Maximally Recoverable Codes with Hierarchical Locality

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

Aaditya M Nair
201302161
aaditya.mnair@research.iiit.ac.in

Signal Processing and Communications Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2020

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Maximally Recoverable Codes with Hierarchical Locality" by Aaditya M Nair, has been carried out under my supervision and is not submitted elsewhere for a degree.

14th July, 2020	V. Lalitha
Date	Adviser: Dr. Lalitha Vadlamani

To my Family, Friends and me.

Acknowledgments

First and foremost, I am grateful to my thesis advisor, Dr Lalitha Vadlamani for her guidance. Ever since I started working with her in summer of 2015, she has been a source of constant encouragement and support. It is not an overstatement to say that all this work would not have been possible without her.

Next, my gratitude goes towards my friends and family. There were times when frustration and hopelessness got to me. I am grateful that they were there and supported me through those times.

Finally, I am thankful to all my professors and peers at IIIT. They showed me all that an engineer could do and should be. My time here wouldn't have been the same without them.

Abstract

With the exponential increase in amount of data being stored in the cloud, a lot of distributed storage systems have started moving to erasure coding based storage schemes due to their advantage of providing the same reliability as replication with a smaller storage overhead. Locally recoverable codes have started being the defacto code of choice for their ability to correct a small number of erasures by only accessing a small number of other coordinates.

Maximally recoverable codes are a class of codes which recover from all potentially recoverable erasure patterns given the locality constraints of the code. In earlier works, these codes have been studied in the context of codes with locality. The notion of locality has been extended to hierarchical locality, which allows for locality to gradually increase in levels with the increase in the number of erasures. We consider the locality constraints imposed by codes with two-level hierarchical locality and define maximally recoverable codes with data-local and local hierarchical locality.

We derive certain properties related to their punctured codes and minimum distance. We give a procedure to construct hierarchical data-local MRCs from hierarchical local MRCs. We then provide a construction of hierarchical local MRCs for all parameters. We then provide an alternate construction using tensor product codes that construct the code in a smaller field. This construction works for a restricted set of parameters.

Research shows that the most commonly deployed LRCs have low number of global parities. Hence, we then consider three cases, with a small (and sometimes fixed) number of global and mid-level parities and provide a separate construction for them. These constructions use smaller fields still.

Contents

List of Figures

Chapter 1

Introduction

Almost any storage system that intends to store large amounts of data adds some form of redundancy to ensure the reliability of data against node failures (also termed as *erasures*). The easiest solution is to store exact copies of all the data (*replication*). This scheme, however, imposes a heavy cost at scale. Even a simple $2x$-replication will require double the storage just to tolerate upto one erasure.

Alternatively, one can opt to use erasure coding. In this scenario, the data is partitioned into k data symbols. Subsequently, using an erasure code, h redundancy symbols are computed as distinct linear combinations of few or all of the k data symbols. Each of the $k + h$ symbols are stored in a separate node. This can lead to a huge savings on storage. One has, to ensure reliability against the erasure of one symbol, add only one extra symbol. This scales really well with the number of data symbols. Only one extra symbols is required to ensure reliability against a single erasure irrespective of the value of k. Compare this with replication which scales linearly with k. Erasure codes [38] ensure that data can be recovered if a limited number of symbols are erased. If the code can recover from any h erasures, the code is called an MDS code. The figure 1.1 shows an MDS code with $k = 4$ and $h = 2$. This code can recover from the erasure of any two symbols.

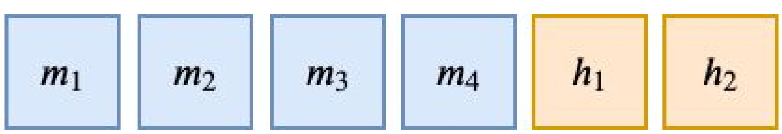

Figure 1.1: Erasure code with $k = 4$ and $h = 2$

MDS erasure codes suffer from one problem though; that of *repair bandwidth*. Repair bandwidth refers to the amount of data that needs to be downloaded into a new node to replace a failed node. Typically, since the h redundant symbols depend on all k data symbols, k symbols need to be accessed for any kind of repair. But this system doesn't take into account the fact that single erasure is a *lot* more common than multiple erasures. Hence it makes sense to optimise repair for small number of erasures. MDS erasure codes as described above require you to access all k symbols, be it a single erasure or of all h symbols. The figure 1.2 shows the repair process for a single failure of code in 1.1.

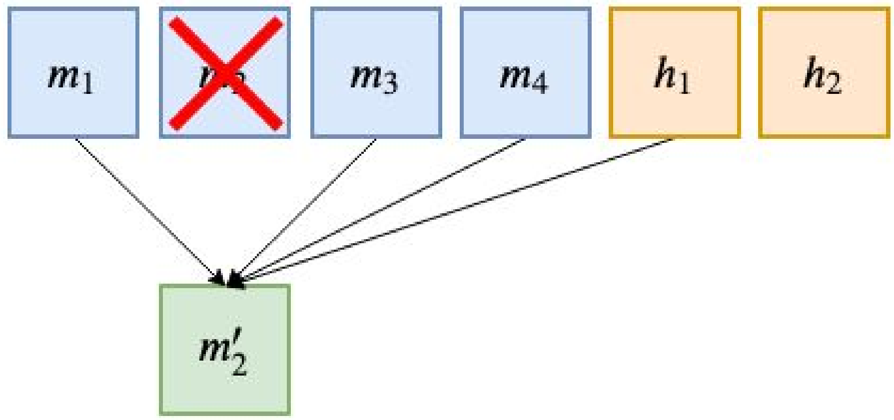

Figure 1.2: Repair for MDS erasure codes

To work around this problem, locality and Locally Recoverable Codes (LRCs) were introduced in [18]. We say that a certain symbol in a code has locality r if the erasure of that symbol can be recovered by accessing at-most r other symbols. One way of having faster repairs is to have $r \ll k$. In recent years, LRCs have emerged to be the erasure codes of choice for large scale systems. Notable users include Microsoft [27] and Hadoop [49] In LRCs, encoding happen in two distinct phases. In the first phase, h global parity symbols are added to k data symbols. Each global parity is a linear combination of all k symbols. In the next phase, these $k+h$ symbols are partitioned into groups of size r and another parity symbol (more generally, δ parity symbols) depending only on the elements of the group is added to each group. This ensures that a single erasure (δ erasures in a general case) can be recovered by only accessing r other symbols in the same group. We show an example of such an erasure code in 1.3 for the case of $\delta = 1$. The local groups are also highlighted. Repair for a single erasure is also shown in 1.4 for the same. For more erasures we have the h global parities and then need to be recovered by accessing all k symbols.

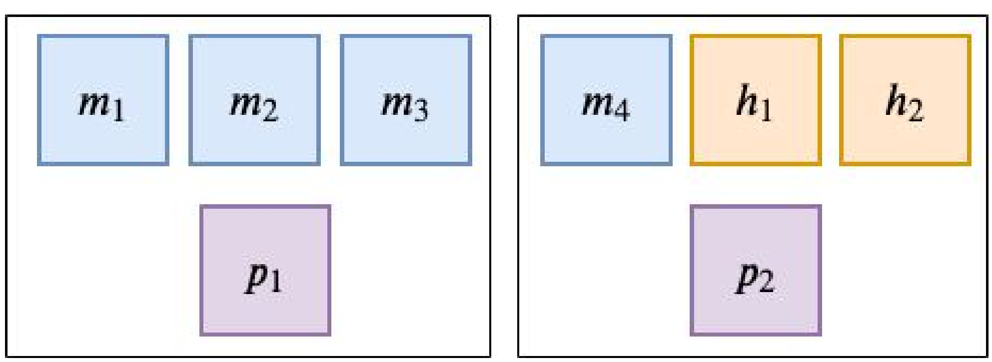

Figure 1.3: Locally Recoverable Code with $k = 4$, $h = 2$, $r = 3$ and $\delta = 1$

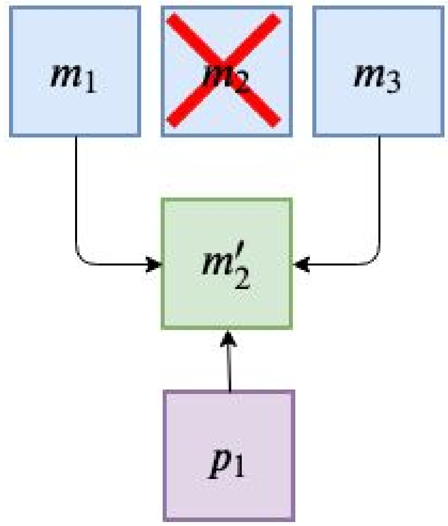

Figure 1.4: Repairing a single erasure

We haven't yet fully defined LRCs yet. Currently, we have only fixed the topology of the codes by specifying for all parity symbols, the other symbols they depend on. We will have to specify the coefficients in global parity symbols. Different choices for the coefficients will lead to codes with different erasure correction capabilities. The best case scenario is that the chosen coefficients ensure that all erasures that are information theoretically recoverable for the topology can be recovered from. Such codes are possible and are called Maximally Recoverable (MR) [10].

Now, with the topology of LRCs, for the first δ erasures, we can recover by accessing only r other symbols. But if we have even one more erasure, we will need to access all k symbols to recover. This is shown for the previous example in figure 1.5 where repairing two erasures requires $k = 4$ symbols. This abrupt jump from r to k can be considered excessive compared to the increase in the number of erasures.

We wanted to have a more fine grained control on the increase of locality with number of erasures. Such codes can be constructed and are called codes with *hierarchical locality* and are the subject of this thesis. These codes are created by extending LRCs themselves. Each r-sized groups from the LRC discussion (and the associated δ parities) is taken and split into sub-groups of size $r' < r$ and for each sub-group, a parity (or more generally, δ' parities) is added. In this case, one (δ') erasure can be corrected by accessing r' symbols. The next erasure however can be corrected by accessing r symbols. Even further erasures can be corrected by accessing all k symbols. This gradual increase from r' to r and then finally to k is what we were looking for.

We show a concrete example in figure 1.6 which extends the LRC in figure 1.3. The initial groups are denoted by the solid box while the sub-groups within it by the dotted box. This is an example of two-level hierarchical local code. In this text, we consider only such codes but we motivate the problem

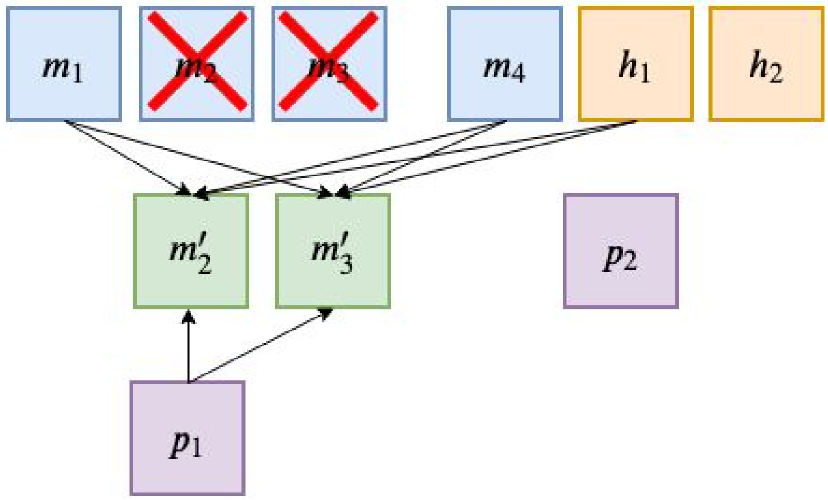

Figure 1.5: Abrupt increase in locality

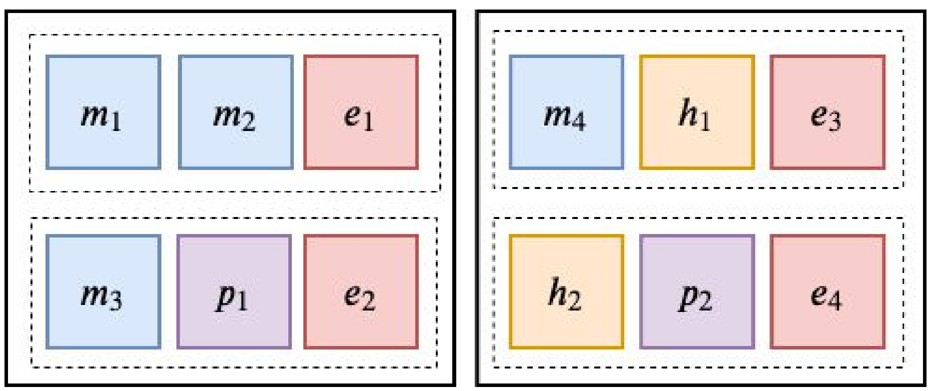

Figure 1.6: Hierarchical Local Code with $r' = 2$ and $\delta' = 1$

for higher levels of locality. We provide explicit constructions for two-level hierarchical local codes and prove that our constructions ensure that the code is also maximally recoverable.

Without explicit constructions, one is left to search for coefficients in a small field and verify that they satisfy the MR property. This process is prohibitively expensive even for small parameter sets. It is however shown that random searches work with high probability over very large fields. [4]

Mathematically, all encoding/decoding operations in erasure codes involve matrix multiplication and solving linear equations. These, in turn, are comprised of numerous finite field arithmetic operations. Therefore, it makes sense to define these codes in fields as small as possible so as to speed up encoding/decoding operations. To this end, we define an alternate construction for general MRCs with hierarchical locality based on tensor product codes that work on smaller fields (and a restricted set of parameters). It is also desirable to have fields of characteristic 2 so that it aligns naturally with the storage primitives on modern disks/SSDs.

We know that most LRCs deployed in practice only have a very small number of global parities typically two or three [27]. Hence we particularly consider these special cases in our work. We provide an alternate construction for these cases that can be constructed in a smaller field. All these constructions satisfy the MR property as well.

Maximally Recoverable Codes with hierarchical locality are also referred as Hierarchical Local MRCs (HLMRCs).

1.1 Organisation

This work is organised into five chapters.

This chapter (Chapter 1) introduces the topic and gives background on the everything that will be presented later.

Chapter 2 intends to present work has already been done on topics related to our work. This includes locally recoverable codes, hierarchical local codes and regenerating codes.

In chapter 3, we define and construct Maximally Recoverable codes with hierarchical locality. We also provide conditions under which such these constructions would be maximally recoverable. We also provide and explain some lemmas that were useful in the proofs provided.

Chapter 4 extends the previous chapter by providing an alternate construction for these codes (based on tensor product codes) which allows us to construct the code in a smaller field and a restricted parameter set. In addition, it also provides three more constructions for hierarchical local MRCs for low number of global parities.

Finally, chapter 5 concludes the results presented in the previous chapters and alludes to some currently open problems in the field that can later be tackled.

1.2 Notation

We use the following notations,

- For any integer n, $[n] = \{1, 2, 3, \ldots, n\}$.
- For any $E \subseteq [n]$, $\bar{E} = [n] - E$.
- For any $[n, k]$ code and $E \subseteq [n]$, $C|_E$ refers to the punctured code obtained by restricting C to the coordinates in E.
- For any $m \times n$ matrix H and $E \subseteq [n]$, $H|_E$ is an $m \times |E|$ matrix formed by restricting H to the columns in E.
- For any $m \times n$ matrix H and $E \subseteq [m]$, $H^{(E)}$ is an $|E| \times n$ matrix formed by restricting H to the rows in E.

- For any matrix H, H^T refers to its transpose.
- For a vector $\mathbf{x}$, $Supp(\mathbf{x})$ denotes the set $\{x_i : x_i \neq 0\}$.
- For a code C, $Supp(C)$ denotes all the non-zero symbols of C.
- $\mathbb{F}_q$ denotes a finite field of size q. Given that, $\mathbb{F}_{q^m}$ denotes an Extension Field of $\mathbb{F}_q$ of degree m.

Also, in several definitions to follow, we implicitly assume certain divisibility conditions that will be clear from the context.

Chapter 2

Related Work

2.1 Codes with locality

Research on codes with locality began with the need for better access efficiency. Although the term 'locality' wasn't used, Codes with locality were first discussed on [26] by Huang *et al.* and [23] by Han *et al.* and later on by Chen *et al.* in [10].

Huang called them Pyramid Codes while Chen called them *multi-protection group* (MPG) codes.

Pyramid Codes can be generated from any pre-existing (preferably MDS) code. The data symbols of the code is partitioned into disjoint groups. Then, redundant symbols are added to each group by projecting some of the redundant symbols from the original code to each of the groups. It is these new redundant symbols that provide the 'locality' aspect to the code.

For MPG codes, protection groups, each made up of a subset of the data symbols, were defined. Then, parity symbols were defined for each protection group by using symbols only from that group. Thus, locality was achieved at a protection group level.

It was only in [18] by Gopalan *et al.* where locality is formally defined. For any symbol $c_i \in C$, locality is defined as the smallest number r such that c_i is a linear combination of r other symbols in C. The code symbol is said to have r-locality. A code is then said to have r-information locality if all data symbols have locality r.

An extension to handle multiple erasures have been by Kamath *et al.* in [30].

Definition 1 ((r, ε)-Locality [30])**.** *For a code C, a symbol $c_i \in C$ has an (r, ε) locality if there exists a punctured code C_i such that,*

- $c_i \in Supp(C_i)$
- $|Supp(C_i)| \leq r + \varepsilon - 1$, *and*
- $d_{min}(C_i) \geq \varepsilon$

Note that the papers above talked about locality for the data-symbols and not for all the symbols in the code. We say that an $[n, k, d]$ code is said to have (r, ε) **information locality**, if k data symbols have (r, ε) locality. Similar to this, the code has an (r, ε) **all symbol locality** if all the n symbols have (r, ε) locality. Definitions for local and data-local codes were given in [17] in terms of the type (global or local) and the number of parities added to the data instead of the distance of the punctured code. They defined these codes for a single local parity but we extend it to any number.

Definition 2 (Data Local Code)**.** *Let C be a systematic $[n, k, d_{min}]$ code. We say that C is an $[k, r, h, \delta]$ data-local code if the following conditions are satisfied,*

- *$r|k$ and $n = k + \frac{k}{r}\delta + h$*
- *Data symbols are partitioned into $\frac{k}{r}$ **local groups** of size r. For each such group, there are δ local parity symbols.*
- *The remaining h global parity symbols may depend on all k symbols.*

Definition 3 (Local Code)**.** *Let C be a systematic $[n, k, d_{min}]$ code. We say that C is an $[k, r, h, \delta]$ local code if the following conditions are satisfied,*

- *$r|(k + h)$ and $n = k + \frac{k+h}{r}\delta + h$*
- *There are k data symbols and h global parity symbols where each global parity may depend on all data symbols.*
- *These $k + h$ symbols are partitioned into $\frac{k+h}{r}$ **local groups** of size r. For each such group, there are δ local parity symbols.*

Figure 2.1 shows an example lrc with $k = 4$, $r = 3$, $h = 2$ and $\delta = 1$.

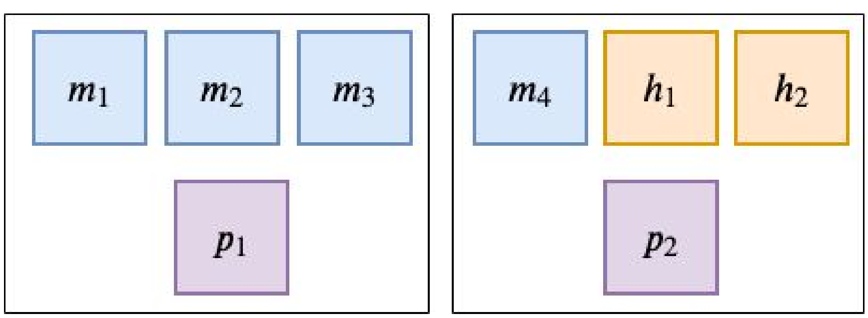

Figure 2.1: A $[4, 3, 2, 1]$ Local Code

The works of [40, 23] investigated all-symbol locality. Oggier *et al.* in [40] call these self-repairing codes. These codes are same as Local Codes albeit defined differently. These code definitions are based on linearized polynomials over a binary field. Similarly, Han *et al.* in [23] talk of local codes in the context of memory protected by erasure codes. They partition a memory 'line' into 'sub-lines' each protected by local codes. The 'line' itself has global parities similar to above definition.

With the exception of MPG codes, all the codes we have considered above only talk about codes with each subgroup having the same locality. But, depending on the application, we would want some subgroups to have a smaller locality than others because they contain hot data. Kadhe *et al.* in [28] called them, unsurprisingly, *codes with unequal locality* and actually show that the construction of Pyramid Codes can be adapted to allow for unequal locality.

To add another level to this, [8, 32] independently investigated codes with different localities and different number of global parities. Both of them define such codes in terms of (r_i, ε_i) code. Each subgroup had its own locality (r_i) and minimum distance (ε_i) associated with them. [8] added an additional constraint for each sub-group to be disjoint but [32] posed no such restriction.

For LRCs in general, simple block codes are extended to provide the final code. But there are some other constructions are derived from various special kinds of codes. Some of those are based on Maximum Rank Distance code [46] while some are based on generic Reed-Solomon codes [56].

A few cyclic codes have also been constructed that are also locally recoverable. The first constructions appeared in [19]. This work is further extended in [55] where the authors provide constructions of codes with $O(n)$ field size. This code also achieves the singleton bound. In addition to this, they study the locality of sub-field subcodes. On the other hand, [35] investigates the use of locality to improve the complexity for decoding a cyclic code.

The concept of locally recoverable codes was extended to non-linear codes as well. Papiliopoulos *et al.* [41] also provides such constructions with all symbol locality. [14] on the other hand, provides constructions only for information symbol locality.

2.1.1 Bounds on LRCs

Over time, a lot of bounds have been given on the structure of LRCs. These bounds extend from those on the minimum field size of the code symbols to those on the parameters of the code itself. Almost every different type of code construction provided above has a singleton like bound associated with it.

The first actual bound on the minimum distance of the code were discussed by Gopalan *et al.* in [18]. They provided a *singleton like* bound on the parameters of the code. Although Gopalan provided the bound for a single local parity, the work of Kamath *et al.* in [29] extended that to any number of local parities.

For an $[n, k, d]$ code with (r, ε) information locality, the parameters are related as follows,

$$d_{min} \leq n - k + 1 - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\varepsilon - 1). \tag{2.1}$$

A code achieving this bound is called an optimal LRC. Gopalan [18] also showed that for their constructions, an optimal code's topology is similar to our codes. An important class of codes that achieve this bound was given by Tamo *et al.* in [53]. They are generally referred to in literature as Tamo-Barg codes. Another feature of these constructions is that they require an $O(n)$ field size.

The above result is only valid for linear codes. An analogous result for $\varepsilon = 2$ and non linear codes are available in [41] and [14]. Another extension, this time to codes with vector alphabets, can be found in [52]. Note that these bounds are independent of the field size.

Contrary to above, [6] takes into account the size of the field and provides a tighter bound on the dimension of a local code. This result is even valid for non-linear code. Similar bounds also exist for cyclic LRCs as well. A Linear Programming based upper bound on the size of the code appears in [55]. Other results exist in [19] and [59].

Some research also provides a bound on the asymptotic rate of of code. An asymptotic Gilbert-Varshamov type lower bound for LRC is given in [54]. The constructions provided in [6] achieves this bound and for some select parameters. For a large enough field size, the authors of [36] provide a better bound than the one in [54]. They also prove that codes exist which meet their new bound.

2.2 Maximal Recoverability

Maximum Recoverability refers to the property of a code wherein all the erasures that should be information theoretically correctable can actually be corrected by the code. Various code constructions mentioned above also prove Maximal Recoverability(MR) under various conditions. For example, Huang *et al.* in [26] provided necessary conditions for a Pyramid Code to be MR. When the underlying code was an MDS one, this condition also become sufficient. The MR property was studied in MPG codes and various conditions (alphabet size and otherwise) were specified for a given code to be MR.

It is important to note that an optimal LRC is different from an MR LRC. Maximal Recoverability is a stronger condition to fulfill than ‘optimality’. An MR LRC is optimal as well as recovers from all erasures that are information theoretically recoverable. An ‘optimal’ LRC isn’t guaranteed to do so.

Maximally recoverable codes with locality are also known in literature as Partial-MDS (PMDS) codes. MRCs have been studied in the context of distributed storage systems and PMDS codes in the context of Solid State Drives. Although their definitions differ, both of them refer to the same thing.

Most of the previous literature studied the MR property as an extension to their previous constructions under restricted settings. Explicit constructions of were initially provided by Blaum *et al.* in [4] and [3] which worked for a small number of global parities ($h = 1, 2$) and a single local parity. This work was extended in [5] which provides the same solution for two local parities. This work was extended by Chen in [9] that provided constructions for ($h = 3$). In addition to this, Blaum in [4] also provided a solution that used a computer search to look for coefficients that satisfied the property.Finally, the work of Chen in MPG codes also proved that one can construct MRCs by picking the coefficients for global parities from a large enough field.

The first explicit constructions for MRCs for any parameters was provided in [17]. These codes also had the advantage of working over a much smaller field than the ones formed by randomly picking coefficients or a computer search. These codes worked on smaller field still in case of low global parities. They also provided the formal definition for MR property which we also use in our work.

Definition 4 (Maximum Recoverability)**.** *A $[k, r, h, \delta]$ local code (or data-local code) is also an MRC if for every $E \subset [n]$ such that E is obtained by picking δ coordinates from each of the local groups, $C|_{\bar{E}}$ yields an $[k + h, k]$ MDS code.*

Figure 2.2 shows the criteria for code in figure 2.1 to be maximally recoverable.

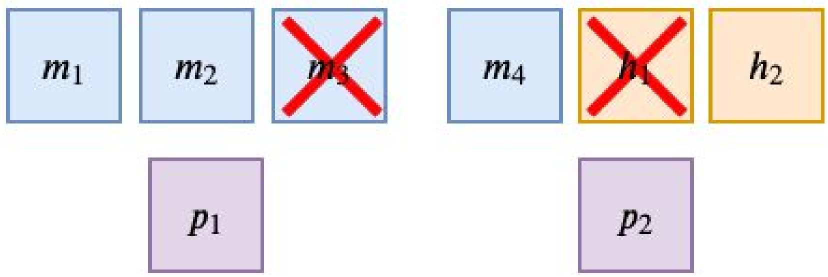

Figure 2.2: Puncture $\delta = 1$ points per local code. The remaining code should be MDS

$[k, r, h, \delta]$ local MRC is optimum with respect to minimum distance bound in (2.1). The minimum distance of a $[k, r, h, \delta]$ local MRC is given by

$$d_{min} = h + \delta + 1 + \left\lfloor \frac{h}{r} \right\rfloor \delta. \tag{2.2}$$

This is obtained by replacing the value of n in equation (2.1) from the one in definition 3 and $\epsilon = \delta + 1$.

This work was later extended by other people. [7] also provided a general construction using sum-rank codes and linearized polynomials. Gabrys *et al.* in [15] improved Gopalan's work [17] by providing constructions in a smaller field. Other works include [42] which study only data local MRCs while [24, 21] study codes in small field sizes but can only correct a single local erasure.

The first lower bound on field size of MR codes in any topology was given in [16] and was continued in [31] and then in [20].

2.3 Hierarchical Locality

Hierarchical Codes were first discussed in [13] in the context of peer-to-peer networks. In such networks, nodes leaving and joining is considered business as usual and happens a lot more often than in the case of distributed storage. Hence, vanilla erasure coding algorithms are quite inefficient in terms of bandwidth used because every time a node leaves, all k data symbols will need to be transferred to reconstruct a lost piece. Hence Hierarchical Codes were discussed as a means of reducing repair bandwidth. They validated their work by taking real network traces from PlanetLab and KAD networks.

The first theoretical study of a Hierarchical Code topology was undertaken in the work of Sasidharan *et al.* in [48]. They first defined the notion of Hierarchical Code and provided a bound on minimum distance for such a code. These will be discussed in upcoming chapters. In addition to that they provided explicit constructions for local and data local hierarchical codes. The all-symbol local constructions involved finding a working polynomial field which contained message polynomials and evaluating them at appropriate points. The data symbol code hierarchical code, on the other hand, involved partitioning the parity symbols for a regular erasure code. These codes are optimal under certain parameter sets. In addition to this, separate constructions are provided in [1] by Ballentine *et al.*. These codes use covering maps on algebraic curves to generate codes. These codes are also optimal under certain conditions.

As far as we are aware of, our work [39] is the only work that discusses Hierarchical Codes in the context of their Maximal Recoverability Property. We provide explicit constructions for code Hierarchical MRCs and provide a code in smaller field size in case of low global parities. In an independent and parallel work by Martinez-Penas *et al.* [37], a class of MRCs called multi-layer MRCs have been introduced. Our work on Hierarchical local MRCs can be considered to form a subclass of these multi-layer MRCs. The field size bounds provided in both the papers is also analogous.

2.4 Regenerating Codes

Regenerating Codes represent an entire different way to optimise the repair for common failure scenarios. Instead of downloading data from a smaller number of nodes, regenerating codes aim to download a smaller amount of data from various nodes thereby reducing the *repair bandwidth* all the while retaining the properties of MDS codes. There are two principal classes of such codes, MBR and MSR codes representing the two extremes of the trade-off known as *storage-repair bandwidth* (SRB) tradeoff [12].

A *Minimum Storage Regenerating* (MSR) code aims to minimise the number of code symbols that need to be stored in a single node. These code have received a lot of attention due them being MDS codes and the fact that the storage overhead can be minimised as much as desired. Shah *et al.* in [50] talk about *Interference Alignment* and how they are necessary in exact-repair MSR codes. They also use IA techniques to construct MSR codes. The product matrix framework is introduced in [44] and provides constructions for some sets of parameters.

A *Minimum Bandwidth Regenerating* (MBR) code aims to minimise the amount of data transferred between nodes during repair. The first constructions of explicit MBR codes was provided in [43]. The product matrix construction from [44] provides the second, general construction of such codes. Construction for codes that asymptotically reach the MSR or MBR point as k increases is provided in [45]. These codes can be constructed over any field size given that the *file-size* is large enough.

This naturally leads to the question of whether LRCs can be made of regenerating codes which benefit from both the small repair degree for LRCs and the low repair bandwidth for regenerating codes.

The Locally Regenerating Codes independently introduced in [30] and [46] answer the question in affirmative. These codes can be viewed as LRCs but the local codes are actually regenerating codes.

Some other interesting works aimed at reducing the repair bandwidth of MDS codes. These techniques rely on viewing scalar symbols of an MDS code as a vectors with entries in a sub-field. This line of work began with the work of Shanmugam *et al.* in [51] who showed the existence of an improved repair scheme for systematic MDS codes with only two parities. One of the more interesting work is on Generalised Reed Solomon codes by Guruswami *et al.* [22] for all symbol repair. They used techniques of polynomial interpolation to bring down the repair bandwidth for a single erasure to the order of number of data symbols. Following this, [11] addresses bandwidth efficient repair of Reed Solomon codes from multiple erasures. This result is extended to include general MDS codes in [2].

2.5 Codes in Practice

A lot of these codes have been implemented in practice too. The work of [27] investigated how LRCs have been implemented in Windows Azure Storage which uses both an erasure code (*for cold data*) and 3-replication (*for hot data*). Similarly in [49], the authors provide an LRC for the Hadoop File System that can locally repair a single erasure. They have also released a module implementing the same.

HDFS implementation of a class of array MDS codes called HashTag codes have been discussed in [33]. These codes are designed for efficient repair of systematic codes and allows for low sub-packetization levels at the cost of higher repair bandwidth. The theoretical foundations for these codes were presented in [34].

One of the first works that investigated performance of regenerating codes is NCCloud [25]. This was built on top of a functional MSR code with two parities. Ceph is a popular distributed block and object store. Ceph has various erasure codes builtin that provided different levels of reliability. Codes include vanilla reed solomon codes as well as LRCs. In addition to that, they have one implementation of a regenerating code by Vajha *et al.* [57]. In this work, the authors create a regenerating code by using 'pairwise coupling across multiple stacked layers of any single MDS code'. These codes happen to be the first practical implementation of regenerating codes that are optimal, have uniform repair for data or parity symbols and can repair multiple erasures.

Chapter 3

Codes with Hierarchical Locality

In this chapter, we define and look at the properties of codes with hierarchical locality. We narrow down on MRCs with hierarchical locality and provide a general construction for them. We provide requirements for our construction to be a MRC with hierarchical locality. In the final section, we provide values that satisfy those requirements using BCH codes.

The concept of *locality* has been extended to hierarchical locality in [48]. In the case of (r, ε) locality, if there are more than $\varepsilon - 1$ erasures, then the code offers no locality. In the case of codes with hierarchical locality, the locality constraints are such that with the increase in the number of erasures, the locality increases in steps. The following is the definition of code with two-level hierarchical locality.

Definition 5. *An* $[n, k, d_{min}]$ *linear code* $\mathcal{C}$ *is a code with* hierarchical locality *having parameters* $[(r_1, \varepsilon_1), (r_2, \varepsilon_2)]$ *if for every symbol* c_i*,* $1 \leq i \leq n$*, there exists a punctured code* $\mathcal{C}_i$ *such that* $c_i \in Supp(C_i)$ *and the following conditions hold,*

- $|Supp(\mathcal{C}_i)| \leq r_1 + \varepsilon_1 - 1$
- $d_{min}(\mathcal{C}_i) \geq \varepsilon_1$ *and*
- $\mathcal{C}_i$ *is a code with* (r_2, ε_2) *locality.*

An upper bound on the minimum distance of a code with two-level hierarchical locality is given by

$$d \leq n - k + 1 - (\left\lceil \frac{k}{r_2} \right\rceil - 1)(\varepsilon_2 - 1) - (\left\lceil \frac{k}{r_1} \right\rceil - 1)(\varepsilon_1 - \varepsilon_2). \tag{3.1}$$

We also redefine hierarchical local codes very similar to how [17] defines codes with locality. We also provide conditions for maximal recoverability

Definition 6 (Hierarchical Data Local Code)**.** *We define a* $[k, r_1, r_2, h_1, h_2, \delta]$ *hierarchical data local (HDL) code of length* $n = k + h_1 + \frac{k}{r_1}(h_2 + \frac{r_1}{r_2}\delta)$ *as follows:*

- *The code symbols $c_1, \dots, c_n$ satisfy h_1 global parities given by*

$$\sum_{j=1}^{n} u_j^{(\ell)} c_j = 0, \;\; 1 \leq \ell \leq h_1$$

.

- *The first $n - h_1$ code symbols are partitioned into $t_1 = \frac{k}{r_1}$ groups $A_i, 1 \leq i \leq t_1$ such that $|A_i| = r_1 + h_2 + \frac{r_1}{r_2}\delta = n_1$. The code symbols in the i^{th} group, $1 \leq i \leq t_1$ satisfy the following h_2 mid-level parities*

$$\sum_{j=1}^{n_1} v_{i,j}^{(\ell)} c_{(i-1)n_1+j} = 0, \;\; 1 \leq \ell \leq h_2$$

.

- *The first $n_1 - h_2$ code symbols of the i^{th} group, $1 \leq i \leq t_1$ are partitioned into $t_2 = \frac{r_1}{r_2}$ groups $B_{i,s}, 1 \leq i \leq t_1, 1 \leq s \leq t_2$ such that $|B_{i,s}| = r_2 + \delta = n_2$. The code symbols in the $(i,s)^{th}$ group, $1 \leq i \leq t_1, 1 \leq s \leq t_2$ satisfy the following δ local parities*

$$\sum_{j=1}^{n_2} w_{i,s,j}^{(\ell)} c_{(i-1)n_1+(s-1)n_2+j} = 0, \;\; 1 \leq \ell \leq \delta$$

.

Definition 7 (Hierarchical Local Code)**.** *We define a $[k, r_1, r_2, h_1, h_2, \delta]$ hierarchical local (HL) code of length $n = k + h_1 + \frac{k+h_1}{r_1}(h_2 + \frac{r_1+h_2}{r_2}\delta)$ as follows:*

- *The code symbols $c_1, \dots, c_n$ satisfy h_1 global parities given by*

$$\sum_{j=1}^{n} u_j^{(\ell)} c_j = 0, \;\; 1 \leq \ell \leq h_1$$

  The figure 3.1 shows an example of such a code.

- *The n code symbols are partitioned into $t_1 = \frac{k+h_1}{r_1}$ groups $A_i, 1 \leq i \leq t_1$ such that $|A_i| = r_1 + h_2 + \frac{r_1+h_2}{r_2}\delta = n_1$. The code symbols in the i^{th} group, $1 \leq i \leq t_1$ satisfy the following h_2 mid-level parities*

$$\sum_{j=1}^{n_1} v_{i,j}^{(\ell)} c_{(i-1)n_1+j} = 0, \;\; 1 \leq \ell \leq h_2$$

  Figure 3.2 shows how this partition can be made.

- *The n_1 code symbols of the i^{th} group, $1 \leq i \leq t_1$ are partitioned into $t_2 = \frac{r_1+h_2}{r_2}$ groups $B_{i,s}, 1 \leq i \leq t_1, 1 \leq s \leq t_2$ such that $|B_{i,s}| = r_2 + \delta = n_2$. The code symbols in the $(i,s)^{th}$ group, $1 \leq i \leq t_1, 1 \leq s \leq t_2$ satisfy the following δ local parities*

$$\sum_{j=1}^{n_2} w_{i,s,j}^{(\ell)} c_{(i-1)n_1+(s-1)n_2+j} = 0, \ \ 1 \leq \ell \leq \delta$$

Figure 3.3 shows a local code for a mid-level partition.

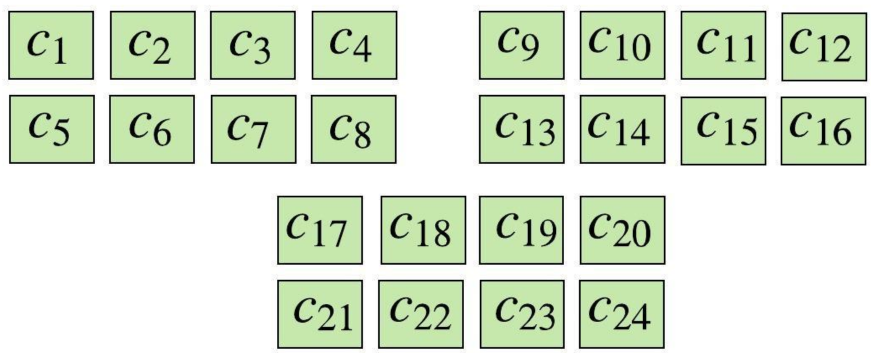

Figure 3.1: A $[9, 4, 3, 3, 2, 1]$ code. All symbols satisfy $h_1 = 3$ parities

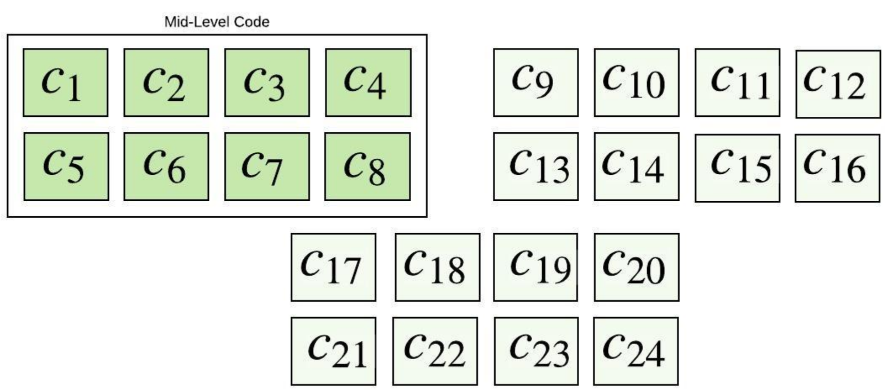

Figure 3.2: All symbols are partitioned into $t_1 = 3$ groups each satisfying $h_2 = 2$ mid-level parities.

Example 1. *We demonstrate the structure of the parity check matrix for an $[k = 5, r_1 = 3, r_2 = 2, h_1 = 1, h_2 = 1, \delta = 2]$ HL code. The length of the code is $n = k + h_1 + \frac{k+h_1}{r_1}(h_2 + \frac{r_1+h_2}{r_2}\delta) = 16$. The*

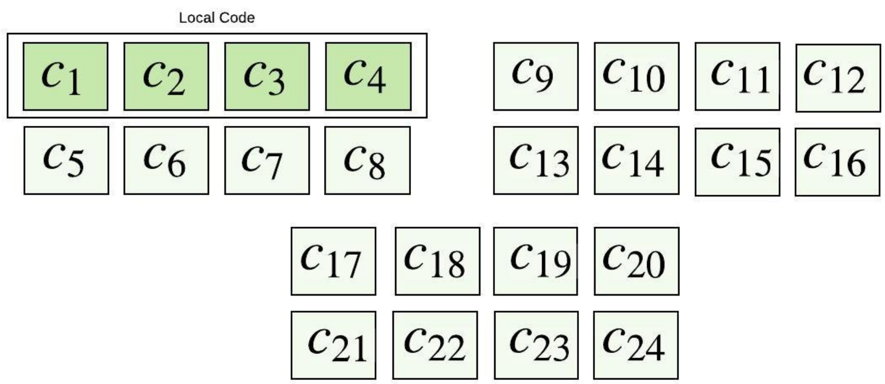

Figure 3.3: A mid-level code is partitioned into $t_2 = 2$ groups each satisfying $\delta = 1$ parities

parity check matrix of the code is given below:

$$H = \left[\begin{array}{c|c} M_{1,1} \qquad & \\ \qquad M_{1,2} & \\ \cline{1-1} N_1 & \\ \hline & M_{2,1} \qquad \\ & \qquad M_{2,2} \\ \cline{2-2} & N_2 \\ \hline \multicolumn{2}{c}{P} \end{array}\right]$$

$$M_{i,j} = \begin{bmatrix} w_{i,j,1}^{(1)} & w_{i,j,2}^{(1)} & w_{i,j,3}^{(1)} & w_{i,j,4}^{(1)} \\ w_{i,j,1}^{(2)} & w_{i,j,2}^{(2)} & w_{i,j,3}^{(2)} & w_{i,j,4}^{(2)} \end{bmatrix},$$

$$N_i = \begin{bmatrix} v_{i,1}^{(1)} & v_{i,2}^{(1)} & \dots & v_{i,8}^{(1)} \end{bmatrix}$$

$$P = \begin{bmatrix} u_1^{(1)} & \dots & u_{16}^{(1)} \end{bmatrix}$$

In this example, sub-matrix P denotes the global parity condition. The code is partitioned into $t_1 = 2$ groups and mid-level parities are added. This is denoted by the matrices $N_i,\ i = 1, 2$ for each group. Each mid-level group is then partitioned into $t_2 = 2$ groups and $\delta = 2$ local parities are added. This is denoted by the matrices $M_{i,j},\ i, j = 1, 2$

Definition 8 (Maximum Recoverability)**.** *Let $\mathcal{C}$ be a $[k, r_1, r_2, h_1, h_2, \delta]$ HDL/HL code. Then C is maximally recoverable if for any set $E \subset [n]$ such that*

- $|E| = k + h_1$,

- $|E \bigcap B_{i,s}| \leq r_2 \; \forall \; i, s$ *and*

- $|E \bigcap A_i| = r_1 \; \forall \; i$,

the punctured code $\mathcal{C}|_E$ *is a* $[k + h_1, k, h_1 + 1]$ *MDS code.*

Figure 3.4 shows one such E.

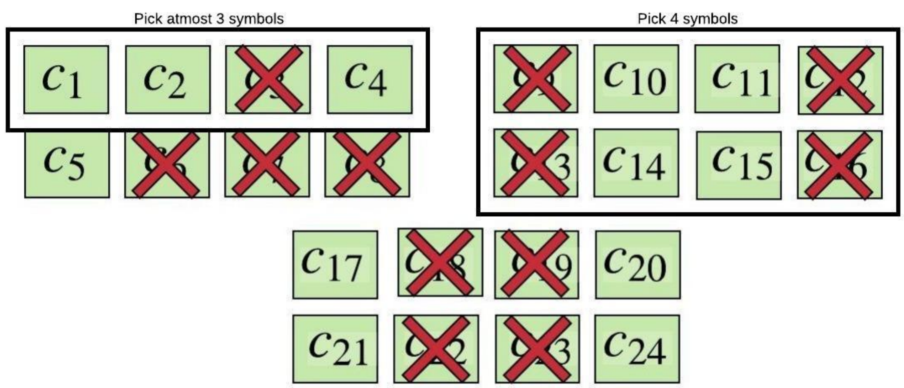

Figure 3.4: MR condition for a $[9, 4, 3, 3, 2, 1]$ code. The remaining symbols form a MDS code

Theorem 3.0.1. *The erasure pattern described in Definition 8 are all the erasures that are information theoretically correctible.*

Proof: We show the proof for an Hierarchical Local MRC. The proof will remain the same for a data local one.

Definition 8 means that if we puncture h_2 coordinates per A_i after we puncture δ coordinates per $B_{i,s}$, the leftover code is an MDS one.

Assume that some local group can handle one more erasure. Hence the total number of erasures that the code can handle equals,

$$\begin{aligned}
nerasures =& \delta\left(\frac{k+h_1}{r_1}\right)\left(\frac{r_1+h_2}{r_2}\right)+1 && \delta \text{ erasures per local code and one more} \\
&+ h_2\left(\frac{k+h_1}{r_1}\right) && (h_1 \text{ mid level parities}) \\
&+ h_1 && (\text{global parities}) \\
=& h_1 + \left(\frac{k+h_1}{r_1}\right)\left(h_2+\frac{r_1+h_2}{r_2}\right)+1
\end{aligned}$$

But note that the total redundancy in the code is

$$n-k = h_1 + \left(\frac{k+h_1}{r_1}\right)\left(h_2+\frac{r_1+h_2}{r_2}\right)+1$$

By our assumption, the code can now recover from more errors that there is redundancy available in the code. Which is an obvious contradiction,information theoretically. Hence the pattern described by Definition 8 are the only ones correctible. $\square$

3.1 Properties of MRCs with Hierarchical Locality

In this section, we will derive two properties of MRC with hierarchical locality. We will show that the middle codes of a HDL/HL-MRC have to be data-local and local MRC respectively. Also, we derive the minimum distance of HDL MRC.

Lemma 3.1.1. *Consider a $[k, r_1, r_2, h_1, h_2, \delta]$ HDL-MRC $\mathcal{C}$. Let $A_i, 1 \leq i \leq t_1$ be the supports of the middle codes as defined in Definition 6. Then, for each i, $\mathcal{C}_{A_i}$ is a $[r_1, r_2, h_2, \delta]$ data-local MRC.*

Proof: Suppose not. This means that for some i, the middle code $\mathcal{C}_{A_i}$ is not a $[r_1, r_2, h_2, \delta]$ data-local MRC. By the definition of data-local MRC, we have that there exists a set $E_1 \subset A_i$ such that $|E_1| = r_1 + h_2$ and $\mathcal{C}_{E_1}$ is not an $[r_1 + h_2, r_1, h_2 + 1]$ MDS code. This implies that there exists a subset $E' \subset E_1$ such that $|E'| = r_1$ and $\text{rank}(G|_{E'}) < r_1$. We can extend the set E' to obtain a set $E \subset [n]$, $|E| = k + h_1$ which satisfies the conditions in the definition of HDL-MRC. The resulting punctured code $\mathcal{C}_E$ cannot be MDS since there exists an $r_1 < k$ sized subset of E such that $\text{rank}(G|_{E'}) < r_1$. $\square$

Lemma 3.1.2. *Consider a $[k, r_1, r_2, h_1, h_2, \delta]$ HL-MRC $\mathcal{C}$. Let $A_i, 1 \leq i \leq t_1$ be the supports of the middle codes as defined in Definition 7. Then, for each i, $\mathcal{C}_{A_i}$ is a $[r_1, r_2, h_2, \delta]$ local MRC.*

Proof: Proof is similar to the proof of Lemma 3.1.1. $\square$

3.1.1 Minimum Distance of HDL-MRC

Lemma 3.1.3. *The minimum distance of a $[k, r_1, r_2, h_1, h_2, \delta]$ HDL-MRC is given by $d = h_1+h_2+\delta+1$.*

Proof: Based on the definition of HDL-MRC, it can be seen that the $[k, r_1, r_2, h_1, h_2, \delta]$ HDL-MRC is a code with hierarchical locality as per Definition 5 with k, r_1, r_2 being the same, $\delta_2 - 1 = \delta$, $\delta_1 = h_2 + \delta + 1$ and $n = k + h_1 + \frac{k}{r_1}(h_2 + \frac{r_1}{r_2}\delta)$ Substituting these parameters in the minimum distance bound in (3.1), we have that $d \leq h_1 + h_2 + \delta + 1$.

By Lemma 3.1.1, we know that $\mathcal{C}_{A_i}$ is a $[r_1, r_2, h_2, \delta]$ data-local MRC. The minimum distance of $\mathcal{C}_{A_i}$ (from (2.2)) is $h_2 + \delta + 1$. Thus, the middle code itself can recover from any $h_2 + \delta$ erasures. The additional h_1 erasures can be shown to be extended to a set E (consisting of k additional non-erased symbols) which satisfies the conditions in Definition 8. Since, the punctured code $\mathcal{C}|_E$ is a $[k + h_1, k, h_1 + 1]$ MDS code, it can be used to recover the h_1 erasures. Hence, $[k, r_1, r_2, h_1, h_2, \delta]$ HDL-MRC can recover from any $h_1 + h_2 + \delta$ erasures. This means that $d \geq h_1 + h_2 + \delta + 1$.

Combining the two results we get what we propose in the lemma. □

3.1.2 Deriving HDL-MRC from HL-MRC

In this section, we give a method to derive any HDL-MRC from a HL-MRC. Assume an $[k, r_1, r_2, h_1, h_2, \delta]$ HL-MRC $\mathcal{C}$. Consider a particular set E of $k + h_1$ symbols satisfying the conditions given in Definition 8. We will refer to the elements of set E as "primary symbols". By the definition of HL-MRC, the code $\mathcal{C}$ when punctured to E results in a $[k + h_1, k, h_1 + 1]$ MDS code. Hence, any k subset of E forms an information set. We will refer to the first k symbols of E as "data symbols" and the rest h_1 symbols as global parities. The symbols in $[n] \setminus E$ will be referred to as parity symbols (mid-level parities and local parities) and it can be observed that the parity symbols can be obtained as linear combinations of data symbols.

- If $r_1 \mid h_1$ and $r_2 \mid h_2$,

  1. For A_i, $\frac{k}{r_1} < i \leq \frac{k+h_1}{r_1}$, drop all the parity symbols, including h_2 mid-level parities per A_i as well as the δ local parities per $B_{i,s} \subset A_i$. As a result, we would be left with h_1 "primary symbols" in the local groups A_i, $\frac{k}{r_1} < i \leq \frac{k+h_1}{r_1}$. These form the global parities of the HDL-MRC. This step ensures that mid-level and local parities formed from global parities are dropped.
  2. For each $B_{i,s}$, $1 \leq i \leq \frac{k}{r_1}$, $s > \frac{r_1}{r_2}$, drop the δ local parities. This step ensures that local parities formed from mid-level parities are dropped.

  This results in an $[k, r_1, r_2, h_1, h_2, \delta]$ HDL-MRC.

- If $r_1 \nmid h_1$ and $r_2 \mid h_2$,

1. From the groups $A_i, \lfloor \frac{k}{r_1} \rfloor + 1 < i \leq \frac{k+h_1}{r_1}$, drop all the parity symbols, including h_2 mid-level parities per A_i as well as the δ local parities per $B_{i,s} \subset A_i$.
2. For each $B_{i,s},\ 1 \leq i \leq \lfloor \frac{k}{r_1} \rfloor,\ s > \frac{r_1}{r_2}$, drop the δ local parities.
3. Drop the $k - \lfloor \frac{k}{r_1} \rfloor r_1$ data symbols in $A_i, i = \lfloor \frac{k}{r_1} \rfloor + 1$ and recalculate all the parities (local, mid-level and global) by setting these data symbols as zero in the linear combinations.

This results in an $[\lfloor \frac{k}{r_1} \rfloor r_1, r_1, r_2, h_1, h_2, \delta]$ HDL-MRC.

For the case of $r_2 \nmid h_2$, HDL-MRC can be derived from HL-MRC using similar techniques as above. Hence, in the rest of the chapter, we will discuss the construction of HL-MRC.

3.2 General Construction

In this section, we will present a general construction of $[k, r_1, r_2, h_1, h_2, \delta]$ HL-MRC. First, we will provide the structure of the code and then derive necessary and sufficient conditions for the code to be HL-MRC. Finally, we will apply a known result of BCH codes to complete the construction.

Definition 9 (k-wise independence)**.** *A multiset $S \subseteq \mathbb{F}$ is k-wise independent over $\mathbb{F}$ if for every set $T \subseteq S$ such that $|T| \leq k$, T is linearly independent over $\mathbb{F}$*

Lemma 3.2.1. *Let $\mathbb{F}_{q^t}$ be an extension of $\mathbb{F}_q$. Let $a_1, a_2, \ldots, a_n$ be elements of $\mathbb{F}_{q^t}$. The following matrix*

$$\begin{bmatrix} a_1 & a_2 & a_3 & \ldots & a_n \\ a_1^q & a_2^q & a_3^q & \ldots & a_n^q \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ a_1^{q^{k-1}} & a_2^{q^{k-1}} & a_3^{q^{k-1}} & \ldots & a_n^{q^{k-1}} \end{bmatrix}$$

is the generator matrix of a $[n, k]$ MDS code if and only if $a_1, a_2, \ldots, a_n$ are k-wise linearly independent over $\mathbb{F}_q$.

Proof: Directly follows from [15, Lemma 3] □

Construction 3.2.2. *The structure of the parity check matrix(H) of a $[k, r_1, r_2, h_1, h_2, \delta]$ HL-MRC is given by*

$$H = \begin{bmatrix} H_0 & & & \\ & H_0 & & \\ & & \ddots & \\ & & & H_0 \\ H_1 & H_2 & \ldots & H_{t_1} \end{bmatrix}$$

where,

$$H_0 = \begin{bmatrix} M_0 & & & \\ & M_0 & & \\ & & \ddots & \\ & & & M_0 \\ M_1 & M_2 & \dots & M_{t_2} \end{bmatrix}$$

Here, H_0 is an $(t_2\delta + h_2) \times n_1$ matrix and $H_i, 1 \le i \le t_1$ are an $h_1 \times n_1$ matrix. H_0 is then further subdivided into M_i. M_0 has the dimensions $\delta \times n_2$ and $M_i, 1 \le i \le t_2$ is an $h_2 \times n_2$ matrix.

Assume q to be a prime power such that $q \ge n$, $\mathbb{F}_{q^{m_1}}$ be an extension field of $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$ is an extension field of $\mathbb{F}_{q^{m_1}}$, where $m_1 \mid m$.

In this case, the construction is given by the following.

$$M_0 = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & \beta & \beta^2 & \dots & \beta^{n_2-1} \\ 0 & \beta^2 & \beta^4 & \dots & \beta^{2(n_2-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \beta^{\delta-1} & \beta^{2(\delta-1)} & \dots & \beta^{(\delta-1)(n_2-1)} \end{bmatrix},$$

where $\beta \in \mathbb{F}_q$ is a primitive element.

$$M_i = \begin{bmatrix} \alpha_{i,1} & \alpha_{i,2} & \dots & \alpha_{i,n_2} \\ \alpha_{i,1}^q & \alpha_{i,2}^q & \dots & \alpha_{i,n_2}^q \\ \vdots & \vdots & \dots & \vdots \\ \alpha_{i,1}^{q^{h_2-1}} & \alpha_{i,2}^{q^{h_2-1}} & \dots & \alpha_{i,n_2}^{q^{h_2-1}} \end{bmatrix},$$

where $i \in [t_2]$, $\alpha_{i,j} \in \mathbb{F}_{q^{m_1}}, 1 \le i \le t_2, 1 \le j \le n_2$.

$$H_i = [H_{i,1}\ H_{i,2} \dots H_{i,t_2}]$$

$$H_{i,s} = \begin{bmatrix} \lambda_{i,s,1} & \lambda_{i,s,2} & \dots & \lambda_{i,s,n_2} \\ \lambda_{i,s,1}^{q^{m_1}} & \lambda_{i,s,2}^{q^{m_1}} & \dots & \lambda_{i,s,n_2}^{q^{m_1}} \\ \vdots & \vdots & \dots & \vdots \\ \lambda_{i,s,1}^{q^{m_1(h_1-1)}} & \lambda_{i,s,2}^{q^{m_1(h_1-1)}} & \dots & \lambda_{i,s,n_2}^{q^{m_1(h_1-1)}} \end{bmatrix},$$

where $i \in [t_1], s \in [t_2]$, $\lambda_{i,s,j} \in \mathbb{F}_{q^m}, 1 \le i \le t_1, 1 \le s \le t_2, 1 \le j \le n_2$.

A (δ, h_2) erasure pattern is defined by the following two sets:

- Δ is a three dimensional array of indices with the first dimension i indexing the middle code and hence $1 \leq i \leq t_1$, the second dimension s indexing the local code and hence $1 \leq s \leq t_2$. The third dimension j varies from 1 to δ and used to index the δ coordinates which are erased in the $(i,s)^{\text{th}}$ group. Let $e \in [n]$ denote the actual index of the erased coordinate in the code and $e \in B_{i,s}$, then we set $\Delta_{i,s,j} = (e \mod n_2) + 1$. $\Delta_{i,s}$ is used to denote the vector of δ coordinates which are erased in the $(i,s)^{\text{th}}$ group. $\bar{\Delta}_{i,s}$ is used to denote the complement of $\Delta_{i,s}$ in the set $[n_2]$.
- Γ is a two dimensional array of indices with the first dimension i indexing the middle code and hence $1 \leq i \leq t_1$. The second dimension j varies from 1 to h_2 and used to index the additional h_2 coordinates which are erased in the i^{th} group. Let $e \in [n]$ denote the actual index of the erased coordinate in the code and $e \in A_i$, then we set $\Gamma_{i,j} = (e \mod n_1) + 1$. Γ_i is used to denote the vector of h_2 coordinates which are erased in the i^{th} group. $\bar{\Gamma}_i$ is used to denote the complement of Γ_i in the set $[n_1] \setminus (\cup_{s=1}^{t_2} \Delta_{i,s})$.

We define some matrices and sets based on the parameters of the construction, which will be useful in proving the subsequent necessary and sufficient condition for the construction to be HL-MRC. Here, $\alpha_{s,\Delta_{i,s}}$ denotes the set $\{\alpha_{s,j} \mid j \in \Delta_{i,s}\}$.

$$\begin{aligned} L_{i,s} &= (M_0|_{\Delta_{i,s}})^{-1} M_0|_{\bar{\Delta}_{i,s}} \\ \Psi_i &= \{\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s}, 1 \leq s \leq t_2\} \\ &= \{\Psi_{i,\Gamma_i},\ \Psi_{i,\bar{\Gamma}_i}\} \\ &= \{\psi_{i,1}, \ldots, \psi_{i,h_2}, \psi_{i,h_2+1}, \ldots, \psi_{i,r_1+h_2}\} \end{aligned}$$

The above equalities follow by noting that the $\cup_{s=1}^{t_2} \bar{\Delta}_{i,s} = \Gamma_i \cup \bar{\Gamma}_i$. We will refer to the elements in Ψ_{i,Γ_i} by $\{\psi_{i,1}, \ldots, \psi_{i,h_2}\}$ and those in $\Psi_{i,\bar{\Gamma}_i}$ by $\{\psi_{i,h_2+1}, \ldots, \psi_{i,r_1+h_2}\}$. Consider the following matrix based on the elements of Ψ_i,

$$\begin{aligned} F_i &= [F_i|_{\Gamma_i}\ \ F_i|_{\bar{\Gamma}_i}] \\ &= \begin{bmatrix} \psi_{i,1} & \psi_{i,2} & \cdots & \psi_{i,r_1+h_2} \\ \psi_{i,1}^q & \psi_{i,2}^q & \cdots & \psi_{i,r_1+h_2}^q \\ \vdots & \vdots & \cdots & \vdots \\ \psi_{i,1}^{q^{h_2-1}} & \psi_{i,2}^{q^{h_2-1}} & \cdots & \psi_{i,r_1+h_2}^{q^{h_2-1}} \end{bmatrix}, \end{aligned}$$

And

$$\begin{aligned} \Phi_i &= \{\lambda_{i,s,\bar{\Delta}_{i,s}} + \lambda_{i,s,\Delta_{i,s}} L_{i,s}, 1 \leq s \leq t_2\} \\ &= \{\Phi_{i,\Gamma_i},\ \Phi_{i,\bar{\Gamma}_i}\} \\ &= \{\phi_{i,1}, \ldots, \phi_{i,h_2}, \phi_{i,h_2+1}, \ldots, \phi_{i,r_1+h_2}\} \end{aligned}$$

Let

$$Z_i = (F_i|_{\Gamma_i})^{-1} F_i|_{\bar{\Gamma}_i}$$

. Finally, the set

$$\Theta = \{\Phi_{i,\bar{\Gamma}_i} + \Phi_{i,\Gamma_i} Z_i, 1 \leq i \leq t_1\}$$

.

Theorem 3.2.3. *The code described in Construction 3.2.2 is a $[k, r_1, r_2, h_1, h_2, \delta]$ HL-MRC only if, for any (δ, h_2) erasure pattern, the following two conditions are satisfied:*

1. *Each $\Psi_i, 1 \leq i \leq t_1$ is h_2-wise independent over $\mathbb{F}_q$.*

2. *Θ is h_1-wise independent over $\mathbb{F}_{q^{m_1}}$.*

Proof: By Lemma 3.1.2, we have that $\mathcal{C}$ is a HL-MRC only if the $\mathcal{C}|_{A_i}$ is a $[r_1, r_2, h_2, \delta]$ local MRC. By the definition of local MRC, a code is a $[r_1, r_2, h_2, \delta]$ local MRC, if after puncturing δ coordinates in each of the $\frac{r_1+h_2}{r_2}$ local groups, the resultant code is $[r_1 + h_2, r_1, h_2 + 1]$ MDS code.

The puncturing on a set of coordinates in the code is equivalent to shortening on the same set of coordinates in the dual code. Shortening on a set of coordinates in the dual code can be performed by zeroing the corresponding coordinates in the parity check matrix by row reduction. To prove that $\mathcal{C}|_{A_i}$ is a $[r_1, r_2, h_2, \delta]$ local MRC, we need to show that certain punctured codes are MDS (Definition 4). We will equivalently that the shortened codes of the dual code are MDS.

Consider the coordinates corresponding to $(i, s)^{\text{th}}$ group in the parity check matrix. The sub-matrix of interest in this case is the following:

$$\left[\begin{array}{c|c} M_0|_{\Delta_{i,s}} & M_0|_{\bar{\Delta}_{i,s}} \\ \hline \alpha_{s,\Delta_{i,s}} & \alpha_{s,\bar{\Delta}_{i,s}} \\ \alpha^q_{s,\Delta_{i,s}} & \alpha^q_{s,\bar{\Delta}_{i,s}} \\ \vdots & \vdots \\ \alpha^{q^{h_2-1}}_{s,\Delta_{i,s}} & \alpha^{q^{h_2-1}}_{s,\bar{\Delta}_{i,s}} \end{array}\right],$$

Where $\alpha^q_{s,\Delta_{i,s}}$ is the vector obtained by taking q^{th} power of each element in the vector. Applying row reduction to the above matrix, we have

$$\left[\begin{array}{c|c} M_0|_{\Delta_{i,s}} & M_0|_{\bar{\Delta}_{i,s}} \\ \hline \mathbf{0} & \alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s} \\ \mathbf{0} & (\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s})^q \\ \vdots & \vdots \\ \mathbf{0} & (\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s})^{q^{h_2-1}} \end{array}\right].$$

Note that $L_{i,s}$ can be pushed into the power of q since the elements of $L_{i,s}$ are in $\mathbb{F}_q$. After row reducing δ coordinates from each of the $\frac{r_1+h_2}{r_2}$ local groups in A_i, the resultant parity check matrix is

F_i. Applying Lemma 3.2.1, F_i forms the generator matrix of an MDS code if and only if the set Ψ_i is h_2-wise independent over $\mathbb{F}_q$. The shortening of the code above is applicable to mid-level parities. Now, we will apply similar shortening in two steps to global parities. The sub-matrix of interest in this case is the following:

$$\left[\begin{array}{c|c} M_0|_{\Delta_{i,s}} & M_0|_{\bar{\Delta}_{i,s}} \\ \hline \alpha_{s,\Delta_{i,s}} & \alpha_{s,\bar{\Delta}_{i,s}} \\ \alpha^{q}_{s,\Delta_{i,s}} & \alpha^{q}_{s,\bar{\Delta}_{i,s}} \\ \vdots & \vdots \\ \alpha^{q^{h_2-1}}_{s,\Delta_{i,s}} & \alpha^{q^{h_2-1}}_{s,\bar{\Delta}_{i,s}} \\ \hline \lambda_{i,s,\Delta_{i,s}} & \lambda_{i,s,\bar{\Delta}_{i,s}} \\ \lambda^{q^{m_1}}_{i,s,\Delta_{i,s}} & \lambda^{q^{m_1}}_{i,s,\bar{\Delta}_{i,s}} \\ \vdots & \vdots \\ \lambda^{q^{m_1(h_1-1)}}_{i,s,\Delta_{i,s}} & \lambda^{q^{m_1(h_1-1)}}_{i,s,\bar{\Delta}_{i,s}} \end{array}\right]$$

Applying row reduction to the above matrix, we have

$$\left[\begin{array}{c|c} M_0|_{\Delta_{i,s}} & M_0|_{\bar{\Delta}_{i,s}} \\ \hline \mathbf{0} & \alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s} \\ \mathbf{0} & (\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s})^{q} \\ \vdots & \vdots \\ \mathbf{0} & (\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s})^{q^{h_2-1}} \\ \hline \mathbf{0} & \lambda_{i,s,\bar{\Delta}_{i,s}} + \lambda_{i,s,\Delta_{i,s}} L_{i,s} \\ \mathbf{0} & (\lambda_{i,s,\bar{\Delta}_{i,s}} + \lambda_{i,s,\Delta_{i,s}} L_{i,s})^{q^{m_1}} \\ \vdots & \vdots \\ \mathbf{0} & (\lambda_{i,s,\bar{\Delta}_{i,s}} + \lambda_{i,s,\Delta_{i,s}} L_{i,s})^{q^{m_1(h_1-1)}} \end{array}\right].$$

To apply row reduction again, we consider the following sub-matrix obtained by deleting the zero columns and aggregating the non-zero columns from the $\frac{r_1+h_2}{r_2}$ groups,

$$\left[\begin{array}{c|c} F_i|_{\Gamma_i} & F_i|_{\bar{\Gamma}_i} \\ \hline \Phi_{i,\Gamma_i} & \Phi_{i,\bar{\Gamma}_i} \\ \Phi^{q^{m_1}}_{i,\Gamma_i} & \Phi^{q^{m_1}}_{i,\bar{\Gamma}_i} \\ \vdots & \vdots \\ \Phi^{q^{m_1(h_1-1)}}_{i,\Gamma_i} & \Phi^{q^{m_1(h_1-1)}}_{i,\bar{\Gamma}_i} \end{array}\right].$$

Applying row reduction to the above matrix, we have

$$\left[\begin{array}{c|c} F_i|_{\Gamma_i} & F_i|_{\bar{\Gamma}_i} \\ \hline \mathbf{0} & \Phi_{i,\bar{\Gamma}_i} + \Phi_{i,\Gamma_i} Z_i \\ \mathbf{0} & (\Phi_{i,\bar{\Gamma}_i} + \Phi_{i,\Gamma_i} Z_i)^{q^{m_1}} \\ \vdots & \vdots \\ \mathbf{0} & (\Phi_{i,\bar{\Gamma}_i} + \Phi_{i,\Gamma_i} Z_i)^{q^{m_1(h_1-1)}} \end{array}\right].$$

Note that Z_i can be pushed into the power of q^{m_1} since the elements of Z_i are in $\mathbb{F}_{q^{m_1}}$. Applying Lemma 3.2.1, the row reduced matrix above forms the generator matrix of an MDS code if and only if the set Θ is h_1-wise independent over $\mathbb{F}_{q^{m_1}}$. □

Lemma 3.2.4. *For any (δ, h_2) erasure pattern,*

- *For each i, $\Psi_i = \{\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s}, 1 \le s \le t_2\}$ is h_2-wise independent over $\mathbb{F}_q$ if the set*

  $$\{\alpha_{s,j}, 1 \le s \le t_2, 1 \le j \le n_2\}$$

  is $(\delta+1)h_2$-wise independent over $\mathbb{F}_q$.

- *$\Theta = \{\Phi_{i,\bar{\Gamma}_i} + \Phi_{i,\Gamma_i} Z_i, 1 \le i \le t_1\}$ is h_1-wise independent over $\mathbb{F}_{q^{m_1}}$ if the set*

  $$\{\lambda_{i,s,j}, 1 \le i \le t_1, 1 \le s \le t_2, 1 \le j \le n_2\}$$

  is $(\delta+1)(h_2+1)h_1$-wise independent over $\mathbb{F}_{q^{m_1}}$.

Proof: Since the size of matrix $L_{i,s}$ is $\delta \times (n_2 - \delta)$, each element of Ψ_i can be a $\mathbb{F}_q$-linear combination of at-most $\delta + 1$ different $\alpha_{s,j}$. Consider $\mathbb{F}_q$-linear combination of h_2 elements in Ψ_i. The linear combination will have at most $(\delta+1)h_2$ different $\alpha_{s,j}$. Thus, if the set $\{\alpha_{s,j}\}$ is $(\delta+1)h_2$-wise independent over $\mathbb{F}_q$, then Ψ_i is h_2-wise independent over $\mathbb{F}_q$.

To prove the second part, we note that each element of Φ_i is a linear combination of at most $\delta + 1$ different $\lambda_{i,s,j}$. Since the size of the matrix Z_i is $h_2 \times r_1$, each element of Θ can be a $\mathbb{F}_{q^{m_1}}$-linear combination of at-most $(\delta+1)(h_2+1)$ different $\lambda_{i,s,j}$. Consider $\mathbb{F}_{q^{m_1}}$-linear combination of h_1 elements in Θ. The linear combination will have at most $(\delta+1)(h_2+1)h_1$ different $\lambda_{i,s,j}$. Thus, if the set $\{\lambda_{i,s,j}\}$ is $(\delta+1)(h_2+1)h_1$-wise independent over $\mathbb{F}_{q^{m_1}}$, then Θ is h_1-wise independent over $\mathbb{F}_{q^{m_1}}$. □

3.3 Picking Values

We will design the $\{\alpha_{s,j}\}$ and $\{\lambda_{i,s,j}\}$ based on the Lemma 3.2.4 so that the field size is minimum possible. We will pick these based on the following two properties:

- **Property 1:** The columns of parity check matrix of an $[n, k, d]$ linear code over $\mathbb{F}_q$ can be interpreted as n elements over $\mathbb{F}_{q^{n-k}}$ which are $(d-1)$-wise linear independent over $\mathbb{F}_q$.

- **Property 2:** There exists $[n = q^t - 1, k, d]$ BCH codes over $\mathbb{F}_q$ [47], where the parameters are related as
$$n - k = 1 + \left\lceil \frac{q-1}{q}(d-2) \right\rceil \lceil \log_2(n) \rceil. \tag{3.2}$$

Theorem 3.3.1. *The code in Construction 3.2.2 is a $[k, r_1, r_2, h_1, h_2, \delta]$ HL-MRC if the parameters are picked as follows:*

1. *q is the smallest prime power greater than n_2.*

2. *m_1 is chosen based on the following relation:*
$$m_1 = 1 + \left\lceil \frac{q-1}{q}((\delta+1)h_2 - 1) \right\rceil \lceil \log_q(n_2 t_2) \rceil.$$

3. *$n_2 t_2$ elements $\{\alpha_{s,j}\}$ over $\mathbb{F}_{q^{m_1}}$ are set to be the columns of parity check matrix of the BCH code over $\mathbb{F}_q$ with parameters $[n = q^{\lceil \log_q(n_2t_2) \rceil} - 1, q^{\lceil \log_q(n_2t_2) \rceil} - 1 - m_1, (\delta+1)h_2 + 1]$.*

4. *m is chosen to be the smallest integer dividing m_1 based on the following relation:*
$$m \geq 1 + \left\lceil \frac{q^{m_1}-1}{q^{m_1}}((\delta+1)(h_2+1)h_1 - 1) \right\rceil \lceil \log_{q^{m_1}}(n) \rceil.$$

5. *n elements $\{\lambda_{i,s,j}\}$ over $\mathbb{F}_{q^m}$ are set to be the columns of parity check matrix of the BCH code over $\mathbb{F}_{q^{m_1}}$ with parameters $[n = q^{m_1 \lceil \log_{q^{m_1}}(n) \rceil} - 1, q^{m_1 \lceil \log_{q^{m_1}}(n) \rceil} - 1 - m, (\delta+1)(h_2+1)h_1 + 1]$*
.

Proof: The proof follows from Lemma 3.2.4 and Properties 1 and 2. □

Example 2. *Let $\alpha_{i,j} \in \mathbb{F}_{q^{m_1}}$ with $\mathbb{F}_{q^{m_1}}$ being an extension of $\mathbb{F}_q$. Assume we need 8 values of $\{\alpha_{i,j}\}$ and for them to be 6 wise independent. We will pick $\alpha_{i,j}$ as the columns of the parity check matrix of an $[n, k, d]$ code.*

We know from Property 1 that for a code in $\mathbb{F}_q$ to have minimum distance d, the columns of the PCM must be $d-1$ wise independent. Hence for this code, we set $d = 7$.

The columns of the PCM can be considered to be elements $\in \mathbb{F}_{q^{n-k}}$ that are $d-1$-wise independent in $\mathbb{F}_q$. With $\alpha_{i,j} \in \mathbb{F}_{q^{m_1}}$, we set $n - k = m_1$.

Since we want 8 values of $\alpha_{i,j}$ we set the value of $t = \lceil log_q 8 \rceil$ so as to have the smallest possible code as mentioned in Property 1. With this, we now have all the values to fill in equation 3.2. We do that to find the value for m_1.

Chapter 4

More Optimal Constructions

In the previous chapter we used BCH codes to generate the values that work with the construction. In this chapter, we describe the erasure correcting properties of tensor product codes and show how we can use them to generate those values in a much smaller field. We then go on to provide bespoke constructions for the following configurations of global and mid-level parities:

- 1 Global parity and any number of mid-level parities.
- 1 Global parity and 1 mid-level parity
- 2 Global parities and 1 mid-level parity.

These constructions use even smaller fields.

4.1 Product Constructions

[58] talks about codes that are formed by taking a tensor product of two distinct codes. Such codes are shown to correct a special class of erasures. We use that to find $\{\alpha_{s,j}\}$ and $\{\lambda_{i,s,j}\}$ that satisfy Theorem 3.2.3.

We first define a Tensor Product Code and its error correcting capability.

Definition 10. *Let C_1 be an $[n, n-\rho]$ linear code in $\mathbb{F}_q$ which can correct e_1 erasures. Also, C_2 is an $[m, m-s]$ code in $\mathbb{F}_{q^\rho}$ that can correct e_2 erasures. An $[nm, nm-s\rho]$ code C in $\mathbb{F}_q$ is called the tensor product code of C_1 and C_2 if*

$$\forall x \in C_1 \textit{ and } y \in C_2, \ \ y \otimes x \in C$$

where $y \otimes x$ is the tensor product of x and y in $\mathbb{F}_q$.

If the parity check matrices for C_1 and C_2 are H_1 and H_2 respectively and they are written as,

$$H_1 = \begin{bmatrix} h^{(1)}_{1,1} & h^{(1)}_{1,2} & \dots & h^{(1)}_{1,n} \end{bmatrix} \text{ and } H_2 = \begin{bmatrix} h^{(2)}_{1,1} & \dots & h^{(2)}_{1,m} \\ \vdots & \ddots & \vdots \\ h^{(2)}_{s,1} & \dots & h^{(2)}_{s,m} \end{bmatrix}.$$

Then the parity check matrix for C is as follows,

$$H = \begin{bmatrix} h^{(2)}_{1,1}h^{(1)}_{1,1} & h^{(2)}_{1,1}h^{(1)}_{1,2} & \dots & h^{(2)}_{1,1}h^{(1)}_{1,n} & \dots & \dots & h^{(2)}_{1,m}h^{(1)}_{1,1} & h^{(2)}_{1,m}h^{(1)}_{1,2} & \dots & h^{(2)}_{1,m}h^{(1)}_{1,n} \\ \vdots & \vdots & \dots & \vdots & \dots & \dots & \vdots & \vdots & \dots & \vdots \\ h^{(2)}_{s,1}h^{(1)}_{1,1} & h^{(2)}_{s,1}h^{(1)}_{1,2} & \dots & h^{(2)}_{s,1}h^{(1)}_{1,n} & \dots & \dots & h^{(2)}_{s,m}h^{(1)}_{1,1} & h^{(2)}_{s,m}h^{(1)}_{1,2} & \dots & h^{(2)}_{s,m}h^{(1)}_{1,n} \end{bmatrix}. \quad (4.1)$$

Theorem 4.1.1. *C_1, C_2 and C are as defined above. If the code-words in C are considered to be consisting of m sub-blocks with each sub-block containing n symbols, C will correct all erasure patterns where,*

- *Atmost e_2 sub-blocks are affected.*
- *Atmost e_1 erasures in each affected sub-block.*

Proof: For a code C defined as above, let E be an erasure pattern such that,

- e_2 sub-blocks are affected. Let the sublocks be $k_1, k_2, \dots k_{e_2}$.
- There are e_1 erasures in each affected sub-block. Let the erased coordinates in subblock k_i be $l^{(k_i)}_1, l^{(k_i)}_2, \dots l^{(k_i)}_{e_1}$.

The sub-matrix $H^{(E)}$ associated with this erasure pattern will be,

$$\begin{bmatrix} h^{(2)}_{1,k_1}h^{(1)}_{1,l^{(k_1)}_1} & h^{(2)}_{1,k_1}h^{(1)}_{1,l^{(k_1)}_2} & \dots & h^{(2)}_{1,k_1}h^{(1)}_{1,l^{(k_1)}_{e_1}} & \dots & \dots & h^{(2)}_{1,k_{e_2}}h^{(1)}_{1,l^{(k_{e_2})}_1} & h^{(2)}_{1,k_{e_2}}h^{(1)}_{1,l^{(k_{e_2})}_2} & \dots & h^{(2)}_{1,k_{e_2}}h^{(1)}_{1,l^{(k_{e_2})}_{e_1}} \\ \vdots & \vdots & \dots & \vdots & \dots & \dots & \vdots & \vdots & \dots & \vdots \\ h^{(2)}_{s,k_1}h^{(1)}_{1,l^{(k_1)}_1} & h^{(2)}_{s,k_1}h^{(1)}_{1,l^{(k_1)}_2} & \dots & h^{(2)}_{s,k_1}h^{(1)}_{1,l^{(k_i)}_{e_1}} & \dots & \dots & h^{(2)}_{s,n}h^{(1)}_{1,l^{(k_{e_2})}_1} & h^{(2)}_{s,n}h^{(1)}_{1,l^{(k_{e_2})}_2} & \dots & h^{(2)}_{s,n}h^{(1)}_{1,l^{(k_{e_2})}_{e_1}} \end{bmatrix}.$$

If this matrix is invertible, then the erasure pattern is correctible. This implies that the columns of this matrix should be linearly independent. Hence,

$$\sum_{i=1}^{e_2}\sum_{j=1}^{e_1} a_{k_i,l_j} \begin{pmatrix} h^{(2)}_{1,k_i}h^{(1)}_{1,l^{(k_i)}_j} \\ \vdots \\ h^{(2)}_{s,k_i}h^{(1)}_{1,l^{(k_i)}_j} \end{pmatrix} \neq 0, \; \forall\, a_{i,j} \neq 0$$

$$\Rightarrow \sum_{i=1}^{e_2} \begin{pmatrix} h^{(2)}_{1,k_i} \\ \vdots \\ h^{(2)}_{s,k_i} \end{pmatrix} \cdot \sum_{j=1}^{e_1} a_{k_i,l_j} \begin{pmatrix} h^{(1)}_{1,l^{(k_i)}_j} \\ \vdots \\ h^{(1)}_{1,l^{(k_i)}_j} \end{pmatrix} \neq 0.$$

In the above equation, the inner summation is the linear combinations of e_1 columns of the Parity Check Matrix for C_1. This means that the inner summation is never equal to zero since C_1 can correct e_1 erasures.

With that, the outer summation becomes the linear combination of e_2 columns of the PCM for C_2. Hence this sum is not equal to zero as well. Hence, the matrix is invertible and the erasure pattern, correctable. $\square$

We say that an $[nm, nm - s\rho]$ tensor product code $C \subseteq \mathbb{F}_q^{m \times n}$ is an $[m, n; e_2, e_1]$ erasure correcting code if it can correct any erasure pattern of the form $\mathbf{E} = (E_1, \ldots, E_m)$ where for $i \in [m]$ and $E_i \subseteq [n]$ and

- $|\{i : E_i \neq \emptyset\}| \leq e_2$.
- for $i \in [m], |E_i| \leq e_1$.

Theorem 4.1.2. *Let C_{TP} be an $[t_2, n_2; h_2, h_2+\delta]$ erasure correcting code with $t_2 > h_2$ and $n_2 > h_2+\delta$ over $\mathbb{F}_q$ with redundancy m_1 and the parity check matrix $H_{TP} = (\alpha_{1,1}, \ldots, \alpha_{t_2,n_2}) \in (\mathbb{F}_{q^{m_1}})^{t_2 n_2}$. Then the set $\{\alpha_{i,j}\}$ chosen as columns of H_{TP} guarantees that $\Psi_i, 1 \leq i \leq t_1$ is h_2 wise independent over $\mathbb{F}_q$.*

Proof: $\Psi_i = \{\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s}, 1 \leq s \leq t_2\}$ where s denotes the local group for the mid level code i.

We also know that $\alpha_{s,\bar{\Delta}_{i,s}}$ and $\alpha_{s,\Delta_{i,s}}$ are vectors of size r_2 and δ respectively. Hence expanding them into their component vectors,

$$\alpha_{s,\bar{\Delta}_{i,s}} = [\alpha_{s,1'}, \ldots, \alpha_{s,r_2'}]$$
$$\alpha_{s,\Delta_{i,s}} = [\alpha_{s,1}, \ldots, \alpha_{s,\delta}]$$

Also,

$$L_{i,s} = \begin{bmatrix} a_{s,1,1} & \ldots & a_{s,r_2,1} \\ \vdots & \vdots & \vdots \\ a_{s,1,\delta} & \ldots & a_{s,r_2,\delta} \end{bmatrix}$$

Therefore,

$$\alpha_{s,\bar{\Delta}_{i,s}} + \alpha_{s,\Delta_{i,s}} L_{i,s} = \{a_{s,j,1}\alpha_{s,1} + \ldots + a_{s,j,\delta}\alpha_{s,\delta} + \alpha_{s,j'},\ 1 \leq j \leq r_2\}\}$$

Finally,

$$\Psi_i = \{a_{s,j,1}\alpha_{s,1} + \ldots + a_{s,j,\delta}\alpha_{s,\delta} + \alpha_{s,j'},\ 1 \leq j \leq r_2,\ 1 \leq s \leq t_2\}$$

Now, we need to show that this set is h_2 wise independent. We pick $[e_1; e_2]$ for the code C_{TP} in such a way that the $\alpha_{i,j}$ in the linear combination of h_2 elements of Ψ_i fall within the correctable erasure pattern for C_{TP} and hence will be linearly independent.

To do this we consider two distinct edge cases of picking elements for Ψ_i and combine the results:

1. In this case, we pick elements from distinct local groups (the s parameter for each element is different). Hence we have h_2 distinct groups and $(\delta + 1)$ $\alpha_{i,j}$ from each group. Hence this pattern can be corrected by an $[t_2, n_2; h_2, \delta + 1]$ code.

2. Now we pick elements from the same local group (s parameter remaining same in each element). Here note that there are $h_2 + \delta$ distinct $\alpha_{i,j}$. Hence, this pattern can be corrected by a $[t_2, n_2; 1, h_2 + \delta]$ code.

Combining both of these, we can say that an $[t_2, n_2; h_2, h_2 + \delta]$ code ensures that Ψ_i is h_2-wise independent. □

Now, for C as an $[t_2, n_2; h_2, h_2 + \delta]$ code, we can also enumerate the constituent codes C_1 and C_2 whose tensor products will give C.

- Let the first code C_1 be an $[n_2, n_2 - (h_2 + \delta), (h_2 + \delta) + 1]$ MDS code in $\mathbb{F}_q$. This implies $n_2 > (h_2 + \delta)$.
- The second code C_2 is an $[t_2, t_2 - h_2, h_2 + 1]$ MDS code in $\mathbb{F}_{q^{h_2+\delta}}$. Hence, $t_2 > h_2$.

With this, we know the value of $m_1 = h_2(h_2 + \delta)$.

Theorem 4.1.3. *Let C_{TP} be an $[t_1, t_2 n_2; h_1, (h_1 + h_2)(\delta + 1)]$ erasure correcting code with $t_1 > h_1$ and $t_2 n_2 > (h_1 + h_2)(\delta + 1)$ over $\mathbb{F}_{q^{m_1}}$ with redundancy m and the parity check matrix $H_{TP} = (\lambda_{1,1,1}, \ldots, \lambda_{t_1,t_2,n_2}) \in (\mathbb{F}_{q^m})^{t_1 t_2 n_2}$. Then the set $\{\lambda_{i,j,k}\}$ chosen as columns of H_{TP} ensures that Θ is h_1 wise independent over $\mathbb{F}_{q^{m_1}}$.*

Proof: We know that,

$$\Theta = \{\Phi_{i,\bar{\Gamma}_i} + \Phi_{i,\Gamma_i} Z_i, 1 \le i \le t_1\}$$

and,

$$\begin{aligned}
\Phi_i &= \{\lambda_{i,s,\bar{\Delta}_{i,s}} + \lambda_{i,s,\Delta_{i,s}} L_{i,s}, 1 \le s \le t_2\} \\
&= \{\Phi_{i,\Gamma_i},\ \Phi_{i,\bar{\Gamma}_i}\} \\
&= \{\phi_{i,1}, \ldots, \phi_{i,h_2}, \phi_{i,h_2+1}, \ldots, \phi_{i,r_1+h_2}\}
\end{aligned}$$

Hence, each $\phi_{i,l}$ is a linear combination of $\delta + 1$ distinct $\lambda_{i,j,k}$

Hence, similar to previous proof,

$$\Theta = \{b_{1,1}\phi_{i,1} + \ldots + b_{1,h_2}\phi_{i,h_2} + \phi_{i,h_2+v}, 1 \le i \le t_1, 1 \le v \le r_2\}$$

now, expanding $\phi_{i,l}$, we get

$$\phi_{i,l} = a_{k_l,1}\lambda_{i,s_l,\Delta_{i,s_l,1}} + \ldots + a_{k_l,\delta}\lambda_{i,s_l,\Delta_{i,s_l,\delta}} + \lambda_{i,s_l,\bar{\Delta}_{i,s_l,k_l}}$$

applying this expansion in Θ and following the procedure similar to the last proof, we get the results mentioned in this theorem. □

Similar to the previous theorem, we can show, for an $[t_1, t_2 n_2; h_1, (h_1 + h_2)(\delta + 1)]$ code C, the constituent codes C_1 and C_2 whose tensor product makes C.

- Let C_1 be an $[t_2 n_2, t_2 n_2 - (h_1 + h_2)(\delta + 1), (h_1 + h_2)(\delta + 1) + 1]$ MDS code in $\mathbb{F}_{q^{m_1}}$ where $m_1 = h_2(h_2 + \delta)$. This implies that $t_2 n_2 > (h_1 + h_2)(\delta + 1)$.

- Let C_2 be an $[t_1, t_1 - h_1, h_1 + 1]$ MDS code in $\mathbb{F}_{q^{m_1(h_1+h_2)(\delta+1)}}$. Hence, $t_1 > h_1$.

Then the value of $m = m_1 h_1 (h_1 + h_2)(\delta + 1)$.

4.2 Construction for $h_1 = 1$

In this section, we present a construction of HL-MRC for the case when $h_1 = 1$ over a field size lower than that provided by Construction 3.2.2.

Construction 4.2.1. *The structure of the parity check matrix for the present construction is the same as that given in Construction 3.2.2. In addition, the matrices M_0 and $M_i,\ 1 \leq i \leq t_2$ also remain the same. We modify the matrix $H_i,\ 1 \leq i \leq t_1$ as follows:*

$$H_i = \begin{bmatrix} \alpha_{1,1}^{q^{h_2}} & \alpha_{1,2}^{q^{h_2}} & \ldots \alpha_{t_2,n_2}^{q^{h_2}} \end{bmatrix},$$

where $\{\alpha_{s,j} \in \mathbb{F}_{q^{m_1}}, 1 \leq s \leq t_2, 1 \leq j \leq n_2\}$ are chosen to be $(\delta + 1)(h_2 + 1)$-wise independent over $\mathbb{F}_q$ based on Theorem 3.3.1.

Theorem 4.2.2. *The code C given by Construction 4.2.1 is a $[k, r_1, r_2, h_1 = 1, h_2, \delta]$ HL-MRC.*

Proof: We show that H can be used to correct all erasure patterns defined in Definition 8. From the definition the code should recover from δ erasures per $B_{i,s}$, h_2 additional erasures per A_i and 1 more erasure anywhere in the entire code.

Now, with $h_1 = 1$, the last erasure can be part of one group. Thus, effectively the code should recover from $h_2 + 1$ erasures per group. Suppose that the last erasure is in the i^{th} group. The sub-matrix of interest for the $(i, s)^{th}$ local group is

$$\left[\begin{array}{c|c} M_0|_{\Delta_{i,s}} & M_0|_{\bar{\Delta}_{i,s}} \\ \hline \alpha_{s,\Delta_{i,s}} & \alpha_{s,\bar{\Delta}_{i,s}} \\ \alpha_{s,\Delta_{i,s}}^{q} & \alpha_{s,\bar{\Delta}_{i,s}}^{q} \\ \vdots & \vdots \\ \alpha_{s,\Delta_{i,s}}^{q^{h_2-1}} & \alpha_{s,\bar{\Delta}_{i,s}}^{q^{h_2-1}} \\ \hline \alpha_{s,\Delta_{i,s}}^{q^{h_2}} & \alpha_{s,\bar{\Delta}_{i,s}}^{q^{h_2}} \end{array}\right].$$

Following the proof of Theorem 3.2.3 and performing row reduction of δ coordinates, the resultant matrix is

$$\begin{bmatrix} \psi_{i,1} & \psi_{i,2} & \cdots & \psi_{i,r_1+h_2} \\ \psi_{i,1}^{q} & \psi_{i,2}^{q} & \cdots & \psi_{i,r_1+h_2}^{q} \\ \vdots & \vdots & \cdots & \vdots \\ \psi_{i,1}^{q^{h_2-1}} & \psi_{i,2}^{q^{h_2-1}} & \cdots & \psi_{i,r_1+h_2}^{q^{h_2-1}} \\ \psi_{i,1}^{q^{h_2}} & \psi_{i,2}^{q^{h_2}} & \cdots & \psi_{i,r_1+h_2}^{q^{h_2}} \end{bmatrix}.$$

Now, by Lemma 3.2.1, it is the generator matrix of an MDS code if and only if Ψ_i is $(h_2 + 1)$-wise independent over $\mathbb{F}_q$. □

Diagrammatic Representation of Erasure Patterns

In the next few constructions, to prove that it works, we list out all the erasure patterns and show that each of them are correctible. To that end, it helps to visualise the erasure pattern itself. A typical code will look like the one in figure 4.1.

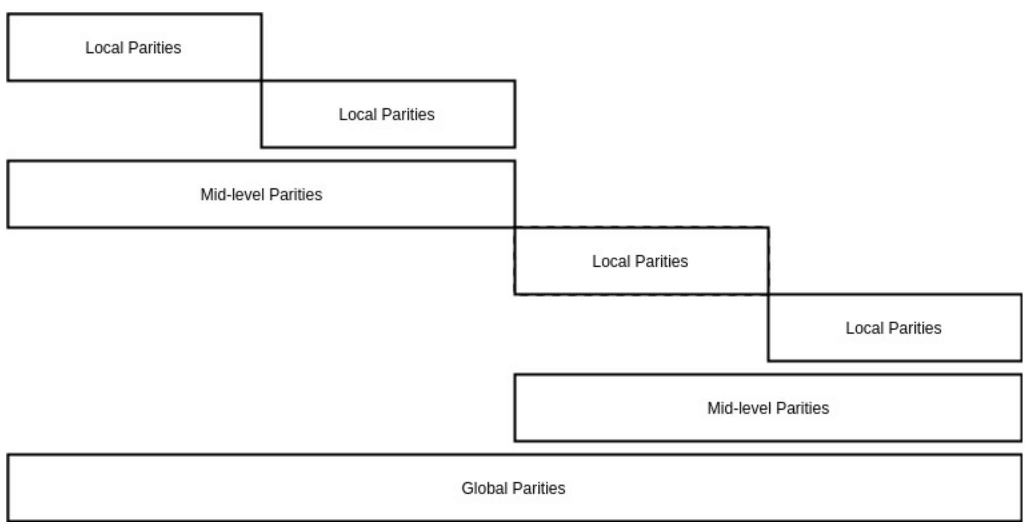

Figure 4.1: Generic structure of Code

To display erasure, we will mark vertical lines to signify which coordinates have been erased. As you will see in the following sections, δ symbols in each local code will, in almost all of the cases, be erased. Hence, for brevity's sake, we will not be displaying them. It will be clear from the context when the δ symbols are erased or not. Instead, we will show the erasures associated with h_1 and h_2 as the variations in the erasure pattern are actually caused by their placements. An example of such is shown in figure 4.2:

Here the solid line denotes the erasure associated with h_1 "global" erasures, while the dotted line shows the ones associated with h_2 "mid-level" erasures. The diagrammatic representation helps in

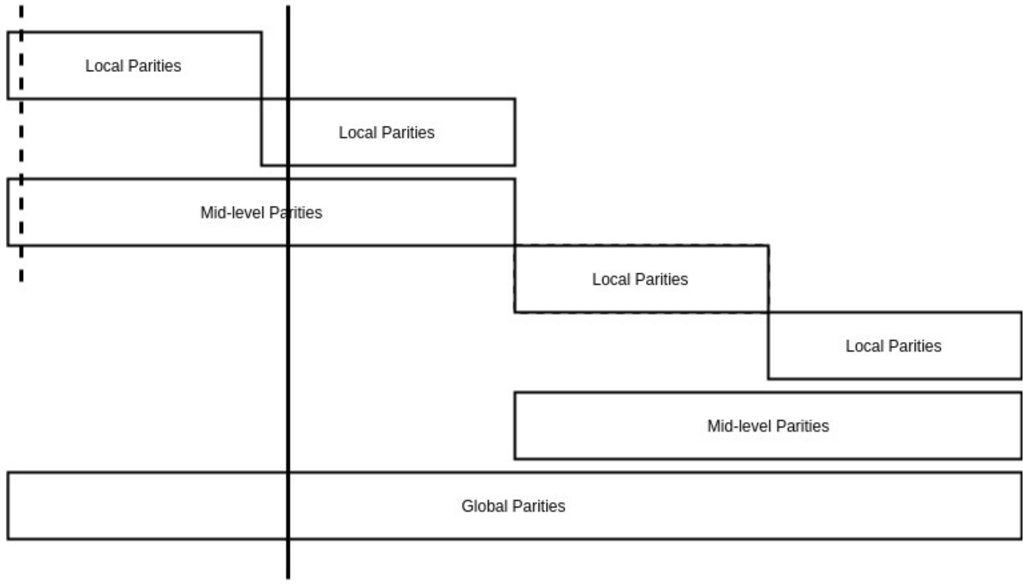

Figure 4.2: An example erasure pattern which indicates mid-level and global erasures.

identifying the structure of the submatrix of the parity check matrix which corresponds to the erasure patterns. The structure of the submatrix is different for different erasure patterns and we argue that the erasure pattern is correctible based on this structure.

4.3 Construction for $h_1 = 1$ and $h_2 = 1$

In this special case, we give a construction which requires a smaller field size. We use determinantal identities to show that the matrix formed by the columns of the parity check matrix corresponding to the erased positions are invertible and hence can be recovered.

Lemma 4.3.1. *Let $C_1, \cdots, C_h$ be $a \times (a+1)$ dimensional matrices and $D_1, \cdots, D_h$ be $h \times (a+1)$ dimensional matrices over a field and let $D_i^{(j)}$ be the j^{th} row of D_i. Then,*

$$\det \left[\begin{array}{c|c|c|c} C_1 & 0 & \cdots & 0 \\ \hline 0 & C_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & C_h \\ \hline D_1 & D_2 & \cdots & D_h \end{array}\right] = (-1)^{\frac{ah(h-1)}{2}} \det \begin{bmatrix} \det \begin{pmatrix} C_1 \\ D_1^{(1)} \end{pmatrix} & \cdots & \det \begin{pmatrix} C_h \\ D_h^{(1)} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ \det \begin{pmatrix} C_1 \\ D_1^{(h)} \end{pmatrix} & \cdots & \det \begin{pmatrix} C_h \\ D_h^{(h)} \end{pmatrix} \end{bmatrix}.$$

Proof: Proof directly follows as a result of [20, Lemmas B.1 and B.2] □

Theorem 4.3.2. *The matrix H as shown below,*

$$H = \begin{bmatrix} H_0 & & & \\ & H_0 & & \\ & & \ddots & \\ & & & H_0 \\ H_1 & H_2 & \dots & H_{t_1} \end{bmatrix} \qquad H_0 = \begin{bmatrix} M_0 & & & \\ & M_0 & & \\ & & \ddots & \\ & & & M_0 \\ M_1 & M_2 & \dots & M_{t_2} \end{bmatrix}$$

$$M_0 = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{n_2} \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_{n_2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^\delta & \alpha_2^\delta & \dots & \alpha_{n_2}^\delta \end{bmatrix} \qquad M_i = \begin{bmatrix} \lambda_i & \lambda_i & \dots & \lambda_i \end{bmatrix}$$

$$H_i = \begin{bmatrix} H_{i,1} & H_{i,2} & \dots & H_{i,t_2} \end{bmatrix}$$
$$H_{i,s} = \begin{bmatrix} \alpha_1^{\delta+1} & \alpha_2^{\delta+1} & \dots & \alpha_{n_2}^{\delta+1} \end{bmatrix}$$

is a parity check matrix for an $[k, r_1, r_2, 1, 1, \delta]$ HL-MRC if the following conditions are satisfied:

- *q is a prime power such that there exists a subgroup G of $\mathbb{F}_q^*$ of size atleast n_2 and with atleast t_2 cosets.*
- *$\alpha_1, \alpha_2, \dots \alpha_{n_2} \in G$ and $\alpha_i \neq \alpha_j$.*
- *$\lambda_1, \lambda_2, \dots, \lambda_{t_2} \in \mathbb{F}_q^*$ be elements from distinct cosets of G.*

Proof: To show that the code is a $[k, r_1, r_2, 1, 1, \delta]$ HL-MRC, we consider erasure patterns where there are δ erasures per local code, one erasure per mid-level code and one more erasure anywhere in the global code. We will show that any such erasure pattern is correctible.

Since, there is only one global erasure and it can be in one mid-level code, we consider that the mid-level code which has additional global erasure has index l and for all $j \neq l$, there are no global erasures associated with these mid-level codes.

Correcting each mid-level code will, in the end, correct the original code.

We show how to correct each of these mid-level codes.

1. For all j^{th} mid-level codes ($j \neq l$), the corresponding erasure pattern is shown in Figure 4.3. Let the mid-level code where the erasure occurs be j'.

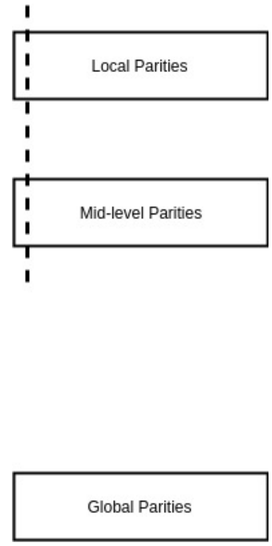

Figure 4.3: Erasure pattern for Case 1

The submatrix B_j of the parity-check matrix which is used to recover the erasures within the j^{th} mid-level code is given by,

$$B_j = \begin{bmatrix} \alpha_{j'_1} & \alpha_{j'_2} & \dots & \alpha_{j'_\delta} & \alpha_{j'_{\delta+1}} \\ \alpha^2_{j'_1} & \alpha^2_{j'_2} & \dots & \alpha^2_{j'_\delta} & \alpha^2_{j'_{\delta+1}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^\delta_{j'_1} & \alpha^\delta_{j'_2} & \dots & \alpha^\delta_{j'_\delta} & \alpha^\delta_{j'_{\delta+1}} \\ \lambda_{j'} & \lambda_{j'} & \dots & \lambda_{j'} & \lambda_{j'} \end{bmatrix}$$

where $\{j'_1, \dots j'_{\delta+1}\}$ denote the $\delta+1$ erased coordinates in the local group j'. We can clearly see that this matrix is a Vandermonde matrix after scaling and permuting rows. Hence $\det(B_j) \neq 0$.

2. For the l^{th} mid-level code, we will also involve the global parity. This case can again be divided into two subcases depending on the local group where the extra erasure happens:

   (a) Both the mid-level erasure and the global erasure occur in the same local code, l' (figure 4.4).

   The matrix formed will be,

$$B_l = \begin{bmatrix} \alpha_{l'_1} & \alpha_{l'_2} & \dots & \alpha_{l'_\delta} & \alpha_{l'_{\delta+1}} & \alpha_{l'_{\delta+2}} \\ \alpha^2_{l'_1} & \alpha^2_{l'_2} & \dots & \alpha^2_{l'_\delta} & \alpha^2_{l'_{\delta+1}} & \alpha^2_{l'_{\delta+2}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \alpha^\delta_{l'_1} & \alpha^\delta_{l'_2} & \dots & \alpha^\delta_{l'_\delta} & \alpha^\delta_{l'_{\delta+1}} & \alpha^\delta_{l'_{\delta+2}} \\ \lambda_{l'} & \lambda_{l'} & \dots & \lambda_{l'} & \lambda_{l'} & \lambda_{l'} \\ \alpha^{\delta+1}_{l'_1} & \alpha^{\delta+1}_{l'_2} & \dots & \alpha^{\delta+1}_{l'_\delta} & \alpha^{\delta+1}_{l'_{\delta+1}} & \alpha^{\delta+1}_{l'_{\delta+2}} \end{bmatrix}.$$

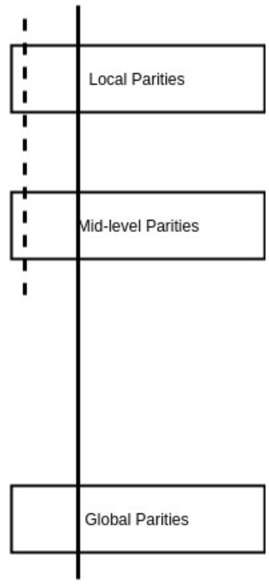

Figure 4.4: Erasure pattern for case 2.a

This is similar to above where B_l after scaling and permuting the rows is also a vandermonde matrix. Hence $\det(B_l) \neq 0$.

(b) The mid-level and global erasure occur in different local codes (figure 4.5). Let those local codes be l' and l''. The matrix B_l,

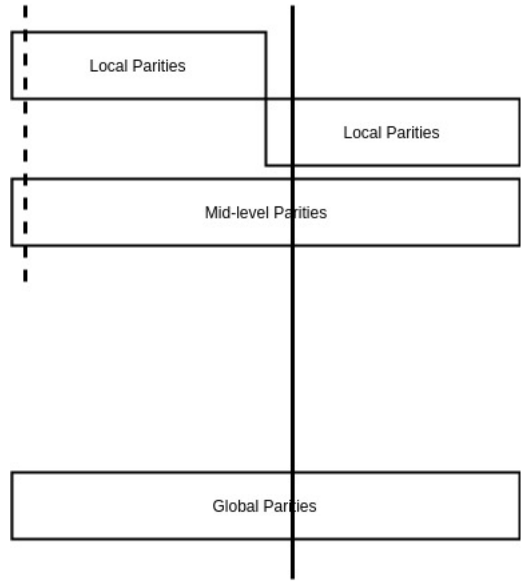

Figure 4.5: Erasure pattern for case 2.b

$$
\det(B_l) = \det \begin{bmatrix}
\alpha_{l'_1} & \dots & \alpha_{l'_{\delta+1}} & & & \\
\alpha^2_{l'_1} & \dots & \alpha^2_{l'_{\delta+1}} & & & \\
\vdots & \ddots & \vdots & & & \\
\alpha^{\delta}_{l'_1} & \dots & \alpha^{\delta}_{l'_{\delta+1}} & & & \\
 & & & \alpha_{l''_1} & \dots & \alpha_{l''_{\delta+1}} \\
 & & & \alpha^2_{l''_1} & \dots & \alpha^2_{l''_{\delta+1}} \\
 & & & \vdots & \ddots & \vdots \\
 & & & \alpha^{\delta}_{l''_1} & \dots & \alpha^{\delta}_{l''_{\delta+1}} \\
\lambda_{l'} & \dots & \lambda_{l'} & \lambda_{l''} & \dots & \lambda_{l''} \\
\alpha^{\delta+1}_{l'_{\delta+1}} & \dots & \alpha^{\delta+1}_{l'_{\delta+1}} & \alpha^{\delta+1}_{l''_{\delta+1}} & \dots & \alpha^{\delta+1}_{l''_{\delta+1}}
\end{bmatrix}
$$

$$
= \det \begin{bmatrix}
\det \begin{pmatrix}
\alpha_{l'_1} & \dots & \alpha_{l'_{\delta+1}} \\
\alpha^2_{l'_1} & \dots & \alpha^2_{l'_{\delta+1}} \\
\vdots & \ddots & \vdots \\
\alpha^{\delta}_{l'_1} & \dots & \alpha^{\delta}_{l'_{\delta+1}} \\
\lambda_{l'} & \dots & \lambda_{l'}
\end{pmatrix} & \det \begin{pmatrix}
\alpha_{l''_1} & \dots & \alpha_{l''_{\delta+1}} \\
\alpha^2_{l''_1} & \dots & \alpha^2_{l''_{\delta+1}} \\
\vdots & \ddots & \vdots \\
\alpha^{\delta}_{l''_1} & \dots & \alpha^{\delta}_{l''_{\delta+1}} \\
\lambda_{l''} & \dots & \lambda_{l''}
\end{pmatrix} \\
\det \begin{pmatrix}
\alpha_{l'_1} & \dots & \alpha_{l'_{\delta+1}} \\
\alpha^2_{l'_1} & \dots & \alpha^2_{l'_{\delta+1}} \\
\vdots & \ddots & \vdots \\
\alpha^{\delta}_{l'_1} & \dots & \alpha^{\delta}_{l'_{\delta+1}} \\
\alpha^{\delta+1}_{l'_{\delta+1}} & \dots & \alpha^{\delta+1}_{l'_{\delta+1}}
\end{pmatrix} & \det \begin{pmatrix}
\alpha_{l''_1} & \dots & \alpha_{l''_{\delta+1}} \\
\alpha^2_{l''_1} & \dots & \alpha^2_{l''_{\delta+1}} \\
\vdots & \ddots & \vdots \\
\alpha^{\delta}_{l''_1} & \dots & \alpha^{\delta}_{l''_{\delta+1}} \\
\alpha^{\delta+1}_{l''_{\delta+1}} & \dots & \alpha^{\delta+1}_{l''_{\delta+1}}
\end{pmatrix}
\end{bmatrix}
$$

This means that if,

$$\det(B_l) = 0$$

Then,

$$\det \begin{bmatrix} \lambda_{l'} & \lambda_{l''} \\ \prod_{i=1}^{\delta+1} \alpha_{l'_i} & \prod_{i=1}^{\delta+1} \alpha_{l''_i} \end{bmatrix} = 0.$$

Where we factored out the non-zero Vandermonde determinants from each column. Since $\alpha_{l'_i}, \alpha_{l''_i} \in G$ and $\lambda_{l'}, \lambda_{l''}$ are in different cosets of G, the last determinant cannot be zero.

With this, we can say that the provided erasure code can corrects all possible erasure patterns. $\square$

4.4 Construction for $h_1 = 2$ and $h_2 = 1$

Similar to above, we can provide an optimal construction for an $[k, r_1, r_2, 2, 1]$ code. We would again require some determinantal identities prove the validity of our construction.

Lemma 4.4.1. *Let C_1 be an $a \times (a+1)$ matrix, C_2 be an $a \times (a+2)$ matrix, D_1 be a $3 \times (a+1)$ matrix and D_2 be a $3 \times (a+2)$ matrix and let $D_i^{(j)}$ be the j^{th} row of D_i. Then,*

$$\begin{aligned} \det \left[\begin{array}{c|c} C_1 & 0 \\ \hline 0 & C_2 \\ \hline D_1 & D_2 \end{array}\right] =& (-1)^a \cdot \left(\det \begin{pmatrix} C_1 \\ D_1^{(1)} \end{pmatrix} \cdot \det \begin{pmatrix} C_2 \\ D_2^{(2)} \\ D_2^{(3)} \end{pmatrix} - \det \begin{pmatrix} C_1 \\ D_1^{(2)} \end{pmatrix} \cdot \det \begin{pmatrix} C_2 \\ D_2^{(1)} \\ D_2^{(3)} \end{pmatrix} \right. \\ & \left. + \det \begin{pmatrix} C_1 \\ D_1^{(3)} \end{pmatrix} \cdot \det \begin{pmatrix} C_2 \\ D_2^{(1)} \\ D_2^{(2)} \end{pmatrix} \right) \end{aligned}$$

Lemma 4.4.2. *Given C_1 and C_2 to be $a \times a+1$ matrices and C_3 to be an $a \times a+2$ matrix. Also, D_1 and D_2 are $4 \times a+1$ matrices while D_3 is a $4 \times a+2$ matrix. It is also given that $D_3^{(1)}, D_1^{(2)}, D_2^{(2)} = [0]$.*

Then,

$$\det\left[\begin{array}{c|c|c} C_1 & 0 & 0 \\ \hline 0 & C_2 & 0 \\ \hline 0 & 0 & C_3 \\ \hline D_1 & D_2 & D_3 \end{array}\right] = (-1)^a \cdot \left(\det\begin{pmatrix} C_1 \\ D_1^{(1)} \end{pmatrix} \cdot \det\begin{pmatrix} C_2 \\ D_2^{(3)} \end{pmatrix} \cdot \det\begin{pmatrix} C_3 \\ D_3^{(2)} \\ D_3^{(4)} \end{pmatrix} \right.$$

$$+ \det\begin{pmatrix} C_1 \\ D_1^{(1)} \end{pmatrix} \cdot \det\begin{pmatrix} C_2 \\ D_2^{(4)} \end{pmatrix} \cdot \det\begin{pmatrix} C_3 \\ D_3^{(2)} \\ D_3^{(3)} \end{pmatrix}$$

$$+ \det\begin{pmatrix} C_1 \\ D_1^{(3)} \end{pmatrix} \cdot \det\begin{pmatrix} C_2 \\ D_2^{(1)} \end{pmatrix} \cdot \det\begin{pmatrix} C_3 \\ D_3^{(2)} \\ D_3^{(4)} \end{pmatrix}$$

$$\left. - \det\begin{pmatrix} C_1 \\ D_1^{(4)} \end{pmatrix} \cdot \det\begin{pmatrix} C_2 \\ D_2^{(1)} \end{pmatrix} \cdot \det\begin{pmatrix} C_3 \\ D_3^{(2)} \\ D_3^{(3)} \end{pmatrix} \right)$$

Proof: Lemmas 4.4.1 and 4.4.2 directly follow as a result of [20, Lemmas B.1 and B.2]. □

We also define a cauchy matrix here.

Lemma 4.4.3 (Cauchy Matrix [47]). *Let $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n \in \mathbb{F}_q$ be all distinct. Then,*

$$\det\begin{bmatrix} \frac{1}{a_1-b_1} & \frac{1}{a_2-b_1} & \cdots & \frac{1}{a_n-b_1} \\ \frac{1}{a_1-b_2} & \frac{1}{a_2-b_2} & \cdots & \frac{1}{a_n-b_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_1-b_n} & \frac{1}{a_2-b_n} & \cdots & \frac{1}{a_n-b_n} \end{bmatrix} = \frac{\prod_{i>j}(a_i-a_j)(b_i-b_j)}{\prod_{i,j}(a_i-a_j)}.$$

Such a matrix is called an Cauchy Matrix. Every minor of a Cauchy matrix is also an Cauchy matrix.

Theorem 4.4.4. *The matrix H as shown below,*

$$H = \begin{bmatrix} H_0 & & & \\ & H_0 & & \\ & & \ddots & \\ & & & H_0 \\ H_1 & H_2 & \ldots & H_{t_1} \end{bmatrix} \qquad H_0 = \begin{bmatrix} M_0 & & & \\ & M_0 & & \\ & & \ddots & \\ & & & M_0 \\ M_1 & M_2 & \ldots & M_{t_2} \end{bmatrix}$$

$$M_0 = \begin{bmatrix} \frac{1}{\alpha_1-\beta_1} & \frac{1}{\alpha_2-\beta_1} & \cdots & \frac{1}{\alpha_{n_2}-\beta_1} \\ \frac{1}{\alpha_1-\beta_2} & \frac{1}{\alpha_2-\beta_2} & \cdots & \frac{1}{\alpha_{n_2}-\beta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_1-\beta_\delta} & \frac{1}{\alpha_2-\beta_\delta} & \cdots & \frac{1}{\alpha_{n_2}-\beta_\delta} \end{bmatrix} \qquad M_i = \begin{bmatrix} \frac{1}{\alpha_1-\beta_{\delta+1}} & \frac{1}{\alpha_2-\beta_{\delta+1}} & \cdots & \frac{1}{\alpha_{n_2}-\beta_{\delta+1}} \end{bmatrix}$$

$$H_i = \begin{bmatrix} H_{i,1} & H_{i,2} & \ldots & H_{i,t_2} \end{bmatrix}$$

$$H_{i,s} = \begin{bmatrix} \frac{\lambda_{s+(i-1)t_2}}{\alpha_1-\beta_{\delta+2}} & \frac{\lambda_{s+(i-1)t_2}}{\alpha_2-\beta_{\delta+2}} & \cdots & \frac{\lambda_{s+(i-1)t_2}}{\alpha_{n_2}-\beta_{\delta+2}} \\ \frac{\mu_{s+(i-1)t_2}}{\alpha_1-\beta_{\delta+3}} & \frac{\mu_{s+(i-1)t_2}}{\alpha_2-\beta_{\delta+3}} & \cdots & \frac{\mu_{s+(i-1)t_2}}{\alpha_{n_2}-\beta_{\delta+3}} \end{bmatrix}$$

is a parity check matrix for an $[k, r_1, r_2, 2, 1, \delta]$ *code if the following conditions are satisfied:*

- $q_0 \geq 2(n_2 + \delta) + 3$ *is a prime power.*
- *There exists a subgroup* G *of* $\mathbb{F}_{q_0}^*$ *of size at least* $n_2 + 2$ *with atleast* $t_1 t_2$ *cosets.*
- $\mathbb{F}_q$ *is an extension field of* $\mathbb{F}_{q_0}$*.*
- $\mu_1, \ldots, \mu_{t_1 t_2}$ *are picked from distinct cosets of* G*.*
- *Choose distinct* $\beta_{\delta+1}, \beta_{\delta+2}, \beta_{\delta+3} \in \mathbb{F}_{q_0}$*.*
- *Pick* $\alpha_1, \ldots \alpha_{n_2} \in \mathbb{F}_{q_0}$ *such that,* $\frac{\alpha_i-\beta_{\delta+2}}{\alpha_i-\beta_{\delta+3}}, \frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} \in G$*.*
- *Pick distinct* $\beta_1, \ldots, \beta_\delta \in \mathbb{F}_{q_0} \setminus \{\alpha_1, \ldots, \alpha_{n_2}, \beta_{\delta+1}, \beta_{\delta+2}, \beta_{\delta+3}\}$*.*
- $\lambda_1, \lambda_2, \ldots, \lambda_{t_1 t_2} \in \mathbb{F}_q$ *are picked 4 wise-independent over* $\mathbb{F}_{q_0}$*.*

Proof: Again as in previous proof, we consider the case when there are δ erasures per mid-level code, one erasure per mid-level code and two more global erasures anywhere in the code. We again look at the erasure patterns within each mid-level codes. There are three distinct patterns possible

1. No global erasures occur in that mid-level code.
2. Either one or both of the global erasures occur in the mid-level code.

We that each of the above are correctible.

Let $\gamma_{i,j} = \frac{1}{\alpha_j - \beta_i}$.

1. When no global erasures occur in the mid-level code, there are δ erasures per local code and one more erasure per mid-level code (figure 4.6).

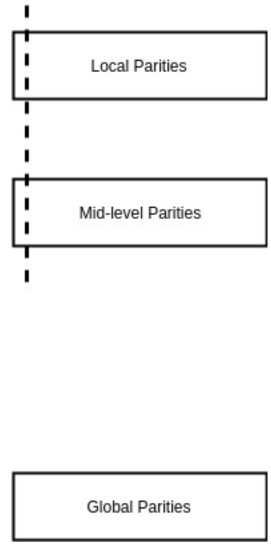

Figure 4.6: Erasure pattern for case 1

In this scenario, we involve the mid-level parities. Let l be the affected mid-level code and l' be the local code within the mid-level code where the erasure occurs. The matrix, B_l

$$B_l = \begin{bmatrix} \gamma_{1,l'_1} & \gamma_{1,l'_2} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \gamma_{2,l'_1} & \gamma_{2,l'_2} & \cdots & \gamma_{2,l'_{\delta+1}} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \gamma_{\delta,l'_2} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \gamma_{\delta+1,l'_1} & \gamma_{\delta+1,l'_2} & \cdots & \gamma_{\delta+1,l'_{\delta+1}} \end{bmatrix}.$$

Where $\{l'_1, l'_2, \ldots, l'_{\delta+1}\}$ are the erased coordinates in local code l'. This is a Cauchy matrix and hence $\det(B_l) \neq 0$.

2. When there are global erasures, there are δ erasures per local code, one erasure per mid-level code and two more erasures anywhere in the code

   Here we have a lot more sub-cases.

   (a) Both global erasures are in the same local code as the mid-level code. Let l be the affected mid-level code and l' be the local code in the mid-level code where the erasure happens (figure 4.7). The matrix B_l in that case,

$$B_l = \begin{bmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+3}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+3}} \\ \lambda_{l'} \cdot \gamma_{\delta+2,l'_1} & \cdots & \lambda_{l'} \cdot \gamma_{\delta+2,l'_{\delta+3}} \\ \mu_{l'} \cdot \gamma_{\delta+3,l'_1} & \cdots & \mu_{l'} \cdot \gamma_{\delta+3,l'_{\delta+3}} \end{bmatrix}.$$

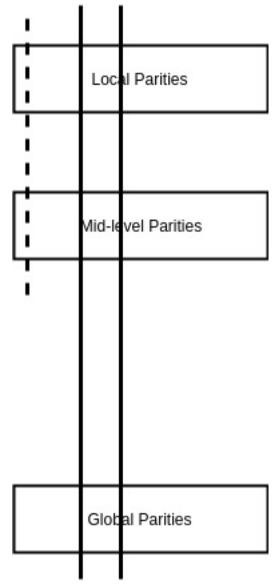

Figure 4.7: Erasure pattern for case case 2.a

This is also a Cauchy matrix with the last two rows scaled to $\lambda_{l'}$ and $\mu_{l'}$ respectively. Hence $\det(B_l) \neq 0$ and this erasure pattern is correctible.

(b) Both global erasures are in the same local code but different one from the mid-level erasure for that mid-level code (figure 4.8).

Assume that the l^{th} mid-level code is affected. Let l'' be the local code with two erasures while l' be the other one within this mid-level code.

$$B_l = \begin{bmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} & & & \\ \vdots & \ddots & \vdots & & & \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} & & & \\ & & & \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ & & & \vdots & \ddots & \vdots \\ & & & \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+1}} & \gamma_{\delta+1,l''_1} & \cdots & \gamma_{\delta+1,l''_{\delta+2}} \\ \lambda_{l'} \cdot \gamma_{\delta+2,l'_1} & \cdots & \lambda_{l'} \cdot \gamma_{\delta+2,l'_{\delta+1}} & \lambda_{l''} \cdot \gamma_{\delta+2,l''_1} & \cdots & \lambda_{l''} \cdot \gamma_{\delta+2,l''_{\delta+2}} \\ \mu_{l'} \cdot \gamma_{\delta+3,l'_1} & \cdots & \mu_{l'} \cdot \gamma_{\delta+3,l'_{\delta+1}} & \mu_{l''} \cdot \gamma_{\delta+3,l''_1} & \cdots & \mu_{l''} \cdot \gamma_{\delta+3,l''_{\delta+2}} \end{bmatrix}.$$

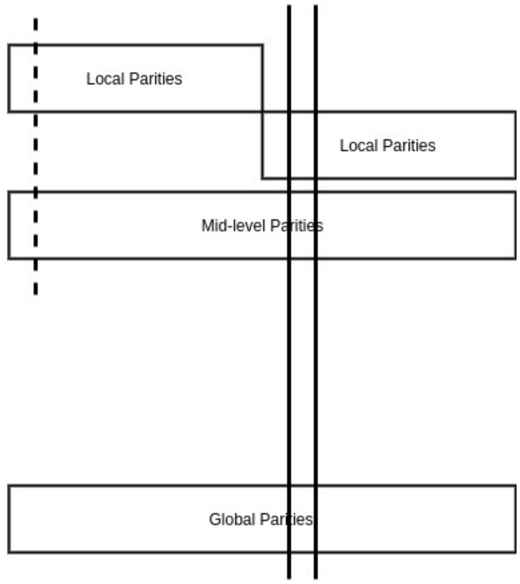

Figure 4.8: Erasure pattern for case 2.b

Expanding this via the lemma 4.4.1,

$$
\begin{aligned}
\det(B_l) = &\det\begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+1}} \end{pmatrix} \cdot \det\begin{pmatrix} \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \lambda_{l''}\cdot\gamma_{\delta+2,l''_1} & \cdots & \lambda_{l''}\cdot\gamma_{\delta+2,l''_{\delta+2}} \\ \mu_{l''}\cdot\gamma_{\delta+3,l''_1} & \cdots & \mu_{l''}\cdot\gamma_{\delta+3,l''_{\delta+2}} \end{pmatrix} \\
&- \det\begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \lambda_{l'}\cdot\gamma_{\delta+2,l'_1} & \cdots & \lambda_{l'}\cdot\gamma_{\delta+2,l'_{\delta+1}} \end{pmatrix} \cdot \det\begin{pmatrix} \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \gamma_{\delta+1,l''_1} & \cdots & \gamma_{\delta+1,l''_{\delta+2}} \\ \mu_{l''}\cdot\gamma_{\delta+3,l''_1} & \cdots & \mu_{l''}\cdot\gamma_{\delta+3,l''_{\delta+2}} \end{pmatrix} \\
&+ \det\begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \mu_{l'}\cdot\gamma_{\delta+3,l'_1} & \cdots & \mu_{l'}\cdot\gamma_{\delta+3,l'_{\delta+1}} \end{pmatrix} \cdot \det\begin{pmatrix} \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \gamma_{\delta+1,l''_1} & \cdots & \gamma_{\delta+1,l''_{\delta+2}} \\ \lambda_{l''}\cdot\gamma_{\delta+2,l''_1} & \cdots & \lambda_{l''}\cdot\gamma_{\delta+2,l''_{\delta+2}} \end{pmatrix}
\end{aligned}
$$

$$\det(B_l) = \lambda_{l''}\mu_{l''} \cdot \det\begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+1}} \end{pmatrix} \cdot \det\begin{pmatrix} \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \gamma_{\delta+2,l''_1} & \cdots & \gamma_{\delta+2,l''_{\delta+2}} \\ \gamma_{\delta+3,l''_1} & \cdots & \gamma_{\delta+3,l''_{\delta+2}} \end{pmatrix}$$

$$- \lambda_{l'}\mu_{l''} \cdot \det\begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \gamma_{\delta+2,l'_1} & \cdots & \gamma_{\delta+2,l'_{\delta+1}} \end{pmatrix} \cdot \det\begin{pmatrix} \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \gamma_{\delta+1,l''_1} & \cdots & \gamma_{\delta+1,l''_{\delta+2}} \\ \gamma_{\delta+3,l''_1} & \cdots & \gamma_{\delta+3,l''_{\delta+2}} \end{pmatrix}$$

$$+ \lambda_{l''}\mu_{l'} \cdot \det\begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+1}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l'_1} & \cdots & \gamma_{\delta,l'_{\delta+1}} \\ \gamma_{\delta+3,l'_1} & \cdots & \gamma_{\delta+3,l'_{\delta+1}} \end{pmatrix} \cdot \det\begin{pmatrix} \gamma_{1,l''_1} & \cdots & \gamma_{1,l''_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l''_1} & \cdots & \gamma_{\delta,l''_{\delta+2}} \\ \gamma_{\delta+1,l''_1} & \cdots & \gamma_{\delta+1,l''_{\delta+2}} \\ \gamma_{\delta+2,l''_1} & \cdots & \gamma_{\delta+2,l''_{\delta+2}} \end{pmatrix}$$

Each term in this determinant is $\lambda_i\mu_j$ multiplied by a Cauchy matrix $\in \mathbb{F}_{q_0}$. The determinant is again a linear combination of $\lambda_{l'}$ and $\lambda_{l''}$. Again, this determinant cannot be zero because λ's are 4-wise independent.

(c) Both global and the one mid-level erasures are in different local code but the same mid-level code (figure 4.9).

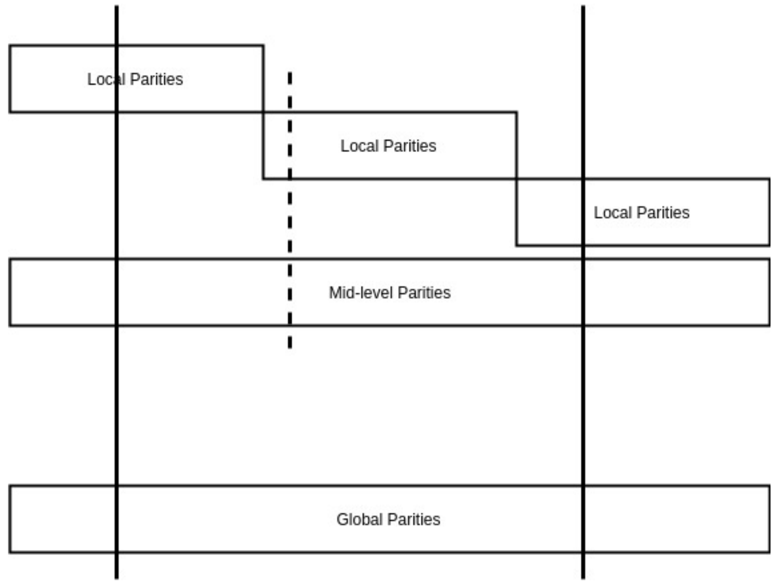

Figure 4.9: Erasure pattern for case 2.c

Let the affected mid-level code be l and the local codes within, where the erasure occurs, be $l^{(1)}, l^{(2)}$ and $l^{(3)}$. The matrix B_l,

$$B_l = \begin{bmatrix} \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+1}^{(1)}} & & & & & & \\ \vdots & \ddots & \vdots & & & & & & \\ \gamma_{\delta,l_1^{(1)}} & \cdots & \gamma_{\delta,l_{\delta+1}^{(1)}} & & & & & & \\ & & & \gamma_{1,l_1^{(2)}} & \cdots & \gamma_{1,l_{\delta+1}^{(2)}} & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & \gamma_{\delta,l_1^{(2)}} & \cdots & \gamma_{\delta,l_{\delta+1}^{(2)}} & & & \\ & & & & & & \gamma_{1,l_1^{(3)}} & \cdots & \gamma_{1,l_{\delta+1}^{(3)}} \\ & & & & & & \vdots & \ddots & \vdots \\ & & & & & & \gamma_{\delta,l_1^{(3)}} & \cdots & \gamma_{\delta,l_{\delta+1}^{(3)}} \\ \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+1}^{(1)}} & \gamma_{\delta+1,l_1^{(2)}} & \cdots & \gamma_{\delta+1,l_{\delta+1}^{(2)}} & \gamma_{\delta+1,l_1^{(3)}} & \cdots & \gamma_{\delta+1,l_{\delta+1}^{(3)}} \\ \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_1^{(1)}} & \cdots & \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_{\delta+1}^{(1)}} & \lambda_{l^{(2)}} \cdot \gamma_{\delta+2,l_1^{(2)}} & \cdots & \lambda_{l^{(2)}} \cdot \gamma_{\delta+2,l_{\delta+1}^{(2)}} & \lambda_{l^{(3)}} \cdot \gamma_{\delta+2,l_1^{(3)}} & \cdots & \lambda_{l^{(3)}} \cdot \gamma_{\delta+2,l_{\delta+1}^{(3)}} \\ \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_1^{(1)}} & \cdots & \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_{\delta+1}^{(1)}} & \mu_{l^{(2)}} \cdot \gamma_{\delta+3,l_1^{(2)}} & \cdots & \mu_{l^{(2)}} \cdot \gamma_{\delta+3,l_{\delta+1}^{(2)}} & \mu_{l^{(3)}} \cdot \gamma_{\delta+3,l_1^{(3)}} & \cdots & \mu_{l^{(3)}} \cdot \gamma_{\delta+3,l_{\delta+1}^{(3)}} \end{bmatrix}.$$

$\det(B_l)$ can be expanded via lemma 4.3.1. After doing that and setting the determinant to zero,

$$\det(B_l) = 0,$$

we get,

$$\det\begin{bmatrix} \frac{c_{l^{(1)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+1})}{e_{l^{(1)}}\prod_{i\in l_S^{(1)}}(\alpha_i-\beta_{\delta+1})} & \frac{c_{l^{(2)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+1})}{e_{l^{(2)}}\prod_{i\in l_S^{(2)}}(\alpha_i-\beta_{\delta+1})} & \frac{c_{l^{(3)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+1})}{e_{l^{(3)}}\prod_{i\in l_S^{(3)}}(\alpha_i-\beta_{\delta+1})} \\ \lambda_{l^{(1)}}\cdot\frac{c_{l^{(1)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+2})}{e_{l^{(1)}}\prod_{i\in l_S^{(1)}}(\alpha_i-\beta_{\delta+2})} & \lambda_{l^{(2)}}\cdot\frac{c_{l^{(2)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+2})}{e_{l^{(2)}}\prod_{i\in l_S^{(2)}}(\alpha_i-\beta_{\delta+2})} & \lambda_{l^{(3)}}\cdot\frac{c_{l^{(3)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+2})}{e_{l^{(3)}}\prod_{i\in l_S^{(3)}}(\alpha_i-\beta_{\delta+2})} \\ \mu_{l^{(1)}}\cdot\frac{c_{l^{(1)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+3})}{e_{l^{(1)}}\prod_{i\in l_S^{(1)}}(\alpha_i-\beta_{\delta+3})} & \mu_{l^{(2)}}\cdot\frac{c_{l^{(2)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+3})}{e_{l^{(2)}}\prod_{i\in l_S^{(2)}}(\alpha_i-\beta_{\delta+3})} & \mu_{l^{(3)}}\cdot\frac{c_{l^{(3)}} d\prod_{i\in[\delta]}(\beta_i-\beta_{\delta+3})}{e_{l^{(3)}}\prod_{i\in l_S^{(3)}}(\alpha_i-\beta_{\delta+3})} \end{bmatrix} = 0$$

$$\det\begin{bmatrix} 1 & 1 & 1 \\ \lambda_{l^{(1)}}\prod_{i\in l_S^{(1)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} & \lambda_{l^{(2)}}\prod_{i\in l_S^{(2)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} & \lambda_{l^{(3)}}\prod_{i\in l_S^{(3)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} \\ \mu_{l^{(1)}}\prod_{i\in l_S^{(1)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} & \mu_{l^{(2)}}\prod_{i\in l_S^{(2)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} & \mu_{l^{(3)}}\prod_{i\in l_S^{(3)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} \end{bmatrix} = 0$$

Where,

- $l_S^{(i)} = \{l_1^{(i)}, \ldots, l_{\delta+1}^{(i)}\}$.
- $c_{l^{(i)}} = \prod_{f>g, f,g\in l_S^{(i)}}(\alpha_f - \alpha_g)$.
- $d = \prod_{f>g,f,g\in[\delta]}(\beta_f - \beta_g)$.
- $e_{l^{(i)}} = \prod_{f\in l_S^{(i)}, g\in[\delta]}(\alpha_f - \beta_g)$.

Now, by the choice of α's, $\prod_{i\in l_S^{(k)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} \in G$. And because μ_i belong to different cosets in G, the last row in the above matrix consists of distinct elements. This determinant is a linear combination in the three λ's. Hence the determinant is non-zero because the λ's are 4-wise independent.

(d) Both global erasures are in different mid-level code but share that local code with the mid-level parities for that mid-level code (figure 4.10).

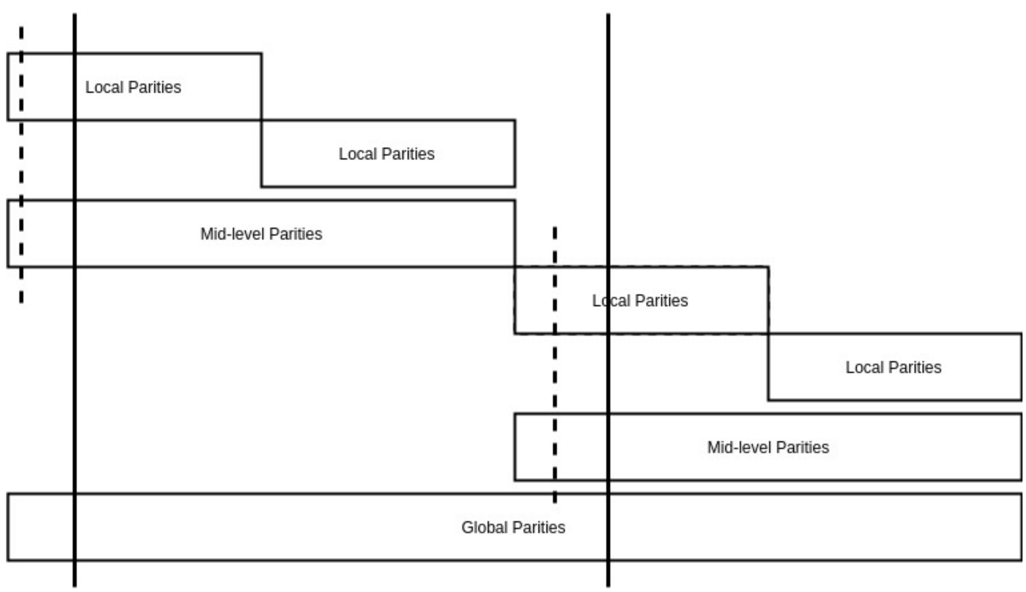

Figure 4.10: Erasure pattern for case 2.d

Assume k^{th} and l^{th} mid-level codes are affected. The local codes within them, where the erasure occurs, are k' and l'. The matrix $B_{k,l}$,

$$B_{k,l} = \begin{bmatrix} \gamma_{1,k'_1} & \cdots & \gamma_{1,k'_{\delta+2}} & & & \\ \vdots & \ddots & \vdots & & & \\ \gamma_{\delta+1,k'_1} & \cdots & \gamma_{\delta+1,k'_{\delta+2}} & & & \\ & & & \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+2}} \\ & & & \vdots & \ddots & \vdots \\ & & & \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+2}} \\ \lambda_{k'}\cdot\gamma_{\delta+2,k'_1} & \dots & \lambda_{k'}\cdot\gamma_{\delta+2,k'_{\delta+2}} & \lambda_{l'}\cdot\gamma_{\delta+2,l'_1} & \dots & \lambda_{l'}\cdot\gamma_{\delta+2,l'_{\delta+2}} \\ \mu_{k'}\cdot\gamma_{\delta+3,k'_1} & \dots & \mu_{k'}\cdot\gamma_{\delta+3,k'_{\delta+2}} & \mu_{l'}\cdot\gamma_{\delta+3,l'_1} & \dots & \mu_{l'}\cdot\gamma_{\delta+3,l'_{\delta+2}} \end{bmatrix}.$$

Therefore, for $\det(B_{k,l}) = 0$,

$$\det(B_{k,l}) = \det \begin{bmatrix} \gamma_{1,k'_1} & \cdots & \gamma_{1,k'_{\delta+2}} & & & \\ \vdots & \ddots & \vdots & & & \\ \gamma_{\delta+1,k'_1} & \cdots & \gamma_{\delta+1,k'_{\delta+2}} & & & \\ & & & \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+2}} \\ & & & \vdots & \ddots & \vdots \\ & & & \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+2}} \\ \lambda_{k'}\cdot\gamma_{\delta+2,k'_1} & \dots & \lambda_{k'}\cdot\gamma_{\delta+2,k'_{\delta+2}} & \lambda_{l'}\cdot\gamma_{\delta+2,l'_1} & \dots & \lambda_{l'}\cdot\gamma_{\delta+2,l'_{\delta+2}} \\ \mu_{k'}\cdot\gamma_{\delta+3,k'_1} & \dots & \mu_{k'}\cdot\gamma_{\delta+3,k'_{\delta+2}} & \mu_{l'}\cdot\gamma_{\delta+3,l'_1} & \dots & \mu_{l'}\cdot\gamma_{\delta+3,l'_{\delta+2}} \end{bmatrix} = 0$$

$$\Rightarrow \det \begin{bmatrix} \det \begin{pmatrix} \gamma_{1,k'_1} & \cdots & \gamma_{1,k'_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta+1,k'_1} & \cdots & \gamma_{\delta+1,k'_{\delta+2}} \\ \lambda_{k'}\cdot\gamma_{\delta+2,k'_1} & \cdots & \lambda_{k'}\cdot\gamma_{\delta+2,k'_{\delta+2}} \end{pmatrix} & \det \begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+2}} \\ \lambda_{l'}\cdot\gamma_{\delta+2,l'_1} & \cdots & \lambda_{l'}\cdot\gamma_{\delta+2,l'_{\delta+2}} \end{pmatrix} \\ \det \begin{pmatrix} \gamma_{1,k'_1} & \cdots & \gamma_{1,k'_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta+1,k'_1} & \cdots & \gamma_{\delta+1,k'_{\delta+2}} \\ \mu_{k'}\cdot\gamma_{\delta+3,k'_1} & \cdots & \mu_{k'}\cdot\gamma_{\delta+3,k'_{\delta+2}} \end{pmatrix} & \det \begin{pmatrix} \gamma_{1,l'_1} & \cdots & \gamma_{1,l'_{\delta+2}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta+1,l'_1} & \cdots & \gamma_{\delta+1,l'_{\delta+2}} \\ \mu_{l'}\cdot\gamma_{\delta+3,l'_1} & \cdots & \mu_{l'}\cdot\gamma_{\delta+3,l'_{\delta+2}} \end{pmatrix} \end{bmatrix} = 0$$

$$\Rightarrow \det \begin{bmatrix} \lambda_{k'}\cdot\frac{\prod_{i\in[\delta+1]}(\beta_i-\beta_{\delta+2})}{\prod_{i\in k'_S}(\alpha_i-\beta_{\delta+2})} & \lambda_{l'}\cdot\frac{\prod_{i\in[\delta+1]}(\beta_i-\beta_{\delta+2})}{\prod_{i\in l'_S}(\alpha_i-\beta_{\delta+2})} \\ \mu_{k'}\cdot\frac{\prod_{i\in[\delta+1]}(\beta_i-\beta_{\delta+3})}{\prod_{i\in k'_S}(\alpha_i-\beta_{\delta+3})} & \mu_{l'}\cdot\frac{\prod_{i\in[\delta+1]}(\beta_i-\beta_{\delta+3})}{\prod_{i\in l'_S}(\alpha_i-\beta_{\delta+3})} \end{bmatrix} = 0$$

$$\Rightarrow \det \begin{bmatrix} \lambda_{k'} & \lambda_{l'} \\ \mu_{k'}\prod_{i\in k'_S}\frac{(\alpha_i-\beta_{\delta+2})}{(\alpha_i-\beta_{\delta+3})} & \mu_{l'}\prod_{i\in l'_S}\frac{(\alpha_i-\beta_{\delta+2})}{(\alpha_i-\beta_{\delta+3})} \end{bmatrix} = 0$$

Where $k'_S = \{k'_1, \ldots, k'_{\delta+2}\}$ and $l'_S = \{l'_1, \ldots, l'_{\delta+2}\}$. The terms $c_{l^{(i)}}$, d and $e_{l^{(i)}}$ were factored out from the above determinant where,

- $c_{l^{(i)}} = \prod_{f>g, f,g \in l_S^{(i)}} (\alpha_f - \alpha_g)$.
- $d = \prod_{f>g, f,g \in [\delta+1]} (\beta_f - \beta_g)$.
- $e_{l^{(i)}} = \prod_{f \in l_S^{(i)}, g \in [\delta+1]} (\alpha_f - \beta_g)$.

By the choice of α_i's, $\prod_{i \in x} \frac{(\alpha_i - \beta_{\delta+2})}{(\alpha_i - \beta_{\delta+3})} \in G$ for $x = k'_S, l'_S$. This yet again is a linear combination of two λ's. Hence this determinant is non-zero and the erasure pattern correctable.

(e) Each global erasure is in their own different local code and do not share with the mid-level erasures (figure 4.11).

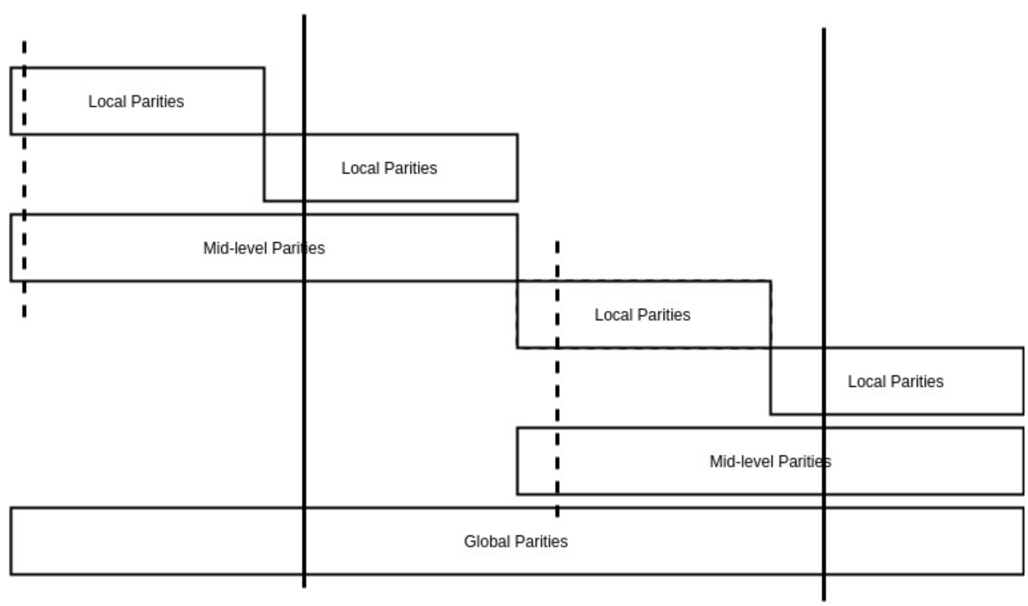

Figure 4.11: Erasure pattern for case 2.e

There are four local groups where the erasure occurs, two in each mid-level code. Let the affected mid-level codes be k and l while the local codes within, where the erasure occurs, be $k^{(1)}$ and $k^{(2)}$ and $l^{(1)}$ and $l^{(2)}$ respectively. The matrix $B_{k,l}$ is similar to lemma 4.4.1

$$B_{k,l} = \begin{bmatrix} A & \\ & B \\ C & D \end{bmatrix} \Rightarrow \det(B_{k,l}) = \det \begin{bmatrix} \det \begin{pmatrix} A \\ C^{(1)} \end{pmatrix} & \det \begin{pmatrix} B \\ D^{(1)} \end{pmatrix} \\ \det \begin{pmatrix} A \\ C^{(2)} \end{pmatrix} & \det \begin{pmatrix} B \\ D^{(2)} \end{pmatrix} \end{bmatrix} = 0$$

$$A = \begin{bmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} & & & \\ \vdots & \ddots & \vdots & & & \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} & & & \\ & & & \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ & & & \vdots & \ddots & \vdots \\ & & & \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ \gamma_{\delta+1,k_1^{(1)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(1)}} & \gamma_{\delta+1,k_1^{(2)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(2)}} \end{bmatrix}$$

$$B = \begin{bmatrix} \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+1}^{(1)}} & & & \\ \vdots & \ddots & \vdots & & & \\ \gamma_{\delta,l_1^{(1)}} & \cdots & \gamma_{\delta,l_{\delta+1}^{(1)}} & & & \\ & & & \gamma_{1,l_1^{(2)}} & \cdots & \gamma_{1,l_{\delta+1}^{(2)}} \\ & & & \vdots & \ddots & \vdots \\ & & & \gamma_{\delta,l_1^{(2)}} & \cdots & \gamma_{\delta,l_{\delta+1}^{(2)}} \\ \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+1}^{(1)}} & \gamma_{\delta+1,l_1^{(2)}} & \cdots & \gamma_{\delta+1,l_{\delta+1}^{(2)}} \end{bmatrix}$$

$$C = \begin{bmatrix} \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_1^{(1)}} & \cdots & \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(1)}} & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_1^{(2)}} & \cdots & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(2)}} \\ \mu_{k^{(1)}} \cdot \gamma_{\delta+3,k_1^{(1)}} & \cdots & \mu_{k^{(1)}} \cdot \gamma_{\delta+3,k_{\delta+1}^{(1)}} & \mu_{k^{(2)}} \cdot \gamma_{\delta+3,k_1^{(2)}} & \cdots & \mu_{k^{(2)}} \cdot \gamma_{\delta+3,k_{\delta+1}^{(2)}} \end{bmatrix}$$

$$D = \begin{bmatrix} \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_1^{(1)}} & \cdots & \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_{\delta+1}^{(1)}} & \lambda_{l^{(2)}} \cdot \gamma_{\delta+2,l_1^{(2)}} & \cdots & \lambda_{l^{(2)}} \cdot \gamma_{\delta+2,l_{\delta+1}^{(2)}} \\ \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_1^{(1)}} & \cdots & \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_{\delta+1}^{(1)}} & \mu_{l^{(2)}} \cdot \gamma_{\delta+3,l_1^{(2)}} & \cdots & \mu_{l^{(2)}} \cdot \gamma_{\delta+3,l_{\delta+1}^{(2)}} \end{bmatrix}$$

To calculate the whole determinant, We consider the first element,

$$\det\begin{pmatrix} A \\ C^{(1)} \end{pmatrix} = \det\begin{bmatrix} & \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} & & & & \\ & \vdots & \ddots & \vdots & & & & \\ & \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} & & & & \\ & & & & & \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ & & & & & \vdots & \ddots & \vdots \\ & & & & & \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ & \gamma_{\delta+1,k_1^{(1)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(1)}} & & \gamma_{\delta+1,k_1^{(2)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(2)}} \\ \lambda_{k^{(1)}} \cdot & \gamma_{\delta+2,k_1^{(1)}} & \cdots & \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(1)}} & \lambda_{k^{(2)}} \cdot & \gamma_{\delta+2,k_1^{(2)}} & \cdots & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(2)}} \end{bmatrix}$$

$$= \det\begin{bmatrix} \det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \gamma_{\delta+1,k_1^{(1)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(1)}} \end{pmatrix} & \det\begin{pmatrix} \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ \gamma_{\delta+1,k_1^{(2)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(2)}} \end{pmatrix} \\ \det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_1^{(1)}} & \cdots & \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(1)}} \end{pmatrix} & \det\begin{pmatrix} \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_1^{(2)}} & \cdots & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(2)}} \end{pmatrix} \end{bmatrix}$$

$$= \det\begin{bmatrix} \frac{c_{k^{(1)}} d \prod_{i\in[\delta]}(\beta_i-\beta_{\delta+1})}{e_{k^{(1)}} \prod_{i\in k_S^{(1)}}(\alpha_i-\beta_{\delta+1})} & \frac{c_{k^{(2)}} d \prod_{i\in[\delta]}(\beta_i-\beta_{\delta+1})}{e_{k^{(2)}} \prod_{i\in k_S^{(2)}}(\alpha_i-\beta_{\delta+1})} \\ \lambda_{k^{(1)}} \cdot \frac{c_{k^{(1)}} d \prod_{i\in[\delta]}(\beta_i-\beta_{\delta+2})}{e_{k^{(1)}} \prod_{i\in k_S^{(1)}}(\alpha_i-\beta_{\delta+2})} & \lambda_{k^{(2)}} \cdot \frac{c_{k^{(2)}} d \prod_{i\in[\delta]}(\beta_i-\beta_{\delta+2})}{e_{k^{(2)}} \prod_{i\in k_S^{(2)}}(\alpha_i-\beta_{\delta+2})} \end{bmatrix}$$

$$= \frac{c_{k^{(1)}} c_{k^{(2)}} d^2}{e_{k^{(1)}} e_{k^{(2)}}} \prod_{i\in[\delta]}(\beta_i-\beta_{\delta+1})(\beta_i-\beta_{\delta+2}) \cdot \det\begin{bmatrix} \prod_{i\in k_S^{(1)}} \frac{1}{\alpha_i-\beta_{\delta+1}} & \prod_{i\in k_S^{(2)}} \frac{1}{\alpha_i-\beta_{\delta+1}} \\ \lambda_{k^{(1)}} \cdot \prod_{i\in k_S^{(1)}} \frac{1}{\alpha_i-\beta_{\delta+2}} & \lambda_{k^{(2)}} \cdot \prod_{i\in k_S^{(2)}} \frac{1}{\alpha_i-\beta_{\delta+2}} \end{bmatrix}.$$

Where,

- $k_S^{(i)} = \{k_1^{(i)}, \dots, k_{\delta+2}^{(i)}\}$.
- $c_{k^{(i)}} = \prod_{f>g, f,g \in k_S^{(i)}} (\alpha_f - \alpha_g)$.
- $d = \prod_{f>g, f,g\in[\delta]} (\beta_f - \beta_g)$.
- $e_{k^{(i)}} = \prod_{f\in k_S^{(i)}, g\in[\delta]} (\alpha_f - \beta_g)$.

Applying all this in the main determinant and setting,

$$\det(B_{k,l}) = 0$$

and factoring out the common multiples, we get

$$\det\begin{bmatrix} \det\begin{pmatrix} \prod_{i\in k_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+1}} & \prod_{i\in k_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+1}} \\ \lambda_{k^{(1)}}\cdot\prod_{i\in k_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+2}} & \lambda_{k^{(2)}}\cdot\prod_{i\in k_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+2}} \end{pmatrix} & \det\begin{pmatrix} \prod_{i\in l_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+1}} & \prod_{i\in l_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+1}} \\ \lambda_{l^{(1)}}\cdot\prod_{i\in l_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+2}} & \lambda_{l^{(2)}}\cdot\prod_{i\in l_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+2}} \end{pmatrix} \\ \det\begin{pmatrix} \prod_{i\in k_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+1}} & \prod_{i\in k_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+1}} \\ \mu_{k^{(1)}}\cdot\prod_{i\in k_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+3}} & \mu_{k^{(2)}}\cdot\prod_{i\in k_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+3}} \end{pmatrix} & \det\begin{pmatrix} \prod_{i\in l_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+1}} & \prod_{i\in l_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+1}} \\ \mu_{l^{(1)}}\cdot\prod_{i\in l_S^{(1)}}\frac{1}{\alpha_i-\beta_{\delta+3}} & \mu_{l^{(2)}}\cdot\prod_{i\in l_S^{(2)}}\frac{1}{\alpha_i-\beta_{\delta+3}} \end{pmatrix} \end{bmatrix} = 0$$

$$\det\begin{bmatrix} \lambda_{k^{(2)}}\cdot\prod_{i\in k_S^{(2)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} - \lambda_{k^{(1)}}\cdot\prod_{i\in k_S^{(1)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} & \lambda_{l^{(2)}}\cdot\prod_{i\in l_S^{(2)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} - \lambda_{l^{(1)}}\cdot\prod_{i\in l_S^{(1)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+2}} \\ \mu_{k^{(2)}}\cdot\prod_{i\in k_S^{(2)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} - \mu_{k^{(1)}}\cdot\prod_{i\in k_S^{(1)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} & \mu_{l^{(2)}}\cdot\prod_{i\in l_S^{(2)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} - \mu_{l^{(1)}}\cdot\prod_{i\in l_S^{(1)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} \end{bmatrix} = 0$$

Where similarly, $l_S^{(i)} = \{l_1^{(i)}, \ldots, l_{\delta+2}^{(i)}\}$.

Now, since the λ_i's are 4-wise independent over $\mathbb{F}_{q_0}$, the first row is never zero. Similarly, all the μ_j's are in different cosets of G and by choice of α's $\prod_{i\in l_S^{(j)},k_S^{(j)}}\frac{\alpha_i-\beta_{\delta+1}}{\alpha_i-\beta_{\delta+3}} \in G$. Hence the last row isn't zero either. Then this determinant resolves into a linear combination for 4 different values of λ_is. Hence, by linear independence rules of λ, this determinant is also non-zero.

(f) In this case, one of the global erasure shares the local code with a mid-level code while the other does not (figure 4.12). Assume that the k^{th} and l^{th} mid-level codes are affected. Let the local codes within, where the erasure occurs, be $k^{(1)}$, $k^{(2)}$ and $l^{(1)}$. The matrix $B_{k,l}$,

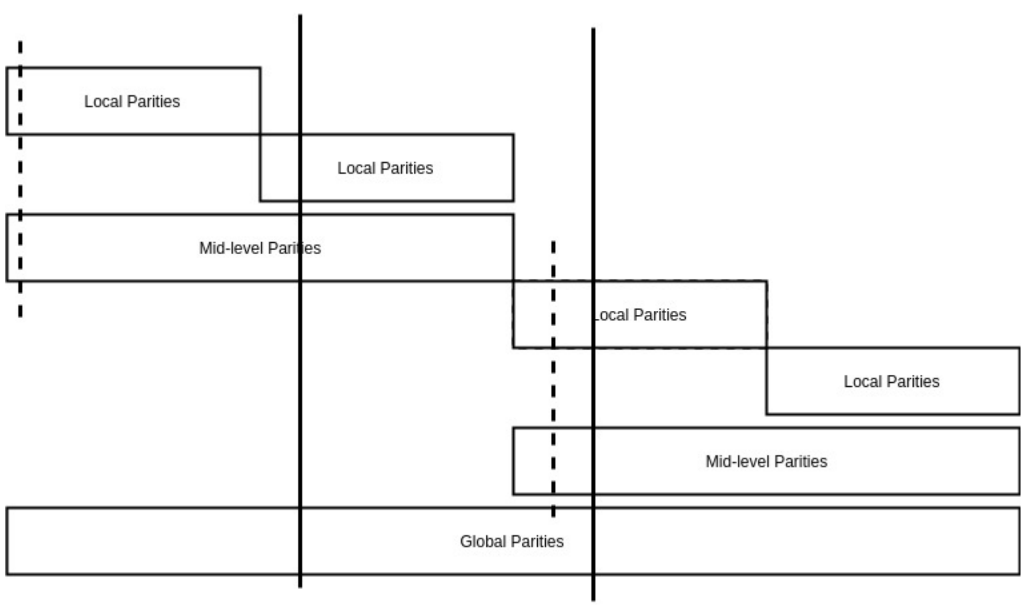

Figure 4.12: Erasure pattern for case 2.f

$$B_{k,l} = \begin{bmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} & & & & & & \\ \vdots & \ddots & \vdots & & & & & & \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} & & & & & & \\ & & & \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} & & & \\ \gamma_{\delta+1,k_1^{(1)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(1)}} & \gamma_{\delta+1,k_1^{(2)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(2)}} & & & \\ & & & & & & \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+2}^{(1)}} \\ & & & & & & \vdots & \ddots & \vdots \\ & & & & & & \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+2}^{(1)}} \\ \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_1^{(1)}} & \cdots & \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(1)}} & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_1^{(2)}} & \cdots & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(2)}} & \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_1^{(1)}} & \cdots & \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_{\delta+2}^{(1)}} \\ \mu_{k^{(1)}} \cdot \gamma_{\delta+3,k_1^{(1)}} & \cdots & \mu_{k^{(1)}} \cdot \gamma_{\delta+3,k_{\delta+1}^{(1)}} & \mu_{k^{(2)}} \cdot \gamma_{\delta+3,k_1^{(2)}} & \cdots & \mu_{k^{(2)}} \cdot \gamma_{\delta+3,k_{\delta+1}^{(2)}} & \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_1^{(1)}} & \cdots & \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_{\delta+2}^{(1)}} \end{bmatrix}$$

Now, after permuting one row, we can apply 4.4.2 to expand the matrix for the determinant,

$$\det(B_{k,l}) =$$

$$\det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \gamma_{\delta+1,k_1^{(1)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(1)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_1^{(2)}} & \cdots & \lambda_{k^{(2)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(2)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+2}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l_1^{(1)}} & \cdots & \gamma_{\delta,l_{\delta+2}^{(1)}} \\ \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+2}^{(1)}} \\ \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_1^{(1)}} & \cdots & \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_{\delta+2}^{(1)}} \end{pmatrix} +$$

$$\det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \gamma_{\delta+1,k_1^{(1)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(1)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ \mu_{k^{(2)}} \cdot \gamma_{\delta+3,k_1^{(2)}} & \cdots & \mu_{k^{(2)}} \cdot \gamma_{\delta+3,k_{\delta+1}^{(2)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+2}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l_1^{(1)}} & \cdots & \gamma_{\delta,l_{\delta+2}^{(1)}} \\ \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+2}^{(1)}} \\ \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_1^{(1)}} & \cdots & \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_{\delta+2}^{(1)}} \end{pmatrix} +$$

$$\det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_1^{(1)}} & \cdots & \lambda_{k^{(1)}} \cdot \gamma_{\delta+2,k_{\delta+1}^{(1)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \gamma_{\delta+1,k_1^{(2)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(2)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+2}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l_1^{(1)}} & \cdots & \gamma_{\delta,l_{\delta+2}^{(1)}} \\ \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+2}^{(1)}} \\ \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_1^{(1)}} & \cdots & \mu_{l^{(1)}} \cdot \gamma_{\delta+3,l_{\delta+2}^{(1)}} \end{pmatrix} -$$

$$\det\begin{pmatrix} \gamma_{1,k_1^{(1)}} & \cdots & \gamma_{1,k_{\delta+1}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(1)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(1)}} \\ \mu_{k^{(1)}} \cdot \gamma_{\delta+3,k_1^{(1)}} & \cdots & \mu_{k^{(1)}} \cdot \gamma_{\delta+3,k_{\delta+1}^{(1)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,k_1^{(2)}} & \cdots & \gamma_{1,k_{\delta+1}^{(2)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,k_1^{(2)}} & \cdots & \gamma_{\delta,k_{\delta+1}^{(2)}} \\ \gamma_{\delta+1,k_1^{(2)}} & \cdots & \gamma_{\delta+1,k_{\delta+1}^{(2)}} \end{pmatrix} \det\begin{pmatrix} \gamma_{1,l_1^{(1)}} & \cdots & \gamma_{1,l_{\delta+2}^{(1)}} \\ \vdots & \ddots & \vdots \\ \gamma_{\delta,l_1^{(1)}} & \cdots & \gamma_{\delta,l_{\delta+2}^{(1)}} \\ \gamma_{\delta+1,l_1^{(1)}} & \cdots & \gamma_{\delta+1,l_{\delta+2}^{(1)}} \\ \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_1^{(1)}} & \cdots & \lambda_{l^{(1)}} \cdot \gamma_{\delta+2,l_{\delta+2}^{(1)}} \end{pmatrix}$$

Now, in this massive expansion, we can take λ_i and μ_j out of the determinants. What we will find is that each term is $\lambda_i \mu_j$ multiplied by the product of the determinant of three Cauchy matrices. Each of those determinant $\in \mathbb{F}_{q_0}$.

Hence the final determinant is actually the linear combination of three λ_i in $\mathbb{F}_{q_0}$. Hence $\det(B_{k,l}) \neq 0$.

$\square$

Chapter 5

Conclusions

In the previous chapters, you saw what Hierarchical Codes are and how to construct such codes with the MR property. In addition to that we proved some properties regarding hierarchical codes, especially regarding local and data-local versions and conversion between them. We provided a general construction for all parameter sets. That construction required a high field size. So we go on to provide a slightly different construction for a restricted set of parameters using tensor product codes which allowed for smaller field size. Since a lot of practically used erasure codes have a small number of global parities, we provide alternate constructions for the codes that use even smaller fields still.

But research on Hierarchical Codes is far from over. There are a lot of open problems awaiting attention.

- We still haven't considered under what bounds our codes work in. It would be interesting to attain some field size bounds on generic codes and see how well our constructions fare against them. We would also be interested in having field specific bounds on the parameters we can have.
- Our work defined Hierarchical Codes and their MR versions only for two levels of locality. The concept can be extended to a higher y-levels of locality. This will allow for a more fine grained control over the increase in locality with the number of erasures.
- All our constructions focus on reducing the number of symbols accessed for repair. But we haven't looked into actually optimising the repair process itself. [22] shows one way to optimise repair. It would be great if those techniques can be adapted to our case as well.

Related Publications

1. Nair, Aaditya M. and V. Lalitha. "Maximally Recoverable Codes with Hierarchical Locality." accepted in *IEEE National Conference on Communications (NCC) 2019* **(Best Paper Award - Runner Up)**

2. Nair, Aaditya M. and V. Lalitha. "Maximally Recoverable Codes with Hierarchical Locality." available at *arxiv:1901.02867*,2019

Bibliography

[1] S. Ballentine and A. Barg. Codes on curves with hierarchical locality. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 1201–1205. IEEE, 2018.

[2] B. Bartan and M. Wootters. Repairing multiple failures for scalar mds codes. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1145–1152. IEEE, 2017.

[3] M. Blaum. Construction of pmds and sd codes extending raid 5. *arXiv preprint arXiv:1305.0032*, 2013.

[4] M. Blaum, J. L. Hafner, and S. Hetzler. Partial-mds codes and their application to raid type of architectures. *IEEE Transactions on Information Theory*, 59(7):4510–4519, 2013.

[5] M. Blaum, J. S. Plank, M. Schwartz, and E. Yaakobi. Construction of partial mds and sector-disk codes with two global parity symbols. *IEEE Transactions on Information Theory*, 62(5):2673–2681, 2016.

[6] V. R. Cadambe and A. Mazumdar. Bounds on the size of locally recoverable codes. *IEEE transactions on information theory*, 61(11):5787–5794, 2015.

[7] G. Calis and O. O. Koyluoglu. A general construction for pmds codes. *IEEE Communications Letters*, 21(3):452–455, 2017.

[8] B. Chen, S.-T. Xia, and J. Hao. Locally repairable codes with multiple (r i, δ i)-localities. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2038–2042. IEEE, 2017.

[9] J. Chen, K. W. Shum, Q. Yu, and C. W. Sung. Sector-disk codes and partial mds codes with up to three global parities. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1876–1880. IEEE, 2015.

[10] M. Chen, C. Huang, and J. Li. On the maximally recoverable property for multi-protection group codes. In *2007 IEEE International Symposium on Information Theory*, pages 486–490. IEEE, 2007.

[11] H. Dau, I. M. Duursma, H. M. Kiah, and O. Milenkovic. Repairing reed-solomon codes with multiple erasures. *IEEE Transactions on Information Theory*, 64(10):6567–6582, 2018.

[12] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE transactions on information theory*, 56(9):4539–4551, 2010.

[13] A. Duminuco and E. Biersack. Hierarchical codes: How to make erasure codes attractive for peer-to-peer storage systems. In *2008 Eighth International Conference on Peer-to-Peer Computing*, pages 89–98. IEEE, 2008.

[14] M. Forbes and S. Yekhanin. On the locality of codeword symbols in non-linear codes. *Discrete mathematics*, 324:78–84, 2014.

[15] R. Gabrys, E. Yaakobi, M. Blaum, and P. H. Siegel. Constructions of partial mds codes over small fields. *IEEE Transactions on Information Theory*, 2018.

[16] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, and S. Yekhanin. Maximally recoverable codes for grid-like topologies. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2092–2108. Society for Industrial and Applied Mathematics, 2017.

[17] P. Gopalan, C. Huang, B. Jenkins, and S. Yekhanin. Explicit maximally recoverable codes with locality. *IEEE Transactions on Information Theory*, 60(9):5245–5256, 2014.

[18] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin. On the locality of codeword symbols. *IEEE Transactions on Information Theory*, 58(11):6925–6934, 2012.

[19] S. Goparaju and R. Calderbank. Binary cyclic codes that are locally repairable. In *2014 IEEE International Symposium on Information Theory*, pages 676–680. IEEE, 2014.

[20] S. Gopi, V. Guruswami, and S. Yekhanin. Maximally recoverable lrcs: A field size lower bound and constructions for few heavy parities. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2154–2170. SIAM, 2019.

[21] V. Guruswami, L. Jin, and C. Xing. Constructions of maximally recoverable local reconstruction codes via function fields. *arXiv preprint arXiv:1808.04539*, 2018.

[22] V. Guruswami and M. Wootters. Repairing reed-solomon codes. *IEEE transactions on Information Theory*, 63(9):5684–5698, 2017.

[23] J. Han and L. A. Lastras-Montano. Reliable memories with subline accesses. In *2007 IEEE International Symposium on Information Theory*, pages 2531–2535. IEEE, 2007.

[24] G. Hu and S. Yekhanin. New constructions of sd and mr codes over small finite fields. *arXiv preprint arXiv:1605.02290*, 2016.

[25] Y. Hu, H. C. Chen, P. P. Lee, and Y. Tang. Nccloud: applying network coding for the storage repair in a cloud-of-clouds. In *FAST*, page 21, 2012.

[26] C. Huang, M. Chen, and J. Li. Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems. In *Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007)*.

[27] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin. Erasure coding in windows azure storage. In *Presented as part of the 2012 {USENIX} Annual Technical Conference ({USENIX}{ATC} 12)*, pages 15–26, 2012.

[28] S. Kadhe and A. Sprintson. Codes with unequal locality. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 435–439. IEEE, 2016.

[29] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar. Codes with local regeneration. In *2013 Information Theory and Applications Workshop (ITA)*, pages 1–5. IEEE, 2013.

[30] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar. Codes with local regeneration and erasure correction. *IEEE Transactions on Information Theory*, 60(8):4637–4660, 2014.

[31] D. M. Kane, S. Lovett, and S. Rao. Labeling the complete bipartite graph with no zero cycles. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 24, page 33, 2017.

[32] G. Kim and J. Lee. Locally repairable codes with unequal local erasure correction. *IEEE Transactions on Information Theory*, 64(11):7137–7152, 2018.

[33] K. Kralevska, D. Gligoroski, R. E. Jensen, and H. Øverby. Hashtag erasure codes: From theory to practice. *IEEE Transactions on Big Data*, 4(4):516–529, 2017.

[34] K. Kralevska, D. Gligoroski, and H. Øverby. General sub-packetized access-optimal regenerating codes. *IEEE Communications Letters*, 20(7):1281–1284, 2016.

[35] M. N. Krishnan, B. Puranik, P. V. Kumar, I. Tamo, and A. Barg. Exploiting locality for improved decoding of binary cyclic codes. *IEEE Transactions on Communications*, 66(6):2346–2358, 2018.

[36] X. Li, L. Ma, and C. Xing. Construction of asymptotically good locally repairable codes via automorphism groups of function fields. *IEEE Transactions on Information Theory*, 65(11):7087–7094, 2019.

[37] U. Martnez-Penas and F. R. Kschischang. Universal and dynamic locally repairable codes with maximal recoverability via sum-rank codes. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 792–799. IEEE, 2018.

[38] T. K. Moon. Error correction coding. *Mathematical Methods and Algorithms. Jhon Wiley and Son*, pages 2001–2006, 2005.

[39] A. M. Nair and V. Lalitha. Maximally recoverable codes with hierarchical locality. *arXiv preprint arXiv:1901.02867*, 2019.

[40] F. Oggier and A. Datta. Self-repairing homomorphic codes for distributed storage systems. In *2011 Proceedings IEEE INFOCOM*, pages 1215–1223. IEEE, 2011.

[41] D. S. Papailiopoulos and A. G. Dimakis. Locally repairable codes. *IEEE Transactions on Information Theory*, 60(10):5843–5855, 2014.

[42] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar. Optimal linear codes with a local-error-correction property. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2776–2780. IEEE, 2012.

[43] K. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran. Explicit construction of optimal exact regenerating codes for distributed storage. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1243–1249. IEEE, 2009.

[44] K. V. Rashmi, N. B. Shah, and P. V. Kumar. Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction. *IEEE Transactions on Information Theory*, 57(8):5227–5239, 2011.

[45] N. Raviv. Asymptotically optimal regenerating codes over any field. *IEEE Transactions on Information Theory*, 64(11):7178–7187, 2018.

[46] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath. Optimal locally repairable and secure codes for distributed storage systems. *IEEE Transactions on Information Theory*, 60(1):212–236, 2013.

[47] R. Roth. *Introduction to coding theory*. Cambridge University Press, 2006.

[48] B. Sasidharan, G. K. Agarwal, and P. V. Kumar. Codes with hierarchical locality. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1257–1261. IEEE, 2015.

[49] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur. Xoring elephants: Novel erasure codes for big data. In *Proceedings of the VLDB Endowment*, volume 6, pages 325–336. VLDB Endowment, 2013.

[50] N. B. Shah, K. Rashmi, P. V. Kumar, and K. Ramchandran. Interference alignment in regenerating codes for distributed storage: Necessity and code constructions. *IEEE Transactions on Information Theory*, 58(4):2134–2158, 2011.

[51] K. Shanmugam, D. S. Papailiopoulos, A. G. Dimakis, and G. Caire. A repair framework for scalar mds codes. *IEEE Journal on Selected Areas in Communications*, 32(5):998–1007, 2014.

[52] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath. Optimal locally repairable codes via rank-metric codes. In *2013 IEEE International Symposium on Information Theory*, pages 1819–1823. IEEE, 2013.

[53] I. Tamo and A. Barg. A family of optimal locally recoverable codes. *IEEE Transactions on Information Theory*, 60(8):4661–4676, 2014.

[54] I. Tamo, A. Barg, and A. Frolov. Bounds on the parameters of locally recoverable codes. *IEEE Transactions on information theory*, 62(6):3070–3083, 2016.

[55] I. Tamo, A. Barg, S. Goparaju, and R. Calderbank. Cyclic lrc codes, binary lrc codes, and upper bounds on the distance of cyclic codes. *arXiv preprint arXiv:1603.08878*, 2016.

[56] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis. Optimal locally repairable codes and connections to matroid theory. *IEEE Transactions on Information Theory*, 62(12):6661–6671, 2016.

[57] M. Vajha, V. Ramkumar, B. Puranik, G. Kini, E. Lobo, B. Sasidharan, P. V. Kumar, A. Barg, M. Ye, S. Narayanamurthy, et al. Clay codes: Moulding {MDS} codes to yield an {MSR} code. In *16th {USENIX} Conference on File and Storage Technologies ({FAST} 18)*, pages 139–154, 2018.

[58] J. Wolf. On codes derivable from the tensor product of check matrices. *IEEE Transactions on Information Theory*, 11(2):281–284, 1965.

[59] A. Zeh and E. Yaakobi. Optimal linear and cyclic locally repairable codes over small fields. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.