RJ10521 (ALM1406-009) June 19, 2014
Computer Science

IBM Research Report

Generalized Concatenated Types of Codes for Erasure Correction

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

Research Division
Almaden – Austin – Beijing – Cambridge – Dublin - Haifa – India – Melbourne - T.J. Watson – Tokyo - Zurich

Generalized Concatenated Types of Codes for Erasure Correction

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

June 9, 2014

Abstract

Generalized Concatenated (GC), also known as Integrated Interleaved (II) Codes, are studied from an erasure correction point of view making them useful for Redundant Arrays of Independent Disks (RAID) types of architectures combining global and local properties. The fundamental erasure-correcting properties of the codes are proven and efficient encoding and decoding algorithms are provided. Although less powerful than the recently developed PMDS codes, this implementation has the advantage of allowing generalization to any range of parameters while the size of the field is much smaller than the one required for PMDS codes.

Keywords: Error-correcting codes, Reed-Solomon codes, Generalized Concatenated codes, Integrated Interleaved Codes, Maximally Recoverable codes, MDS codes, PMDS codes, Redundant Arrays of Independent Disks (RAID), local and global parities, heavy parities.

1 Introduction

Considerable interest has arisen lately in coding schemes that combine local and global properties. Applications like Redundant Arrays of Independent Disks (RAID) architectures [2][11][12][15] are an example of this interest. In effect, given an array of disks, a regular RAID architecture like, say, RAID 5, protects against a catastrophic disk (or, more in general, a storage device) failure. This is simply done by XORing the data devices in order to obtain a parity device (in this paper, we do not distinguish between RAID 4 and RAID 5, since this distinction is not relevant to our discussion). Then, if a storage device fails, its contents can be recovered by XORing the surviving devices.

A problem with this approach is that there may be individual sectors in the surviving devices that have failed (what is known as silent failures), a problem that is common in

Solid State Devices (SSDs), that decline as a function of time and of usage. In that case, one individual sector that has failed will cause data loss in the presence of a catastrophic device failure.

A method around this situation is using RAID 6: adding a second parity device allows for correction of most individual sector failures in the presence of a catastrophic device failure. The drawback of this approach is that it is wasteful: if for example a few extra sectors need to be recovered in addition to all the sectors corresponding to the failed device, it is desirable to optimize the redundancy necessary for doing so.

Codes dealing with this problem are the Partial MDS (PMDS) codes [1][2][4][5][8][11] (in [8][11], PMDS codes are called Maximally Recoverable codes), sector-disk (SD) codes [14][15], Locally Recoverable Codes (LRC) [17] and STAIR codes [12].

In general, we consider an $m \times n$ array. The parameter n represents the number of devices and m represents the size of a stripe: m is repeated a number of times throughout the array and each $m \times n$ stripe is decoded independently of the others.

The codes to be described in this paper are weaker than those in [2][8][11], in the sense that there are some erasure patterns that they cannot correct for the same amount of redundancy. However, they can be generalized to any set of parameters and, more importantly, they are simpler to implement, since they require a finite field $GF(2^b)$ of size $2^b > n$, the length of the rows, while the codes in [11] require size $2^b \geq mn$, the total length of the array (and the known constructions require much larger fields [2][8][11]). Similar considerations inspired the recent STAIR codes [12]. In [16], different combinations of local and global failures, involving either erasures and errors, are corrected using probabilistic methods by exploiting the rank of the error arrays. In [17], the data is encoded using a global RS code, and it is divided into parity groups that are independently encoded from the RS symbols. The Zigzag codes [18] keep the MDS property and optimize the minimum number of updates in the presence of one failure, but the parameter m is exponential on the number of devices n. In [7], a new probabilistic method is studied for decoding arrays using two-dimensional LDPC codes.

In order to illustrate our discussion, consider a (1,2) PMDS code over 4×5 arrays [1]. The code can correct an erasure in each row, and in addition two extra erasures anywhere. Below are two examples of erasure-patterns that can be corrected, where the erasures are indicated by X:

X				
	X			X
				X
		X		X

X				
	X		X	X
				X
				X

The array on the left has two rows with two erasures each, while the array on the right has a row with three erasures. The remaining rows have one erasure each, that is corrected by a horizontal parity-check code. The PMDS codes dealing with these type of errors, as presented in [1], require a field of size at least $2mn$ (these codes were extended in [5]). The codes to be presented will require a field of size at least $n+1$ only, one more than the length of the rows, but will correct, in this example, either the arrays on the left, or those on the right, but not both simultaneously (or, they can correct both simultaneously by using more redundancy). However, the codes can be extended to any set of parameters.

Actually, codes having the desired characteristics were created for a different application. Those are the so called Generalized Concatenated (GC) codes [6][21]. GC codes were presented in a form more suitable for implementation by the so called Integrated Interleaved (II) codes [10][19]. Here we want to adapt an II type of approach as an erasure-correcting code to deal with the problem of local and global parities. Some of the uses of GC codes for erasure-correction in RAID type of architectures were presented in [3]. Our description of the codes is based on their parity-check matrices.

In the next section we give the formal definition of the codes, we illustrate them with several examples and then we prove their basic property in Theorem 2.1. In Section 3 we present efficient encoding and decoding algorithms that are based on a divide and conquer approach: at each step an individual Reed-Solomon (RS) code [13] of length n is decoded for erasures, starting by the rows of the array having the less erasures. The procedure is much faster than by solving at once all the erasures using a linear system of equations based on the parity-check matrix. In Section 4 we discuss extending the codes presented in order to adapt them to different applications. We end the paper by drawing some conclusions.

2 Generalized Concatenated (GC) Codes as Erasure-Correcting Codes

The GC codes that we describe in this section are $m \times n$ array codes with symbols in a finite field GF(2^b), where $2^b > n$. In fact, the codes can be described over any finite field of characteristic p, p a prime number, but we keep $p=2$ for simplicity and because it is the case more relevant in applications. Reading the symbols horizontally in a row-wise manner gives a code of length mn. We will describe the GC codes by providing their parity-check matrices. We will then give the erasure-correcting capability of the codes by referring to erasures per row. We will use interchangeably the array and the row-wise vector structure of the code throughout the paper.

Denote by I_m the $m \times m$ identity matrix and by $A \otimes B$ the Kronecker product [20] of matrices A and B. Next we give a formal definition of t-level GC codes.

Definition 2.1 Let $m \leq n$ be integers, and $\alpha \in GF(2^b)$ an element of order $\mathcal{O}(\alpha) \geq n$ (if α is primitive, $\mathcal{O}(\alpha) = 2^b - 1$). Consider the matrices

$$H(u,n;\ell) = \begin{pmatrix} \alpha^{(n-1)\ell} & \alpha^{(n-2)\ell} & \dots & \alpha^{2\ell} & \alpha^{\ell} & 1 \\ \alpha^{(n-1)(\ell+1)} & \alpha^{(n-2)(\ell+1)} & \dots & \alpha^{2(\ell+1)} & \alpha^{\ell+1} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \\ \alpha^{(n-1)(\ell+u-1)} & \alpha^{(n-2)(\ell+u-1)} & \dots & \alpha^{2(\ell+u-1)} & \alpha^{\ell+u-1} & 1 \end{pmatrix} \quad (1)$$

and

$$\hat{H}(s,m;\ell) = \begin{pmatrix} 1 & \alpha^{-\ell} & \alpha^{-2\ell} & \ldots & \alpha^{-(m-2)\ell} & \alpha^{-(m-1)\ell} \\ 1 & \alpha^{-(\ell+1)} & \alpha^{-2(\ell+1)} & \ldots & \alpha^{-(m-2)(\ell+1)} & \alpha^{-(m-1)(\ell+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{-(\ell+s-1)} & \alpha^{-2(\ell+s-1)} & \ldots & \alpha^{-(m-2)(\ell+s-1)} & \alpha^{-(m-1)(\ell+s-1)} \end{pmatrix}. \quad (2)$$

Let $\underline{u}$ be a vector of non-decreasing integers and length $m = s_0 + s_1 + \cdots + s_{t-1}$ as follows:

$$\underline{u} = \left(\overbrace{u_0, u_0, \ldots, u_0}^{s_0}, \overbrace{u_1, u_1, \ldots, u_1}^{s_1}, \ldots, \overbrace{u_{t-1}, u_{t-1}, \ldots, u_{t-1}}^{s_{t-1}}\right), \quad (3)$$

where $t \geq 1$, $s_i \geq 1$ for $0 \leq i \leq t-1$ and $1 \leq u_0 < u_1 < \ldots < u_{t-1} \leq n-1$. Let $\hat{s}_i = \sum_{j=i}^{t-1} s_j$, $0 \leq i \leq t-1$ (notice that $m = \hat{s}_0$). We say that the $[mn, mn - \sum_{i=0}^{t-1} u_i s_i]$ code $\mathcal{C}(n;\underline{u})$ whose parity-check matrix is given by the $\left(\sum_{i=0}^{t-1} u_i s_i\right) \times mn$ matrix

$$\mathbf{H}(n;\underline{u}) = \begin{pmatrix} I_m & \otimes & H(u_0, n; 0) \\ \hat{H}(s_{t-1}, m; 0) & \otimes & H(u_{t-1} - u_0, n; u_0) \\ \hat{H}(s_{t-2}, m; \hat{s}_{t-1}) & \otimes & H(u_{t-2} - u_0, n; u_0) \\ \hat{H}(s_{t-3}, m; \hat{s}_{t-2}) & \otimes & H(u_{t-3} - u_0, n; u_0) \\ & \vdots & \\ \hat{H}(s_1, m; \hat{s}_2) & \otimes & H(u_1 - u_0, n; u_0) \end{pmatrix} \quad (4)$$

is a t-level GC code.

It would remain to be proven that the $\sum_{i=0}^{t-1} u_i s_i$ rows of matrix $\mathbf{H}(n;\underline{u})$ are linearly independent, but this will arise as a consequence of Theorem 2.1 to be stated below.

Although (4) provides for a compact description of the parity-check matrix $\mathbf{H}(n;\underline{u})$, it is not easy to visualize. Below we give a more explicit form of (4). Let $H_0 = H(u_0, n; 0)$ and

$H_j = H(u_j - u_0, n; u_0)$ as given by (1) for $1 \leq j \leq t-1$. Then,

$$\mathbf{H}(n;\underline{u}) = \left(\begin{array}{c|c|c|c}
H_0 & \underline{0} & \cdots & \underline{0} \\
\underline{0} & H_0 & \cdots & \underline{0} \\
\vdots & \vdots & \ddots & \vdots \\
\underline{0} & \underline{0} & \cdots & H_0 \\
\hline
H_{t-1} & H_{t-1} & \cdots & H_{t-1} \\
H_{t-1} & \alpha^{-1}H_{t-1} & \cdots & \alpha^{-(m-1)}H_{t-1} \\
\vdots & \vdots & \ddots & \vdots \\
H_{t-1} & \alpha^{-(\hat{s}_{t-1}-1)}H_{t-1} & \cdots & \alpha^{-(m-1)(\hat{s}_{t-1}-1)}H_{t-1} \\
\hline
H_{t-2} & \alpha^{-\hat{s}_{t-1}}H_{t-2} & \cdots & \alpha^{-(m-1)\hat{s}_{t-1}}H_{t-2} \\
H_{t-2} & \alpha^{-(\hat{s}_{t-1}+1)}H_{t-2} & \cdots & \alpha^{-(m-1)(\hat{s}_{t-1}+1)}H_{t-2} \\
\vdots & \vdots & \ddots & \vdots \\
H_{t-2} & \alpha^{-(\hat{s}_{t-2}-1)}H_{t-2} & \cdots & \alpha^{-(m-1)(\hat{s}_{t-2}-1)}H_{t-2} \\
\hline
\vdots & \vdots & \ddots & \vdots \\
\hline
H_i & \alpha^{-\hat{s}_{i+1}}H_i & \cdots & \alpha^{-(m-1)\hat{s}_{i+1}}H_i \\
H_i & \alpha^{-(\hat{s}_{i+1}+1)}H_i & \cdots & \alpha^{-(m-1)(\hat{s}_{i+1}+1)}H_i \\
\vdots & \vdots & \ddots & \vdots \\
H_i & \alpha^{-(\hat{s}_i-1)}H_i & \cdots & \alpha^{-(m-1)(\hat{s}_i-1)}H_i \\
\hline
\vdots & \vdots & \ddots & \vdots \\
\hline
H_1 & \alpha^{-\hat{s}_2}H_1 & \cdots & \alpha^{-(m-1)\hat{s}_2}H_1 \\
H_1 & \alpha^{-(\hat{s}_2+1)}H_1 & \cdots & \alpha^{-(m-1)(\hat{s}_2+1)}H_1 \\
\vdots & \vdots & \ddots & \vdots \\
H_1 & \alpha^{-(\hat{s}_1-1)}H_1 & \cdots & \alpha^{-(m-1)(\hat{s}_1-1)}H_1
\end{array}\right) \tag{5}$$

Let us illustrate the construction of $\mathbf{H}(n;\underline{u})$ with some examples.

Example 2.1 Assume $t=1$, i.e., $\underline{u} = \left(\overbrace{u_0, u_0, \ldots, u_0}^{s_0}\right)$ and $\mathcal{C}(n;\underline{u})$ is a 1-level GC code. Then, according to (4) and (5),

$$\begin{aligned}
\mathbf{H}(n;\underline{u}) &= \left(\; I_{s_0} \otimes H(u_0, n; 0) \;\right) \\
&= \left(\begin{array}{c|c|c|c}
H_0 & \underline{0} & \cdots & \underline{0} \\
\underline{0} & H_0 & \cdots & \underline{0} \\
\vdots & \vdots & \ddots & \vdots \\
\underline{0} & \underline{0} & \cdots & H_0
\end{array}\right).
\end{aligned}$$

This is a trivial case, since it corresponds to s_0 RS codewords of length n one after the other, each codeword having u_0 parity symbols.

Example 2.2 Assume $t=2$, i.e., $\underline{u}=\left(\overbrace{u_0,u_0,\ldots,u_0}^{s_0},\overbrace{u_1,u_1,\ldots,u_1}^{s_1}\right)$ and $\mathcal{C}(n;\underline{u})$ is a 2-level GC code. Then, according to (4) and (5),

$$
\begin{aligned}
\mathbf{H}(n;\underline{u}) &= \begin{pmatrix} I_{s_0+s_1} & \otimes & H(u_0,n;0) \\ \hat{H}(s_1,m;0) & \otimes & H(u_1-u_0,n;u_0) \end{pmatrix} \\
&= \left(\begin{array}{c|c|c|c}
H_0 & \underline{0} & \ldots & \underline{0} \\
\underline{0} & H_0 & \ldots & \underline{0} \\
\vdots & \vdots & \ddots & \vdots \\
\underline{0} & \underline{0} & \ldots & H_0 \\
\hline
H_1 & H_1 & \ldots & H_1 \\
H_1 & \alpha^{-1}H_1 & \ldots & \alpha^{-(m-1)}H_1 \\
H_1 & \alpha^{-2}H_1 & \ldots & \alpha^{-2(m-1)}H_1 \\
\vdots & \vdots & \ddots & \vdots \\
H_1 & \alpha^{-(s_1-1)}H_1 & \ldots & \alpha^{-(m-1)(s_1-1)}H_1
\end{array}\right)
\end{aligned} \tag{6}
$$

The parity-check matrix of a 2-level GC code was also presented in [9].

Let us take now some concrete examples of a 2-level GC code. Take $\underline{u}=(1,1,3,3)$, i.e., $u_0=1$, $u_1=3$, $s_0=s_1=2$. Then, according to (6), the parity-check matrix $\mathbf{H}(5;(1,1,3,3))$ of the 2-level code $\mathcal{C}(5;(1,1,3,3))$ is given by

$$
\mathbf{H}(5;(1,1,3,3)) = \begin{pmatrix} I_4 & \otimes & H(1,5;0) \\ \hat{H}(2,4;0) & \otimes & H(2,5;1) \end{pmatrix}.
$$

Notice that

$$
H(1,5;0) = H_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix},
$$

$$
H(2,5;1) = H_1 = \begin{pmatrix} \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \\ \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \end{pmatrix}
$$

and

$$
\hat{H}(2,4;0) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3} \end{pmatrix}.
$$

Explicitly, according to (6),

$$\mathbf{H}(5;(1,1,3,3)) \;=\; \left(\begin{array}{c|c|c|c} H_0 & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & H_0 & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & H_0 & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & H_0 \\ \hline H_1 & H_1 & H_1 & H_1 \\ H_1 & \alpha^{-1}H_1 & \alpha^{-2}H_1 & \alpha^{-3}H_1 \end{array}\right),$$

thus, $\mathbf{H}(5;(1,1,3,3))$ is the matrix

$$\left(\begin{array}{ccccc|ccccc|ccccc|ccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \\
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3} \\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^7 & \alpha^5 & \alpha^3 & \alpha & \alpha^{-1} & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^{-2} & \alpha^5 & \alpha^3 & \alpha & \alpha^{-1} & \alpha^{-3}
\end{array}\right)$$

assuming that α is an element in a finite field of order at least 5. For instance, we may take the finite field GF(8) and α a primitive root in GF(8), which has order 7.

Similarly,

$$\mathbf{H}(5;(2,2,3,3)) \;=\; \left(\begin{array}{rcl} I_4 & \otimes & H(2,5;0) \\ \hat{H}(2,4;0) & \otimes & H(1,5;1) \end{array}\right),$$

which explicitly is given by

$$\left(\begin{array}{ccccc|ccccc|ccccc|ccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \\
\hline
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^7 & \alpha^5 & \alpha^3 & \alpha & \alpha^{-1} & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^{-2} & \alpha^5 & \alpha^3 & \alpha & \alpha^{-1} & \alpha^{-3}
\end{array}\right).$$

As another example, take

$$\mathbf{H}(5;(2,2,4,4)) \;=\; \left(\begin{array}{rcl} I_4 & \otimes & H(2,5;0) \\ \hat{H}(2,4;0) & \otimes & H(2,5;1) \end{array}\right),$$

which gives the following explicit value for $\mathbf{H}(5;(2,2,4,4))$:

$$\left(\begin{array}{ccccc|ccccc|ccccc|ccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \\
\hline
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \\
\alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 \\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^7 & \alpha^5 & \alpha^3 & \alpha & \alpha^{-1} & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^{-2} & \alpha^5 & \alpha^3 & \alpha & \alpha^{-1} & \alpha^{-3} \\
\alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & \alpha^{11} & \alpha^8 & \alpha^5 & \alpha^2 & \alpha^{-1} & \alpha^{10} & \alpha^7 & \alpha^4 & \alpha & \alpha^{-2} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & \alpha^{-3}
\end{array}\right).$$

Example 2.3 Assume now $t=3$, i.e., $\underline{u}=\left(\overbrace{u_0,u_0,\ldots,u_0}^{s_0},\overbrace{u_1,u_1,\ldots,u_1}^{s_1},\overbrace{u_2,u_2,\ldots,u_2}^{s_2}\right)$ and $\mathcal{C}(n;\underline{u})$ is a 3-level GC code.. Then, according to (4) and (5),

$$\begin{aligned}
\mathbf{H}(n;\underline{u}) &= \left(\begin{array}{rcl}
I_{s_0+s_1+s_2} & \otimes & H(u_0,n;0) \\
\hat{H}(s_2,m;0) & \otimes & H(u_2-u_0,n;u_0) \\
\hat{H}(s_1,m;s_2) & \otimes & H(u_1-u_0,n;u_0)
\end{array}\right) \\
&= \left(\begin{array}{c|c|c|c}
H_0 & \underline{0} & \ldots & \underline{0} \\
\underline{0} & H_0 & \ldots & \underline{0} \\
\vdots & \vdots & \ddots & \vdots \\
\underline{0} & \underline{0} & \ldots & H_0 \\
\hline
H_2 & H_2 & \ldots & H_2 \\
H_2 & \alpha^{-1}H_2 & \ldots & \alpha^{-(m-1)}H_2 \\
\vdots & \vdots & \ddots & \vdots \\
H_2 & \alpha^{-(s_1-1)}H_2 & \ldots & \alpha^{-(m-1)(s_1-1)}H_2 \\
\hline
H_1 & \alpha^{-s_1}H_1 & \ldots & \alpha^{-(m-1)s_1}H_1 \\
H_1 & \alpha^{-(s_1+1)}H_1 & \ldots & \alpha^{-(m-1)(s_1+1)}H_1 \\
\vdots & \vdots & \ddots & \vdots \\
H_1 & \alpha^{-(s_1+s_2-1)}H_1 & \ldots & \alpha^{-(m-1)(s_1+s_2-1)}H_1
\end{array}\right).
\end{aligned} \tag{7}$$

If we take $\underline{u}=(1,1,2,3)$, then the parity-check matrix of the 3-level code $\mathcal{C}(5;(1,1,2,3))$, is given by

$$\mathbf{H}(5;(1,1,2,3)) = \left(\begin{array}{rcl}
I_4 & \otimes & H(1,5;0) \\
\hat{H}(1,4;0) & \otimes & H(2,5;1) \\
\hat{H}(1,4;1) & \otimes & H(1,5;1)
\end{array}\right),$$

which explicitly gives

$$\left(\begin{array}{ccccc|ccccc|ccccc|ccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1\\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1\\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1\\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3}
\end{array}\right),$$

while if we take $\underline{u}=(1,2,2,3)$, then the parity-check matrix $\mathbf{H}(5;(1,2,2,3))$ of the 3-level code $\mathcal{C}(5;(1,2,2,3))$, is given by

$$\mathbf{H}(5;(1,2,2,3)) \quad = \quad \left(\begin{array}{rcl} I_4 & \otimes & H(1,5;0)\\ \hat{H}(1,4;0) & \otimes & H(2,5;1)\\ \hat{H}(2,4;1) & \otimes & H(1,5;1)\end{array}\right),$$

which explicitly gives

$$\left(\begin{array}{ccccc|ccccc|ccccc|ccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1\\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1\\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1\\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3}\\
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3} & \alpha^{-4} & \alpha^{-2} & \alpha^{-3} & \alpha^{-4} & \alpha^{-5} & \alpha^{-6}
\end{array}\right),$$

again assuming that α is an element in a finite field with order at least 5.

We give next the main property of t-level GC codes.

Theorem 2.1 Consider the integers $n \leq 2^b - 1$, $t \geq 1$, $s_i \geq 1$ for $0 \leq i \leq t-1$ and $1 \leq u_0 < u_1 < \ldots < u_{t-1} \leq n-1$. Let $m = s_0 + s_1 + \cdots + s_{t-1}$ and $\underline{u}$ be given by (3). Then the t-level GC code $\mathcal{C}(n;\underline{u})$ whose parity-check matrix $\mathbf{H}(n;\underline{u})$ is given by (4) can correct up to u_i erasures in any s_i rows, $0 \leq i \leq t-1$, of an $m \times n$ array corresponding to a codeword in $\mathcal{C}(n;\underline{u})$.

Theorem 2.1 will be proved in Section 3, where we will show that there is a decoding algorithm correcting the erasure instances described in the theorem. Next we illustrate it with an example.

Example 2.4 Consider code $\mathcal{C}(5;(1,1,3,3))$ given in Example 2.2 corresponding to 4×5 arrays. According to Theorem 2.1, up to three erasures will be corrected in any pair of rows, while the remaining rows can correct up to one erasure. For example, denoting erasures by X, the following arrays are correctable in $\mathcal{C}(5;(1,1,3,3))$:

	X			
X			X	X
				X
	X	X	X	

X	X			X
X				
	X			
	X		X	X

A way to correct the erasures above is by using the parity-check matrix $\mathbf{H}(5;(1,1,3,3))$ of the code given in Example 2.2: syndromes are computed, and first the rows that experienced one erasure are corrected (using single parity). Once they are corrected, the syndromes are updated. To correct the two rows with 3 erasures each, it is needed to solve a linear system of 6 equations with 6 unknowns, which can be easily done, for instance, by Gaussian elimination (we will present a much more efficient decoding algorithm in Section 3).

As is the case in general with erasure decoding, encoding is a special case of decoding. For example, for $\mathcal{C}(5;(1,1,3,3))$, we may choose to place the parities at the end of each row, like below, in either increasing or decreasing order on the number of erasures (the STAIR codes [12] use such an encoding ordering):

				X
				X
		X	X	X
		X	X	X

		X	X	X
		X	X	X
				X
				X

Knowing a priori the erased entries allows for shortcuts in the processing time by precomputing certain operations. We will give some details in Section 3.

Similarly, $\mathcal{C}(5;(1,1,2,3))$ corresponds to a 4×5 array such that one row can correct up to three erasures, one of the remaining three rows can correct up to two erasures, and the remaining rows can correct up to one erasure. For example, the following arrays are correctable in $\mathcal{C}(5;(1,1,2,3))$:

	X			
X			X	X
				X
	X		X	

	X			X
X				
	X			
	X		X	X

Let us examine more closely the array in the left above. Consider its parity-check matrix $\mathbf{H}(5;(1,1,2,3))$ as given in Example 2.3. The rows with only one erasure are corrected using single parity, so we are left with the array

X			X	X
	X		X	

By writing the array as a vector row-wise, the erased entries correspond to locations 5, 8, 9, 16 and 18. The 5×5 matrix from $\mathbf{H}(5;(1,1,2,3))$ corresponding to these locations is

$$\tilde{H} \quad = \quad \left(\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ \hline \alpha^4 & \alpha & 1 & \alpha^3 & \alpha \\ \alpha^8 & \alpha^2 & 1 & \alpha^6 & \alpha^2 \\ \hline \alpha^3 & 1 & \alpha^{-1} & 1 & \alpha^{-2} \end{array} \right),$$

which we must prove is invertible. To see this, let $H_0(3) = (1\ 1\ 1)$, $H_0(2) = (1\ 1)$, $H_1(3) = (\alpha^4\ \alpha\ 1)$, $H_1(2) = (\alpha^3\ \alpha)$, $H_2'(3) = (\alpha^8\ \alpha^2\ 1)$ and $H_2'(2) = (\alpha^6\ \alpha^2)$. Then, we can write $\tilde{H}$ as

$$\tilde{H} \quad = \quad \left(\begin{array}{c|c} H_0(3) & \underline{0} \\ \underline{0} & H_0(2) \\ \hline H_1(3) & H_1(2) \\ H_2'(3) & H_2'(2) \\ \hline \alpha^{-1}H_1(3) & \alpha^{-3}H_1(3) \end{array} \right).$$

Since

$$\left(\begin{array}{cc} 1 & 1 \\ \alpha^{-1} & \alpha^{-3} \end{array} \right)$$

is a Vandermonde matrix, in particular it is invertible and there is a linear combination of its rows that transforms it into an upper triangular matrix with 1s in the diagonal, i.e.,

$$\left(\begin{array}{cc} 1 & \gamma \\ 0 & 1 \end{array} \right)$$

(we are not interested in the value γ at this point). Applying this linear combination to the rows of $\tilde{H}$ corresponding to $H_1(3)$ and $H_1(2)$, we obtain

$$\tilde{H}^{'} \quad = \quad \left(\begin{array}{c|c} H_0(3) & \underline{0} \\ \underline{0} & H_0(2) \\ \hline H_1(3) & \gamma H_1(2) \\ H_2'(3) & H_2'(2) \\ \hline \underline{0} & H_1(2) \end{array} \right).$$

Permuting the rows of $\tilde{H}^{'}$, we have

$$\tilde{H}^{''} \quad = \quad \left(\begin{array}{c|c} H_0(3) & \underline{0} \\ H_1(3) & \gamma H_1(2) \\ H_2'(3) & H_2'(2) \\ \hline \underline{0} & H_0(2) \\ \underline{0} & H_1(2) \end{array} \right).$$

By properties of determinants, the determinant of $\tilde{H}''$ is the product of the determinants of

$$\begin{pmatrix} H_0(3) \\ H_1(3) \\ H_2'(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \alpha^4 & \alpha & 1 \\ \alpha^8 & \alpha^2 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} H_0(2) \\ H_1(2) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \alpha^3 & \alpha \end{pmatrix}.$$

Since these determinants are both Vandermonde determinants they are non-zero, thus, their product is non-zero.

The decoding algorithm proving Theorem 2.1 to be presented in the next section develops the idea presented in Example 2.4.

The following result was given without proof in [19]:

Corollary 2.1 Consider the t-level GC code $\mathcal{C}(n;\underline{u})$ of Theorem 2.1. Then, if $\hat{s}_t = 0$ and $\hat{s}_i = \sum_{j=i}^{t} s_j$ for $0 \le i \le t-1$, the minimum distance of $\mathcal{C}(n;\underline{u})$ is given by

$$d \;=\; \min\left\{(\hat{s}_{i+1}+1)\,(u_i+1)\;,\; 0 \le i \le t-1\right\}.$$

Proof: Assume that there is a codeword that has exactly $\hat{s}_{i+1}$ rows of weight $u_i + 1$ and one row of weight u_i, while all the other rows are zero (notice that when $i = t-1$, this simply means that there is a codeword consisting of a row of weight u_{t-1}, while all the other rows are zero). By Theorem 2.1, such a codeword would be corrected by the code as the zero codeword, thus

$$d \;>\; \min\left\{(\hat{s}_{i+1})\,(u_i+1) + u_i\;,\; 0 \le i \le t-1\right\},$$

or,

$$d \;\ge\; \min\left\{(\hat{s}_{i+1}+1)\,(u_i+1)\;,\; 0 \le i \le t-1\right\}.$$

In order to show equality, we need to prove that for each i, $1 \le i \le t-1$, there is a codeword in $\mathcal{C}(n;\underline{u})$ of weight $(\hat{s}_{i+1}+1)\,(u_i+1)$.

Consider first the case $i = t-1$, thus, we have to prove that there is a codeword of weight $u_{t-1}+1$. Let $\underline{u}$ be a codeword of weight $u_{t-1}+1$ in the $[n, n-u_{t-1}, u_{t-1}+1]$ RS code whose parity-check matrix is given by $H(u_{t-1}, n; 0)$, and $\underline{0}_n$ a zero vector of length n. Then, according to (4) and (5), vector

$$(\underline{u}, \overbrace{\underline{0}_n, \underline{0}_n, \ldots, \underline{0}_n}^{m-1})$$

is a codeword in $\mathcal{C}(n;\underline{u})$ of weight $u_{t-1}+1$.

Next consider $0 \le i \le t-2$. Let $\underline{u}$ be a codeword of weight u_i+1 in the $[n, n-u_i, u_i+1]$ RS code whose parity-check matrix is given by $H(u_i, n; 0)$. Let $\underline{v}$ be a codeword of weight $\hat{s}_{i+1}+1$ in the RS code whose parity-check matrix is given by $\hat{H}(\hat{s}_{i+1}, \hat{s}_{i+1}+1; 0)$. Explicitly, let $\underline{v} = (v_0, v_1, \ldots, v_{\hat{s}_{i+1}})$. Consider the following vector of length mn:

$$\underline{w} \;=\; \left(v_0\underline{u}, v_1\underline{u}, \ldots, v_{\hat{s}_{i+1}}\underline{u}, \underline{0} \right),$$

where $\underline{0}$ is a vector of length $n\left(\sum_{j=0}^{i} s_j\right)$. According to (4) and (5), we have to show that vector $\underline{w}$ is a codeword in $\mathcal{C}(n;\underline{u})$. Certainly, since $H(u_i, n; 0)\underline{u}^T = \underline{0}$, we have that, according to (5), the inner product of the first mu_0 and the last $\sum_{j=1}^{i} u_j s_j$ rows of $\mathbf{H}(n;\underline{u})$ with $\underline{w}$ are zero. Now, take any of the rows corresponding to $H_{t-1}, H_{t-2}, \ldots, H_{i+1}$ in (5). The inner product of such a row with $\underline{w}$ is also zero, since it is a constant times the inner product of $\underline{v}$ with a row of the parity-check matrix $\hat{H}(\hat{s}_{i+1}, \hat{s}_{i+1}+1; 0)$, which is zero by construction. □

The following example illustrates Corollary 2.1.

Example 2.5 Consider code $\mathcal{C}(5; (1,2,2,3))$ as given in Example 2.3. Corollary 2.1 states that the minimum distance of $\mathcal{C}(5; (1,2,2,3))$ is given by

$$d \;=\; \min\{(4)(2)\;,\;(2)(3)\;,\;4\} \;=\; 4.$$

Certainly there are no codewords of weight 3. According to (5), the parity-check matrix $\mathbf{H}(5; (1,2,2,3))$ is given by

$$\mathbf{H}(5;(1,2,2,3)) \;=\; \left(\begin{array}{c|c|c|c} H_0 & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & H_0 & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & H_0 & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & H_0 \\ \hline H_2 & H_2 & H_2 & H_2 \\ \hline H_1 & \alpha^{-1}H_1 & \alpha^{-2}H_1 & \alpha^{-3}H_1 \\ H_1 & \alpha^{-2}H_1 & \alpha^{-4}H_1 & \alpha^{-6}H_1 \end{array}\right),$$

where

$$H_0 = \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \end{array}\right),$$
$$H_1 = \left(\begin{array}{ccccc} \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \end{array}\right)$$

and

$$H_2 = \left(\begin{array}{ccccc} \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \\ \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \end{array}\right).$$

Consider the [5,2,4] RS code whose parity-check matrix is $\binom{H_0}{H_2}$. Let $\underline{u}$ be a codeword of weight 4 in such a code. Then, $(\underline{u}\,,\,\underline{0}\,,\,\underline{0}\,,\,\underline{0})$ is a codeword of weight 4 in $\mathcal{C}(5;(1,2,2,3))$, since we easily see that its inner product with the rows of $\mathbf{H}(5;(1,2,2,3))$ is zero.

Let us show next the existence of a codeword of weight $(2)(3)=6$ with two non-zero rows of weight 3. Take a codeword $\underline{u}$ of weight 3 in the $[5,3,3]$ code whose parity-check matrix is given by $\binom{H_0}{H_1}$. Consider a codeword of weight 2 in the $[5,4,2]$ code whose parity-check matrix is $\left(\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right)$, say, (1,1,0,0). Then, we can see that $\underline{w}=(\underline{u},\underline{u},\underline{0},\underline{0})$ is a codeword of weight (2)(3) in $\mathcal{C}(5;(1,2,2,3))$. In effect, the inner product of $\underline{w}$ with the first 5 and the last 2 rows of $\mathbf{H}(5;(1,2,2,3))$ is zero, since the inner product of the rows of H_0 and of H_1 with $\underline{u}$ are zero by construction. Now, if the inner product of $\underline{u}$ with the second row of H_2 is, say, γ, then the inner product of $\underline{w}$ with the sixth row of $\mathbf{H}(5;(1,2,2,3))$ is $\gamma\oplus\gamma=0$.

Finally, let us show that there is a codeword of weight $(4)(2)=8$, with four non-zero rows of weight 2. Take a codeword $\underline{u}$ of weight 2 in the $[5,4,2]$ code whose parity-check matrix is given by H_0, for instance, $\underline{u}=(1,1,0,0,0)$ is such a codeword. Take a codeword $\underline{v}=(v_0,v_1,v_2,v_3)$ of weight 4 in the $[4,1,4]$ code whose parity-check matrix is

$$\hat{H}(3,4;0) \;=\; \left(\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3} \\ 1 & \alpha^{-2} & \alpha^{-4} & \alpha^{-6} \end{array}\right).$$

Take $\underline{w}=\,(v_0\underline{u},v_1\underline{u},v_2\underline{u},v_3\underline{u})$. Then, $\underline{w}$ is a codeword of weight (4)(2) in $\mathcal{C}(5;(1,2,2,3))$. In effect, the inner product of $\underline{w}$ with any of the first four rows of $\mathbf{H}(5;(1,2,2,3))$ is zero, since the inner product of $\underline{u}$ with the row of H_0 is zero. Next take any of the remaining rows, and assume that the inner product of $\underline{u}$ with the first 5 coordinates of such row is γ. Then the inner product of $\underline{w}$ with the row is given by γ times the inner product of $\underline{v}$ with a row of $\hat{H}(3,4;0)$, which is zero by construction.

3 Encoding and Decoding

In erasure decoding, encoding is a special case of the decoding. The decoding algorithm to be presented next also proves Theorem 2.1.

Assume that we have a t-level GC-code $\mathcal{C}(n;\underline{u})$ as given by Definition 2.1. The codewords are $m\times n$ arrays. As before, let $\underline{u}$ be given by (3), $\underline{v}$ be a received $m\times n$ array with erasures, and without loss of generality, assume that there are s_{t-1} rows of $\underline{v}$ with u_{t-1} erasures each, s_{t-2} rows of $\underline{v}$ with u_{t-2} erasures each, and so on, until finally there are s_0 rows of $\underline{v}$ with u_0 erasures each. Let $\sigma:\{0,1,\ldots,m-1\}\to\{0,1,\ldots,m-1\}$ be a permutation of the rows of $\underline{v}$ and $\underline{v}_\sigma$ the array $\underline{v}$ with the rows permuted according to σ, such that the first s_{t-1} rows of $\underline{v}_\sigma$ have u_{t-1} erasures each, the next s_{t-2} rows of $\underline{v}_\sigma$ have u_{t-2} erasures each, and so on, until finally the last s_0 rows of $\underline{v}_\sigma$ have s_0 erasures each.

We permute accordingly the columns of the parity-check matrix $\mathbf{H}(n;\underline{u})$ of $\mathcal{C}(n;\underline{u})$ to give the permuted parity-check matrix $\mathbf{H}_\sigma(n;\underline{u})$ corresponding to a permuted code $\mathcal{C}_\sigma(n;\underline{u})$.

Specifically, if we write the $\left(\sum_{i=0}^{t-1} u_i s_i\right) \times mn$ matrix $\mathbf{H}(n;\underline{u})$ as

$$\mathbf{H}(n;\underline{u}) \quad = \quad \left(\begin{array}{cccc} \mathbf{H}_0 & \mathbf{H}_1 & \ldots & \mathbf{H}_{m-1} \end{array} \right), \tag{8}$$

where each $\mathbf{H_i}$ is a $\left(\sum_{i=0}^{t-1} u_i s_i\right) \times n$ matrix, and let $i_0, i_1, \ldots, i_{m-1}$ be such that $\sigma(i_j) = j$ for $0 \leq j \leq m-1$, then

$$\mathbf{H}_{\sigma}(n;\underline{u}) \quad = \quad \left(\begin{array}{cccc} \mathbf{H}_{i_0} & \mathbf{H}_{i_1} & \ldots & \mathbf{H}_{i_{m-1}} \end{array} \right) \tag{9}$$

and $\mathcal{C}_{\sigma}(n;\underline{u})$ is the permuted code given by the parity-check matrix $\mathbf{H}_{\sigma}(n;\underline{u})$. We will see how to use this permuted parity-check matrix to implement an efficient decoding algorithm.

Based on $\mathbf{H}(n;\underline{u})$ as given by (5), $\mathbf{H}_{\sigma}(n;\underline{u})$ is given by

$$\mathbf{H}_{\sigma}(n;\underline{u}) \quad = \quad \left(\begin{array}{c|c|c|c}
H_0 & \underline{0} & \ldots & \underline{0} \\
\underline{0} & H_0 & \ldots & \underline{0} \\
\vdots & \vdots & \ddots & \vdots \\
\underline{0} & \underline{0} & \ldots & H_0 \\
\hline
H_{t-1} & H_{t-1} & \ldots & H_{t-1} \\
\alpha^{-i_0} H_{t-1} & \alpha^{-i_1} H_{t-1} & \ldots & \alpha^{-i_{m-1}} H_{t-1} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_{t-1}-1)} H_{t-1} & \alpha^{-i_1(\hat{s}_{t-1}-1)} H_{t-1} & \ldots & \alpha^{-i_{m-1}(\hat{s}_{t-1}-1)} H_{t-1} \\
\hline
\alpha^{-i_0 \hat{s}_{t-1}} H_{t-2} & \alpha^{-i_1 \hat{s}_{t-1}} H_{t-2} & \ldots & \alpha^{-i_{m-1} \hat{s}_{t-1}} H_{t-2} \\
\alpha^{-i_0(\hat{s}_{t-1}+1)} H_{t-2} & \alpha^{-i_1(\hat{s}_{t-1}+1)} H_{t-2} & \ldots & \alpha^{-i_{m-1}(\hat{s}_{t-1}+1)} H_{t-2} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_{t-2}-1)} H_{t-2} & \alpha^{-i_1(\hat{s}_{t-2}-1)} H_{t-2} & \ldots & \alpha^{-i_{m-1}(\hat{s}_{t-2}-1)} H_{t-2} \\
\hline
\vdots & \vdots & \ddots & \vdots \\
\hline
\alpha^{-i_0 \hat{s}_{i+1}} H_i & \alpha^{-i_1 \hat{s}_{i+1}} H_i & \ldots & \alpha^{-i_{m-1} \hat{s}_{i+1}} H_i \\
\alpha^{-i_0(\hat{s}_{i+1}+1)} H_i & \alpha^{-i_1(\hat{s}_{i+1}+1)} H_i & \ldots & \alpha^{-i_{m-1}(\hat{s}_{i+1}+1)} H_i \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_i-1)} H_i & \alpha^{-i_1(\hat{s}_i-1)} H_i & \ldots & \alpha^{-i_{m-1}(\hat{s}_i-1)} H_i \\
\hline
\vdots & \vdots & \ddots & \vdots \\
\hline
\alpha^{-i_0 \hat{s}_2} H_1 & \alpha^{-i_1 \hat{s}_2} H_1 & \ldots & \alpha^{-i_{m-1} \hat{s}_2} H_1 \\
\alpha^{-i_0(\hat{s}_2+1)} H_1 & \alpha^{-i_1(\hat{s}_2+1)} H_1 & \ldots & \alpha^{-i_{m-1}(\hat{s}_2+1)} H_1 \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_1-1)} H_1 & \alpha^{-i_1(\hat{s}_1-1)} H_1 & \ldots & \alpha^{-i_{m-1}(\hat{s}_1-1)} H_1
\end{array} \right) \tag{10}$$

Consider next the $\hat{s}_1 \times m$ matrix

$$\left(\begin{array}{cccc}
1 & 1 & \dots & 1 \\
\alpha^{-i_0} & \alpha^{-i_1} & \dots & \alpha^{-i_{m-1}} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_{t-1}-1)} & \alpha^{-i_1(\hat{s}_{t-1}-1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{t-1}-1)} \\
\hline
\alpha^{-i_0\hat{s}_{t-1}} & \alpha^{-i_1\hat{s}_{t-1}} & \dots & \alpha^{-i_{m-1}\hat{s}_{t-1}} \\
\alpha^{-i_0(\hat{s}_{t-1}+1)} & \alpha^{-i_1(\hat{s}_{t-1}+1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{t-1}+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_{t-2}-1)} & \alpha^{-i_1(\hat{s}_{t-2}-1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{t-2}-1)} \\
\hline
\vdots & \vdots & \ddots & \vdots \\
\hline
\alpha^{-i_0\hat{s}_{i+1}} & \alpha^{-i_1\hat{s}_{i+1}} & \dots & \alpha^{-i_{m-1}\hat{s}_{i+1}} \\
\alpha^{-i_0(\hat{s}_{i+1}+1)} & \alpha^{-i_1(\hat{s}_{i+1}+1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{i+1}+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_{i}-1)} & \alpha^{-i_1(\hat{s}_{i}-1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{i}-1)} \\
\hline
\vdots & \vdots & \ddots & \vdots \\
\hline
\alpha^{-i_0\hat{s}_{2}} & \alpha^{-i_1\hat{s}_{2}} & \dots & \alpha^{-i_{m-1}\hat{s}_{2}} \\
\alpha^{-i_0(\hat{s}_{2}+1)} & \alpha^{-i_1(\hat{s}_{2}+1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{2}+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha^{-i_0(\hat{s}_{1}-1)} & \alpha^{-i_1(\hat{s}_{1}-1)} & \dots & \alpha^{-i_{m-1}(\hat{s}_{1}-1)}
\end{array}\right) \tag{11}$$

Since this one is a (rectangular) Vandermonde matrix and $\hat{s}_1 < \hat{s}_0 = m$, there is a linear combination that transforms the matrix above into an upper triangular form (for instance, by doing Gaussian elimination). Specifically, let the upper triangular form be

$$\begin{pmatrix}
1 & 1 & 1 & \dots & 1 & 1 & \dots & 1 & 1 \\
0 & 1 & \gamma_{1,2} & \cdots & \gamma_{1,\underline{s}_1-2} & \gamma_{1,\underline{s}_1-1} & \cdots & \gamma_{1,m-2} & \gamma_{1,m-1} \\
0 & 0 & 1 & \dots & \gamma_{2,\underline{s}_1-2} & \gamma_{2,\underline{s}_1-1} & \cdots & \gamma_{2,m-2} & \gamma_{2,m-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 1 & \gamma_{\underline{s}_1-2,\underline{s}_1-1} & \cdots & \gamma_{\underline{s}_1-2,m-2} & \gamma_{\underline{s}_1-2,m-1} \\
0 & 0 & 0 & \dots & 0 & 1 & \cdots & \gamma_{\underline{s}_1-1,m-2} & \gamma_{\underline{s}_1-1,m-1}
\end{pmatrix} \tag{12}$$

Since the rows of H_i are contained in the rows of H_j for $i < j$, this means that the parity-check matrix $\mathbf{H}_\sigma(n;\underline{u})$ as given by (10), by row operations and after some rearrangement of the rows, can be transformed into the pseudo upper-triangular matrix $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$ given by (13) below using (12):

$$\left(\begin{array}{c|c|c|c|c|c|c|c}
H_0 & \underline{0} & \dots & \underline{0} & \underline{0} & \dots & \underline{0} & \underline{0} \\
H_{t-1} & H_{t-1} & \dots & H_{t-1} & H_{t-1} & \dots & H_{t-1} & H_{t-1} \\
\hline
\underline{0} & H_0 & \dots & \underline{0} & \underline{0} & \dots & \underline{0} & \underline{0} \\
\underline{0} & H_{t-1} & \dots & \gamma_{1,\underline{s}_1-1}H_{t-1} & \gamma_{1,\underline{s}_1}H_{t-1} & \dots & \gamma_{1,m-2}H_{t-1} & \gamma_{1,m-1}H_{t-1} \\
\hline
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\hline
\underline{0} & \underline{0} & \dots & H_0 & \underline{0} & \dots & \underline{0} & \underline{0} \\
\underline{0} & \underline{0} & \dots & H_1 & \gamma_{\underline{s}_1-1,\underline{s}_1}H_1 & \dots & \gamma_{\underline{s}_1-1,m-2}H_1 & \gamma_{\underline{s}_1-1,m-1}H_1 \\
\hline
\underline{0} & \underline{0} & \dots & \underline{0} & H_0 & \dots & \underline{0} & \underline{0} \\
\hline
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\hline
\underline{0} & \underline{0} & \dots & \underline{0} & \underline{0} & \dots & H_0 & \underline{0} \\
\hline
\underline{0} & \underline{0} & \dots & \underline{0} & \underline{0} & \dots & \underline{0} & H_0
\end{array}\right) \tag{13}$$

Using the pseudo upper-triangular parity-check matrix $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$ given by (13), we can decode the (permuted) received array $\underline{v}_\sigma$ by successive decoding of individual RS codes. Notice that H_0 is the parity-check matrix of a RS code that can correct up to u_0 erasures, and each $\binom{H_0}{H_i}$, $1 \leq i \leq t-1$, is the parity-check matrix of a RS code that can correct up to u_i erasures. The first step is computing the $\sum_{i=0}^{t-1} u_i s_i$ syndromes of $\underline{v}_\sigma$ (the permuted version of the received array $\underline{v}$) with respect to $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$ (erasures are assumed to be zero in syndrome computation). Since the number of erasures of $\underline{v}_\sigma$ is in non-increasing order, the up to u_0 erasures in the last row of $\underline{v}_\sigma$ are corrected by using the parity-check matrix H_0. Once this has been done, the remaining $(\sum_{i=0}^{t-1} u_i s_i) - u_0$ syndromes are updated using the corrected information. The process is repeated with each of the last s_0 rows of $\underline{v}_\sigma$, which contain up to u_0 erasures each. Once finished with correction of the last s_0 rows, the next row, containing up to u_1 erasures, is corrected using the parity-check matrix $\binom{H_0}{H_1}$. The process continues by induction, until the first row, which contains up to u_{t-1} erasures, is corrected. Finally, the inverse permutation σ^{-1} is applied to the rows of the corrected version of $\underline{v}_\sigma$ to obtain the corrected version of $\underline{v}$.

Let us write formally the algorithm arising from the discussion above.

Algorithm 3.1 (Decoding Algorithm) Consider a t-level GC-code $\mathcal{C}(n;\underline{u})$ as given by Definition 2.1. Let $\underline{v}$ be a received $m \times n$ array with erasures.

Let $\sigma : \{0, 1, \dots, m-1\} \to \{0, 1, \dots, m-1\}$ be a permutation of the rows of $\underline{v}$ and $\underline{v}_\sigma$ the array $\underline{v}$ with the rows permuted according to σ, such that the number of erasures in each row of $\underline{v}_\sigma$ is in non-increasing order.

If the parity-check matrix of $\mathbf{H}(n;\underline{u})$ of $\mathcal{C}(n;\underline{u})$ is given by (8), consider the permuted parity-check matrix $\mathbf{H}_\sigma(n;\underline{u})$ given by (9), or, more in detail, by (10), which corresponds to a permuted code $\mathcal{C}_\sigma(n;\underline{u})$.

Let $\sigma^{-1}(j)=i_j$ for $0\leq j\leq m-1$. Take the rectangular Vandermonde matrix given by (11) and, by row operations, transform it into the upper triangular form given by (12). Use this upper triangular matrix to transform the parity-check matrix $\mathbf{H}_\sigma(n;\underline{u})$ as given by (10) into the pseudo upper triangular parity-check matrix $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$ given by (13). Then proceed as follows:

1. Compute the $\sum_{i=0}^{m-1}u_is_i$ syndromes of $\underline{v}_\sigma$ with respect to the parity-check matrix $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$.
2. Correct the erasures in the last row of $\underline{v}_\sigma$ using the RS parity-check matrix H_0 and the last u_0 syndromes. Then the next to last row of $\underline{v}_\sigma$ using the RS parity-check matrix H_0 and the next to last u_0 syndromes, and so on until correcting the last s_0 rows. If any of these last rows had more than u_0 erasures, then declare an uncorrectable error.
3. Using the corrected locations and values in the last s_0 rows of $\underline{v}_\sigma$, update the first $\sum_{i=1}^{m-1}u_is_i$ syndromes of $\underline{v}_\sigma$ with respect to $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$.
4. Next, consider the last of the first $\sum_{i=1}^{m-1}s_i$ rows of $\underline{v}_\sigma$. If there are more than u_1 erasures in such row, declare an uncorrectable error. Otherwise, correct up to u_1 erasures in the last of these $\sum_{i=1}^{m-1}s_i$ rows using the last u_1 of the $\sum_{i=1}^{m-1}u_is_i$ syndromes with respect to the RS code whose parity-check matrix is given by $\left(\begin{smallmatrix}H_0\\H_1\end{smallmatrix}\right)$. Update then the first $\left(\sum_{i=1}^{m-1}u_is_i\right)-u_1$ syndromes.
5. Repeat the process until the first row, which contains up to u_{m-1} erasures, is corrected using the first u_{m-1} syndromes with respect to the RS code whose parity-check matrix is given by $\left(\begin{smallmatrix}H_0\\H_{m-1}\end{smallmatrix}\right)$.
6. Obtain the corrected array $\underline{v}$ by applying the permutation σ^{-1} to the rows of the corrected array $\underline{v}_\sigma$.

The next example illustrates the decoding algorithm.

Example 3.1 Let $n=5$ and $\underline{u}=(1,2,2,4)$. Take the code $\mathcal{C}(5;(1,2,2,4))$ over the finite field $GF(8)$ given by the primitive polynomial $1+x+x^3$. According to (4) and (5), $\mathbf{H}(5;(1,2,2,4))$ is given by the 9×20 matrix

$$\left(\begin{array}{c|c|c|c} H_0 & \underline{0} & \underline{0} & \underline{0}\\ \underline{0} & H_0 & \underline{0} & \underline{0}\\ \underline{0} & \underline{0} & H_0 & \underline{0}\\ \underline{0} & \underline{0} & \underline{0} & H_0\\ \hline H_2 & H_2 & H_2 & H_2\\ \hline H_1 & \alpha^{-1}H_1 & \alpha^{-2}H_1 & \alpha^{-3}H_1\\ H_1 & \alpha^{-2}H_1 & \alpha^{-4}H_1 & \alpha^{-6}H_1 \end{array}\right),$$

where

$$H_0 \ = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$H_2 \ = \begin{pmatrix} \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \\ \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \\ \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 \end{pmatrix}.$$

and H_1 corresponds to the first row of H_2. Notice that $\alpha^8 = \alpha$, $\alpha^9 = \alpha^2$ and $\alpha^{12} = \alpha^5$ since $\alpha^7 = 1$.

The codewords in the code are 4 × 5 arrays. Assume that the following array has been received:

$$\underline{v} \ = \begin{array}{|c|c|c|c|c|} \hline E & \alpha^3 & 1 & E & 0 \\ \hline \alpha^6 & E & E & E & E \\ \hline \alpha^6 & E & \alpha^5 & E & 1 \\ \hline \alpha^4 & 0 & \alpha & E & \alpha^3 \\ \hline \end{array}$$

where E denotes an erasure. We can see that there are 2 erasures in the first row, 4 in the second, 2 in the third and one in the fourth. If we take the permutation

$$\sigma : \{0,1,2,3\} \to \{0,1,2,3\}$$

such that $\sigma(0) = 1$, $\sigma(1) = 0$, $\sigma(2) = 2$ and $\sigma(3) = 3$, the permuted array is given by

$$\underline{v}_\sigma \ = \begin{array}{|c|c|c|c|c|} \hline \alpha^6 & E & E & E & E \\ \hline E & \alpha^3 & 1 & E & 0 \\ \hline \alpha^6 & E & \alpha^5 & E & 1 \\ \hline \alpha^4 & 0 & \alpha & E & \alpha^3 \\ \hline \end{array}.$$

We can see that the number of erasures in $\underline{v}_\sigma$ appears now in non-increasing order: the first row has 4 erasures, the next two have two, and the last row has one erasure. The parity-check matrix $\mathbf{H}_\sigma(5; (1,2,2,4))$ corresponding to the permuted code $\mathcal{C}_\sigma(5; (1,2,2,4))$ is given by

$$\left(\begin{array}{c|c|c|c} H_0 & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & H_0 & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & H_0 & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & H_0 \\ \hline H_2 & H_2 & H_2 & H_2 \\ \hline \alpha^{-1}H_1 & H_1 & \alpha^{-2}H_1 & \alpha^{-3}H_1 \\ \alpha^{-2}H_1 & H_1 & \alpha^{-4}H_1 & \alpha^{-6}H_1 \end{array} \right)$$

and the matrix given by (11) is in this example

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ \alpha^{-1} & 1 & \alpha^{-2} & \alpha^{-3} \\ \alpha^{-2} & 1 & \alpha^{-4} & \alpha^{-6} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \alpha^{6} & 1 & \alpha^{5} & \alpha^{4} \\ \alpha^{5} & 1 & \alpha^{3} & \alpha \end{pmatrix}.$$

Triangulating this last matrix, for instance, by Gaussian elimination, we obtain the matrix given by (12)

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & \alpha^{6} & \alpha \\ 0 & 0 & 1 & \alpha^{3} \end{pmatrix}.$$

Now, with this matrix, we can obtain $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$ given by (13) as follows:

$$\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4)) = \left(\begin{array}{c|c|c|c} H_0 & \underline{0} & \underline{0} & \underline{0} \\ H_2 & H_2 & H_2 & H_2 \\ \hline \underline{0} & H_0 & \underline{0} & \underline{0} \\ \underline{0} & H_1 & \alpha^6 H_1 & \alpha H_1 \\ \hline \underline{0} & \underline{0} & H_0 & \underline{0} \\ \underline{0} & \underline{0} & H_1 & \alpha^3 H_1 \\ \hline \underline{0} & \underline{0} & \underline{0} & H_0 \end{array}\right).$$

Next we compute the 9 syndromes of $\underline{v}_\sigma$ with respect to $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$. Explicitly, these 9 syndromes are

$$\begin{array}{rcl} \mathbf{S}_0 & = & \alpha^6 \\ \mathbf{S}_1 & = & \alpha^3 \\ \mathbf{S}_2 & = & \alpha^2 \\ \mathbf{S}_3 & = & \alpha^3 \\ \mathbf{S}_4 & = & \alpha \\ \mathbf{S}_5 & = & 1 \\ \mathbf{S}_6 & = & \alpha^3 \\ \mathbf{S}_7 & = & \alpha^6 \\ \mathbf{S}_8 & = & \alpha^5 \end{array}$$

The first step is decoding one erasure in the fourth coordinate of $\underline{v}_\sigma$, which corresponds to coordinate 18 of $\underline{v}_\sigma$ when written as a vector. Since there is only one erased coordinate, such erased coordinate has to equal the syndrome $\mathbf{S}_8=\alpha^5$. Thus, the last row of $\underline{v}_\sigma$ becomes

$$\begin{pmatrix} \alpha^4 & 0 & \alpha & \alpha^5 & \alpha^3. \end{pmatrix}$$

The next step is updating the first 8 syndromes. Notice that $\mathbf{S}_0$, $\mathbf{S}_4$ and $\mathbf{S}_6$ remain the same since coordinate 18 of the corresponding rows in $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$ are zero. As for the rest, using $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$, we have

$$\begin{array}{rclclcl}
\mathbf{S}_1 & = & \mathbf{S}_1\oplus(\alpha)(\alpha^5) & = & \alpha^3\oplus\alpha^6 & = & \alpha^4\\
\mathbf{S}_2 & = & \mathbf{S}_2\oplus(\alpha^2)(\alpha^5) & = & \alpha^2\oplus 1 & = & \alpha^6\\
\mathbf{S}_3 & = & \mathbf{S}_3\oplus(\alpha^3)(\alpha^5) & = & \alpha^3\oplus\alpha & = & 1\\
\mathbf{S}_5 & = & \mathbf{S}_5\oplus(\alpha^2)(\alpha^5) & = & 1\oplus 1 & = & 0\\
\mathbf{S}_7 & = & \mathbf{S}_7\oplus(\alpha^4)(\alpha^5) & = & \alpha^6\oplus\alpha^2 & = & 1
\end{array}$$

Next we have to decode the two erasures corresponding to the third row of $\underline{v}_\sigma$ using the parity-check matrix $\binom{H_0}{H_1}$ and the two syndromes $\mathbf{S}_6$ and $\mathbf{S}_7$. Specifically, since erasures have occurred in locations 1 and 3 of the third row, we have to solve the following system of two linear equations with two unknowns:

$$\begin{array}{rclcl}
X\oplus Y & = & \mathbf{S}_6 & = & \alpha^3\\
\alpha^3X\oplus\alpha Y & = & \mathbf{S}_7 & = & 1.
\end{array}$$

Solving this system, for instance by triangulation, gives $X=\alpha^5$ and $Y=\alpha^2$. Replacing in the third row of $\underline{v}_\sigma$ gives

$$\begin{pmatrix} \alpha^6 & \alpha^5 & \alpha^5 & \alpha^2 & 1 \end{pmatrix}.$$

Next we need to update the first 6 syndromes, but as before, syndromes $\mathbf{S}_0$ and $\mathbf{S}_4$ do not need to be updated. The corrected erased coordinates correspond to coordinates 11 and 13 of $\underline{v}_\sigma$ when regarded as a vector. Again using $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$, we have

$$\begin{array}{rclclcl}
\mathbf{S}_1 & = & \mathbf{S}_1\oplus(\alpha^3)(\alpha^5)\oplus(\alpha)(\alpha^2) & = & \alpha^4\oplus\alpha\oplus\alpha^3 & = & \alpha^5\\
\mathbf{S}_2 & = & \mathbf{S}_2\oplus(\alpha^6)(\alpha^5)\oplus(\alpha^2)(\alpha^2) & = & \alpha^6\oplus\alpha^4\oplus\alpha^4 & = & \alpha^6\\
\mathbf{S}_3 & = & \mathbf{S}_3\oplus(\alpha^2)(\alpha^5)\oplus(\alpha^3)(\alpha^2) & = & 1\oplus 1\oplus\alpha^5 & = & \alpha^5\\
\mathbf{S}_5 & = & \mathbf{S}_5\oplus(\alpha^2)(\alpha^5)\oplus(1)(\alpha^2) & = & 0\oplus 1\oplus\alpha^2 & = & \alpha^6
\end{array}$$

Now we have to decode the two erasures corresponding to the second row of $\underline{v}_\sigma$ using the parity-check matrix $\binom{H_0}{H_1}$ and the two syndromes $\mathbf{S}_4$ and $\mathbf{S}_5$. Since erasures have occurred in locations 0 and 3 of the second row, we have to solve the following system of two linear equations with two unknowns:

$$\begin{aligned} X \oplus Y &= \mathbf{S}_4 = \alpha \\ \alpha^4 X \oplus \alpha Y &= \mathbf{S}_5 = \alpha^6. \end{aligned}$$

Solving this system gives $X = \alpha^5$ and $Y = \alpha^6$. Replacing in the second row of $\underline{v}_\sigma$ gives

$$\left(\begin{array}{ccccc} \alpha^5 & \alpha^3 & 1 & \alpha^6 & 0 \end{array}\right).$$

Next we need to update the first 4 syndromes, but syndrome $\mathbf{S}_0$ does not need to be updated. The corrected erased coordinates correspond to coordinates 5 and 8 of $\underline{v}_\sigma$ when regarded as a vector. Using $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$, we have

$$\begin{array}{lllllll} \mathbf{S}_1 & = & \mathbf{S}_1 \oplus (\alpha^4)(\alpha^5) \oplus (\alpha)(\alpha^6) & = & \alpha^5 \oplus \alpha^2 \oplus 1 & = & \alpha \\ \mathbf{S}_2 & = & \mathbf{S}_2 \oplus (\alpha)(\alpha^5) \oplus (\alpha^2)(\alpha^6) & = & \alpha^6 \oplus \alpha^6 \oplus \alpha & = & \alpha \\ \mathbf{S}_3 & = & \mathbf{S}_3 \oplus (\alpha^5)(\alpha^5) \oplus (\alpha^3)(\alpha^6) & = & \alpha^5 \oplus \alpha^3 \oplus \alpha^2 & = & 0 \end{array}$$

Finally we have to decode the four erasures corresponding to the first row of $\underline{v}_\sigma$ using the parity-check matrix $\binom{H_0}{H_2}$ and the four syndromes $\mathbf{S}_0$, $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$. Since erasures have occurred in locations 1, 2, 3 and 4 of the first row, we have to solve the following system of four linear equations with four unknowns:

$$\begin{aligned} X \oplus Y \oplus Z \oplus W &= \mathbf{S}_0 = \alpha^6 \\ \alpha^3 X \oplus \alpha^2 Y \oplus \alpha Z \oplus W &= \mathbf{S}_1 = \alpha \\ \alpha^6 X \oplus \alpha^4 Y \oplus \alpha^2 Z \oplus W &= \mathbf{S}_2 = \alpha \\ \alpha^9 X \oplus \alpha^6 Y \oplus \alpha^3 Z \oplus W &= \mathbf{S}_3 = 0 \end{aligned}$$

Solving this system, we obtain $X = 0$, $Y = \alpha^3$, $Z = 1$ and $W = \alpha^5$. Replacing in the first row of $\underline{v}_\sigma$ gives

$$\left(\begin{array}{ccccc} \alpha^6 & 0 & \alpha^3 & 1 & \alpha^5 \end{array}\right).$$

Finally, we apply σ^{-1} (which in this particular case coincides with σ) to the rows of the decoded version of $\underline{v}_\sigma$ to obtain the decoded version of $\underline{v}$, giving the decoded array

$$\underline{v} \;=\; \begin{array}{|c|c|c|c|c|}\hline \alpha^5 & \alpha^3 & 1 & \alpha^6 & 0 \\ \hline \alpha^6 & 0 & \alpha^3 & 1 & \alpha^5 \\ \hline \alpha^6 & \alpha^5 & \alpha^5 & \alpha^2 & 1 \\ \hline \alpha^4 & 0 & \alpha & \alpha^5 & \alpha^3 \\ \hline \end{array}$$

It can be verified that the syndromes of this array with respect to the parity-check matrix $\mathbf{H}(5;(1,2,2,4))$ are zero, otherwise an uncorrectable error would be declared.

Let us point out that the decoding algorithm can be adapted to correct errors as well as erasures (or combinations of both), but in this paper we concentrate on the erasure problem only.

3.1 Encoding

The encoding is a special case of the decoding, where the parities correspond to erasures. We can place the parities wherever we want as long as the erasure-correcting capability of the code is not exceeded. A natural choice is to put the parities in non-increasing order with respect to their number in the last entries of each row. For example, if $\underline{u} = (1,2,2,4)$ like in Example 3.1, the parities may be placed as follows (assuming $n = 5$ as in the example):

$$\begin{array}{|c|c|c|c|c|}\hline D & P & P & P & P \\ \hline D & D & D & P & P \\ \hline D & D & D & P & P \\ \hline D & D & D & D & P \\ \hline \end{array},$$

where D denotes data and P parity. In this case, the permutation σ is the identity. Knowing a priori where the parities are allows for precomputing the pseudo-triangular parity-check matrix $\overset{\triangle}{\mathbf{H}}_\sigma(n;\underline{u})$ given by (13). Then the encoding follows the steps of the decoding to compute the parities. Let us retake the case of Example 3.1 to illustrate the encoding.

Example 3.2 Assume that we want to encode the following array in $\mathcal{C}(5;(1,2,2,4))$ over the finite field $GF(8)$, where the entries denoted by P are the parities and are considered as erasures.

$$\underline{v} \;=\; \begin{array}{|c|c|c|c|c|}\hline \alpha^5 & P & P & P & P \\ \hline \alpha^6 & 0 & \alpha^3 & P & P \\ \hline \alpha^6 & \alpha^5 & \alpha^5 & P & P \\ \hline \alpha^4 & 0 & \alpha & \alpha^5 & P \\ \hline \end{array}$$

Following the decoding algorithm, as in Example 3.1, we need to find the pseudo-triangular parity-check matrix $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4)$, where in this case σ is the identity (the number of erasures in each row are already in non-increasing order). Thus, $\mathbf{H}_\sigma(5;(1,2,2,4))=\mathbf{H}(5;(1,2,2,4))$ and the matrix given by (11) is

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3} \\ 1 & \alpha^{-2} & \alpha^{-4} & \alpha^{-6} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha^{6} & \alpha^{5} & \alpha^{4} \\ 1 & \alpha^{5} & \alpha^{3} & \alpha \end{pmatrix}$$

Triangulating this last matrix, for instance, by Gaussian elimination, we obtain the matrix of (12)

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & \alpha^{2} & \alpha^{3} \\ 0 & 0 & 1 & \alpha^{3} \end{pmatrix}.$$

Now, with this matrix, we can obtain $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$ given by (13) as follows:

$$\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4)) = \left(\begin{array}{c|c|c|c} H_0 & \underline{0} & \underline{0} & \underline{0} \\ H_2 & H_2 & H_2 & H_2 \\ \hline \underline{0} & H_0 & \underline{0} & \underline{0} \\ \underline{0} & H_1 & \alpha^2 H_1 & \alpha^3 H_1 \\ \hline \underline{0} & \underline{0} & H_0 & \underline{0} \\ \underline{0} & \underline{0} & H_1 & \alpha^3 H_1 \\ \hline \underline{0} & \underline{0} & \underline{0} & H_0 \end{array}\right).$$

The encoding now proceeds like the decoding using this parity-check matrix $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$. Doing so, it can be verified that the encoded array coincides with the decoded array of Example 3.1. Since $\overset{\triangle}{\mathbf{H}}_\sigma(5;(1,2,2,4))$ is precomputed, the encoding starts at this point, saving the time necessary to produce this matrix, as in the general decoding algorithm.

4 Extending the codes

Next we discuss extending the length of the rows of the codes discussed in Section 2. Consider a t-level GC code $\mathcal{C}(n;\underline{u})$ as given by Definition 2.1. The codewords are $m \times n$ arrays over a field $GF(2^b)$, where $n < 2^b$. However, we can relax this requirement to $n-1 < 2^b$ and to $n-2 < 2^b$, by using extended and doubly extended RS codes. Specifically,

Definition 4.1 Let $m \le n$ be integers, and $\alpha \in GF(2^b)$ an element of order $\mathcal{O}(\alpha) \ge n$ (if α is primitive, $\mathcal{O}(\alpha) = 2^b - 1$). Consider the $u \times (n+1)$ and $u \times (n+2)$ matrices

$$H^{(1)}(u,n) = \left(\begin{array}{c|c} & 1 \\ & 0 \\ H(u,n;0) & 0 \\ & \vdots \\ & 0 \end{array} \right), \tag{14}$$

$$H^{(2)}(u,n) = \left(\begin{array}{c|cc} & 1 & 0 \\ & 0 & 1 \\ H(u,n;0) & 0 & 0 \\ & \vdots & \\ & 0 & 0 \end{array} \right), \tag{15}$$

$$H^{(1)}(u,n;\ell) = \left(\begin{array}{c|c} & 0 \\ & 0 \\ H(u,n;\ell) & \vdots \\ & 0 \end{array} \right) \tag{16}$$

and

$$H^{(2)}(u,n;\ell) = \left(\begin{array}{c|cc} & 0 & 0 \\ & 0 & 0 \\ H(u,n;\ell) & \vdots & \\ & 0 & 0 \end{array} \right), \tag{17}$$

where $H(u,n;\ell)$ is given by (1). Notice that $H^{(1)}(u,n)$ and $H^{(2)}(u,n)$ correspond to extended and doubly extended RS codes respectively.

Let $\underline{u}$ be a vector of non-decreasing integers and length $m = s_0 + s_1 + \cdots + s_{t-1}$ as given by (3). We say that, for $1 \le j \le 2$, the $[m(n+j), m(n+j) - \sum_{i=0}^{t-1} u_i s_i]$ code $\mathcal{C}^{(j)}(n;\underline{u})$ whose parity-check matrix is given by the $\left(\sum_{i=0}^{t-1} u_i s_i\right) \times m(n+j)$ matrix

$$\mathbf{H}^{(j)}(n;\underline{u}) = \left(\begin{array}{rcl} I_m & \otimes & H^{(j)}(u_0,n) \\ \hat{H}(s_{t-1},m;0) & \otimes & H^{(j)}(u_{t-1}-u_0,n;u_0) \\ \hat{H}(s_{t-2},m;\hat{s}_{t-1}) & \otimes & H^{(j)}(u_{t-2}-u_0,n;u_0) \\ \hat{H}(s_{t-3},m;\hat{s}_{t-2}) & \otimes & H^{(j)}(u_{t-3}-u_0,n;u_0) \\ & \vdots & \\ \hat{H}(s_1,m;\hat{s}_2) & \otimes & H^{(j)}(u_1-u_0,n;u_0) \end{array} \right) \tag{18}$$

is a t-level extended GC code for $j=1$ and a doubly extended GC code for $j=2$.

The extended and doubly extended codes $\mathcal{C}^{(j)}(n;\underline{u})$, $1 \le j \le 2$, have the same erasure-correcting as code $\mathcal{C}(n;\underline{u})$, since the decoding algorithm in this case involves repeated erasure-correction of extended and doubly extended RS codes, using an upper triangular matrix similar to (13).

Let us illustrate the extension with an example.

Example 4.1 Consider the case of Example 2.3, that is, assume that $t=3$, i.e.,

$$\underline{u}=\left(\overbrace{u_0,u_0,\ldots,u_0}^{s_0},\overbrace{u_1,u_1,\ldots,u_1}^{s_1},\overbrace{u_2,u_2,\ldots,u_2}^{s_2}\right)$$

and $\mathcal{C}(n;\underline{u})$ is a 3-level GC code.. Then, according to (18),

$$\mathbf{H}^{(j)}(n;\underline{u}) \quad = \quad \left(\begin{array}{rcl} I_{s_0+s_1+s_2} & \otimes & H^{(j)}(u_0,n) \\ \hat{H}(s_2,m;0) & \otimes & H^{(j)}(u_2-u_0,n;u_0) \\ \hat{H}(s_1,m;s_2) & \otimes & H^{(j)}(u_1-u_0,n;u_0) \end{array}\right),$$

where $H^{(j)}(u_0,n)$ and $H^{(j)}(u_i{-}u_0,n;u_0)$ are given by (14), (15), (16) and (17), and $\hat{H}(s_1,m;s_2)$ and $\hat{H}(s_2,m;0)$ are given by (2).

If we take $\underline{u}=(1,1,2,3)$, then the parity-check matrix of the 3-level extended GC code $\mathcal{C}^{(1)}(5;(1,1,2,3))$ is given by

$$\mathbf{H}^{(1)}(5;(1,1,2,3)) \quad = \quad \left(\begin{array}{rcl} I_4 & \otimes & H^{(1)}(1,5) \\ \hat{H}(1,4;0) & \otimes & H^{(1)}(2,5;1) \\ \hat{H}(1,4;1) & \otimes & H^{(1)}(1,5;1) \end{array}\right),$$

which explicitly gives

$$\left(\begin{array}{cccccc|cccccc|cccccc|cccccc}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 \\
\alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 \\
\hline
\alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & \alpha^3 & \alpha^2 & \alpha & 1 & \alpha^{-1} & 0 & \alpha^2 & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & 0 & \alpha & 1 & \alpha^{-1} & \alpha^{-2} & \alpha^{-3} & 0
\end{array}\right).$$

The extension gives more versatility in the choice of codes, and in some cases the advantages are crucial, as seen in the next example.

Example 4.2 Consider the situation of the LRC codes described in [17]. There, a [16,10,5] code over $GF(16)$ is presented. The data is divided into two sets of 5 symbols each. To each of these set of symbols a parity symbol is added, so that in each group, whenever a symbol is erased, it can be recovered using the remaining 5 symbols (in [17], this is called locality 5). In addition, independently of these two parity symbols, the 10 data symbols are encoded into a [14,10] RS code. It is not difficult to see that the [16,10] code so obtained has minimum distance 5 (Theorem 3 in [17]).

Now, consider code $\mathcal{C}^{(2)}(6;(2,4))$ over $GF(8)$ as given by Definition 4.1. According to Corollary 2.1, the minimum distance of this code is 5 also. By (18), its parity-check matrix is given by

$$\mathbf{H}^{(2)}(8;(2,4)) = \begin{pmatrix} I_2 & \otimes & H^{(2)}(2,6) \\ \hat{H}(1,2;0) & \otimes & H^{(2)}(2,6;2) \end{pmatrix}$$

$$= \left(\begin{array}{cccccc|cc||cccccc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 1 \\
\hline
\alpha^{10} & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 & 0 & \alpha^{10} & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 & 0 \\
\alpha^{15} & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & 0 & 0 & \alpha^{15} & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 & 1 & 0 & 0
\end{array}\right)$$

This code has locality 6 as opposed to 5 with respect to the LRC code. But it operates over $GF(8)$ as opposed to $GF(16)$, and the smaller field translates into faster and less complex operations. Also, the LRC code has locality with respect to only one erasure in the data, while the code presented has locality with respect to two erasures, without distinguishing between data and parity. In addition, although both codes have the same minimum distance $d=5$, the LRC cannot correct most combinations of 5 erasures, mainly, when the 5 erasures affect the data and the RS symbols, while $\mathcal{C}^{(2)}(6;(2,4))$ can correct most: the only case that it cannot handle occurs when the 5 erasures occur in the same half of the array, as illustrated below:

E		E	E	E		E	

Then, given 5 erasures, the probability that the LRC cannot correct them is $\binom{14}{5}/\binom{16}{5}=.46$, while the probability that $\mathcal{C}^{(2)}(6;(2,4))$ cannot correct 5 erasures is given by $(2)\binom{8}{5}/\binom{16}{5}=.026$.

Similarly, $\mathcal{C}^{(2)}(6;(2,4))$ can correct 6 erasures when there are 4 erasures in one row and two in the other.

5 Conclusions

We have presented a method of implementing Generalized Concatenated Codes as erasure-correcting codes over $m \times n$ arrays. We proved the fundamental properties of the codes and gave efficient encoding and decoding algorithms.

References

[1] M. Blaum, “Construction of PMDS and SD Codes extending RAID 5,” arXiv:1305.0032, April 2013.

[2] M. Blaum, J. L. Hafner and S. R. Hetzler, “Partial-MDS Codes and their Application to RAID Type of Architectures,” IEEE Trans. on Information Theory, vol.IT-59, pp. 4510-19, July 2013.

[3] M. Blaum, J. L. Hafner and S. R. Hetzler, “Nested multiple erasure correcting codes for storage arrays,” US Patent 8,433,979, April 2013.

[4] M. Blaum and J. S. Plank, “Construction of two SD Codes,” arXiv:1305.1221, May 2013.

[5] M. Blaum, J. S. Plank, M. Schwartz and E. Yaakobi, “Construction of Partial MDS (PMDS) and Sector-Disk (SD) Codes with Two Global Parity Symbols,” arXiv:1401.4715, January 2014.

[6] E. L. Blokh and V. V. Zyablov, “Coding of Generalized Concatenated Codes,” Problemy Peredachii Informatsii, Vol. 10(3), pp. 218–222, 1974.

[7] Y. Cassuto and A. Shokrollahi, “LDPC Codes for 2D Arrays,” IEEE Trans. on Information Theory, vol. IT-60, pp. 3279-91, June 2014.

[8] P. Gopalan, C. Huang, B. Jenkins and S. Yekhanin, “Explicit Maximally Recoverable Codes with Locality,” arXiv:1307.4150, July 2013.

[9] 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.

[10] 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.

[11] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S. Yekhanin, “Erasure Coding in Windows Azure Storage,” 2012 USENIX Annual Technical Conference, Boston, Massachussetts, June 2012.

[12] M. Li and P. C. Lee, “STAIR Codes: A General Family of Erasure Codes for Tolerating Device and Sector Failures in Practical Storage Systems,” 12th USENIX Conference on File and Storage Technologies (FAST 14), Santa Clara, CA, February 2014.

[13] F. J. MacWilliams and N. J. A. Sloane, “The Theory of Error-Correcting Codes,” North Holland, Amsterdam, 1977.

[14] J. S. Plank, M. Blaum and J. L. Hafner, "SD Codes: Erasure Codes Designed for How Storage Systems Really Fail," 11th USENIX Conference on File and Storage Technologies (FAST 13), Santa Clara, CA, February 2013.

[15] J. S. Plank and M. Blaum, "Sector-Disk (SD) Erasure Codes for Mixed Failure Modes in RAID Systems," ACM Transactions on Storage, Vol. 10, No. 1, Article 4, January 2014.

[16] R. M. Roth and P. O. Vontobel, "Coding for Combined Block-Symbol Error Correction," IEEE Trans. on Information Theory, vol.IT-60, pp. 2697-2713, May 2014.

[17] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data," Proceedings of VLDB, Vol. 6, No. 5, pp. 325–336, August 2013.

[18] I. Tamo, Z. Wang and J. Bruck, "Zigzag Codes: MDS Array Codes With Optimal Rebuilding," IEEE Trans. on Information Theory, vol. IT-59, pp. 1597–616, March 2013.

[19] 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.

[20] Wikipedia, http://en.wikipedia.org/wiki/Kronecker_product.

[21] V. A. Zinoviev, "Generalized cascade codes," Probl. Pered. Inform., vol. 12, no. 1, pp. 5-15, 1976.