDS but $\mathcal{C}^{(1)}(3, 7, 1, 3; M_p(x))$ is not, for $p=41$, $\mathcal{C}(5, 8, 1, 3; M_p(x))$ is PMDS but $\mathcal{C}^{(1)}(5, 8, 1, 3; M_p(x))$ is not, and for $p=113$, $\mathcal{C}(3, 7, 10, 11; M_p(x))$, $\mathcal{C}(3, 7, 11, 10; M_p(x))$ and $\mathcal{C}(3, 7, 12, 9; M_p(x))$ are PMDS but $\mathcal{C}^{(1)}(3, 7, 10, 11; M_p(x))$, $\mathcal{C}^{(1)}(3, 7, 11, 10; M_p(x))$ and $\mathcal{C}^{(1)}(3, 7, 12, 9; M_p(x))$ are not.

7 A Simplified Construction

In this section we present a construction that is an alternative to codes $\mathcal{C}(m, n, 1, s; f(x))$ for $1 \leq s \leq 2$. In the case of $s=2$, the new construction can correct the situation depicted at the left of Figure 2, that is, two pairs of erasures in two different rows. It cannot correct the situation at the right of Figure 2, i.e., three erasures in the same row. This is a tradeoff, since the new construction, as we will see, uses a smaller finite field or ring. Explicitly:

Construction 7.1 Consider the binary polynomials modulo $f(x)$, where either $f(x)$ is irreducible or $f(x) = M_p(x)$, and let $\max\{m, n\} \leq e(f(x))$, where $e(f(x))$ is the exponent of $f(x)$. Let $\mathcal{C}^{(2)}(m, n, 1, 2; f(x))$ be the code whose $(m+2) \times mn$ parity-check matrix is

$$\mathcal{H}^{(2)}(m, n, 1, 2) = \left(\begin{array}{cccc|cccc|c|cccc} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 1 & 1 & \dots & 1 \\ \hline 1 & \alpha & \dots & \alpha^{n-1} & 1 & \alpha & \dots & \alpha^{n-1} & \dots & 1 & \alpha & \dots & \alpha^{n-1} \\ 1 & \alpha & \dots & \alpha^{n-1} & \alpha & \alpha^2 & \dots & \alpha^n & \dots & \alpha^{m-1} & \alpha^{m-2} & \dots & \alpha^{m+n-2} \end{array} \right)$$

$\mathcal{C}^{(2)}(m, n, 1, 1; f(x))$ is the code whose $(m+1) \times mn$ parity-check matrix is given by the first $m+1$ rows of $\mathcal{H}^{(2)}(m, n, 1, 2)$.

The following example illustrates Construction 7.1.

Example 7.1 Consider codes $\mathcal{C}^{(2)}(3, 5, 1, 2; M_5(x))$ and $\mathcal{C}^{(2)}(5, 3, 1, 2; M_5(x))$. Then, since $\alpha^5 = 1$, their respective parity-check matrices are

$$\mathcal{H}^{(2)}(3, 5, 1, 2) = \left(\begin{array}{ccccc|ccccc|c|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & 1 & \alpha^2 & \alpha^3 & \alpha^4 & 1 & \alpha \end{array} \right)$$

$$\mathcal{H}^{(2)}(5, 3, 1, 2) = \left(\begin{array}{ccc|ccc|ccc|ccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 \\ 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha^3 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^3 & \alpha^4 & 1 & \alpha^4 & 1 & \alpha \end{array} \right)$$

The following lemma is immediate:

Lemma 7.1 The code $\mathcal{C}^{(2)}(m, n, 1, 1; f(x))$ given by Construction 7.1 is PMDS.

Comparing lemmas 5.1 and 7.1, both $\mathcal{C}(m, n, 1, 1; f(x))$ and $\mathcal{C}^{(2)}(m, n, 1, 1; g(x))$ are PMDS, where $f(x)$ and $g(x)$ are either irreducible or have the form $M_p(x)$ for some prime number p . However, the conditions on $\mathcal{C}^{(2)}(m, n, 1, 1; g(x))$ are less stringent. For instance, if we consider the codes of Example 7.1 for $M_p(x)$, we can see that we need to consider at least $p=17$ for $\mathcal{C}(3, 5, 1, 1; M_p(x))$ and $\mathcal{C}(5, 3, 1, 1; M_p(x))$, while we may take $p=5$ for $\mathcal{C}^{(2)}(3, 5, 1, 1; M_p(x))$ and $\mathcal{C}^{(2)}(5, 3, 1, 1; M_p(x))$. Thus, although we are using a smaller field or ring, the PMDS property is not lost. This is not the case for codes $\mathcal{C}^{(2)}(m, n, 1, 2; f(x))$: we immediately see that the codes are not (1;2)-erasure correcting (and hence are not PMDS). However, they are (1;1,1)-erasure correcting, as stated in the following lemma:

Lemma 7.2 The code $\mathcal{C}^{(2)}(m, n, 1, 2; f(x))$ given by Construction 7.1 is (1;1,1)-erasure correcting.

Proof: Assume that we have two erasures in locations j_0 and j_1 of row i_0 and two erasures in locations ℓ_0 and ℓ_1 of row i_1 , where $0 \leq i_0 < i_1 \leq m-1$, $0 \leq j_0 < j_1 \leq n-1$ and $0 \leq \ell_0 < \ell_1 \leq n-1$. Using the parity-check matrix $\mathcal{H}^{(2)}(m, n, 1, 2)$ as given in Construction 7.1, these four erasures can be recovered if and only if

$$\det \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \alpha^{j_0} & \alpha^{j_1} & \alpha^{\ell_0} & \alpha^{\ell_1} \\ \alpha^{i_0+j_0} & \alpha^{i_0+j_1} & \alpha^{i_1+\ell_0} & \alpha^{i_1+\ell_1} \end{pmatrix}$$

is invertible. By row operations, we find out that this determinant is invertible if and only if $1 \oplus \alpha^{j_1-j_0}$, $1 \oplus \alpha^{\ell_1-\ell_0}$ and $1 \oplus \alpha^{i_1-i_0}$ are invertible. But this is certainly the case if $f(x)$ is irreducible or $f(x) = M_p(x)$, since $j_1 - j_0$, $\ell_1 - \ell_0$ and $i_1 - i_0$ are smaller than the exponent of $f(x)$. \square

Lemma 7.2 is important in applications. Let us compare it with $\mathcal{C}(m, n, 1, 2; f(x))$ codes that are PMDS as given in Subsection 5.2. For the sake of discussion, let us assume

that $\max\{m, n\} \leq 15$, a situation that covers some practical applications. In the case of $\mathcal{C}(m, n, 1, 2; f(x))$, using Table 1, if $n = 15$, we would need to operate on the field $GF(2^{16})$. If we use a code $\mathcal{C}^{(2)}(m, n, 1, 2; f(x))$, we can take the finite field $GF(2^4)$ as given by a primitive polynomial, which has exponent 15. If we use $GF(2^5)$ with a primitive polynomial, we can increase m to 16, a value convenient in applications. If we use rings generated by $M_p(x)$ and we want $m \leq 17$, $n \leq 17$, by Table 2, we may use $p = 257$ for a PMDS code $\mathcal{C}(m, n, 1, 2; M_{257}(x))$. If we just implement a $(1;1,1)$ -erasure correcting code for the same values of m and n , we can do it with a code $\mathcal{C}^{(2)}(m, n, 1, 2; M_{17}(x))$.

Let us point out that Construction 7.1 is closely related to Generalized Concatenated (GC) codes [8][44]. For descriptions of GC codes, see also [9][13][41] and the references therein. Implementations of GC codes are given by two-level ECC schemes [1][10][34], later improved in the two-level [17][18] and the multilevel [38] Integrated Interleaving schemes.

Using ideas similar to the ones of GC codes we can extend Construction 7.1 to codes that are $(1; \overbrace{1, 1, \dots, 1}^s)$ -erasure correcting (that we denote $\mathcal{C}^{(2)}(m, n, 1, s; f(x))$) as well as other combinations by using horizontal and vertical codes, but for reasons of space we omit them here. Moreover, as we will see in the next section, there is not much gain for codes $\mathcal{C}^{(2)}(m, n, 1, s; f(x))$ and $s \geq 3$ with respect to codes $\mathcal{C}^{(2)}(m, n, 1, 2; f(x))$ in a mixed environment of catastrophic failures and hard errors.

8 Probability of Data Loss After One Disk Failure

In this section, we assume that a catastrophic device failure has occurred. We will make a number of assumptions and we will compute the probability of data loss for the different schemes presented in the paper as a function of the raw error probability p (a parameter that, as we have discussed in the Introduction, degrades with time and with the number of writes for SSDs). Specifically, we will compare $(1; 2)$ PMDS codes and $(1;1,1)$ -erasure correcting codes, since both have the same redundancy (but the former is implemented over a larger field). We assume that the information in each SSD is stored in pages, where each page has size 4K and there are eight 512B sectors per page. Further, we assume that each sector is protected by a t -bit error-correcting code, like a BCH code (for instance, $t = 15$). Each SSD device has M pages. For example, if a device has size 32 G, it has 8 million pages. We assume that stripes are rows of pages in an $m \times n$ block.

Since we assume that one of the n devices has failed, if exactly one hard error has occurred in at least three different stripes of an m -stripe block, we will have data loss. If a $(1; 2)$ PMDS code is used, three hard errors in the same stripe will also cause data loss, while if a $(1; 1, 1)$ -erasure correcting code is used, two hard errors in the same stripe are enough to cause data loss.

As stated above, each codeword is in a BCH code with 512B information bytes (4096 bits). The BCH code can correct up to t bit errors, so the redundancy is $13t$ bits, giving 195 bits for $t = 15$.

We want to compute first the probability P that a codeword cannot be decoded. This will occur each time $t + 1$ or more errors occur, and this event we assume is always detected either by the BCH code itself or by the CRC. Therefore, we have:

$$P = \sum_{i=t+1}^{4096+13t} \binom{4096+13t}{i} p^i (1-p)^{4096+13t-i}$$

Thus, since there are 8 sectors per page, the probability that in a page at least a codeword is not corrected (i.e., a hard error) is

$$P_H = \sum_{i=1}^8 \binom{8}{i} P^i (1-P)^{8-i} = 1 - (1-P)^8.$$

The probability of exactly one hard error in a stripe is

$$P_{HR=1} = (n-1)P_H(1-P_H)^{n-2}$$

The probability of more than j hard errors in a stripe is

$$P_{HR>j} = \sum_{i=j+1}^{n-1} \binom{n-1}{i} (P_H)^i (1-P_H)^{n-i-1}$$

The probability of exactly one hard error in at least three of the m stripes in a block is then

$$P_{S_{m,1,3}} = \sum_{i=3}^m \binom{m}{i} (P_{HR=1})^i (1-P_H)^{(n-1)(m-i)}$$

The probability of at least $j + 1$ hard errors in any of the m stripes of the block is

$$P_{S_{m,j+1,1}} = mP_{HR>j}$$

The probability of data loss in an m -stripe block of a (1;1,1)-erasure correcting code is then given by

$$P_{S_{(1;1,1)EC}} = P_{S_{m,1,3}} + P_{S_{m,2,1}},$$

while the probability of data loss in an m -stripe block of a (1,2) PMDS code is given by

$$P_{S_{(1;2)PMDS}} = P_{S_{m,1,3}} + P_{S_{m,3,1}}.$$

We can now compute the probability of data loss for both a (1;1,1)-erasure correcting code and a (1;2) PMDS code. That will occur each time at least one m -stripe block has experienced data loss. Thus, since we had assumed that there are M pages per device and that each block has m stripes, there are M/m blocks (for 32G SSDs, and $m = 16$, $M/m = 500,000$), we obtain for a (1;1,1)-erasure correcting code,

$$P_{\text{DL}(1;1,1)\text{EC}} = \sum_{i=1}^{M/m} \binom{M/m}{i} \left(P_{\text{S}(1;1,1)\text{EC}}\right)^i \left(1 - P_{\text{S}(1;1,1)\text{EC}}\right)^{(M/m)-i}$$

and for a (1;2) PMDS code,

$$P_{\text{DL}(1;2)\text{PMDS}} = \sum_{i=1}^{M/m} \binom{M/m}{i} \left(P_{\text{S}(1;2)\text{PMDS}}\right)^i \left(1 - P_{\text{S}(1;2)\text{PMDS}}\right)^{(M/m)-i}$$

Looking at the probabilities of data loss in an m -stripe block for $m = 16$ and 32G devices in Table 5, we can see that in general $P_{\text{S}(1;1,1)\text{EC}}$ is dominated by $P_{\text{S}m,2,1}$ (i.e., $P_{\text{S}(1;1,1)\text{EC}} \approx P_{\text{S}m,2,1}$), while $P_{\text{S}(1;2)\text{PMDS}}$ is dominated by $P_{\text{S}m,1,3}$ (i.e., $P_{\text{S}(1;2)\text{PMDS}} \approx P_{\text{S}m,1,3}$). For that reason, by increasing s , the probability of data loss of a $(1; \overbrace{1, 1, \dots, 1}^s)$ -erasure correcting code is basically the same for any $s \geq 2$ when a whole device has failed. In particular, for $s = m$, we have RAID 6. Of course RAID 6 can tolerate a second device failure, but any hard error in the case of two device failures will cause data loss.

Another conclusion from Table 5 is the advantage of using a (1;2) PMDS code over a (1;1,1)-erasure correcting code, both codes having the same number of parity entries. As stated in the Introduction, as the system ages, the bit error probability p degrades. So, a natural question is, if we are monitoring p , which value allows us a reasonable expectation of not experiencing data loss? For instance, when we reach $p = .0007$, according to Table 5, the probability of miscorrection in case a device fails is 7.8E-5. This may be viewed as, less than one in ten thousand systems will have data loss provided a device has failed, which may be acceptable (depending on the application). However, if we used a (1;2) PMDS code, when $p = .0008$, the probability of data loss is 6.3E-6, more than an order of magnitude better than the (1;1,1)-erasure correcting code. So, the system is more reliable and allows further degradation of the parameter p , increasing its lifetime.

9 Conclusions

We have presented two constructions of codes that are suitable for a flash array type of architecture, in which hard errors co-exist with catastrophic device failures. We have presented specific codes that are useful in applications. Necessary and sufficient conditions for codes satisfying an optimality criterion were given.

p	.0001	.0002	.0003	.0004	.0005	.0006	.0007	.0008	.0009	.001
P	4.1E-20	1.8E-15	7.9E-13	5.3E-11	1.3E-9	1.6E-8	1.2E-7	7.0E-7	3.1E-6	1.1E-5
P_H	3.3E-19	1.4E-14	6.3E-12	7.9E-13	1.0E-8	1.3E-7	9.9E-7	5.6E-6	2.5E-5	9.0E-5
$P_{S_{16,1,3}}$	2.5E-51	2.1E-37	1.8E-29	5.3E-24	7.2E-20	1.4E-16	6.8E-14	1.3E-11	1.1E-9	5.2E-8
$P_{S_{16,2,1}}$	5.3E-35	3.3E-26	6.4E-21	2.9E-17	1.6E-14	2.5E-12	1.6E-10	5.1E-9	1.0E-9	1.3E-6
$P_{S_{16,3,1}}$	5.7E-54	4.8E-40	4.1E-32	1.2E-26	3.1E-19	3.1E-19	1.6E-16	2.9E-14	2.5E-12	1.2E-10
$P_{S_{(1;1,1)EC}}$	1.7E-35	3.3E-26	6.4E-21	2.9E-17	1.6E-14	2.5E-12	1.6E-10	5.1E-9	1.0E-7	1.4E-6
$P_{S_{(1;2)PMDS}}$	2.5E-51	2.1E-37	1.8E-29	5.3E-24	7.2E-20	1.4E-16	6.8E-14	1.3E-11	1.1E-9	5.2E-8
$P_{DL_{(1;1,1)EC}}$	8.6E-30	1.7E-20	3.2E-15	1.4E-11	8.2E-9	1.3E-6	7.8E-5	2.5E-3	.05	.5
$P_{DL_{(1;2)PMDS}}$	1.2E-45	1.1E-31	8.9E-24	2.7E-18	3.6E-14	6.9E-11	3.4E-8	6.3E-6	5.4E-4	.026

Table 5: Probabilities of data loss for (1;1,1)-erasure correcting codes and (1;2) PMDS codes for different values of bit error probability p in the presence of a catastrophic device failure

A Appendix

Lemma A.1 Let $\gamma_0, \gamma_1, \dots, \gamma_{s-1}$ be distinct elements in a field or ring of characteristic 2. Consider the $s \times s$ matrix

$$\Gamma = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{s-1} \\ \gamma_0^2 & \gamma_1^2 & \cdots & \gamma_{s-1}^2 \\ \gamma_0^4 & \gamma_1^4 & \cdots & \gamma_{s-1}^4 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_0^{2^{s-1}} & \gamma_1^{2^{s-1}} & \cdots & \gamma_{s-1}^{2^{s-1}} \end{pmatrix}$$

Then,

$$\det \Gamma = \prod_{S \subseteq \{\gamma_0, \gamma_1, \dots, \gamma_{s-1}\}} \bigoplus_{i \in S} \gamma_i$$

Proof: We will do induction on s . The result is certainly true for $s=1$. Consider the determinant of the matrix obtained by replacing γ_0 in the first column of Γ by x , i.e.,

$$h(x) = \det \begin{pmatrix} x & \gamma_1 & \cdots & \gamma_{s-1} \\ x^2 & \gamma_1^2 & \cdots & \gamma_{s-1}^2 \\ x^4 & \gamma_1^4 & \cdots & \gamma_{s-1}^4 \\ \vdots & \vdots & \ddots & \vdots \\ x^{2^{s-1}} & \gamma_1^{2^{s-1}} & \cdots & \gamma_{s-1}^{2^{s-1}} \end{pmatrix}$$

Since $h(x)$ has degree 2^{s-1} , it has at most 2^{s-1} zeros. Notice that if S is one of the 2^{s-1} subsets of $\{\gamma_1, \gamma_2, \dots, \gamma_{s-1}\}$ (including the empty subset), then $\bigoplus_{i \in S} \gamma_i$ is a zero of $h(x)$ (the

element 0 corresponding to the empty set), due to the linearity of the square operation in a field of characteristic 2. Therefore, we can write,

$$h(x) = C \prod_{S \subseteq \{\gamma_1, \gamma_2, \dots, \gamma_{s-1}\}} \left(x + \bigoplus_{i \in S} \gamma_i \right),$$

where

$$C = \det \begin{pmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_{s-1} \\ \gamma_1^2 & \gamma_2^2 & \cdots & \gamma_{s-1}^2 \\ \gamma_1^4 & \gamma_2^4 & \cdots & \gamma_{s-1}^4 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1^{2^{s-2}} & \gamma_2^{2^{s-2}} & \cdots & \gamma_{s-1}^{2^{s-2}} \end{pmatrix}$$

The result follows from the fact that $h(\gamma_0) = \det(\Gamma)$ and by induction on the expression of C above. \square

References

- [1] K. Abdel-Ghaffar and M. Hassner, “Multilevel codes for data storage channels,” *IEEE Trans. on Information Theory*, vol. IT-37, pp. 735–41, May 1991.
- [2] M. Balakrishnan, A. Kadav, V. Prabhakaran and D. Malkhi, “Differential RAID: Rethinking RAID for SSD reliability,” *ACM Transactions on Storage (TOS)*, Vol. 6, Issue 2, Article 1, July 2010.
- [3] A. Barg and A. Mazumdar, “Codes in Permutations and Error Correction for Rank Modulation,” *IEEE Trans. on Information Theory*, vol. IT-56, pp. 3158–65, July 2010.
- [4] M. Blaum, J. Brady, J. Bruck and J. Menon, “EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures,” *IEEE Trans. on Computers*, vol. C-44, pp. 192–202, February 1995.
- [5] M. Blaum, P. G. Farrell and H. C. A. van Tilborg, “Array Codes,” *Handbook of Coding Theory*, edited by V. S. Pless and W. C. Huffman, Elsevier Science B. V., Chapter 22, 1998.
- [6] M. Blaum and R. J. McEliece, “Coding protection for magnetic tapes: a generalization of the Patel-Hong code,” *IEEE Trans. on Information Theory*, vol. IT-31, pp. 690–693, September 1985.
- [7] M. Blaum and R. M. Roth, “New Array Codes for Multiple Phased Burst Correction,” *IEEE Trans. on Information Theory*, vol. IT-39, pp. 66–77, January 1993.

- [8] E. L. Blokh and V. V. Zyablov, “Coding of Generalized Concatenated Codes,” *Problemy Peredachii Informatsii*, Vol. 10(3), pp. 218–222, 1974.
- [9] M. Bossert, “Channel Coding for Telecommunications,” Ch. 9, Wiley, 1999.
- [10] O. Collins, “Exploiting the Cannibalistic Traits of Reed-Solomon Codes,” *IEEE Trans. on Communications*, Vol. 43, No. 11, pp. 2696–703, November 1995.
- [11] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, “Row-diagonal parity for double disk failure correction,” *Proc. 3rd Conf. File and Storage Technologies - FAST’04*, San Francisco, CA, March/April 2004.
- [12] P. Delsarte, “Bilinear Forms over a Finite Field, with Applications to Coding Theory,” *J. Comb. Th. Series A*, 25, pp. 226–241, 1978.
- [13] I. Dumer, “Concatenated Codes and Their Multilevel Generalizations,” *Handbook of Coding Theory*, edited by V. S. Pless and W. C. Huffman, Elsevier Science B. V., Chapter 23, 1998.
- [14] E. M. Gabidulin, “Theory of codes with maximum rank distance,” *Prob. Info. Trans.*, Vol. 21, No. 1, pp. 3–16, 1985.
- [15] G. A. Gibson, “Redundant Disk Arrays,” MIT Press, 1992.
- [16] K. M. Greenan, D. D. Long, E. L. Miller, T. J. E. Schwarz and A. Wildani, “Building Flexible, Fault-Tolerant Flash-based Storage Systems,” *Fifth Workshop on Hot Topics in Dependability (HotDep09)*, Lisbon, Portugal, June 2009.
- [17] J. Han and L. A. Lastras-Montaño, “Reliable Memories with Subline Accesses,” *ISIT 2007*, *IEEE International Symposium on Information Theory*, pp. 2531–35, June 2007.
- [18] M. Hassner, K. Abdel-Ghaffar, A. Patel, R. Koetter and B. Trager, “Integrated Interleaving – A Novel ECC Architecture,” *IEEE Transactions on Magnetics*, Vol. 37, No. 2, pp. 773–5, March 2001.
- [19] Q. Huang, S. Lin and K. A. S. Abdel-Ghaffar, “Error-Correcting Codes for Flash Coding,” *IEEE Trans. on Information Theory*, vol. IT-57, pp. 6097–108, September 2011.
- [20] W. Hutsell, “An In-depth Look at the RamSan-500 Cached Flash Solid State Disk,” <http://www.texmemsys.com/files/f000233.pdf>.
- [21] S. Im and D. Shin, “Flash-Aware RAID Techniques for Dependable and High-Performance Flash Memory SSD,” *IEEE Trans. on Computers*, vol. C-60, pp. 80–922, January 2011.

- [22] A. Jiang, V. Bohossian and J. Bruck, “Rewriting Codes for Joint Information Storage in Flash Memories,” *IEEE Trans. on Information Theory*, vol. IT-56, pp. 5300-13, September 2010.
- [23] A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, “Rank Modulation for Flash Memories,” *IEEE Trans. on Information Theory*, vol. IT-55, pp. 2659-73, June 2009.
- [24] A. Jiang, M. Schwartz and J. Bruck, “Correcting Charge-Constrained Errors in the Rank-Modulation Scheme,” *IEEE Trans. on Information Theory*, vol. IT-56, pp. 2112-20, April 2010.
- [25] T. Klove, B. Bose and N. Eliaref, “Systematic, Single Limited Magnitude Error Correcting Codes for Flash Memories,” *IEEE Trans. on Information Theory*, vol. IT-57, pp. 4477-87, July 2011.
- [26] T. Klove, J. Luo, I. Naydenova and S. Yari, “Some Codes Correcting Asymmetric Errors of Limited Magnitude,” *IEEE Trans. on Information Theory*, vol. IT-57, pp. 7459-72, November 2011.
- [27] L. A. Lastras-Montaña, P. J. Meaney, E. Stephens, B. M. Trager, J. O’Connor and L. C. Alves, “A new class of array codes for memory storage,” *Information Theory and Applications Workshop (ITA)*, La Jolla, California, February 2011.
- [28] M. Li and J. Shu, “C-Codes: Cyclic Lowest-Density MDS Array Codes Constructed Using Starters for RAID 6,” *IBM Research Report*, RC25218, October 2011.
- [29] S. Lin and D. J. Costello, “Error Control Coding: Fundamentals and Applications,” Prentice Hall, 1983.
- [30] S. Lin and D. J. Costello, “Error Control Coding (2nd Edition),” Prentice Hall, 2004.
- [31] F. J. MacWilliams and N. J. A. Sloane, “The Theory of Error-Correcting Codes,” North Holland, Amsterdam, 1977.
- [32] H. MahdaviFar, P. H. Siegel, A. Vardy, J. K. Wolf and E. Yaakobi, “A nearly optimal construction of flash codes,” *ISIT 2009, IEEE International Symposium on Information Theory*, pp. 1239–43, July 2009.
- [33] Micron, “N-29-17: NAND Flash Design and Use Considerations Introduction,” <http://download.micron.com/pdf/technotes/nand/tn2917.pdf>.
- [34] A. Patel, “Two-Level Coding for Error-Control in Magnetic Disk Storage Products,” *IBM Journal of Research and Development*, vol. 33, pp. 470–84, 1989.
- [35] J. S. Plank, “A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems,” *Software – Practice & Experience*, 27(9), pp. 995-1012, September 1997.

- [36] J. S. Plank, “The RAID-6 Liberation codes,” 6th USENIX Conference on File and Storage Technologies, San Francisco, CA, pp. 97-110, 2008.
- [37] R. M. Roth, “Maximum rank array codes and their application to crisscross error correction,” IEEE Trans. on Information Theory, vol. IT-37, pp. 328–336, March 1991.
- [38] X. Tang and R. Koetter, “A Novel Method for Combining Algebraic Decoding and Iterative Processing,” ISIT 2006, IEEE International Symposium on Information Theory, pp. 474–78, July 2006.
- [39] A. Thomasian and M. Blaum, “Higher Reliability Redundant Disk Arrays: Organization, Operation, and Coding,” ACM Transactions on Storage (TOS), vol. 5, No 3. Article 7, November 2009.
- [40] W. Wesley Peterson and E. J. Weldon, Jr., “Error-Correcting Codes,” MIT Press, Second Edition, 1984.
- [41] J. Wu and D. Costello, Jr., “New Multilevel Codes over $GF(q)$,” IEEE Transactions on Information Theory, vol. IT-38, pp. 933-939, May 1992.
- [42] 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. IT-45, pp. 1817–26, September 1999.
- [43] L. Xu and J. Bruck, “X-code: MDS array codes with optimal encoding,” IEEE Trans. on Information Theory, vol. IT-45, pp. 272–76, January 1999.
- [44] V. A. Zinoviev, “Generalized cascade codes,” Probl. Pered. Inform., vol. 12, no. 1, pp. 5-15, 1976