Optimal Locally Repairable Codes with Local Minimum Storage Regeneration via Rank-Metric Codes

Ankit S. Rawat*, Natalia Silberstein*, O. Ozan Koyluoglu, and Sriram Vishwanath

***Abstract*—This paper presents a new explicit construction for locally repairable codes (LRCs) for distributed storage systems. The codes possess all-symbols locality and maximal possible minimum distance, or equivalently, can tolerate the maximal number of node failures. This construction, based on maximum rank distance (MRD) Gabidulin codes, provides minimum distance optimal vector and scalar LRCs for a wide range of parameters. In addition, vector LRCs that allow for efficient local repair of failed nodes are considered. Towards this, the paper derives an upper bound on the amount of data that can be stored on DSS employing minimum distance optimal LRCs with given repair bandwidth, and presents codes which attain this bound by combining MRD and minimum storage regenerating (MSR) codes.**

***Index Terms*—Coding for distributed storage systems, locally repairable codes, repair bandwidth efficient codes.**

I. Introduction

In distributed storage systems (DSS), it is desirable that data be reliably stored over a network of nodes in such a way that a user (*data collector*) can retrieve the stored data even if some nodes fail. To achieve such a resilience against node failures, DSS introduce data redundancy based on different coding techniques. For example, erasures codes are widely used in such systems: When using an (n, k) code, data to be stored is first divided into k blocks; subsequently, these k information blocks are encoded into n blocks stored on n distinct nodes in the system. In addition, when a single node fails, the system reconstructs the data stored in the failed node to keep the required level of redundancy. This process of data reconstruction for a failed node is called *node repair process* [1]. During a node repair process, the node which is added to the system to replace the failed node downloads data from a set of appropriate and accessible nodes.

There are two important goals that guide the design of codes for DSS: reducing the *repair bandwidth*, i.e. the amount of data downloaded from system nodes during the node repair process, and achieving *locality*, i.e. reducing the number of nodes participating in the node repair process. These goals underpin the design of two families of codes for DSS called *regenerating codes* (see [1]–[8] and references therein) and *locally repairable codes* (see [9]–[20]), respectively.

The authors are with the Laboratory of Informatics, Networks and Communications, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78751 USA. E-mail: ankitsr@utexas.edu, {natalys, ozan, sriram}@austin.utexas.edu.

* A. S. Rawat and N. Silberstein contributed equally to this work.

In this paper we focus on the locally repairable codes (LRCs). Recently, these codes have drawn significant attention within the research community. Oggier et al. [12], [13] present coding schemes which facilitate local node repair. In [9], Gopalan et al. establish an upper bound on the minimum distance of scalar LRCs, which is analogous to the Singleton bound. The paper also showes that pyramid codes, presented in [17], achieve this bound with information symbols locality. Subsequently, the work by Prakash et al. extends the bound to a more general definition of scalar LRCs [11]. (Han and Lastras-Montano [18] provide a similar upper bound which is coincident with the one in [11] for small minimum distances, and also present codes that attain this bound in the context of reliable memories.) In [10], Papailiopoulos and Dimakis generalize the bound in [9] to vector codes, and present locally repairable coding schemes which exhibits MDS property at the cost of small amount of additional storage per node.

The main contributions of this paper are as follows. First, in Section II, we generalize the definition of *scalar* locally repairable codes, presented in [11] to *vector* locally repairable codes. For such codes, every node storing α symbols from a given field $\mathbb{F}$, can be locally repaired by using data stored in at most r other nodes from a group of nodes of size $r + \delta - 1 < n$, which we call a *local group*, where n is the number of system nodes, and r and δ are the given locality parameters. Subsequently, in Section III, we derive an upper bound on the minimum distance $d_{\min}$ of the vector codes that satisfy a given locality constraint, which establishes a trade off between node failure resilience (i.e., $d_{\min}$) and per node storage α. [1] The bound presented in [10] can be considered as a special case of our bound with $\delta = 2$. Further, we present an explicit construction for LRCs which attain this bound on minimum distance. This construction is based on maximum rank distance (MRD) Gabidulin codes, which are a rank-metric analog of Reed-Solomon codes. The *scalar* and *vector* LRCs that are obtained by this construction are the first explicit optimal locally repairable codes with $(r + \delta - 1) \nmid n$.

Finally, in Section IV, we discuss a hybrid construction, which optimizes repair bandwidth for given locality parameters (r, δ). In particular, we focus on locally repairable codes with local minimum storage regeneration (MSR-LRCs), where the code allows for a) having the maximal possible

[1] In a parallel and independent work, [19], Kamath et al. also provide upper bounds on minimum distance together with constructions and existence results for vector LRCs.

minimum distance for given locality constraints; b) minimizing storage overhead at each node given locality constraints; and c) minimizing the repair bandwidth for local repairs. We first provide an upper bound on the amount of data that can be stored on minimum distance optimal DSS given a fixed repair bandwidth. We then present codes, based on the combination of MRD and MSR codes, that attain this bound. We conclude the paper with Section V.

II. Background

A. System Parameters

Let $\mathbf{f}$ be a file of size $\mathcal{M}$ over finite field $\mathbb{F}$ that needs to be stored on a DSS with n nodes. Each node is assumed to store α symbols over $\mathbb{F}$.

B. Vector Codes

A linear $[n, \mathcal{M}, d_{\min}, \alpha]_q$ vector code C over $\mathbb{F}_q$ of length n is defined as a linear subspace of $\mathbb{F}_q^{\alpha n}$ of dimension $\mathcal{M}$. The symbols $\mathbf{c}_i$, $1 \leq i \leq n$, of a codeword $\mathbf{c} \in C$ belong to $\mathbb{F}_q^{\alpha}$. The minimum distance $d_{\min}$ of C is defined as the minimum Hamming distance over $\mathbb{F}_q^{\alpha}$. An alternative definition for the minimum distance of an $[n, \mathcal{M}, d_{\min}, \alpha]_q$ vector code is as follows:

Definition 1. *The minimum distance $d_{\min}$ of a vector code C of dimension $\mathcal{M}$ is defined as*

$$d_{\min} = n - \max_{\mathcal{A} \subseteq [n]: H(\mathbf{c}_{\mathcal{A}}) < \mathcal{M}} |\mathcal{A}|, \tag{1}$$

where $\mathcal{A} = \{i_1, \ldots, i_{|\mathcal{A}|}\} \subseteq [n]$ and $\mathbf{c}_{\mathcal{A}} = (\mathbf{c}_{i_1}, \ldots, \mathbf{c}_{i_{|\mathcal{A}|}})$.

Vector codes are also known as *array* codes. An $[n, \mathcal{M}, d_{\min}, \alpha]_q$ array code is called *MDS array code* if $d_{\min} = n - \mathcal{M} + 1$. Constructions for MDS array codes can be found e.g. in [25]–[27].

C. Locally Repairable Codes

In this subsection, we generalize the definition of *scalar* LRCs, presented in [11] to *vector* LRCs.

Definition 2. *We say that an $[n, \mathcal{M}, d_{\min}, \alpha]_q$ vector code C has (r, δ)* locality *if for each symbol $\mathbf{c}_i \in \mathbb{F}_q^{\alpha}$, $1 \leq i \leq n$, of a codeword $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n) \in C$, there exists a set of nodes $\Gamma(i)$ such that*

- $i \in \Gamma(i)$
- $|\Gamma(i)| \leq r + \delta - 1$
- *Minimum distance of $C|_{\Gamma(i)}$ is at least δ, where $C|_{\Gamma(i)}$ denotes the code obtained by puncturing C over set of indices $\Gamma(i) \subseteq [n]$.*

Note that the last two properties imply that each element $j \in \Gamma(i)$ can be written as a function of a set of at most r elements in $\Gamma(i)$ (not containing j) and that $H(\Gamma(i)) \leq r\alpha$.

Codes that satisfy these properties are called (r, δ, α) locally repairable codes *(LRCs).*

Note, that definition of LRCs presented in this paper generalizes the notion of LRCs given in [10], which is restricted to $\delta = 2$.

In order to store a file $\mathbf{f}$ on a DSS using an LRC, $\mathbf{f}$ is first encoded to a codeword of an LRC. Each symbol of the codeword is then stored on a different node. In particular, we have $\mathbf{x}_i = \mathbf{c}_i$, where $\mathbf{x}_i$ denotes the content of ith node. Note that a node i in locally repairable DSS can be repaired by downloading data from at most r nodes in $\Gamma(i) \backslash \{i\}$.

Remark 3. *$(r, \delta, \alpha = 1)$ LRCs are named as (r, δ) scalar LRCs.*

In [11], Prakash et al. present the following upper bound on the minimum distance of an (r, δ) scalar LRC:

$$d_{\min} \leq n - \mathcal{M} + 1 - \left(\left\lceil \frac{\mathcal{M}}{r} \right\rceil - 1 \right) (\delta - 1). \tag{2}$$

It was established in [11] that a family of pyramid codes, presented in [17], attains this bound and has *information locality*, i.e. only information symbols satisfy the locality constraint. However, an explicit construction of optimal scalar LRCs with *all-symbols locality* is known only for the case $n = \left\lceil \frac{\mathcal{M}}{r} \right\rceil (r + \delta - 1)$ [11], [18]. Towards optimal scalar LRCs for broader range of parameters, given field size $|\mathbb{F}| > \mathcal{M} n^{\mathcal{M}}$, [11] establishes the existence of scalar codes with all-symbols locality for the setting when $(r + \delta - 1) | n$. In this paper, we provide an explicit construction of optimal scalar LRCs with all-symbols locality relaxing the restriction of $(r + \delta - 1) | n$.

The following upper bound on the minimum distance of $(r, \delta = 2, \alpha)$ LRCs and a construction of codes that attain this bound was presented in [10]:

$$d_{\min} \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - \left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil + 2 \tag{3}$$

In the sequel, we generalize this bound for any $\delta \geq 2$ and present (r, δ, α) LRCs that attain this bound.

D. Maximum Rank Distance Codes

The construction presented in this paper involves a precoding step, where the file is encoded using an optimal rank-metric code, called maximum rank distance code [21], [22]. In this subsection, we present a brief introduction to rank-metric codes.

Let $\mathbb{F}_{q^m}$ be en extension field of $\mathbb{F}_q$. An element $\gamma \in \mathbb{F}_{q^m}$ can be represented as the vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)^T \in \mathbb{F}_q^m$, such that $\gamma = \sum_{i=1}^{m} \gamma_i b_i$, for a fixed basis $\{b_1, \ldots, b_m\}$ of the extension field $\mathbb{F}_{q^m}$. Using this, a vector $\mathbf{v} = (v_1, \ldots, v_N) \in \mathbb{F}_{q^m}^N$ can be represented by an $m \times N$ matrix $\mathbf{V} = [v_{i,j}]$ over $\mathbb{F}_q$, which is obtained by replacing each v_i of $\mathbf{v}$ by its vector representation $(v_{i,1}, \ldots, v_{i,m})^T$.

Definition 4. *The* rank *of a vector $\mathbf{v} \in \mathbb{F}_{q^m}^N$, denoted by* $\mathrm{rank}(\mathbf{v})$ *is defined as the rank of its $m \times N$ matrix representation $\mathbf{V}$ (over $\mathbb{F}_q$). Similarly, for two vectors $\mathbf{v}, \mathbf{u} \in \mathbb{F}_{q^m}^N$, the* rank distance *is defined by*

$$d_R(\mathbf{v}, \mathbf{u}) = \mathrm{rank}(\mathbf{V} - \mathbf{U}).$$

An $[N, K, D]_{q^m}$ rank-metric code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^N$ is a linear block code over $\mathbb{F}_{q^m}$ of length N with dimension K and minimum rank distance D. A rank-metric code that attains

the Singleton bound $D \leq N - K + 1$ in rank-metric is called *maximum rank distance* (MRD) code. For $m \geq N$, a family of MRD codes, called Gabidulin codes, was presented by Gabidulin [21]. Similar to Reed-Solomon codes, Gabidulin codes can be obtained by evaluation of polynomials, however, for Gabidulin codes a special family of polynomials, *linearized polynomials*, is used:

Definition 5. *A linearized polynomial $f(x)$ over $\mathbb{F}_{q^m}$ of $q-$degree t has the form $f(x) = \sum_{i=0}^{t} a_i x^{q^i}$, where $a_i \in \mathbb{F}_{q^m}$, and $a_t \neq 0$.*

Remark 6. *Note that evaluation of a linearized polynomial is an $\mathbb{F}_q$-linear transformation from $\mathbb{F}_{q^m}$ to itself, i.e., for any $a, b \in \mathbb{F}_q$ and $\gamma_1, \gamma_2 \in \mathbb{F}_{q^m}$, we have $f(a\gamma_1 + b\gamma_2) = af(\gamma_1) + bf(\gamma_2)$ [23].*

A codeword in a $[N, K, D = N - K + 1]_{q^m}$ Gabidulin code $\mathcal{C}^{\text{Gab}}$, $m \geq N$, is defined as $\mathbf{c} = (f(g_1), f(g_2), \ldots, f(g_N)) \in \mathbb{F}_{q^m}^N$, where $f(x)$ is a linearized polynomial over $\mathbb{F}_{q^m}$ of $q-$degree $K-1$ with K message symbols as its coefficients, and $g_1, \ldots, g_N \in \mathbb{F}_{q^m}$ are linearly independent over $\mathbb{F}_q$ [21].

Remark 7. *Given evaluations of $f(\cdot)$ at any K linearly independent (over $\mathbb{F}_q$) points in $\mathbb{F}_{q^m}$, say $(z_1, \ldots, z_K)$, one can get evaluations of $f(\cdot)$ at q^K points spanned by $\mathbb{F}_q$-linear combinations of $(z_1, \ldots, z_K)$ using linearized property of $f(\cdot)$ (Remark 6). This allows one to recover q^{K-1}-degree polynomial $f(\cdot)$, and therefore to reconstruct the message vector, by performing polynomial interpolation.*

An MRD code $\mathcal{C}^{Gab}$ with minimum distance D can correct any $D - 1 = N - K$ erasures, which we will refer as *rank erasures*. An algorithm for erasure correction of Gabidulin codes can be found e.g. in [24].

E. Regenerating Codes

In their seminal work [1], Dimakis et al. consider the setting, where a newcomer, a node that replaces a failed node, contacts d nodes, $k \leq d \leq n-1$, during node repair and downloads $\beta \leq \alpha$ symbols from each of these d nodes. Dimakis et al. model operation of a DSS using a multicasting problem over *information flow graph*. Using this approach, [1] characterizes the information theoretic trade off between repair bandwidth ($\gamma = d\beta$) and per node storage (α) for DSS satisfying the *maximum distance separable* (MDS) or "any k out of n" property:

$$\mathcal{M} \leq \sum_{i=1}^{k} \min\{(d - i + 1)\beta, \alpha\}. \tag{4}$$

The codes that achieve this trade off are termed *regenerating codes*. Two classes of codes that correspond two extreme points of this trade off are known as *minimum storage regenerating (MSR)* codes and *minimum bandwidth regenerating (MBR)* codes, corresponding to minimum storage per node (i.e., $\alpha = \mathcal{M}/k$) and minimum possible repair bandwidth ($\gamma = \alpha$) respectively. The former is obtained by first choosing a minimum storage per node (i.e., $\alpha = \mathcal{M}/k$), and then minimizing repair bandwidth satisfying (4), whereas the latter is obtained by first finding the minimum possible γ and then finding the minimum α in (4). For MSR codes, we have $(\alpha_{\text{msr}}, \beta_{\text{msr}}) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}}{k(d-k+1)}\right)$. On the other hand, MBR codes are characterized by $(\alpha_{\text{mbr}}, \beta_{\text{mbr}}) = \left(\frac{2\mathcal{M}d}{k(2d-k+1)}, \frac{2\mathcal{M}}{k(2d-k+1)}\right)$. Regenerating codes that allow *exact node repair*, where the data on the regenerated node is the same as that stored on the failed node, are of particular interest. In this paper, we term MSR (MBR) codes with the ability to perform exact repair as exact-MSR (MBR) codes.

III. Optimal Locally Repairable Codes

In this section, we first derive an upper bound on the minimum distance of (r, δ, α) LRCs. Next, we propose a general code construction which attains the derived bound on $d_{\min}$. Our approach is to apply a two-stage encoding, where we use Gabidulin codes (a rank-metric analog of Reed-Solomon codes) along with MDS array codes. This construction can be viewed as a generalization of the construction proposed in [28].

A. Upper Bound on $d_{\min}$ for an (r, δ, α) LRC

We state a generic upper bound on the minimum distance $d_{\min}$ of an (r, δ, α) LRC C of length n and dimension $\mathcal{M}$. The bound generalizes the $d_{\min}$ bound given in [10] for LRC with single local parity ($\delta = 2$) to LRC with multiple local parities ($\delta \geq 2$).

Theorem 8. *For an (r, δ, α) LRC C over $\mathbb{F}$ of length n and dimension $\mathcal{M}$, we have*

$$d_{\min}(C) \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil + 1 - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1). \tag{5}$$

Proof: We follow the proof technique of [9], [10]. In particular, the proof involves construction of a set of nodes $\mathcal{A}$ for a locally repairable DSS such that total entropy of the symbols stored in $\mathcal{A}$ is less than $\mathcal{M}$ and

$$|\mathcal{A}| \geq \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - 1 + \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1). \tag{6}$$

Theorem 8 then follows from Definition 1 and (6). Refer [29] for detailed proof. ∎

Remarkably, the above theorem establishes a trade off between node failure resilience ($d_{\min}$) and per node storage (α), where α can be increased to obtain higher $d_{\min}$. This is of particular interest to design codes having both locality and high resilience to node failures.

Remark 9. *For the special case of $\delta = 2$, this bound matches with the bound (3) presented in [10]. For the case of $\alpha = 1$, the bound reduces to $d_{\min} \leq n - \mathcal{M} + 1 + (\lceil \mathcal{M}/r \rceil - 1)(\delta - 1)$, which is coincident with the bound (2) presented in [11].*

B. Construction of $d_{\min}$-Optimal Vector LRCs

In this subsection we present a construction of an (r, δ, α) LRC with length n and dimension $\mathcal{M}$, which attains the bound given in Theorem 8.

Construction I. Consider a file $\mathbf{f}$ over $\mathbb{F} = \mathbb{F}_{q^m}$ of size $\mathcal{M} \geq r\alpha$, where m will be defined in the sequel. We encode the file in two steps before storing it on DSS. First, the file is encoded using a Gabidulin code. The codeword of the Gabidulin code is then partitioned into local groups and each local group is then encoded using an MDS array code over $\mathbb{F}_q$.

In particular, let $\mathcal{M}, n, r, \delta, \alpha$ be the positive integers such that $r + \delta - 1 < n$ and $\mathcal{M} \geq r\alpha$. We denote by $g = \left\lceil \frac{n}{r+\delta-1} \right\rceil$ the number of local groups in the system. We consider the following two cases:

1) $(r + \delta - 1) | n$**:** Let $N = \frac{nr\alpha}{r+\delta-1}$, $m \geq N$, and let $\mathcal{C}^{\mathrm{Gab}}$ be an $[N, \mathcal{M}, D = N - \mathcal{M} + 1]_{q^m}$ Gabidulin code. First, we encode $\mathcal{M}$ to a codeword $\mathbf{c} \in \mathcal{C}^{\mathrm{Gab}}$ and partition $\mathbf{c}$ into $g = \frac{N}{r\alpha}$ disjoint groups, each of size $r\alpha$, and each group is stored on a different set of r nodes, α symbols per node. In other words, the output of the first encoding step generates the encoded data stored on rg nodes, each one containing α symbols of a (folded) Gabidulin codeword. Second, we generate $\delta - 1$ parity nodes per group by applying an $[(r + \delta - 1), r, \delta, \alpha]_q$ MDS array code on each local group of r nodes, treating these r nodes as input data blocks (of length α) for the MDS array code. At the end of the second round of encoding, we have $n = g(r + \delta - 1) = \frac{N}{\alpha} + \frac{N}{r\alpha}(\delta - 1)$ nodes, each storing α symbols over $\mathbb{F}_{q^m}$, partitioned into g local groups, each of size $r - \delta + 1$.

2) $n (\mathrm{mod}\ r + \delta - 1) - (\delta - 1) > 0$**:** Let β_0, $1 \leq \beta_0 \leq r - 1$ be an integer, such that $n = \lfloor \frac{n}{r+\delta-1} \rfloor (r+\delta-1) + \beta_0 + \delta - 1 = (g-1)(r + \delta - 1) + \beta_0 + \delta - 1$. Let $N = (g-1)r\alpha + \beta_0\alpha$, $m \geq N$, and let $\mathcal{C}^{\mathrm{Gab}}$ be an $[N, \mathcal{M}, D = N - \mathcal{M} + 1]_{q^m}$ Gabidulin code. First, we encode $\mathcal{M}$ to a codeword $\mathbf{c} \in \mathcal{C}^{\mathrm{Gab}}$ and partition $\mathbf{c}$ into $g - 1$ disjoint groups of size $r\alpha$ and one additional group of size $\beta_0\alpha$, the first $g-1$ groups are stored on $(g-1)r$ nodes, and the last group is stored on β_0 nodes, each one containing α symbols of a (folded) Gabidulin codeword. Second, we generate $\delta - 1$ parity nodes per group by applying an $[(r + \delta - 1), r, \delta, \alpha]_q$ MDS array code on each of the first $g - 1$ local groups of r nodes, and by applying a $[(\beta_0 + \delta - 1), \beta_0, \delta, \alpha]_q$ MDS array code on the last local group. At the end of the second round of encoding, we have $n = (g - 1)(r + \delta - 1) + (\beta_0 + \delta - 1) = \frac{N}{\alpha} + \left\lceil \frac{N}{r\alpha} \right\rceil (\delta - 1)$ nodes, each storing α symbols over $\mathbb{F}_{q^m}$, partitioned into g local groups, $g - 1$ of which of size $r - \delta + 1$ and one group of size $\beta_0 + \delta - 1$.

We denote the obtained code by C^{loc}.

Remark 10. *Note, that since an MDS array code from Construction I is defined over $\mathbb{F}_q$, any symbol of any node of C^{loc} can be written as $\sum_{j=1}^{r\alpha} a_j c_{i_j} = \sum_{j=1}^{r\alpha} a_j f(g_{i_j}) = f(\sum_{j=1}^{r\alpha} a_j g_{i_j})$, where $a_j \in \mathbb{F}_q$, $c_{i_j} \in \mathbb{F}_{q^m}$ are $r\alpha$ symbols of the same group of* $\mathbf{c}$*, and g_{i_j}, are linearly independent over $\mathbb{F}_q$ evaluation points. Hence,* any *$s \leq r\alpha$ symbols inside a group of C^{loc} are evaluations of $f(x)$ in s linearly independent over $\mathbb{F}_q$ points. (If there is a group with $\beta_0 < r$ elements we have the same result substituting r with β_0). Thus* any *$\delta - 1 + i$* node erasures *in a group correspond to $i\alpha$* rank erasures*. Moreover, if we take any $r\alpha$ symbols of C^{loc} from every group (and $\alpha\beta_0$ symbols from the smallest group, if it exists), we obtain a Gabidulin codeword, for a corresponding choice of evaluation points for a Gabidulin code, which encodes the given data $\mathcal{M}$.*

Next, we provide the conditions for parameters of the code C^{loc} obtained from Construction I to be a $d_{\min}$ optimal (r, δ, α) LRC.

Theorem 11. *Let C^{loc} be an (r, δ, α) LRC obtained by Construction I. Then,*

- *If $(r+\delta-1) | n$, then C^{loc} over $\mathbb{F} = \mathbb{F}_{q^m}$, for $m \geq \frac{nr\alpha}{r+\delta-1}$ and $q \geq (r + \delta - 1)$, attains the bound (5).*
- *If $n (\mathrm{mod}\ r + \delta - 1) - (\delta - 1) \geq \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil (\mathrm{mod}\ r) > 0$, then C^{loc} over $\mathbb{F} = \mathbb{F}_{q^m}$, for $m \geq \alpha \left(n - (\delta - 1) \left(\left\lfloor \frac{n}{r+\delta-1} \right\rfloor + 1 \right) \right)$ and $q \geq (r + \delta - 1)$, attains the bound (5).*

Proof: The proof is based on Remark 10 and the observation that any $n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right) (\delta - 1)$ node erasures correspond to at most $D - 1$ rank erasures which can be corrected by the Gabidulin code $\mathcal{C}^{\mathrm{Gab}}$. See the details in Appendix A. ∎

Specializing the Construction I to scalar case ($\alpha = 1$), we obtain explicit (r, δ) scalar LRCs for the settings of parameters, where only results on existence of scalar LRCs are present in literature. For scalar case, Construction I employs MDS codes instead of MDS array codes after encoding information symbols ($\mathbf{f}$) with Gabidulin codes. The following corollary (of Theorem 11) summarizes our contribution towards scalar LRCs:

Corollary 12. *Let C^{loc} be a (r, δ) scalar LRC obtained by Construction I. Then,*

- *If $(r+\delta-1) | n$, then C^{loc} over $\mathbb{F} = \mathbb{F}_{q^m}$, for $m \geq \frac{nr}{r+\delta-1}$ and $q \geq (r + \delta - 1)$, attains the bound (2).*
- *If $n (\mathrm{mod}\ r+\delta-1) - (\delta-1) \geq \mathcal{M} (\mathrm{mod}\ r) > 0$, then C^{loc} over $\mathbb{F} = \mathbb{F}_{q^m}$, for $m \geq \left(n - (\delta - 1) \left(\left\lfloor \frac{n}{r+\delta-1} \right\rfloor + 1 \right) \right)$ and $q \geq (r + \delta - 1)$, attains the bound (2).*

Remark 13. *The required field size $|\mathbb{F}| = q^m$ for the proposed construction should satisfy $m \geq N$, for any choice of $q \geq (r + \delta - 1)$. So we can assume that $|\mathbb{F}| = q^N$, for N given in Theorem 11. Note that we can reduce the field size to $|\mathbb{F}| = q^{N/\alpha}$ by stacking [19] of α independent optimal scalar LRCs, obtained from Construction I.*

We illustrate the construction of C^{loc} in the following examples. First we consider the scalar case.

Example 14. *Consider the following system parameters:*

$$(\mathcal{M}, n, r, \delta, \alpha) = (9, 14, 4, 2, 1).$$

Since $n = \left\lfloor \frac{14}{4+2-1} \right\rfloor \cdot (4 + 2 - 1) + (3 + 2 - 1)$*, let*

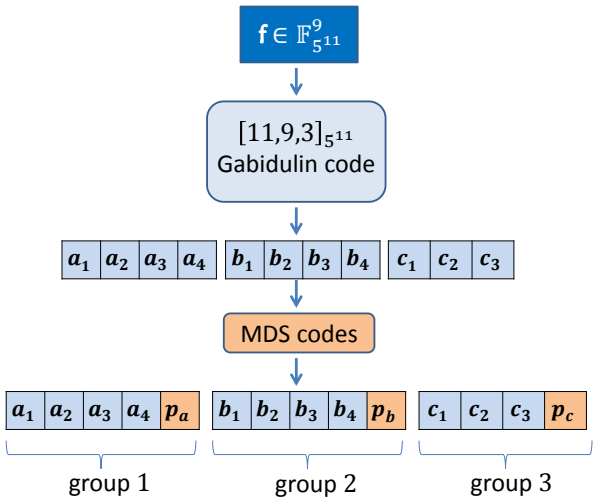

Fig. 1: Illustration of the construction of a scalar $(r = 4, \delta = 2, \alpha = 1)$ LRC for $n = 14, \mathcal{M} = 9$ and $d_{\min} = 4$.

$N = \left\lfloor \frac{14}{4+2-1} \right\rfloor \cdot 4 + 3 = 11$. *First* $\mathcal{M} = 9$ *symbols over* $\mathbb{F} = \mathbb{F}_{5^{11}}$ *are encoded into a codeword* $\mathbf{c}$ *of a* $[11, 9, 3]_{5^{11}}$ *Gabidulin code* $\mathcal{C}^{\text{Gab}}$. *This codeword is partitioned into three groups, two of size* 4 *and one of size* 3, *as follows:* $\mathbf{c} = (a_1, a_2, a_3, a_4 | b_1, b_2, b_3, b_4 | c_1, c_2, c_3)$. *Then, by applying a* $[5, 4, 2]$ *MDS code in the first two groups and a* $[4, 3, 2]$ *MDS code in the last group we add one parity to each group. The symbols of* $\mathbf{c}$ *with three new parities* p_a, p_b, p_c *are stored on 14 nodes as shown in Fig 1. By Theorem 8, the minimum distance* $d_{\min}$ *of this code is at most* 4. *By Remark 10, any* 3 *node erasures correspond to at most* 2 *rank erasures and then can be corrected by* $\mathcal{C}^{\text{Gab}}$, *hence* $d_{\min} = 4$. *In addition, when a single node fails, it can be repaired by using the data stored on all the other nodes from the same group.*

Next, we illustrate Construction I for a vector LRC.

Example 15. *We consider a DSS with the following parameters:*

$$(\mathcal{M}, n, r, \delta, \alpha) = (28, 15, 3, 3, 4).$$

By (5) we have $d_{\min} \leq 5$. *Let* $N = \frac{15 \cdot 3 \cdot 4}{3+3-1} = 36$ *and* $(a_1, \ldots, a_{12}, b_1, \ldots, b_{12}, c_1, \ldots, c_{12})$ *be a codeword of a* $[36, 28, 9]_{q^{36}}$ *code* $\mathcal{C}^{Gab}$, *which is obtained by encoding* $\mathcal{M} = 28$ *symbols over* $\mathbb{F} = \mathbb{F}_{q^{36}}$ *of the original file. The Gabidulin codeword is then partitioned into three groups* $(a_1, \ldots, a_{12})$, $(b_1, \ldots, b_{12})$, *and* $(c_1, \ldots, c_{12})$. *Encoded symbols in each group are stored on three storage nodes as shown in Fig. 2. In the second stage of encoding, a* $[5, 3, 3, 4]_q$ *MDS array code over* $\mathbb{F}_q$ *is applied on each local group to obtain* $\delta - 1 = 2$ *parity nodes per local group. The coding scheme is illustrated in Fig. 2.*

By Remark 10, any 4 *node failures correspond to at most* 8 *rank erasures in the corresponding codeword of* $\mathcal{C}^{Gab}$. *Since the minimum rank distance of* $\mathcal{C}^{Gab}$ *is* 9, *these node erasures can be corrected by* $\mathcal{C}^{Gab}$, *and thus the minimum distance of* C^{loc} *is exactly* 5.

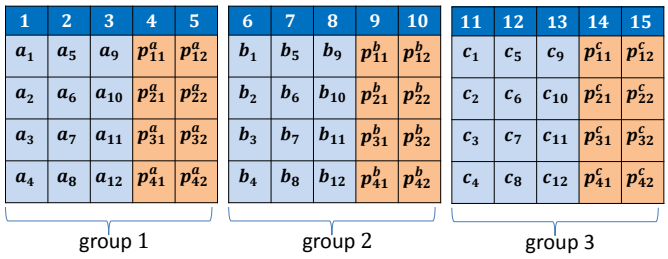

Fig. 2: Example of an $(r = 3, \delta = 3, \alpha = 4)$ LRC with $n = 15$ and $d_{\min} = 5$.

Remark 16. *The efficiency of the decoding of the codes obtained by Construction I depends on the efficiency of the decoding of the MDS codes and the Gabidulin codes.*

IV. Repair Bandwidth Efficient Locally Repairable Codes

In this section, we present the hybrid codes, which allow for local repairs while minimizing repair bandwidth for given locality parameters. We combine MRD codes with regenerating codes to obtain these codes.

As pointed out in Section II-C, LRCs allow for naïve repair process, where a newcomer can repair a failed node by contacting r nodes in its local group and downloading all symbols stored on these r nodes. Following the line of work of bandwidth efficient repair in DSS due to [1], we allow a newcomer to contact $d \geq r$ nodes in its local group and to download only $\beta \leq \alpha$ symbols from each of these d nodes in order to repair the failed node. The motivation behind this is to lower the repair bandwidth of an LRC. The main idea here is to apply a regenerating code in each local group. (We note that, in a parallel and independent work, Kamath et al. [19] also proposed utilizing regenerating codes to perform efficient local repairs.).

First, we provide an upper bound on the amount of data that can be stored in a locally repairable DSS while supporting a given repair bandwidth ($d\beta$) and the maximum possible failure resilience (i.e., maximum minimum distance). Next, we present $d_{\min}$ - optimal codes, which attain this bound by applying an MSR code in each local group instead of an MDS array code in the second step of Construction I.

A. File Size Upper Bound for Repair Bandwidth Efficient LRCs

In the rest of this section, we restrict ourselves to LRCs that have the maximum possible minimum distance as described in (5). We also assume for simplicity that $(r + \delta - 1)|n$. We remark that, for $(r + \delta - 1)|n$, the upper bound on minimum distance for LRCs given in (5) is achievable only if the code have disjoint local groups [29]. Accordingly, we provide a file size bound for the disjoint case. (See Fig. 3.)

Any failed node in a particular local group is repaired by contacting d remaining nodes within that group, where $r \leq d \leq r + \delta - 2$. During the node repair process a newcomer

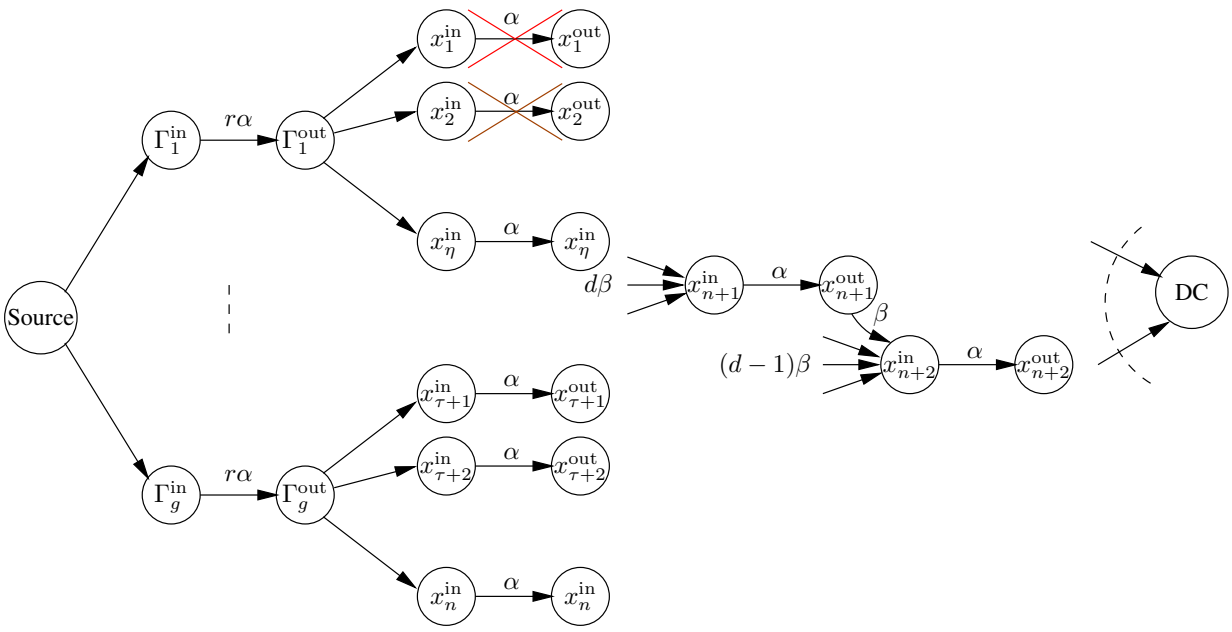

Fig. 3: Flow graph for (r, δ) LRC. In this graph, node pairs $\{\Gamma_i^{\text{in}}, \Gamma_i^{\text{out}}\}_{i=1}^{g}$ with edge of capacity $r\alpha$ enforce the requirement that each local group has at most $r\alpha$ entropy. Here η and τ denote $r+\delta-1$ and $n-(r+\delta-1)$ respectively. The figure illustrates the case where x_1 and x_2 sequentially fail and are replaced by introducing x_{n+1} and x_{n+2} respectively. The data collector (DC) is assumed to contact a set of $n-d_{\min}+1$ nodes for reconstruction of original data.

downloads β symbols from each of these d nodes. In what follows, we denote such LRC by the tuple $(r, \delta, \alpha, d, \beta)$.

Next, we perform the standard min-cut max-flow based analysis for locally repairable DSS by mapping it to a multi-casting problem on a dynamic information flow graph. (The information flow graph representing a locally repairable DSS is a modification of the information flow graph for classical DSS analyzed in [1] and is first introduced in [10] for naïve repair, where the newcomer contacts r nodes.) Each data collector contacts $n-d_{\min}+1$ storage nodes for data reconstruction. Consider $n-d_{\min}+1 = g''(r+\delta-1)+h$, for some $h < r+\delta-1$. Here, to minimize the cut over the flow graph (see Fig. 3), we consider that the DC connects to all the nodes in g'' number of groups, and connects to an additional group with h nodes. Within each group contacted by the DC, we consider a repair scenario similar to [1] in order to obtain a lower cut value. Thus, from (4), we obtain the following file size upper bound for LRCs.

Theorem 17. *For an DSS employing an $(r, \delta, \alpha, d, \beta)$ LRC, such that from any set of $n-d_{\min}+1$ nodes the original file can be recovered, we have*

$$\mathcal{M} \leq \min\left\{r\alpha, \sum_{i=0}^{h-1} \min\{\max\{(d-i)\beta, 0\}, \alpha\}\right\} + \sum_{j=1}^{\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor} \min\left\{r\alpha, \sum_{i=0}^{r+\delta-2} \min\{\max\{(d-i)\beta, 0\}, \alpha\}\right\}, \tag{7}$$

where $h = n-d_{\min}+1-(r+\delta-1)\left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor$.

Note that according to Definition 2 of an (r, δ, α) LRC, any set of r nodes has at most $r\alpha$ independent symbols. Now we assume that any set of r nodes has exactly $r\alpha$ independent symbols. Note that the construction of Section III applying MDS array codes in each local group has this property. However, to have a local repair bandwidth efficient code, we apply an $(r+\delta-1, r, d, \alpha, \beta)$ MSR code with a file of size $r\alpha$ in each local group. Such a code will be called MSR-LRC. Similar to the analysis given in [1] for the classical setup, the parameters of MSR-LRC need to satisfy

$$r\alpha = \sum_{i=0}^{r-1} \min\{(d-i)\beta, \alpha\}, \tag{8}$$

and then $(d-i)\beta \geq \alpha$ for each $i = 0, \cdots, r-1$. Thus, minimum β is obtained as $\beta^* = \frac{\alpha}{d-r+1}$ and the bound in (7) reduces to

$$\mathcal{M} \leq \left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor r\alpha + \min\{h, r\}\alpha \tag{9}$$

where h is as defined in Theorem 17. This establishes the file size upper bound for bandwidth efficient $d_{\min}$-optimal LRCs applying MSR codes in each local group.

B. Optimal MSR-LRC

In the following we prove that the code presented in Section III-B, when an MSR code is employed for the second

encoding stage, achieves the bound (9) if $\alpha|\mathcal{M}$. We establish this claim in the following theorem.

Theorem 18. *Let C^{loc} be a code obtained from Construction I described in Sec. III-B with an MSR code employed in the second encoding stage in Construction I to generate local parities. If $\alpha|\mathcal{M}$, then C^{loc} attains the bound (9).*

Proof: Lets assume that $\alpha|\mathcal{M}$. Then, we can write $\mathcal{M} = \alpha(\alpha_1 r + \beta_1)$, for some integers $0 \leq \alpha_1, \beta_1$, s.t. $\beta_1 \leq r-1$. Then, by (5),

- If $\beta_1 > 0$ then $n - d_{\min} + 1 = (\alpha_1 r + \beta_1) + \alpha_1(\delta - 1) = (r + \delta - 1)\alpha_1 + \beta_1$, hence $h = \beta_1 < r$ and $\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor r\alpha + \min\{h, r\}\alpha = \alpha_1 r\alpha + \beta_1\alpha = \mathcal{M}$.
- If $\beta_1 = 0$ then $n - d_{\min} + 1 = \alpha_1 r + (\alpha_1 - 1)(\delta - 1) = (r+\delta-1)(\alpha_1-1)+r$, hence $h = r$ and $\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor r\alpha + \min\{h, r\}\alpha = (\alpha_1 - 1)r\alpha + r\alpha = \mathcal{M}$.

This establishes that C^{loc}, when an MSR code is used to generate its local parities, attains the bound given in (9). ■

Example 19. *Consider the parameters given in Example 15. Now we apply an $(r + \delta - 1 = 5, r = 3, d = 4, \alpha = 4, \beta = 2)$ exact-MSR code (e.g., (5,3)-zigzag code [6]) in each group instead of an MDS array code. For these parameters, $h = 1$, and by (9), $\mathcal{M} \leq 2 \cdot 3 \cdot 4 + 1 \cdot 4 = 28$, thus the code attains the bound (9). Moreover, each failed node can be repaired bandwidth efficiently as an exact-MSR code is used within each local group.*

V. Conclusion

This paper studies the problem of designing LRCs for distributed storage systems. We characterized the resilience (minimum distance) vs. per node storage trade-off for such codes. We then presented a novel construction for vector LRCs that are optimal in the sense they achieve this trade off for a wide range of parameters. This construction is based on MRD codes. As a special case of these vector LRCs, this construction gives optimal scalar LRCs in the range of parameters, where explicit scalar LRCs were previously unknown.

We then introduced the notion of minimizing repair bandwidth for locally repairable DSS and provided a bound on file size, the amount of data that can be stored on DSS, for a given repair bandwidth. We specialized this bound to MSR-LRC, where restriction of LRC to a local group is an MSR code. We further showed that the MRD based construction for LRCs, presented in this paper, can also be used to design codes that are file size optimal.

A similar construction for vector LRC also allows to design optimal MBR-LRC [30]. On the similar note, following the mechanism given in [19], scalar LRCs presented in this paper can also be used to design novel MBR-LRC.

The MRD (in particular, Gabidulin) precoding utilized in this paper also has applications in the security context. In particular, the issue of designing secure locally repair DSS against passive eavesdropping attack is studied in [29], where classical secret sharing scheme [31] is combined with vector locally repairable codes presented in this paper in order to characterize secrecy capacity of locally repairable DSS. In addition, utilizing MRD precoding, security in DSS with cooperative repairs (where multiple failures repaired simulataneously) against passive eavesdropper is studied in [32], and codes for security against active eavesdroppers are proposed in [28].

References

[1] A. G. Dimakis, P. Godfrey, M. Wainwright and K. Ramachandran, "Network coding for distributed storage system," *IEEE Trans. on Inform. Theory*, vol. 56, no. 9, pp. 4539-4551, Sep. 2010.

[2] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Proc. of IEEE ISIT*, Jul. 2009.

[3] N. B. Shah, K. V. Rashmi, P. V. Kumar and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *Proc. of IEEE ITW*, Jan. 2010.

[4] C. Suh and K. Ramchandran, "Exact-repair MDS codes for distributed storage using interference alignment," in *Proc. of IEEE ISIT*, Jul. 2010.

[5] K. V. Rashmi, N. B. Shah and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR point via a product-matrix construction," *IEEE Trans. on Inform. Theory*, vol. 57, no. 57, pp. 5227-5239, Aug. 2011.

[6] I. Tamo, Z. Wang, and J. Bruck, "Zigzag Codes: MDS Array Codes With Optimal Rebuilding," to appear in *IEEE Trans. Inform. Theory*.

[7] A. Datta and F. Oggier, "An overview of codes tailor-made for networked distributed data storage," *CoRR*, vol. abs/1109.2317, Sep. 2011.

[8] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A Survey on Network Codes for Distributed Storage," Proceedings of the IEEE, Mar. 2011.

[9] P. Gopalan, C. Huang, H. Simitchi and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. on Inform. Theory*, vol. 58, no. 11, pp. 6925-6934, Nov. 2012.

[10] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. of IEEE ISIT*, Jul. 2012.

[11] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, "Optimal linear codes with a local-error-correction property," in *Proc. of IEEE ISIT*, Jul. 2012.

[12] F. E. Oggier and A. Datta, "Homomorphic self-repairing codes for agile maintenance of distributed storage systems," *CoRR*, vol. abs/1107.3129, Jul. 2011.

[13] F. E. Oggier and A. Datta, "Self-repairing codes for distributed storage - A projective geometric construction," *CoRR*, vol. abs/1105.0379, May 2011.

[14] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin,"Erasure coding in windows azure storage," *in Proc. USENIX Annual Technical Conference (ATC)*, Apr. 2012.

[15] A. S. Rawat and S. Vishwanath, "On locality in distributed storage systems," in *Proc. of IEEE ITW*, Sept. 2012.

[16] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data," *CoRR*, vol. abs/1301.3791, Jan. 2013.

[17] C. Huang, M. Chen, and J. Li, "Pyramid code: flexible schemes to trade space for access efficiency in reliable data storage systems," in *Proc. of 6th IEEE NCA*, Mar. 2007.

[18] J. Han and L.A. Lastras-Montano, "Reliable memories with subline accesses," in *Proc. of IEEE ISIT 2007*, Jun. 2007.

[19] G M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration," *CoRR*, vol. abs/1211.1932, Nov. 2012.

[20] H. D. L. Hollmann, "Storage codes – coding rate and repair locality," *CoRR*, vol. abs/1301.4300, Jan. 2013.

[21] E M. Gabidulin, "Theory of codes with maximum rank distance," *Problems of Information Transmission*, vol. 21, pp. 1-12, Jul. 1985.

[22] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. on Inform. Theory*, vol. 37, pp. 328-336, Mar. 1991.

[23] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, North-Holland, 1978.

[24] E. M. Gabidulin and N. I. Pilipchuk, "Error and erasure correcting algorithms for rank codes," *Designs, codes and Cryptography*, vol. 49, pp. 105–122, 2008.
[25] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 192–202, Feb. 1995.
[26] M. Blaum and R. M. Roth, "On lowest density MDS codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 46–59, 1999.
[27] Y. Cassuto and J. Bruck,"Cyclic low-density MDS array codes," in *Proc. of IEEE ISIT*, Jul. 2006.
[28] N. Silberstein, A. S. Rawat and S. Vishwanath, "Error resilience in distributed storage via rank-metric codes," in *Proc. of 50th Allerton*, available in *http://arxiv.org/abs/1202.0800*, Oct. 2012.
[29] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *CoRR*, vol. abs/1302.0744, Oct. 2012.
[30] G M. Kamath, N. Prakash, V. Lalitha, P. V. Kumar, N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Explicit MBR All-Symbol Locality Codes," *CoRR*, vol. abs/1210.6954, Feb. 2013.
[31] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22 no. 11, pp.612–613, Nov. 1979.
[32] O. O. Koyluoglu, A. S. Rawat, and S. Vishwanath, "Secure cooperative regenerating codes for distributed storage systems," *CoRR*, vol. abs/1210.3664, Oct. 2012.

APPENDIX A
PROOF OF THEOREM 11

To prove that $C^{\rm loc}$ attains the bound (5) we need to show that any $E \triangleq n-\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil-\left(\left\lceil\frac{\mathcal{M}}{r\alpha}\right\rceil-1\right)(\delta-1)$ node erasures can be corrected by $C^{\rm loc}$. In order to show this we prove that any E erasures of $C^{\rm loc}$ correspond to at most $D-1$ rank erasures of the underlying $[N,\mathcal{M},D]_{q^m}$ Gabidulin code $\mathcal{C}^{\rm Gab}$, and hence can be corrected by $\mathcal{C}^{\rm Gab}$. Here, we point out the the worst case erasure pattern is when the erasures appear in the smallest possible number of groups and the number of erasures inside a local group is maximal.

First, given $n=\frac{N}{\alpha}+\left\lceil\frac{N}{r\alpha}\right\rceil(\delta-1)$, we can rewrite E in the following way:

$$E=\frac{N}{\alpha}-\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil+\left(\left\lceil\frac{N}{r\alpha}\right\rceil-\left\lceil\frac{\mathcal{M}}{r\alpha}\right\rceil+1\right)(\delta-1). \quad (10)$$

Let $\alpha_1,\beta_1,\gamma_1$ be the integers such that $\mathcal{M}=\alpha(\alpha_1 r+\beta_1)+\gamma_1$, where $1\le\alpha_1\le g$, for $g=\left\lceil\frac{n}{r+\delta-1}\right\rceil$, $0\le\beta_1\le r-1$, and $0\le\gamma_1\le\alpha-1$. Then

1) If $(r+\delta-1)|n$ then $N=gr\alpha$ and

$$D-1=N-\mathcal{M}=(g-\alpha_1)r\alpha-\beta_1\alpha-\gamma_1. \quad (11)$$

- If $\gamma_1=\beta_1=0$ then $\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil=\alpha_1 r$ and $\left\lceil\frac{\mathcal{M}}{r\alpha}\right\rceil=\alpha_1$. Then by (10), $E=(g-\alpha_1)(r+\delta-1)+(\delta-1)$. Hence, in the worst case we have $(g-\alpha_1)$ groups with all the erased nodes and one additional group with $\delta-1$ erased nodes, which by Remark 10 corresponds to $r\alpha$ rank erasures in $(g-\alpha_1)$ groups of the corresponding Gabidulin codeword. Since by (11), $D-1=(g-\alpha_1)r\alpha$, this erasures can be corrected by the Gabidulin code.
- If $\gamma_1=0$, $\beta_1>0$ then $\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil=\alpha_1 r+\beta_1$ and $\left\lceil\frac{\mathcal{M}}{r\alpha}\right\rceil=\alpha_1+1$. Then by (10) we have $E=(g-\alpha_1-1)(r+\delta-1)+(r+\delta-1-\beta_1)$. Hence, in the worst case we have $(g-\alpha_1-1)$ groups with all the erased nodes and one additional group with $r+\delta-1-\beta_1$ erased nodes, which by Remark 10 and by (11) corresponds to $(g-\alpha_1)r\alpha-\beta_1\alpha=D-1$ rank erasures that can be corrected by the Gabidulin code.
- If $\gamma_1>0$ then $\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil=\alpha_1 r+\beta_1+1$ and $\left\lceil\frac{\mathcal{M}}{r\alpha}\right\rceil=\alpha_1+1$. Then by (10) we have $E=(g-\alpha_1-1)(r+\delta-1)+(r+\delta-1-\beta_1-1)$. Hence, in the worst case we have $(g-\alpha_1-1)$ groups with all the erased nodes and one additional group with $r+\delta-1-\beta_1-1$ erased nodes, which by Remark 10 and by (11) corresponds to $(g-\alpha_1)r\alpha-\beta_1\alpha-\alpha<D-1$ rank erasures that can be corrected by the Gabidulin code.

2) If $n(\text{mod } r+\delta-1)-(\delta-1)\ge\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil(\text{mod } r)>0$, then since $n(\text{mod } r+\delta-1)-(\delta-1)\equiv\beta_0$ we have $\beta_0\ge\beta_1>0$, $N=(g-1)r\alpha+\beta_0\alpha$, $\frac{N}{\alpha}=(g-1)r+\beta_0$, $\left\lceil\frac{N}{r\alpha}\right\rceil=g$ and

$$D-1=(g-\alpha_1-1)r\alpha+(\beta_0-\beta_1)\alpha-\gamma_1. \quad (12)$$

- If $\gamma_1=0$ then $\left\lceil\frac{\mathcal{M}}{\alpha}\right\rceil=\alpha_1 r+\beta_1$ and $\left\lceil\frac{\mathcal{M}}{r\alpha}\right\rceil=\alpha_1+1$. Then by (10), we have $E=(g-\alpha_1-1)(r+\delta-1)+(\beta_0-\beta_1+\delta-1)$. Hence, in the worst case we have $(g-\alpha_1-1)$ groups with all the erased nodes and one additional group with $\beta_0-\beta_1+\delta-1$ erased nodes (or $\beta_0+\delta-1$ erased nodes in the smallest group, $(g-\alpha_1-2)$ groups with all the erased nodes and one group with $r+\delta-1-\beta_1$ erased nodes). This by Remark 10 and by (12) corresponds to $(g-\alpha_1-1)r\alpha+(\beta_0-\beta_1)\alpha=D-1$ rank erasures that can be corrected by the Gabidulin code.
- If $\gamma_1>0$ then $\left\lceil\frac{M}{\alpha}\right\rceil=\alpha_1 r+\beta_1+1$ and $\left\lceil\frac{M}{r\alpha}\right\rceil=\alpha_1+1$. Then by (10) we have $E=(g-\alpha_1-1)(r+\delta-1)+(\beta_0-\beta_1-1+\delta-1)$. Hence, in the worst case we have $(g-\alpha_1-1)$ groups with all the erased nodes and one additional group with $\beta_0-\beta_1-1+\delta-1$ erased nodes (or $\beta_0+\delta-1$ erased nodes in the smallest group, $(g-\alpha_1-2)$ groups with all the erased nodes and one group with $r-\beta_1-1+\delta-1$ erased nodes). This by Remark 10 and by (12) corresponds to $(g-\alpha_1-1)r\alpha+(\beta_0-\beta_1)\alpha-\alpha<D-1$ rank erasures that can be corrected by the Gabidulin code.