MDS Array Codes for Correcting Criss-Cross Errors

Mario Blaum
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120
blaum@almaden.ibm.com

Jehoshua Bruck*
Mail Stop 136-93
California Institute of Technology
Pasadena, CA 91125
bruck@paradise.caltech.edu

Abstract

We present a family of MDS array codes of size $(p-1) \times (p-1)$, p a prime number, and minimum criss-cross distance 3, i.e., the code is capable of correcting any row or column in error, without apriori knowledge of what type of error occurred. The complexity of the encoding and decoding algorithms is lower than that of known codes with the same error-correcting power, since our algorithms are based on exclusive-OR operations over lines of different slopes, as opposed to algebraic operations over a finite field. We also provide efficient encoding and decoding algorithms for errors and erasures.

*Supported in part by the NSF Young Investigator Award CCR-9457811 and by the Sloan Research Fellowship.

1 Introduction

In this paper we describe 2-dimensional array codes that can correct errors given by either a row or a column in error (without apriori knowledge of which one occurred). There exist codes that can do so. Moreover, the known codes are stronger in the sense that they can correct the “rank” of an array. The idea of using the rank as a metric comes from Delsarte [4]. See also Gabidulin [6] and Roth [12]. However, these constructions are based on finite field arithmetic, as Reed-Solomon codes. Therefore, for very large arrays, they may become impractical, since they may need a very large look-up table. In this paper, we will present array codes that have the same error-correcting capability in terms of rows and columns (although sometimes they cannot correct the rank) as the ones in [4][6][12], but they have less complexity. The new codes are based on simple parity along lines of different slopes, in the spirit of [3].

There are applications in which information bits are stored in $n \times n$ bit arrays. The error patterns are such that all corrupted bits are confined to at most some pre-specified number t of rows or columns (or both). We will refer to such errors as *criss-cross errors*. Criss-cross errors can be found in memory chip arrays, where row or column failures occur due to the malfunctioning of row drivers, or column amplifiers (see, for instance [5][8][9]). Another application of codes correcting criss-cross errors occurs in multi-track magnetic tapes, where the errors usually occur *along* the tracks, whereas the information units (bytes) are recorded *across* the tracks. Computation of check bits is equivalent to decoding of erasures at the check bit locations, and in this case these erasures are perpendicular to the erroneous tracks. There exist codes for multi-track magnetic recording [2][10][11].

We need some definitions. Let $E = [e_{ij}]_{i,j=0}^{n-1}$ be an $n \times n$ matrix over a field F. A *cover* of E is a pair (X, Y) of sets $X, Y \subseteq \{0, 1, \ldots, n-1\}$, such that $e_{ij} \neq 0 \Rightarrow \big((i \in X) \text{ or } (j \in Y)\big)$ for all $0 \leq i, j \leq n-1$. The size of a cover (X, Y) is defined by $|(X, Y)| = |X| + |Y|$. The *criss-cross weight* of E, denoted by $\mathrm{w}(E)$, is the minimum size $|(X, Y)|$ over all possible covers (X, Y) of E. Note that a minimum-size cover of a given matrix E is not always unique. The rank of E over F is never greater than its criss-cross weight.

A well-known result by König (see [7, Theorem 5.1.4]) states that the minimum size of a cover of a $\{0, 1\}$-matrix is equal to the maximum number of 1's that can be chosen in that matrix with no two on the same row of column. The *criss-cross distance* $d(A, B)$ between two $n \times n$ matrices A and B over F is defined by $d(A, B) = \mathrm{w}(A - B)$.

Example 1.1 Consider the 4×4 array

$$E \quad = \quad \begin{array}{c|c|c|c|c|c}
\cline{1-4} 1 & 0 & 0 & 0 & \scriptstyle 0 \\
\cline{1-4} 0 & 1 & 0 & 0 & \scriptstyle 1 \\
\cline{1-4} 1 & 0 & 1 & 1 & \scriptstyle 2 \\
\cline{1-4} 0 & 1 & 0 & 0 & \scriptstyle 3 \\
\cline{1-4} \multicolumn{1}{c}{\scriptstyle 0} & \multicolumn{1}{c}{\scriptstyle 1} & \multicolumn{1}{c}{\scriptstyle 2} & \multicolumn{1}{c}{\scriptstyle 3} & \multicolumn{1}{c}{}
\end{array}$$

over $GF(2)$. It is easy to verify that E has two covers of size 3, namely, $\big(\{0,2\},\{1\}\big)$ and $\big(\{2\},\{0,1\}\big)$. Furthermore, since the three nonzero elements on the main diagonal of E belong to distinct rows and columns, the criss-cross weight of E must be at least 3. Therefore, $\mathrm{w}(E)=3$.

Let $\Gamma=[c_{ij}]_{i,j=0}^{n-1}$ be an $n\times n$ matrix over F, denoting the correct array to be stored, and let $\Gamma\oplus E$ denote the array actually recorded, with $E=[e_{ij}]_{i,j=0}^{n-1}$ standing for the error array. The criss-cross error model assumes that $\mathrm{w}(E)\le t$ for some pre-specified t.

An $[n\times n,k,d]$ linear array code C over a field F is a k-dimensional linear space of $n\times n$ matrices over F with d being the minimum of all criss-cross distances between pairs of distinct matrices in C. Adopting the terminology of conventional linear codes, we call d the *minimum criss-cross distance* of C. As with regular block codes d equals the minimum criss-cross weight of any nonzero matrix in C. An $[n\times n,k,d]$ array code C can correct any pattern of t criss-cross errors if and only if $t\le(d-1)/2$. The proof is again identical to the proof for block codes.

In this paper, we present array codes with minimum criss-cross distance $d=3$. The constructions in [4][6][12] operate over a field $GF(2^n)$. When n is a large number, like in holographic storage applications, the resulting complexity may be prohibitive. Thus, we want to construct codes with low complexity but still having minimum criss-cross distance 3. To this end, we will consider codes over the ring of polynomials modulo $1+x+x^2+\cdots+x^{p-1}$, p a prime number, as in [3].

We have the following version of the Singleton bound [12]:

Theorem 1.1 For any $[n\times n,k,d]$ array code over a field F,

$$k\le n(n-d+1)\ .$$

In the next section, we will construct $[(p-1)\times(p-1),(p-1)(p-3),3]$ array codes, p a prime number. According to Theorem 1.1, these codes are MDS. In Section 3, we prove the main properties of the codes, mainly, the conditions under which they are MDS with respect to the criss-cross distance. We also show that, in general, our criss-cross distance is not equivalent to the rank distance, as with the codes in [4][6][12]. We also briefly discuss possible generalizations to multiple parities. In Section 4 we present efficient decoding algorithms in the case of errors and erasures. We end the paper by drawing some conclusions.

2 Construction of the codes

We give two descriptions of the codes, one algebraic, the other geometric. In the sequel, p denotes a prime number and l a number such that $2\le l\le p-2$.

Let us start with the algebraic description. The entries of the elements of the code are polynomials modulo $1+x+\cdots+x^{p-1}$. Let α be a root of $1+x+\cdots+x^{p-1}$ and, moreover, assume that $1+x+\cdots+x^{p-1}$ is the minimal polynomial of α. Then, a parity-check matrix of code $\mathcal{C}(p,l)$ is given by

$$H(p,l) \;=\; \left(\begin{array}{ccccc} 1 & \alpha & \alpha^2 & \ldots & \alpha^{p-2} \\ 1 & \alpha^l & \alpha^{2l} & \ldots & \alpha^{(p-2)l} \end{array}\right) \tag{1}$$

Now, assume that $(c_0(\alpha), c_1(\alpha), \ldots, c_{p-2}(\alpha)) \in \mathcal{C}(p,l)$, where $c_j(\alpha) = \sum_{i=0}^{p-2} c_{ij}\alpha^i$. The codewords may be interpreted as $(p-1)\times(p-1)$ arrays $(c_{ij})_{0\le i,j\le p-2}$ such that each symbol in a codeword is given by a column in the array. We denote by $C(p,l)$ the binary code of $(p-1)\times(p-1)$ arrays derived from $\mathcal{C}(p,l)$. Normally, and in order to simplify notation, we will add an imaginary 0-row and an imaginary 0-column to the arrays in $C(p,l)$. So, the codewords may be interpreted as $p\times p$ arrays $(c_{ij})_{0\le i,j\le p-1}$, such that $c_{p-1,j} = c_{i,p-1} = 0$ for $0 \le i,j \le p-1$. Also, from now on, we take all the subindices modulo p. We apologize for this abuse, but the notation is somewhat awkward if we want to denote the modulo p subindices every time.
We will see that a geometric interpretation of code $C(p,l)$ as derived from code $\mathcal{C}(p,l)$ defined by (1), is as the set of arrays having either even or odd parity along lines of slope 1 and l. This will be made clear by the following lemma.

Lemma 2.1 Vector $(c_0(\alpha), c_1(\alpha), \ldots, c_{p-2}(\alpha))$ belongs in $\mathcal{C}(p,l)$, where $c_j(\alpha) = \sum_{i=0}^{p-2} c_{ij}\alpha^i$, if and only if, for each $0 \le i \le p-1$,

$$\sum_{j=0}^{p-1} c_{i-j,j} \;=\; b \tag{2}$$

$$\sum_{j=0}^{p-1} c_{i-lj,j} \;=\; b, \tag{3}$$

where $b \in GF(2)$.

The geometric meaning of Equations (2) and (3) is the following: we have parity in the array along lines of slope 1 and l respectively. This parity can be either even or odd: it is even when $b=0$, and odd when $b=1$.
Before proving Lemma 2.1, let us give an example.

Example 2.1 Let us consider $p=5$ and $l=2$, i.e., code $\mathcal{C}(5,2)$ or $C(5,2)$ as binary 4×4 arrays. According to (1), a parity-check matrix of the code is given by

$$H(5,2) \;=\; \left(\begin{array}{cccc} 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \end{array}\right)$$

where $1+\alpha+\alpha^2+\alpha^3+\alpha^4=0$. Now, consider the following codeword in $\mathcal{C}(5,2)$:

$$c(\alpha) \;=\; (\alpha+\alpha^3, 1+\alpha^2+\alpha^3, 1+\alpha^2+\alpha^3, 1+\alpha+\alpha^3).$$

The reader can easily verify that $c(\alpha)(H(5,2))^T=0$ (H^T denotes the transpose matrix of H), therefore $c(\alpha) \in C(5,2)$. Writing $c(\alpha)$ as the array in $C(5,2)$ in which the columns correspond to the entries of $c(\alpha)$, we obtain (remember that we are adding an extra 0-row and an extra 0-column)

0	1	1	1	0
1	0	0	1	0
0	1	1	0	0
1	1	1	1	0
0	0	0	0	0

The reader can verify that we have odd parity along the lines of slope 1 and of slope 2, as predicted by Equations (2) and (3). We are ready now to prove Lemma 2.1.

Proof of Lemma 2.1: Remember that a 0-row and as 0-column have been added to the $(c_{i,j})$ array, i.e., $c_{p-1,j}=0$ and $c_{i,p-1}=0$. Let us prove (3), (2) is analogous taking $l=1$. If $c(\alpha)=(c_0(\alpha),c_1(\alpha),\ldots,c_{p-2}(\alpha)) \in \mathcal{C}(p,l)$, taking the inner product of $c(\alpha)$ with the second row of $H(p,l)$ and remembering that the subindices are taken modulo p, we obtain

$$\begin{aligned}
0 &= \sum_{j=0}^{p-2} c_j(\alpha)\alpha^{lj} \\
&= \sum_{j=0}^{p-2}(\sum_{i=0}^{p-2} c_{i,j}\alpha^i)\alpha^{lj} \\
&= \sum_{j=0}^{p-2}\sum_{i=0}^{p-2} c_{i,j}\alpha^{i+lj} \\
&= \sum_{t=0}^{p-1}\left(\sum_{i+lj=t} c_{i,j}\right)\alpha^t \qquad (4)\\
&= \sum_{t=0}^{p-1}(\sum_{j=0}^{p-1} c_{t-lj,j})\alpha^t
\end{aligned}$$

where (4) is obtained using the fact that $\alpha^p=1$ and grouping together the terms corresponding to the same power α^t, $0 \le t \le p-1$. Since this element is in the ring modulo $1+\alpha+\cdots+\alpha^{p-1}$, $\sum_{t=0}^{p-1}(\sum_{j=0}^{p-1} c_{t-lj,j})\alpha^t=0$ if and only if $\sum_{j=0}^{p-1} c_{t-lj,j}=b_l$, $b_l \in GF(2)$, for each $0 \le t \le p-1$, which nearly coincides with (3). Similarly, by repeating this procedure with $l=1$, we obtain $\sum_{j=0}^{p-1} c_{t-j,j}=b_1$, $b_1 \in GF(2)$, for each $0 \le t \le p-1$, which nearly coincides with (2). In order to complete the proof, we need to show that $b_1 = b_l$. Notice that, taking the XOR of all the elements in the array in two different ways, we obtain

$$\begin{aligned} b_1 &= pb_1 \\ &= \sum_{t=0}^{p-1} \left(\sum_{j=0}^{p-1} c_{t-j,j} \right) \\ &= \sum_{t=0}^{p-1} \left(\sum_{j=0}^{p-1} c_{t-lj,j} \right) \\ &= pb_l \\ &= b_l, \end{aligned}$$

completing the proof. □

We next consider a code $C'(p,l)$ whose elements are the transposes of the arrays in $C(p,l)$. The following lemma connects the two codes.

Lemma 2.2 Consider the code $C(p,l)$ of binary $(p-1)\times(p-1)$ arrays defined by (1), and let $C'(p,l)$ be the code whose elements are the transposes of the elements in $C(p,l)$. Then, $C'(p,l)=C(p,1/l)$, i.e., the arrays in $C'(p,l)$ have even or odd parity along lines of slope 1 and $1/l$. Algebraically, a parity-check matrix for the corresponding code $\mathcal{C}'(p,l)$ is given by

$$H'(p,l) = \begin{pmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{p-2} \\ 1 & \alpha^{1/l} & \alpha^{2/l} & \ldots & \alpha^{(p-2)/l} \end{pmatrix} \tag{5}$$

Proof: Let $(c_{i,j}) \in C(p,l)$ and $(c'_{i,j}) \in C'(p,l)$ such that $c'_{i,j}=c_{j,i}$. According to (4) and Lemma 2.1, $\sum_{i+lj=t} c_{i,j} = b$, $b \in GF(2)$, for $0 \le t \le p-1$. Dividing the subindices by l, this occurs if and only if $\sum_{j+i/l=t/l} c_{i,j} = b$, $b \in GF(2)$, for $0 \le t \le p-1$, if and only if $\sum_{j+i/l=t} c_{i,j} = b$, $b \in GF(2)$, for $0 \le t \le p-1$, since dividing by l each $0 \le t \le p-1$ is a 1-1 function. This occurs if and only if $\sum_{i=0}^{p-1} c_{i,t-i/l} = b$, $b \in GF(2)$, for $0 \le t \le p-1$, if and only if $\sum_{i=0}^{p-1} c'_{t-i/l,i} = b$, $b \in GF(2)$, for $0 \le t \le p-1$. By Lemma 2.1, $(c'_{i,j}) \in C(p,1/l)$ and (5) is the parity-check matrix corresponding to $\mathcal{C}'(p,l)$. □

The next example illustrates Lemma 2.2.

Example 2.2 Consider code $\mathcal{C}(5,2)$ as in Example 2.1. According to Lemma 2.2, since 1/2=3 modulo 5, a parity-check matrix of the code $\mathcal{C}'(5,2)$ is given by

$$H'(5,2) = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 \end{pmatrix}$$

The transpose of the array given in Example 2.1 is,

0	1	0	1	0
1	0	1	1	0
1	0	1	1	0
1	1	0	1	0
0	0	0	0	0

We can see that the array above has odd parity along lines of slope 1 and 3.

An easy observation is, code $\mathcal{C}(p,l)$ is MDS, i.e., if we erase any two columns in an array $(c_{i,j}) \in C(p,l)$, regarding these two columns as elements modulo $1+x+\cdots+x^{p-1}$, they will be recovered by the code. Of course the same is true for the corresponding code $C'(p,l)$ in which we identify the rows of the arrays with elements modulo $1+x+\cdots+x^{p-1}$: any pair of erased rows will be recovered. Let us prove these facts in the next lemma.

Lemma 2.3 Code $\mathcal{C}(p,l)$ is MDS.

Proof: As in [3], it is enough to show that any 2×2 determinant in $H(p,l)$ as given by (1) is invertible. Notice that

$$\det \begin{pmatrix} \alpha^i & \alpha^j \\ \alpha^{li} & \alpha^{lj} \end{pmatrix} = \alpha^{i+lj} + \alpha^{j+li} = \alpha^{j+li}(\alpha^{(l-1)(j-i)} + 1),$$

and since $1 \le l \le p-3$ and $1 \le j-i \le p-2$, $(l-1)(j-i) \not\equiv 0 \pmod{p}$ and $\alpha^{(l-1)(j-i)} \ne 1$. Moreover, $\gcd(x^t(x^s+1), 1+x+\cdots+x^{p-1}) = 1$ for $s \not\equiv 0 \pmod{p}$ [3], thus, $\alpha^{j+li}(\alpha^{(l-1)(j-i)}+1)$ is invertible. $\square$

However, our goal is to show that the binary code $C(p,l)$ of $(p-1)\times(p-1)$ arrays is MDS with respect to the criss-cross distance. To this end, we have to show that any erased row together with any erased column will be uniquely recovered. This will not happen for every code $C(p,l)$. Actually, it will occur if and only if l is primitive in $GF(p)$. For instance, 2 is primitive in $GF(5)$, but not in $GF(7)$. However, 3 is primitive in $GF(7)$. Thus, $C(7,2)$ is not MDS with respect to the criss-cross distance, but $C(7,3)$ is.
We prove this and other properties in the next section.

3 Main Properties

Before proving our main theorem, let us give some examples of codes $C(p,l)$ for which 2 is not primitive in $GF(p)$.

Example 3.1 Consider the following array in $C(7,2)$ (to which a 0-row and a 0-column have been added):

0	1	1	0	1	0	0	0
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	2
0	0	0	0	0	0	0	3
1	0	0	0	0	0	0	4
0	0	0	0	0	0	0	5
0	0	0	0	0	0	0	6
0	1	2	3	4	5	6	

We can see that the array above has even parity on lines of slope 1 and 2, therefore it belongs in $C(7,2)$. This array has criss-cross weight 2, so $C(7,2)$ is not MDS with respect to the criss-cross distance.

Example 3.2 Consider the following array in $C(17,2)$ (to which a 0-row and a 0-column have been added):

0	0	0	1	0	1	1	1	0	0	1	1	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

As in Example 3.1, we can see that the array above has even parity on lines of slope 1 and 2, therefore it belongs in $C(17,2)$. This array has criss-cross weight 2, so $C(17,2)$ is not MDS with respect to the criss-cross distance.

Notice that 2 is not primitive neither in $GF(7)$ nor in $GF(17)$. The next theorem is our main result.

Theorem 3.1 Code $C(p,l)$ has minimum criss-cross distance 3 if and only if l is primitive in $GF(p)$.

Proof: $\Rightarrow$) Assume that l is not primitive in $GF(p)$. We will exhibit an array with a cover of size 2. Consider the set $S(p,l) = \{1 = l^0, l = l^1, l^2, \ldots, l^{s-1}\}$ such that these powers are taken modulo p and $l^s = 1$. Since l is not primitive in $GF(p)$, $S(p,l) \neq GF(p) - \{0\}$. There are two cases: $p-1 \notin S(p,l)$ and $p-1 \in S(p,l)$. We define a set U as follows:

$$\begin{aligned} U &= S(p,l) \text{ if } p-1 \notin S(p,l) \\ U &= GF(p) - S(p,l) - \{0\} \text{ if } p-1 \in S(p,l) \end{aligned}$$

As an example of the case $p-1 \notin S(p,l)$, consider $p=7$ and $l=2$. Then, $S(7,2) = U = \{1,2,4\}$ (notice that $6 \notin S(7,2)$). The array $(c_{i,j})$ in this case is depicted in Example 3.1.
As an example of the case $p-1 \in S(p,l)$, consider $p=17$ and $l=2$. Then,
$S(17,2) = \{1,2,4,8,9,13,15,16\}$ and $U = \{3,5,6,7,10,11,12,14\}$. The array $(c_{i,j})$ in this case is depicted in Example 3.2.
Next consider the array $(c_{i,j})_{0 \leq i,j \leq p-1}$ of criss-cross weight 2 such that:

$$\begin{aligned} c_{i,0} &= 1 \text{ if } i \in U \\ c_{0,j} &= 1 \text{ if } j \in U \\ c_{i,j} &= 0 \text{ elsewhere} \end{aligned}$$

In order to reach a contradiction, we need to prove that the array $(c_{i,j})$ is in $C(p,l)$. Without loss of generality, it is enough to consider the case $U = S(p,l)$, the other one follows from being the complement of $S(p,l)$ in $GF(p) - \{0\}$. First we make the following observation: since U consist of all the powers of l modulo p the function $i \rightarrow li$ is closed in U and is 1-1. Next we show that all the lines of slopes 1 and l have parity 0 and use Lemma 2.1 to complete the proof. Notice that the only nonzero entries in $(c_{i,j})$ are as defined by U in the first row and first column.
Namely, for every $0 \leq i \leq p-1$,

$$\sum_{j=0}^{p-1} c_{i-j,j} = 0$$

because the first row and first column are identical.
Moreover, for every $0 \leq i \leq p-1$,

$$\sum_{j=0}^{p-1} c_{i-lj,j} = 0,$$

since the function $j \rightarrow lj$ is closed in U and is 1-1. Hence, by Lemma 2.1 $(c_{i,j})$ is in $C(p,l)$.

$\Leftarrow$) Assume now that l is primitive in $GF(p)$. We have already proven in Lemma 2.3 that codes $\mathcal{C}(p,l)$ and $\mathcal{C}'(p,l)$ are MDS, thus, any pair of columns or of rows in an array in $C(p,l)$ can be uniquely retrieved. We need to show now that the same is true for any row and column.

Assume now that i is the only possible non-zero row and j the only possible non-zero column in an array in $C(p,l)$. We will show that row i and column j are also zero.

Let $(c_{s,t})_{0 \le s,t \le p-1}$ be an array in $C(p,l)$, and assume that $c_{s,t} = 0$ whenever $s \ne i$ and $t \ne j$. As usual, assume the last row is an imaginary 0-column as well as the last column. We will show that also $c_{i,t} = 0$ and $c_{s,j} = 0$.

Since $c_{p-1,j} = 0$, then $c_{i,j-(i+1)} = b$, since they belong in the same diagonal, and $b = 0$ or $b = 1$, according to the parity of the diagonals and the lines of slope l. Since $c_{i,j-(i+1)} = b$, then $c_{i-l(i+1),j} = 0$, since they belong in the same line of slope l. By induction, assume that $c_{i-l^{r-1}(i+1),j} = 0$ for $1 \le r \le p-2$. Then, $c_{i,j-l^{r-1}(i+1)} = b$, since $c_{i-l^{r-1}(i+1),j}$ and $c_{i,j-l^{r-1}(i+1)}$ belong in the same diagonal. This implies that $c_{i-l^r(i+1),j} = 0$, since $c_{i,j-l^{r-1}(i+1)}$ and $c_{i-l^r(i+1),j}$ belong in the same line of slope l. Therefore, $c_{i-l^r(i+1),j} = 0$ and $c_{i,j-l^r(i+1)} = b$ for $0 \le r \le p-2$. Since l is primitive in $GF(p)$, then there is an r such that $l^r = (j+1)/(i+1)$. For that r, $j - l^r(i+1) = p-1$, but $c_{i,p-1} = 0 = b$, thus, $c_{i,j-l^r(i+1)} = 0$ for $0 \le r \le p-2$. Again, using the fact that l is primitive in $GF(p)$, we conclude that $c_{s,j} = 0$ for $s \ne i$ and $c_{i,t} = 0$ for $t \ne j$. Finally, $c_{i,j} = 0$ since the diagonals and lines of slope l have even parity. This completes the proof. $\square$

In references [4][6][12], the authors prove that their construction can correct the rank of an array when the rank is used as a metric. We have seen that the rank is a more powerful metric than the criss-cross distance considered here. Therefore, a legitimate question is, can the codes $C(p,l)$ also correct the rank? The answer is no, in general. For instance, consider the following array in $C(7,3)$:

0	0	0	0	1	1	0
0	0	0	0	1	1	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	1	1	1	0
0	0	0	0	0	0	0

This array has criss-cross weight 3 but rank 2.

So, the question is, under which conditions the codes $C(p,l)$ can correct the rank? The answer is, whenever the polynomial $1 + x + \cdots + x^{p-1}$ is irreducible, i.e., when 2 is primitive in $GF(p)$, then the non-zero arrays in $\mathcal{C}(p,l)$ have rank at least 3. In this case, the ring of polynomials modulo $1+x+\cdots+x^{p-1}$ is a field. Explicitly,

Theorem 3.2 Every non-zero array in code $C(p,l)$, 2 and l primitive in $GF(p)$, has rank at least 3.

Proof: Let Γ be a non-zero array in $C(p,l)$ with rank 2, therefore, we can write

$$\Gamma = UD,$$

where $U=(u_{i,j})_{\substack{0\le i\le p-2\\0\le j\le 1}}$ is a $(p-1)\times 2$ array and $D=(d_{i,j})_{\substack{0\le i\le 1\\0\le j\le p-2}}$ is a $2\times(p-1)$ array, and both U and D have rank 2. Looking at each column of Γ as an element modulo $1+x+\cdots+x^{p-1}$, we obtain that the j-th column of Γ is given by

$$\sum_{i=0}^{p-2}\left(\sum_{t=0}^{1}u_{i,t}d_{t,j}\right)\alpha^i = \sum_{t=0}^{1}d_{t,j}\sum_{i=0}^{p-2}u_{i,t}\alpha^i.$$

Using the parity-check matrix $H(p,l)$ defined by (1), we obtain that

$$\begin{aligned}\sum_{j=0}^{p-2}\left(\sum_{t=0}^{1}d_{t,j}\sum_{i=0}^{p-2}u_{i,t}\alpha^i\right)\alpha^j &= 0\\ \sum_{j=0}^{p-2}\left(\sum_{t=0}^{1}d_{t,j}\sum_{i=0}^{p-2}u_{i,t}\alpha^i\right)\alpha^{lj} &= 0,\end{aligned}$$

which can be rearranged as

$$\sum_{t=0}^{1}\left(\sum_{j=0}^{p-2}d_{t,j}\alpha^j\right)\left(\sum_{i=0}^{p-2}u_{i,t}\alpha^i\right) = 0 \tag{6}$$

$$\sum_{t=0}^{1}\left(\sum_{j=0}^{p-2}d_{t,j}\alpha^{lj}\right)\left(\sum_{i=0}^{p-2}u_{i,t}\alpha^i\right) = 0. \tag{7}$$

Since 2 is primitive in $GF(p)$, then l is a power of 2, and thus

$$\sum_{j=0}^{p-2}d_{t,j}\alpha^{lj} = \left(\sum_{j=0}^{p-2}d_{t,j}\alpha^j\right)^l,$$

so (6) and (7) can be written as

$$
\begin{aligned}
\left(\sum_{j=0}^{p-2} d_{0,j}\alpha^j\right)\left(\sum_{i=0}^{p-2} u_{i,0}\alpha^i\right)+\left(\sum_{j=0}^{p-2} d_{1,j}\alpha^j\right)\left(\sum_{i=0}^{p-2} u_{i,1}\alpha^i\right) &= 0 \qquad (8)\\
\left(\sum_{j=0}^{p-2} d_{0,j}\alpha^j\right)^l\left(\sum_{i=0}^{p-2} u_{i,0}\alpha^i\right)+\left(\sum_{j=0}^{p-2} d_{1,j}\alpha^j\right)^l\left(\sum_{i=0}^{p-2} u_{i,1}\alpha^i\right) &= 0. \qquad (9)
\end{aligned}
$$

Let $D_t = \sum_{j=0}^{p-2} d_{t,j}\alpha^j$ and $U_t = \sum_{i=0}^{p-2} u_{i,t}\alpha^i$, then (8) and (9) become

$$
\begin{aligned}
D_0U_0 + D_1U_1 &= 0\\
D_0^lU_0 + D_1^lU_1 &= 0.
\end{aligned}
$$

Since we are in a field, the system above has a non-trivial solution if and only if

$$
\det\begin{pmatrix} D_0 & D_1\\ D_0^l & D_1^l \end{pmatrix} = 0,
$$

and, since $D_0 \neq 0$ and $D_1 \neq 0$, this can only occur if $D_0^{l-1} + D_1^{l-1} = 0 = (D_0 + D_1)^{l-1}$. Thus, $D_0 = D_1$, a contradiction, since we are assuming that matrix D has rank 2. $\square$

We have not found an adequate generalization of the codes $C(p, l)$ to more than two parities. It is an open problem if such a generalization exists. However, if $l = 2$ and 2 is primitive in $GF(p)$, then the code with r parities defined by the parity-check matrix

$$
H = \begin{pmatrix}
1 & \alpha & \alpha^2 & \dots & \alpha^{p-2}\\
1 & \alpha^2 & \alpha^4 & \dots & \alpha^{(p-2)2}\\
1 & \alpha^4 & \alpha^8 & \dots & \alpha^{(p-2)4}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
1 & \alpha^{2^{r-1}} & \alpha^{2(2^{r-1})} & \dots & \alpha^{(p-2)2^{r-1}}
\end{pmatrix}
$$

is MDS with respect to the rank. This code is a particular case of the ones described in [4][6][12].

4 Encoding and Decoding

In this section we give encoding and decoding algorithms for errors and erasures. If there are no indications of erased rows or columns in a received array, the decoder attempts to correct either a column or a row. In the case of erasures, the decoder can correct either two

erased columns, two erased rows, or an erased column together with an erased row. The encoding is a particular case of the decoding of two erased columns. We examine all these cases separately. Let us start with errors.
Assume that $(r_{i,j})$ is a received array, possibly a noisy version of an originally stored array $(c_{i,j}) \in C(p,l)$. Moreover, assume that either a column or a row in $(r_{i,j})$ are in error. The first step is finding the column syndromes using the parity-check matrix $H(p,l)$ given by (1). To this end, we define $r(\alpha) = (r_0(\alpha), r_1(\alpha), \ldots, r_{p-2}(\alpha))$ as $r_j(\alpha) = \sum_{i=0}^{p-2} r_{i,j}\alpha^i$. Estimating $r(\alpha)(H(p,l)^T$, we obtain

$$\sum_{j=0}^{p-2} r_j(\alpha)\alpha^j = S_1(\alpha) \tag{10}$$

$$\sum_{j=0}^{p-2} r_j(\alpha)\alpha^{jl} = S_l(\alpha) \tag{11}$$

Let us point out that multiplying by a power α^t a vector is equivalent to rotating the vector t times modulo $1 + x + \cdots + x^{p-1}$. For details, we refer the reader to [3]. If there was an error E in, say, column t, and all the other columns are correct, (10) and (11) give

$$E\alpha^t = S_1(\alpha) \tag{12}$$

$$E\alpha^{tl} = S_l(\alpha) \tag{13}$$

We need to find the error-location t and the error itself E. Solving (12) and (13), we obtain

$$\alpha^{(l-1)t} S_1(\alpha) = S_l(\alpha) \tag{14}$$

So, the decoder applies repeatedly the operation $\alpha^{(l-1)j} S_1(\alpha)$ for $0 \le j \le p-2$ until it finds a $j = t$ satisfying (14). If there is such a t, then the decoder declares an error in column t, and the value E of the error, from (12), is given by $E = \alpha^{-t} S_1(\alpha)$. The final step is adding E to column t, completing the decoding.
However, if there is no t satisfying (14), the decoder will assume that there was a row error, and will repeat the procedure but this time for rows. Specifically, the decoder now considers $r'(\alpha) = (r'_0(\alpha), r'_1(\alpha), \ldots, r'_{p-2}(\alpha))$ as $r'_i(\alpha) = \sum_{j=0}^{p-2} r_{i,j}\alpha^j$. Estimating $r'(\alpha)(H(p,1/l)^T$, we obtain the row syndromes

$$\sum_{i=0}^{p-2} r'_i(\alpha)\alpha^i = S'_1(\alpha) \tag{15}$$

$$\sum_{i=0}^{p-2} r'_i(\alpha)\alpha^{i/l} = S'_l(\alpha) \tag{16}$$

Notice that

$$S_1'(\alpha)=\sum_{i=0}^{p-2} r_i'(\alpha)\alpha^i=\sum_{i=0}^{p-2}(\sum_{j=0}^{p-2} r_{i,j}\alpha^j)\alpha^i=\sum_{j=0}^{p-2}(\sum_{i=0}^{p-2} r_{i,j}\alpha^i)\alpha^j=\sum_{j=0}^{p-2} r_j(\alpha)\alpha^j=S_1(\alpha),$$

so $S_1'(\alpha)$ does not need to be calculated once $S_1(\alpha)$ is known. If there was an error E' in, say, row s, and all the other rows are correct, (15) and (16) give

$$\begin{aligned} E'\alpha^s &= S_1'(\alpha) \quad (17)\\ E'\alpha^{s/l} &= S_l'(\alpha) \quad (18)\end{aligned}$$

Solving (17) and (18), we obtain

$$\alpha^{((1/l)-1)s}S_1'(\alpha) \;=\; S_l'(\alpha) \quad (19)$$

Now, the decoder applies repeatedly the operation $\alpha^{((1/l)-1)j}S_1'(\alpha)$ for $0\le j\le p-2$ until it finds a $j=s$ satisfying (19). If there is such an s, then the decoder declares an error in row s, and the value E' of the error, from (12), is given by $E'=\alpha^{-s}S_1'(\alpha)$. The final step is adding E' to row s, completing the decoding. If there is no s satisfying (19), then the decoder declares an uncorrectable error.
Let us illustrate the decoding procedure with an example.

Example 4.1 Consider code $C(7,3)$ and assume that the following array is received:

1	0	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	0	1	1	0
0	1	0	1	0	0	0
0	1	0	0	1	1	0
0	0	0	1	0	1	0
0	0	0	0	0	0	0

The corresponding $r(\alpha)$ and $r'(\alpha)$ are given by

$$\begin{aligned} r(\alpha) &= (1,\alpha^2+\alpha^3+\alpha^4,\alpha,\alpha^3+\alpha^5,1+\alpha^2+\alpha^4,\alpha^2+\alpha^4+\alpha^5)\\ r'(\alpha) &= (1+\alpha^4,\alpha^2,\alpha+\alpha^4+\alpha^5,\alpha+\alpha^3,\alpha+\alpha^4+\alpha^5,\alpha^3+\alpha^5)\end{aligned}$$

Notice that $1/3=5$ modulo 7, so,

$$
\begin{aligned}
H(7,3) &= \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha \end{pmatrix} \quad \text{and} \\
H'(7,3)=H(7,5) &= \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ 1 & \alpha^5 & \alpha^3 & \alpha & \alpha^6 & \alpha^4 \end{pmatrix}
\end{aligned}
$$

The values $S_1(\alpha)$, $S_3(\alpha)$, $S_1'(\alpha)$ and $S_3'(\alpha)$ are given by $(S_1(\alpha), S_3(\alpha)) = r(\alpha)(H(7,3))^T$ and $(S_1'(\alpha), S_3'(\alpha)) = r'(\alpha)(H(7,5))^T$. Performing these operations, we obtain

$$
\begin{aligned}
S_1'(\alpha) = S_1(\alpha) &= \alpha^2 + \alpha^3 + \alpha^5 \\
S_3(\alpha) &= 1 + \alpha^2 + \alpha^3 \\
S_3'(\alpha) &= 1 + \alpha + \alpha^3
\end{aligned}
$$

First we check if there is a column error. Using (14), we have to check if there is a t such that $\alpha^{2t} S_1(\alpha) = S_3(\alpha)$. We can verify that there is no such t, so we concentrate next on rows. Using (19), we have to check if there is an s such that $\alpha^{4s} S_1'(\alpha) = S_3'(\alpha)$. We can verify that for $s = 3$,

$$\alpha^{12} S_1'(\alpha) = \alpha^5(\alpha^2 + \alpha^3 + \alpha^5) = 1 + \alpha + \alpha^3 = S_3'(\alpha),$$

so there is an error in row 3. From (12), this error is given by

$$E' = \alpha^{-s} S_1'(\alpha) = \alpha^4(\alpha^2 + \alpha^3 + \alpha^5) = 1 + \alpha^2 + \alpha^6 = \alpha + \alpha^3 + \alpha^4 + \alpha^5.$$

Adding this error value to location 3 of $r'(\alpha)$, we obtain

$$c(\alpha) = (1 + \alpha^4, \alpha^2, \alpha + \alpha^4 + \alpha^5, \alpha^4 + \alpha^5, \alpha + \alpha^4 + \alpha^5, \alpha^3 + \alpha^5)$$

Each of the entries of $c(\alpha)$ represents a row in the array, so the corrected array is given by

1	0	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	0	1	1	0
0	0	0	0	1	1	0
0	1	0	0	1	1	0
0	0	0	1	0	1	0
0	0	0	0	0	0	0

Let us write formally the algorithm described above:

Algorithm 4.1 (Decoding Algorithm for a Row or a Column in Error) Assume that $(r_{i,j})$ is a received array, possibly a noisy version of an originally stored array in $C(p,l)$, where l is primitive in $GF(p)$. Assume that either a column or a row in $(r_{i,j})$ are in error. Define $r(\alpha)=(r_0(\alpha),r_1(\alpha),\ldots,r_{p-2}(\alpha))$, where $r_j(\alpha)=\sum_{i=0}^{p-2} r_{i,j}\alpha^i$, and $r'(\alpha)=(r'_0(\alpha),r'_1(\alpha),\ldots,r'_{p-2}(\alpha))$, where $r'_i(\alpha)=\sum_{j=0}^{p-2} r_{i,j}\alpha^j$. Find $S_1(\alpha)=S'_1(\alpha)$ according to (10), $S_l(\alpha)$ according to (11) and $S'_l(\alpha)$ according to (16). Then:

If there is a t such that $\alpha^{(l-1)t}S_1(\alpha)=S_l(\alpha)$, then:
  Let $E=\alpha^{-t}S_1(\alpha)$, $c_j(\alpha)=r_j(\alpha)$ for $j\neq t$ and $c_t(\alpha)=r_t(\alpha)+E$, where $c_j(\alpha)=\sum_{i=0}^{p-2} c_{i,j}\alpha^i$, output $(c_{i,j})$ and stop.
Else, if there is no t such that $\alpha^{(l-1)t}S_1(\alpha)=S_l(\alpha)$, then:
  If there is an s such that $\alpha^{((1/l)-1)s}S'_1(\alpha)=S'_l(\alpha)$, then:
    Let $E'=\alpha^{-s}S'_1(\alpha)$, $c'_i(\alpha)=r'_i(\alpha)$ for $i\neq s$ and $c'_s(\alpha)=r'_s(\alpha)+E'$, where $c'_j(\alpha)=\sum_{j=0}^{p-2} c_{i,j}\alpha^j$, output $(c_{i,j})$ and stop.
  Else, if there is no s such that $\alpha^{(l-1)s}S'_1(\alpha)=S'_l(\alpha)$, then declare an uncorrectable error and stop.

Next we concentrate on erasures. First, assume that two erasures have occured in columns s and t, $0\leq s<t\leq p-2$. In order to compute the column syndromes according to (10) and (11), we assume that $r_s(\alpha)=r_t(\alpha)=0$. Then, we have to find the missing elements E_s and E_t. In this case, (10) and (11) give

$$\begin{aligned} E_s\alpha^s+E_t\alpha^t &= S_1(\alpha)\\ E_s\alpha^{ls}+E_t\alpha^{lt} &= S_l(\alpha)\end{aligned}$$

Solving the linear system above, we obtain

$$(\alpha^{(l-1)(t-s)}+1)E_s = \alpha^{l(t-s)-t}S_1(\alpha)+\alpha^{-sl}S_l(\alpha) \tag{20}$$
$$(\alpha^{(l-1)(t-s)}+1)E_t = \alpha^{-t}S_1(\alpha)+\alpha^{-(l-1)s-t}S_l(\alpha) \tag{21}$$

Solving efficiently recursions of the type $(\alpha^j+1)A=B$ modulo $1+x+\cdots+x^{p-1}$ was done in detail in [3]. Let us illustrate the case of two erased columns with an example.

Example 4.2 As in Example 4.2, consider code $C(7,3)$ and assume that the following array is received (the "?" signs denote erased bits):

1	0	?	0	?	0	0
1	1	?	0	?	1	0
0	1	?	0	?	1	0
0	1	?	0	?	1	0
0	1	?	0	?	0	0
1	0	?	1	?	1	0
0	0	0	0	0	0	0

Therefore, columns 2 and 4 have been erased. The received vector $r(\alpha)$ can be written as

$$r(\alpha)=(1+\alpha+\alpha^5, \alpha+\alpha^2+\alpha^3+\alpha^4, 0, \alpha^5, 0, \alpha+\alpha^2+\alpha^3+\alpha^5).$$

Performing $(S_1(\alpha), S_3(\alpha))=r(\alpha)(H(7,3))^T$ as in Example 4.1, we obtain

$$\begin{aligned} S_1(\alpha) &= 1+\alpha^3+\alpha^5 \\ S_3(\alpha) &= 1+\alpha+\alpha^2+\alpha^3 \end{aligned}$$

Applying (20) and (21) with $l=3$, $s=2$, $t=4$ and the values of $S_1(\alpha)$ and $S_3(\alpha)$ above, we obtain

$$\begin{aligned} (\alpha^4+1)E_2 &= 1+\alpha+\alpha^2+\alpha^4+\alpha^5 \\ (\alpha^4+1)E_4 &= 1+\alpha^2+\alpha^3 \end{aligned}$$

For the sake of completeness, let us solve the recursion $(\alpha^4+1)E_4=1+\alpha^2+\alpha^3$. For details of the method, see [3]. Let $E_4=x_0+x_1\alpha+x_2\alpha^2+x_3\alpha^3+x_4\alpha^4+x_5\alpha^5$. Then,

$$\begin{aligned} (1+\alpha^4)E_4 &= (x_3+x_2+x_0)+(x_4+x_2+x_1)\alpha+x_5\alpha^2+(x_2+x_3)\alpha^3+(x_0+x_2+x_4)\alpha^4 \\ &\quad +(x_1+x_2+x_5)\alpha^5 \\ &= 1+\alpha^2+\alpha^3. \end{aligned}$$

Solving this system recursively, we obtain

$$\begin{aligned} x_5 &= 1 \\ x_1+x_2 &= 1 \\ x_4 &= 1 \\ x_0+x_2 &= 1 \\ x_3 &= 0 \\ x_2 &= 1 \\ x_1 &= 0 \\ x_0 &= 0, \end{aligned}$$

therefore,

$$E_4=\alpha^2+\alpha^4+\alpha^5.$$

Similarly, if $E_2=x_0+x_1\alpha+x_2\alpha^2+x_3\alpha^3+x_4\alpha^4+x_5\alpha^5$, we have to solve the recursion

$$\begin{aligned}(1+\alpha^4)E_2 &= (x_3+x_2+x_0)+(x_4+x_2+x_1)\alpha+x_5\alpha^2+(x_2+x_3)\alpha^3+(x_0+x_2+x_4)\alpha^4 \\ &\quad +(x_1+x_2+x_5)\alpha^5 \\ &= 1+\alpha+\alpha^2+\alpha^4+\alpha^5.\end{aligned}$$

Proceeding like in the previous case, this gives

$$E_2=\alpha^2.$$

Therefore, the decoded array is

1	0	0	0	0	0	0
1	1	0	0	0	1	0
0	1	1	0	1	1	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
1	0	0	1	1	1	0
0	0	0	0	0	0	0

Notice that the lines of slope 1 and 3 have even parity.

Let us point out once more that the encoding is a particular case of the erasure decoding described above: we choose two columns for the redundancy, say, columns $p-3$ and $p-2$ (the last two columns in the array), and using the information in the first $p-3$ columns, we reconstruct the two redundant columns.
Let us write down formally the algorithm for decoding of two erased columns.

Algorithm 4.2 (Decoding Algorithm for Two Erased Columns) Assume that $(r_{i,j})$ is a received array from an originally stored array in $C(p,l)$, where two columns, $0\le s<t\le p-2$, have been erased. Define $r(\alpha)=(r_0(\alpha),r_1(\alpha),\ldots,r_{p-2}(\alpha))$, where $r_j(\alpha)=\sum_{i=0}^{p-2} r_{i,j}\alpha^i$ for $j\neq s,t$, and $r_s(\alpha)=r_t(\alpha)=0$. Find $S_1(\alpha)$ according to (10) and $S_l(\alpha)$ according to (11). Then, let E_s be the solution of the recursion given by (20) and E_t the solution of the recursion given by (21). Define $c_j(\alpha)=r_j(\alpha)$ for $j\neq s,t$, $c_s(\alpha)=E_s$ and $c_t(\alpha)=E_t$, where $c_j(\alpha)=\sum_{i=0}^{p-2} c_{i,j}\alpha^i$, output $(c_{i,j})$ and stop.

If two rows are erased, the procedure is analogous, except that we have to consider now the syndromes $S'_1(\alpha)$ and $S'_l(\alpha)$ and replace l by $1/l$. Formally,

Algorithm 4.3 (Decoding Algorithm for Two Erased Rows) Assume that $(r_{i,j})$ is a received array from an originally stored array in $C(p,l)$, where two rows, $0\le s<t\le p-2$, have been erased. Define $r'(\alpha)=(r'_0(\alpha),r'_1(\alpha),\ldots,r'_{p-2}(\alpha))$, where $r'_j(\alpha)=\sum_{j=0}^{p-2} r_{i,j}\alpha^i$ for $j\neq s,t$, and $r'_s(\alpha)=r'_t(\alpha)=0$. Find $S'_1(\alpha)$ according to (15) and $S_l(\alpha)$ according to (16). Then, let E'_s and E'_t be the solution of the following recursions:

$$
\begin{aligned}
(\alpha^{((1/l)-1)(t-s)}+1)E'_s &= \alpha^{(1/l)(t-s)-t}S'_1(\alpha)+\alpha^{-s(1/l)}S'_l(\alpha)\\
(\alpha^{((1/l)-1)(t-s)}+1)E'_t &= \alpha^{-t}S'_1(\alpha)+\alpha^{-((1/l)-1)s-t}S'_l(\alpha)
\end{aligned}
$$

Define $c'_i(\alpha)=r'_j(\alpha)$ for $i\neq s,t$, $c'_s(\alpha)=E'_s$ and $c'_t(\alpha)=E'_t$, where $c'_j(\alpha)=\sum_{j=0}^{p-2}c_{i,j}\alpha^i$, output $(c_{i,j})$ and stop.

The last case we need to consider in order to complete the decoding of two erasures, is the case in which a row and a column have been erased. In this case we need to assume that l is primitive in $GF(p)$, an assumption that was not necessary in the decoding of two erased columns or two erased rows.
Assume then that $(r_{i,j})$ is a received array where row s and column t have been erased, $0\leq s,t\leq p-2$. We want to find the values $r_{s,j}$ and $r_{i,t}$. As usual, we assume initially that those values are 0 in order to calculate the syndromes, and we also assume that a 0-row and a 0-column have been added to the array.
From (10) and (11), let us start by estimating, for each $0\leq i\leq p-1$, the p syndromes of slope 1 and l respectively as follows:

$$
\begin{aligned}
S_i^{(1)} &= \sum_{j=0}^{p-1} r_{i-j,j} && (22)\\
S_i^{(l)} &= \sum_{j=0}^{p-1} r_{i-lj,j}. && (23)
\end{aligned}
$$

The reader may ask, what is the relationship between these syndromes, and the column syndromes given by (10) and (11)? Proceeding like in Lemma 2.1, we easily find out that $S_1(\alpha)=\sum_{i=0}^{p-2}\left(S_i^{(1)}+S_{p-1}^{(1)}\right)\alpha^i$ and $S_l(\alpha)=\sum_{i=0}^{p-2}\left(S_i^{(l)}+S_{p-1}^{(l)}\right)\alpha^i$. Therefore, (22) and (23) provide a computationally efficient method to find (10) and (11).
Let b the unknown parity of lines of slope 1 and l. Proceeding inductively like in the "if" part of the proof of Theorem 3.1, for each $0\leq j$, we obtain the following recursion:

$$
\begin{aligned}
r_{s,t-l^j(s+1)} &= S^{(1)}_{s+t-l^j(s+1)}+r_{s-l^j(s+1),t}+b && (24)\\
r_{s-l^{j+1}(s+1),t} &= S^{(l)}_{s+lt-l^{j+1}(s+1)}+r_{s,t-l^j(s+1)}+b && (25)
\end{aligned}
$$

Therefore, applying the recursion repeatedly, for each $0\leq j\leq p-2$, we obtain

$$
r_{s,t-l^i(s+1)} = \left(\sum_{j=0}^{i}S^{(1)}_{s+t-l^j(s+1)}\right)+\left(\sum_{j=1}^{i}S^{(l)}_{s+lt-l^j(s+1)}\right)+b \qquad (26)
$$

In particular, let i be the unique value in $GF(p)$ such that

$$l^i = \frac{t+1}{s+1}. \tag{27}$$

Since l is primitive in $GF(p)$, we know that there exists such an i. Replacing (27) in (26), since $r_{s,t-1}=0$, b is given by

$$b = \left(\sum_{j=0}^{i} S^{(1)}_{s+t-l^j(s+1)}\right) + \left(\sum_{j=1}^{i} S^{(l)}_{s+lt-l^j(s+1)}\right) \tag{28}$$

Thus, the recursion given by (24) and (25) provides the solution, since b is now known. Finally,

$$r_{s,t} = S^{(1)}_{s+t} + b \tag{29}$$

completes the decoding. Let us illustrate the procedure with an example.

Example 4.3 As in Example 4.2, consider again $C(7,3)$. Assume that we receive the following array

1	0	?	0	0	0	0
1	1	?	0	0	1	0
0	1	?	0	1	1	0
0	1	?	0	0	1	0
?	?	?	?	?	?	0
1	0	?	1	1	1	0
0	0	0	0	0	0	0

in which row $s=4$ and column $t=2$ have been erased. The first step is finding the syndromes according to (22) and (23). Taking as zero the symbols denoted by ?, this gives

$$\begin{array}{ll}
S_0^{(1)} = 0 & S_0^{(3)} = 1 \\
S_1^{(1)} = 1 & S_1^{(3)} = 1 \\
S_2^{(1)} = 0 & S_2^{(3)} = 1 \\
S_3^{(1)} = 0 & S_3^{(3)} = 0 \\
S_4^{(1)} = 1 & S_4^{(3)} = 0 \\
S_5^{(1)} = 1 & S_5^{(3)} = 0 \\
S_6^{(1)} = 0 & S_6^{(3)} = 0
\end{array}$$

According to (27),

$$l^i = 3^i = (t+1)/(s+1) = 3/5 = 2,$$

since $1/5 = 3$ in $GF(7)$. Solving for $3^i = 2$, we conclude that $i = 2$. From (28), replacing $l = 3$, $i = 2$, $s = 4$ and $t = 2$, we obtain

$$b = (S_1^{(1)} + S_5^{(1)} + S_3^{(1)}) + (S_2^{(3)} + S_0^{(3)}) = 1 + 1 + 0 + 1 + 1 = 0.$$

We are ready now for the final recursion given by (29), (24) and (25):

$$\begin{aligned}
r_{4,2} &= S_6^{(1)} = 0 \\
r_{4,4} &= S_1^{(1)} + r_{6,2} = 1 \quad (j = 0) \\
r_{3,2} &= S_2^{(3)} + r_{4,4} = 0 \quad (j = 0) \\
r_{4,1} &= S_5^{(1)} + r_{3,2} = 1 \quad (j = 1) \\
r_{1,2} &= S_0^{(3)} + r_{4,1} = 0 \quad (j = 1) \\
r_{4,6} &= S_3^{(1)} + r_{1,2} = 0 \quad (j = 2) \\
r_{2,2} &= S_1^{(3)} + r_{4,6} = 1 \quad (j = 2) \\
r_{4,0} &= S_4^{(1)} + r_{2,2} = 0 \quad (j = 3) \\
r_{5,2} &= S_4^{(3)} + r_{4,0} = 0 \quad (j = 3) \\
r_{4,3} &= S_0^{(1)} + r_{5,2} = 0 \quad (j = 4) \\
r_{0,2} &= S_6^{(3)} + r_{4,3} = 0 \quad (j = 4) \\
r_{4,5} &= S_2^{(1)} + r_{0,2} = 0 \quad (j = 5) \\
r_{6,2} &= S_5^{(3)} + r_{4,5} = 0 \quad (j = 5)
\end{aligned}$$

Notice that, for $j = 2$, we already know that $r_{4,6} = 0$, since this value is in the imaginary 7th 0-column that has been added. However, this fact was exploited in the calculation of the parity bit b given by (28). The final decoded array is thus given by

1	0	0	0	0	0	0
1	1	0	0	0	1	0
0	1	1	0	1	1	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
1	0	0	1	1	1	0
0	0	0	0	0	0	0

which coincides with the one in Example 4.2.

Let us end this section by writing formally the decoding algorithm for an erased row together with an erased column.

Algorithm 4.4 (Decoding Algorithm for an Erased Row and an Erased Column) Assume that $(r_{i,j})$ is a received array from an originally stored array in $C(p,l)$, l primitive in $GF(p)$, where row s and column t, $0 \leq s,t \leq p-2$, have been erased. For each $0 \leq i \leq p-1$, find the syndromes given by (22) and (23). Then, if i is such that $l^i = (t+1)/(s+1)$, determine b according to (28). Finally, find $r_{s,t}$ according to (29) and the rest of the values according to the recursion given by (24) and (25).

5 Conclusions

We presented a family of $(p-1) \times (p-1)$ array codes, p a prime, that can correct any row or any column in error. The construction is based on taking all the arrays with even or odd parity along lines of slope 1 and of slope l, l primitive in $GF(p)$. The new codes differ from previously known codes in literature that can correct the rank of an array. In that sense, our codes are weaker, but they have less complexity, since they are based on simple XOR operations. Although we presented our results for binary codes, they may trivially be extended to any field.

References

[1] M. Blaum, P.G. Farrell, *Array Codes for Cluster-Error Correction, Electronic Letters,* Vol. 30, No. 21 (1994), 1752–3.

[2] M. Blaum, R.J. McEliece, *Coding protection for magnetic tapes: a generalization of the Patel-Hong code, IEEE Trans. Inform. Theory,* IT-31 (1985), 690–693.

[3] M. Blaum, R.M. Roth, *New Array Codes for Multiple Phased Burst Correction, IEEE Trans. Inform. Theory,* IT-39 (1993), 66–77.

[4] P. Delsarte, *Bilinear Forms over a Finite Field, with Applications to Coding Theory, J. Comb. Th. Series A,* 25 (1978), 226–241.

[5] S.A. Elkind, D.P. Siewiorek, *Reliability and performance of error-correcting memory and register codes, IEEE Trans. Computers,* C-29 (1980), 920–927.

[6] E.M. Gabidulin, *Theory of codes with maximum rank distance, Prob. Info. Trans.,* Vol. 21, No. 1, (1985), 3–16.

[7] M. Hall Jr, *Combinatorial Theory,* Blaisdell Publishing Company, Waltham Ma, 1967.

[8] L. Levine, W. Meyers, *Semiconductor memory reliability with error detecting and correcting codes, Computer,* 9 (Oct. 1976), 43–50.

[9] W.F. Mikhail, R.W. Bartoldus, R.A. Rutledge, *The reliability of memory with single-error correction, IEEE Trans. Computers,* C-31 (1983), 560–564.

[10] A.M. Patel, S.J. Hong, *Optimal rectangular code for high density magnetic tape, IBM J. Res. Develop.,* 18 (1974), 579–588.

[11] P. Prusinkiewicz, S. Budkowski, *A double-track error-correction code for magnetic tape, IEEE Trans. on Computers,* C-19 (1976), 642–645.

[12] R.M. Roth, *Maximum rank array codes and their application to crisscross error correction, IEEE Trans. Inform. Theory,* IT-37 (1991), 328–336.