

On the Sub-Packetization Size and the Repair Bandwidth of Reed-Solomon Codes

Weiqli Li, *Student Member, IEEE*,
 Zhiying Wang, *Member, IEEE*
 Hamid Jafarkhani, *Fellow, IEEE*

Abstract—Reed-Solomon (RS) codes are widely used in distributed storage systems. In this paper, we study the repair bandwidth and sub-packetization size of RS codes. The repair bandwidth is defined as the amount of transmitted information from surviving nodes to a failed node. The RS code can be viewed as a polynomial over a finite field $GF(q^\ell)$ evaluated at a set of points, where ℓ is called the sub-packetization size. Smaller bandwidth reduces the network traffic in distributed storage, and smaller ℓ facilitates the implementation of RS codes with lower complexity. Recently, Guruswami and Wootters proposed a repair method for RS codes when the evaluation points are the entire finite field. While the sub-packetization size can be arbitrarily small, the repair bandwidth is higher than the minimum storage regenerating (MSR) bound. Tamo, Ye and Barg achieved the MSR bound but the sub-packetization size grows faster than the exponential function of the number of the evaluation points. In this work, we present code constructions and repair schemes that extend these results to accommodate different sizes of the evaluation points. In other words, we design schemes that provide points in between. These schemes provide a flexible tradeoff between the sub-packetization size and the repair bandwidth. In addition, we generalize our schemes to manage multiple failures.

I. INTRODUCTION

Erasure codes are ubiquitous in distributed storage systems because they can efficiently store data while protecting against failures. Reed-Solomon (RS) code is one of the most commonly used codes because it achieves the Singleton bound [2] and has efficient encoding and decoding methods, see, e.g., [3], [4]. Codes matching the Singleton bound are called maximum distance separable (MDS) codes, and they have the highest possible failure-correction capability for a given redundancy level. In distributed storage, every code word symbol corresponds to a storage node, and communication costs between storage nodes need to be considered when node failures are repaired. In this paper, we study the *repair bandwidth* of RS codes, defined as the

amount of transmission required to repair a single node erasure, or failure, from all the remaining nodes (called helper nodes).

For a given erasure code, when each node corresponds to a single finite field symbol over $\mathbb{F} = GF(q^\ell)$, we say the code is scalar; when each node is a vector of finite field symbols in $\mathbb{B} = GF(q)$ of length ℓ , it is called a vector code or an array code. In both cases, we say the *sub-packetization size* of the code is ℓ . Here q is a power of a prime number. Shanmugam et al. [5] considered the repair of scalar codes for the first time. Recently, Guruswami and Wootters [6] proposed a repair scheme for RS codes. The key idea of both papers is that: rather than directly using the helper nodes as symbols over \mathbb{F} to repair the failed node, one treats them as vectors over the subfield \mathbb{B} . Thus, a helper may transmit less than ℓ symbols over \mathbb{B} , resulting in a reduced bandwidth. For an RS code with length n and dimension k over the field \mathbb{F} , denoted by $RS(n, k)$, [6] achieves the repair bandwidth of $n - 1$ symbols over \mathbb{B} . Moreover, when $n = q^\ell$ (called the full-length RS code) and $n - k = q^{\ell-1}$, the scheme provides the optimal repair bandwidth. Dau and Milenkovic [7] improved the scheme such that the repair bandwidth is optimal for the full-length RS code and any $n - k = q^s, 1 \leq s \leq \log_q(n - k)$.

For the full-length RS code, the schemes in [6] and [7] are optimal for single erasure. However, the repair bandwidth of these schemes still has a big gap from the minimum storage regenerating (MSR) bound derived in [8]. In particular, for an arbitrary MDS code, the repair bandwidth b , measured in the number of symbols over $GF(q)$, is lower bounded by

$$b \geq \frac{\ell(n-1)}{n-k}. \quad (1)$$

An MDS code satisfying the above bound is called an MSR code. In fact, most known MSR codes are vector codes, see [9], [10], [11], [12], [13], [14], [15]. For the repair of RS codes, Ye and Barg proposed a scheme that asymptotically approaches the MSR bound as n grows [16] when the sub-packetization size is $\ell = (n-k)^n$. Tamo et al. [17] provided an RS code repair scheme achieving the MSR bound when the sub-packetization size is $\ell \approx n^n$.

The repair problem for RS codes can also be generalized to multiple erasures. In this case, the schemes in [18] and

Weiqli Li, Zhiying Wang and Hamid Jafarkhani are with the Center for Pervasive Communications and Computing, University of California at Irvine, Irvine, CA 92697 USA (e-mail: {weiqli4,zhiying.hamid}@uci.edu). Part of this work has been presented in the 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2017 [1]

[19] work for the full-length code, [20] and [21] work for centralized repair, and [22] proposed a scheme achieving the multiple-erasure MSR bound.

Motivation: A flexible tradeoff between the sub-packetization size and the repair bandwidth is an open problem: Only the full-length RS code with high repair bandwidth and the MSR-achieving RS code with large sub-packetization are established. Our paper aims to provide more points between the two extremes – the full-length code and the MSR code. One straightforward method is to apply the schemes of [6] and [7] to the case of $\ell > \log_q n$ with fixed (n, k) . However, the resulting normalized repair bandwidth $\frac{b}{\ell(n-1)}$ grows with ℓ , contradictory to our intuition that larger ℓ implies smaller normalized bandwidth.

The need for small repair bandwidth is motivated by reducing the network traffic in distributed storage [8], and the need for the small sub-packetization is due to the complexity in field arithmetic operations, discussed below. It is demonstrated that the time complexity of multiplications in larger fields are much higher than that of smaller fields [23]. Moreover, multiplication in Galois fields are usually done by pre-computed look-up tables and the growing field size has a significant impact on the space complexity of multiplication operations. Larger fields require huge memories for the look-up table. For example, in $GF(2^{16})$, 8 GB are required for the complete table, which is impractical in most current systems [24]. Some logarithm tables and sub-tables are used to alleviate the memory problems for large fields, while increasing the time complexity at the same time [24], [25], [26]. For example, in the Intel SIMD methods, multiplications over $GF(2^{16})$ need twice the amount of operations as over $GF(2^8)$, and multiplications over $GF(2^{32})$ need 4 times the amount of operations compared to $GF(2^8)$, which causes the multiplication speed to drop significantly when the field size grows [25].

To illustrate the impact of the sub-packetization size on the complexity, let us take encoding for example. To encode a single parity check node, we need to do k multiplications and k additions over $GF(q^\ell)$. For a given systematic $RS(n, k)$ code over $GF(q^\ell)$, we can encode $k\ell \log_2 q$ bits of information by multiplications of $(n-k)k\ell \log_2 q$ bits and additions of $(n-k)k\ell \log_2 q$ bits. So, when M bits are encoded into $RS(n, k)$ codes, we need $M/(k\ell \log_2 q)$ copies of the code and we need multiplications of $M(n-k)$ bits and additions of $M(n-k)$ bits in $GF(q^\ell)$ in total. Although the total amount of bits we need to multiply is independent of ℓ , the complexity over a larger field is higher in both time and space. For a simulation of the RS code speed using different field sizes on different platforms, we refer the readers to [27]. The results suggest that RS codes have faster implementation in both encoding and decoding for smaller fields.

Besides the complexity, the small sub-packetization level also has many advantages such as easy system implemen-

tation, great flexibility and bandwidth-efficient access to missing small files [28], [29], which makes it important in distributed storage applications.

As can be seen from the two extremes, a small sub-packetization level also means higher costs in repair bandwidth, and not many other codes are known besides the extremes. For vector codes, Guruswami, Rawat [28] and Li, Tang [30] provided small sub-packetization codes with small repair bandwidth, but only for single erasure. Kravetska et al. [31] also presented a tradeoff between the sub-packetization level and the repair bandwidth for the proposed HashTag codes implemented in Hadoop. For scalar codes, Chowdhury and Vardy [32] extended Ye and Barg's MSR scheme [16] to a smaller sub-packetization size, but it only works for certain redundancy r and single erasure.

Contributions: In this work, we first design three single-erasure RS repair schemes, using the cosets of the multiplicative group of the finite field \mathbb{F} . Note that the RS code can be viewed as n evaluations of a polynomial over \mathbb{F} . The evaluation points of the three schemes are part of one coset, of two cosets, and of multiple cosets, respectively, so that the evaluation point size can vary from a very small number to the whole field size. In the schemes designed in this paper, we have a parameter a that can be tuned, and provides a tradeoff between the sub-packetization size and the repair bandwidth.

- For an $RS(n, k)$ code, our first scheme achieves the repair bandwidth $\frac{\ell}{a}(n-1)(a-s)$ for some a, s such that $n < q^a, r \triangleq n-k > q^s$ and a divides ℓ . Specifically, for the $RS(14, 10)$ code used in Facebook [33], we achieve repair bandwidth of 52 bits with $\ell = 8$, which is 35% better than the naive repair scheme.
- Our second scheme reaches the repair bandwidth of $(n-1)\frac{\ell+a}{2}$ for some a such that $n \leq 2(q^a-1)$, a divides ℓ and $\frac{\ell}{r} < a$.
- The first realization of our third scheme attains the repair bandwidth of $\frac{\ell}{r}(n+1+(r-1)(q^a-2))$ when $n \leq (q^a-1)\log_r \frac{\ell}{a}$. Another realization of the third scheme attains the repair bandwidth of $\frac{\ell}{r}(n-1+(r-1)(q^a-2))$ where $\ell \approx a(\frac{n}{q^a-1})^{(\frac{r-1}{q^a-1})}$. The second realization can also be generalized to any d helpers, for $k \leq d \leq n-1$.

We provide characterizations of linear multiple-erasure repair schemes, and propose two schemes for multiple erasures, where the evaluation points are in one coset and in multiple cosets, respectively. Again, the parameter a is tunable.

- We prove that any linear repair scheme for multiple erasures in a scalar MDS code is equivalent to finding a set of dual codewords satisfying certain rank constraints.
- For an $RS(n, k)$ code with $e < \frac{1}{a-s}\sqrt{\log_q n}$ erasures, our first scheme achieves the repair bandwidth $\frac{e\ell}{a}(n-e)(a-s)$ for some a, s such that $n < q^a, r = n-k > q^s$ and a divides ℓ .

- For an $RS(n, k)$ code, our second scheme works for $e \leq n - k$ erasures and $n - e$ helpers. The repair bandwidth depends on the location of the erasures and in most cases, we achieve $\frac{e\ell}{d-k+e}(n - e + (n - k + e)(q^a - 2))$ where $\ell \approx a \binom{n}{q^a - 1} \binom{n}{q^a - 1}$ and a divides ℓ .
- We demonstrate that repairing multiple erasures simultaneously is advantageous compared to repairing single erasures separately.

The comparison of our schemes, as well as the comparison to previous works, are shown in Tables I and II, and are discussed in more details in Sections III-D and IV-D.

The paper is organized as follows. In Section II, we briefly review the linear repair of RS codes and provide the preliminaries used in this paper. In Section III, we present three RS repair schemes for single erasure. Then, we discuss the repair schemes for multiple erasures in Section IV. In Section V, we provide the conclusion.

Notation: Throughout this paper, for positive integer i , we use $[i]$ to denote the set $\{1, 2, \dots, i\}$. For integers a, b , we use $a \mid b$ to denote that a divides b . For real numbers a_n, b_n , which are functions of n , we use $a \approx b$ to denote $\lim_{n \rightarrow \infty} \frac{a_n}{b_n} = 1$. For sets $A \subseteq B$, we use B/A to denote the difference of A from B . For a finite field \mathbb{F} , we denote by $\mathbb{F}^* = \mathbb{F}/\{0\}$ the corresponding multiplicative group. We write $\mathbb{E} \leq \mathbb{F}$ for \mathbb{E} being a subfield of \mathbb{F} . For element $\beta \in \mathbb{F}$ and E as a subset of \mathbb{F} , we denote $\beta E = \{\beta s, \forall s \in E\}$. A^T denotes the transpose of the matrix A .

II. PRELIMINARIES

In this section, we review the linear repair scheme of RS code in [6], and provide a basic lemma used in our proposed schemes.

The *Reed-Solomon code* $RS(A, k)$ over $\mathbb{F} = GF(q^\ell)$ of dimension k with n evaluation points $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\} \subseteq \mathbb{F}$ is defined as

$$RS(A, k) = \{(f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)) : f \in \mathbb{F}[x], \deg(f) \leq k - 1\},$$

where $\deg()$ denotes the degree of a polynomial, $f(x) = u_0 + u_1x + u_2x^2 + \dots + u_{k-1}x^{k-1}$, and $u_i \in \mathbb{F}, i = 0, 1, \dots, k - 1$ are the messages. Every evaluation symbol $f(\alpha), \alpha \in A$, is called a code word symbol or a storage node. The *sub-packetization size* is defined as ℓ , and $r \triangleq n - k$ denotes the number of parity symbols.

Assume e nodes fail, $e \leq n - k$, and we want to recover them. The number of helper nodes are denoted by d . The amount of information transmitted from the helper nodes is defined as the *repair bandwidth* b , measured in the number of symbols over $GF(q)$. All the remaining $n - e = d$ nodes are assumed to be the helper nodes unless stated otherwise. We define the *normalized repair bandwidth* as $\frac{b}{\ell d}$, which is the average fraction of information transmitted from each

helper. By [8], [34], the minimum storage regenerating (MSR) bound for the bandwidth is

$$b \geq \frac{e\ell d}{d - k + e}. \quad (2)$$

As mentioned before, codes achieving the MSR bound require large sub-packetization sizes. In this section, we focus on the single erasure case.

Assume $\mathbb{B} \leq \mathbb{F}$, namely, \mathbb{B} is a subfield of \mathbb{F} . A linear repair scheme requires some symbols of the subfield \mathbb{B} to be transmitted from each helper node [6]. If the symbols from the same helper node are linearly dependent, the repair bandwidth decreases. In particular, the scheme uses dual code to compute the failed node and uses trace function to obtain the transmitted subfield symbols, as detailed below.

Assume $f(\alpha^*)$ fails for some $\alpha^* \in A$. For any polynomial $p(x) \in \mathbb{F}[x]$ of which the degree is smaller than r , $(v_1p(\alpha_1), v_2p(\alpha_2), \dots, v_np(\alpha_n))$ is a dual codeword of $RS(A, k)$, where $v_i, i \in [n]$ are non-zero constants determined by the set A (see for example [2, Thm. 4 in Ch.10]). We can thus repair the failed node $f(\alpha^*)$ from

$$v_{\alpha^*}p(\alpha^*)f(\alpha^*) = - \sum_{i=1, \alpha_i \neq \alpha^*}^n v_i p(\alpha_i) f(\alpha_i) \quad (3)$$

The summation on the right side means that we add all the i elements from $i = 1$ to $i = n$ except when $\alpha_i \neq \alpha^*$.

The trace function from \mathbb{F} onto \mathbb{B} is defined as

$$tr_{\mathbb{F}/\mathbb{B}}(\beta) = \beta + \beta^q + \dots + \beta^{q^{\ell-1}}, \quad (4)$$

where $\beta \in \mathbb{F}$, $\mathbb{B} = GF(q)$ is called the *base field*, and q is a power of a prime number. It is a linear mapping from \mathbb{F} to \mathbb{B} and satisfies

$$tr_{\mathbb{F}/\mathbb{B}}(\alpha\beta) = \alpha tr_{\mathbb{F}/\mathbb{B}}(\beta) \quad (5)$$

for all $\alpha \in \mathbb{B}$.

We define the rank $rank_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_i\})$ to be the cardinality of a maximal subset of $\{\gamma_1, \gamma_2, \dots, \gamma_i\}$ that is linearly independent over \mathbb{B} . For example, for $\mathbb{B} = GF(2)$ and $\alpha \notin \mathbb{B}$, $rank_{\mathbb{B}}(\{1, \alpha, 1 + \alpha\}) = 2$ because the subset $\{1, \alpha\}$ is the maximal subset that is linearly independent over \mathbb{B} and the cardinality of the subset is 2.

Assume we use polynomials $p_j(x), j \in [\ell]$ to generate ℓ different dual codewords, called *repair polynomials*. Combining the trace function and the dual code, we have

$$\begin{aligned} & tr_{\mathbb{F}/\mathbb{B}}(v_{\alpha^*}p_j(\alpha^*)f(\alpha^*)) \\ &= - \sum_{i=1, \alpha_i \neq \alpha^*}^n tr_{\mathbb{F}/\mathbb{B}}(v_i p_j(\alpha_i) f(\alpha_i)). \end{aligned} \quad (6)$$

In a repair scheme, the helper $f(\alpha_i)$ transmits

$$\{tr_{\mathbb{F}/\mathbb{B}}(v_i p_j(\alpha_i) f(\alpha_i)) : j \in [\ell]\}. \quad (7)$$

TABLE I

Comparison of different schemes for single erasure. When $a = \ell$, our scheme in one coset is the scheme in [6], [7]. When $a = 1$, our schemes in multiple cosets is the schemes in [16], [17].

	repair bandwidth	code length	restrictions
Schemes in [6], [7]	$(n-1)(\ell-s)$	$n \leq q^\ell$	$q^s \leq r$
Scheme in [16]	$< \frac{\ell}{r}(n+1)$	$n = \log_r \ell$	
Scheme in [17]	$\frac{\ell}{r}(n-1)$	$n^n \approx \ell$	
Our scheme in one coset	$\leq \frac{\ell}{a}(n-1)(a-s)$	$n \leq (q^a - 1)$	$q^s \leq r, a \ell$
Our scheme in two cosets	$< (n-1)\frac{\ell+a}{2}$	$n \leq 2(q^a - 1)$	$\frac{\ell}{r} \leq a, a \ell$
Our scheme in multiple cosets 1	$< \frac{\ell}{r}(n+1 + (r-1)(q^a - 2))$	$n \leq (q^a - 1)m$	$\ell/a = r^m$ for some integer m
Our scheme in multiple cosets 2	$\frac{\ell}{r}(n-1 + (r-1)(q^a - 2))$	$n \leq (q^a - 1)m$	$\ell/a \approx r^m$ for some integer m

TABLE II

Comparison of different schemes for multiple erasures. When $a = \ell$ and $s = \ell$ our scheme in one coset is the scheme 1 in [19]. When $a = 1$, our schemes in multiple cosets is the scheme in [22].

	repair bandwidth	code length	restrictions
Scheme 1 in [19]	$\leq (n-e)e - \frac{e(e-1)(q-1)}{2}$	$n \leq q^\ell$	$q^{\ell-1} \leq r, e < \sqrt{\log_q n}$
Scheme 2 in [19]	$\leq \min_{e' \geq e} ((n-e')(\ell - \lfloor \log_q(\frac{n-k+e'-1}{2e'-1}) \rfloor))$	$n \leq q^\ell$	
Scheme in [22]	$\frac{ed\ell}{d-k+e}$	$n^n \approx \ell$	
Our scheme for multiple erasures in one coset	$\leq \frac{e\ell}{a}(n-e)(a-s)$	$n \leq (q^a - 1)$	$q^s \leq r, a \ell, e < \frac{1}{a-s} \sqrt{\log_q n}$
Our scheme for multiple erasures in multiple cosets	$\frac{e\ell}{n-k}(n-e + (n-k+e)(q^a - 2))$	$n \leq (q^a - 1)m$	$\ell/a \approx m^m$ for some integer m

Suppose $\{v_{\alpha^*} p_1(\alpha^*), v_{\alpha^*} p_2(\alpha^*), \dots, v_{\alpha^*} p_\ell(\alpha^*)\}$ is a basis for \mathbb{F} over \mathbb{B} , and assume $\{\mu_1, \mu_2, \dots, \mu_\ell\}$ is its dual basis. Then, $f(\alpha^*)$ can be repaired by

$$f(\alpha^*) = \sum_{j=1}^{\ell} \mu_j t r_{\mathbb{F}/\mathbb{B}}(v_{\alpha^*} p_j(\alpha^*) f(\alpha^*)). \quad (8)$$

Since v_{α^*} is a non-zero constant, we equivalently suppose that $\{p_1(\alpha^*), \dots, p_\ell(\alpha^*)\}$ is a basis.

In fact, by [6] any linear repair scheme of RS code for the failed node $f(\alpha^*)$ is equivalent to choosing $p_j(x), j \in [\ell]$, with degree smaller than r , such that $\{p_1(\alpha^*), \dots, p_\ell(\alpha^*)\}$ forms a basis for \mathbb{F} over \mathbb{B} . We call this the **full rank condition**:

$$\text{rank}_{\mathbb{B}}(\{p_1(\alpha^*), p_2(\alpha^*), \dots, p_\ell(\alpha^*)\}) = \ell. \quad (9)$$

The repair bandwidth can be calculated from (7) and by noting that $v_i f(\alpha_i)$ is a constant:

$$b = \sum_{\alpha \in A, \alpha \neq \alpha^*} \text{rank}_{\mathbb{B}}(\{p_1(\alpha), p_2(\alpha), \dots, p_\ell(\alpha)\}). \quad (10)$$

We call this the **repair bandwidth condition**.

The goal of a good RS code construction and its repair scheme is to choose appropriate evaluation points A and polynomials $p_j(x), j \in [\ell]$, that can reduce the repair bandwidth in (10) while satisfying (9).

The following lemma is due to the structure of the multiplicative group of \mathbb{F} , which will be used for finding

the evaluation points in the code constructions in this paper. Similar statements can be found in [3, Ch. 2.6].

Lemma 1. Assume $\mathbb{E} \leq \mathbb{F} = GF(q^\ell)$, then \mathbb{F}^* can be partitioned to $t \triangleq \frac{q^\ell - 1}{|\mathbb{E}^*| - 1}$ cosets: $\{\mathbb{E}^*, \beta \mathbb{E}^*, \beta^2 \mathbb{E}^*, \dots, \beta^{t-1} \mathbb{E}^*\}$, where β is a primitive element of \mathbb{F} .

Proof: The $q^\ell - 1$ elements in \mathbb{F}^* are $\{1, \beta, \beta^2, \dots, \beta^{q^\ell - 2}\}$ and $\mathbb{E}^* \subseteq \mathbb{F}^*$. Assume that t is the smallest nonzero number that satisfies $\beta^t \in \mathbb{E}^*$, then we know that $\beta^k \in \mathbb{E}^*$ if and only if $t|k$. Also, $\beta^{k_1} \neq \beta^{k_2}$ when $k_1 \neq k_2$ and $k_1, k_2 < q^\ell - 2$. Since there are only $|\mathbb{E}^*| - 1$ nonzero distinct elements in \mathbb{E}^* and $\beta^{q^\ell - 1} = 1$, we have $t = \frac{q^\ell - 1}{|\mathbb{E}^*| - 1}$ and the t cosets are $\mathbb{E}^* = \{1, \beta^t, \beta^{2t}, \dots, \beta^{(|\mathbb{E}^*| - 2)t}\}$, $\beta \mathbb{E}^* = \{\beta, \beta^{t+1}, \beta^{2t+1}, \dots, \beta^{(|\mathbb{E}^*| - 2)t+1}\}$, \dots , $\beta^{t-1} \mathbb{E}^* = \{\beta^{t-1}, \beta^{2t-1}, \beta^{3t-1}, \dots, \beta^{(|\mathbb{E}^*| - 1)t-1}\}$. ■

III. REED-SOLOMON REPAIR SCHEMES FOR SINGLE ERASURE

In this section, we present our schemes in which the evaluation points are part of one coset, two cosets and multiple cosets for a single erasure. From these constructions, we achieve several different points on the tradeoff between the sub-packetization size and the normalized repair bandwidth. The main ideas of the constructions are:

(i) In all our schemes, we take an original RS code, and construct a new code over a larger finite field. Thus, the sub-packetization size ℓ is increased.

(ii) For the schemes using one and two cosets, the code parameters n, k are kept the same as the original code. Hence, for given $n, r = n - k$, the sub-packetization size ℓ increases, but we show that the normalized repair bandwidth remains the same.

(iii) For the scheme using multiple cosets, the code length n is increased and the redundancy r is fixed. Moreover, the code length n grows faster than the sub-packetization size ℓ . Therefore, for fixed n, r , the sub-packetization ℓ decreases, and we show that the normalized repair bandwidth is only slightly larger than the original code.

A. Schemes in one coset

Assume $\mathbb{E} = GF(q^a)$ is a subfield of $\mathbb{F} = GF(q^\ell)$ and $\mathbb{B} = GF(q)$ is the base field, where q is a prime number. The evaluation points of the code over \mathbb{F} that we construct are part of one coset in Lemma 1.

We first present the following lemma about the basis.

Lemma 2. Assume $\{\xi_1, \xi_2, \dots, \xi_\ell\}$ is a basis for $\mathbb{F} = GF(q^\ell)$ over $\mathbb{B} = GF(q)$, then $\{\xi_1^{q^s}, \xi_2^{q^s}, \dots, \xi_\ell^{q^s}\}$, $s \in [\ell]$ is also a basis.

Proof: Assume $\{\xi_1^{q^s}, \xi_2^{q^s}, \dots, \xi_\ell^{q^s}\}$, $s \in [\ell]$ is not a basis for \mathbb{F} over \mathbb{B} , then there exist nonzero $(\alpha_1, \alpha_2, \dots, \alpha_\ell)$, $\alpha_i \in \mathbb{B}$, $i \in [\ell]$, that satisfy

$$\begin{aligned} & \alpha_1 \xi_1^{q^s} + \alpha_2 \xi_2^{q^s} + \dots + \alpha_\ell \xi_\ell^{q^s} \\ &= 0 \\ &= (\alpha_1 \xi_1 + \alpha_2 \xi_2 + \dots + \alpha_\ell \xi_\ell)^{q^s}, \end{aligned} \quad (11)$$

which is in contradiction to the assumption that $\{\xi_1, \xi_2, \dots, \xi_\ell\}$ is a basis for \mathbb{F} over \mathbb{B} . ■

The following theorem shows the repair scheme using one coset for the evaluation points.

Theorem 1. There exists an $RS(n, k)$ code over $\mathbb{F} = GF(q^\ell)$ with repair bandwidth $b \leq \frac{\ell}{a}(n-1)(a-s)$ symbols over $\mathbb{B} = GF(q)$, where q is a prime number and a, s satisfy $n < q^a$, $q^s \leq n - k$, $a|\ell$.

Proof: Assume a field $\mathbb{F} = GF(q^\ell)$ is extended from $\mathbb{E} = GF(q^a)$, $a | \ell$, and β is a primitive element of \mathbb{F} . We focus on the code $RS(A, k)$ of dimension k over \mathbb{F} with evaluation points $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\} \subseteq \beta^m \mathbb{E}^*$ for some $0 \leq m < \frac{q^\ell - 1}{q^a - 1}$, which is one of the cosets in Lemma 1. The base field is $\mathbb{B} = GF(q)$ and (6) is used to repair the failed node $f(\alpha^*)$.

Construction I: Inspired by [6], for $s = a - 1$, we choose

$$p_j(x) = \frac{\text{tr}_{\mathbb{E}/\mathbb{B}}(\xi_j(\frac{x}{\beta^m} - \frac{\alpha^*}{\beta^m}))}{\frac{x}{\beta^m} - \frac{\alpha^*}{\beta^m}}, j \in [a], \quad (12)$$

where $\{\xi_1, \xi_2, \dots, \xi_a\}$ is a basis for \mathbb{E} over \mathbb{B} . The degree of $p_j(x)$ is smaller than r since $q^s \leq r$. When $x = \alpha^*$, by (4) we have

$$p_j(\alpha^*) = \xi_j. \quad (13)$$

So, the polynomials satisfy

$$\text{rank}_{\mathbb{B}}(\{p_1(\alpha^*), p_2(\alpha^*), \dots, p_a(\alpha^*)\}) = a. \quad (14)$$

When $x \neq \alpha^*$, since $\text{tr}_{\mathbb{E}/\mathbb{B}}(\xi_j(\frac{x}{\beta^m} - \frac{\alpha^*}{\beta^m})) \in \mathbb{B}$, and $\frac{x}{\beta^m} - \frac{\alpha^*}{\beta^m}$ is a constant independent of j , we have

$$\text{rank}_{\mathbb{B}}(\{p_1(x), p_2(x), \dots, p_a(x)\}) = 1. \quad (15)$$

Let $\{\eta_1, \eta_2, \eta_3, \dots, \eta_{\ell/a}\}$ be a basis for \mathbb{F} over \mathbb{E} , the ℓ repair polynomials are chosen as

$$\{\eta_1 p_j(x), \eta_2 p_j(x), \dots, \eta_{\ell/a} p_j(x) : j \in [a]\}. \quad (16)$$

Since $p_j(x) \in \mathbb{E}$, we can conclude that

$$\begin{aligned} & \text{rank}_{\mathbb{B}}(\{\eta_1 p_j(\alpha^*), \eta_2 p_j(\alpha^*), \dots, \eta_{\ell/a} p_j(\alpha^*) : j \in [a]\}) \\ &= \frac{\ell}{a} \text{rank}_{\mathbb{B}}(\{p_1(\alpha^*), p_2(\alpha^*), \dots, p_a(\alpha^*)\}) = \ell \end{aligned} \quad (17)$$

satisfies the full rank condition, and for $x \neq \alpha^*$

$$\begin{aligned} & \text{rank}_{\mathbb{B}}(\{\eta_1 p_j(x), \eta_2 p_j(x), \dots, \eta_{\ell/a} p_j(x) : j \in [a]\}) \\ &= \frac{\ell}{a} \text{rank}_{\mathbb{B}}(\{p_1(x), p_2(x), \dots, p_a(x)\}) = \frac{\ell}{a}. \end{aligned} \quad (18)$$

From (10) we can calculate the repair bandwidth

$$b = \frac{\ell}{a}(n-1). \quad (19)$$

Construction II: For $s \leq a - 1$, inspired by [7], we choose

$$p_j(x) = \xi_j \prod_{i=1}^{q^s-1} \left(\frac{x}{\beta^m} - \left(\frac{\alpha^*}{\beta^m} - w_i^{-1} \xi_j \right) \right), j \in [a], \quad (20)$$

where $\{\xi_1, \xi_2, \dots, \xi_a\}$ is a basis for \mathbb{E} over \mathbb{B} , and $W = \{w_0 = 0, w_1, w_2, \dots, w_{q^s-1}\}$ is an s -dimensional subspace in \mathbb{E} , $s < a$, $q^s \leq r$. It is easy to check that the degree of $p_j(x)$ is smaller than r since $q^s \leq r$. When $x = \alpha^*$, we have

$$p_j(\alpha^*) = \xi_j^{q^s} \prod_{i=1}^{q^s-1} w_i^{-1}. \quad (21)$$

Since $\prod_{i=1}^{q^s-1} w_i^{-1}$ is a constant, from Lemma 2 we have

$$\text{rank}_{\mathbb{B}}(\{p_1(\alpha^*), p_2(\alpha^*), \dots, p_a(\alpha^*)\}) = a. \quad (22)$$

For $x \neq \alpha^*$, set $x' = \frac{\alpha^*}{\beta^m} - \frac{x}{\beta^m} \in \mathbb{E}$, we have

$$\begin{aligned} p_j(x) &= \xi_j \prod_{i=1}^{q^s-1} \left(\frac{x}{\beta^m} - \left(\frac{\alpha^*}{\beta^m} - w_i^{-1} \xi_j \right) \right) \\ &= \xi_j \prod_{i=1}^{q^s-1} (w_i^{-1} \xi_j - x') \\ &= \xi_j \prod_{i=1}^{q^s-1} (w_i^{-1} x') \prod_{i=1}^{q^s-1} (\xi_j / x' - w_i) \\ &= (x')^{q^s} \prod_{i=1}^{q^s-1} (w_i^{-1}) \prod_{i=0}^{q^s-1} (\xi_j / x' - w_i). \end{aligned} \quad (23)$$

By [35, p. 4], $g(y) = \prod_{i=0}^{q^s-1} (y - w_i)$ is a linear mapping from \mathbb{E} to itself with dimension $a - s$ over \mathbb{B} . Since $(x')^{q^s} \prod_{i=1}^{q^s-1} (w_i^{-1})$ is a constant independent of j , we have

$$\text{rank}_{\mathbb{B}}(\{p_1(x), p_2(x), \dots, p_a(x)\}) \leq a - s. \quad (24)$$

Let $\{\eta_1, \eta_2, \eta_3, \dots, \eta_{\ell/a}\}$ be a basis for \mathbb{F} over \mathbb{E} , then the ℓ polynomials are chosen as $\{\eta_1 p_j(x), \eta_2 p_j(x), \dots, \eta_{\ell/a} p_j(x), j \in [a]\}$. From (21) and (23) we know that $p_j(x) \in \mathbb{E}$, so we can conclude that

$$\begin{aligned} & \text{rank}_{\mathbb{B}}(\{\eta_1 p_j(\alpha^*), \eta_2 p_j(\alpha^*), \dots, \eta_{\ell/a} p_j(\alpha^*) : j \in [a]\}) \\ &= \frac{\ell}{a} \text{rank}_{\mathbb{B}}(\{p_1(\alpha^*), p_2(\alpha^*), \dots, p_a(\alpha^*)\}) = \ell \end{aligned} \quad (25)$$

satisfies (9), and for $x \neq \alpha^*$

$$\begin{aligned} & \text{rank}_{\mathbb{B}}(\{\eta_1 p_j(x), \eta_2 p_j(x), \dots, \eta_{\ell/a} p_j(x) : j \in [a]\}) \\ &= \frac{\ell}{a} \text{rank}_{\mathbb{B}}(\{p_1(x), p_2(x), \dots, p_a(x)\}) \leq \frac{\ell}{a} (a - s). \end{aligned} \quad (26)$$

Now from (10) we can calculate the repair bandwidth

$$b \leq \frac{\ell}{a} (n - 1)(a - s). \quad (27)$$

Combining (19) and (27) will complete the proof of Theorem 1. \blacksquare

Rather than directly using the schemes in [6] and [7], the polynomials (12) and (20) that we use are similar to [6] and [7], respectively, but are mappings from \mathbb{E} to \mathbb{B} . Moreover, we multiply each polynomial with the basis for \mathbb{F} over \mathbb{E} to satisfy the full rank condition. In this case, our scheme significantly reduces the repair bandwidth when the code length remains the same. Our evaluation points are in a coset rather than the entire field \mathbb{F} as in [6] and [7]. It should be noted that a here can be an arbitrary number that divides ℓ and when $a = \ell$, our schemes are exactly the same as those in [6] and [7]. Note that the normalized repair bandwidth $\frac{b}{\ell(n-1)}$ decreases as a decreases. Therefore, our scheme outperforms those in [6] and [7] when applied to the case of $\ell > \log_q n$.

Example 1. Assume $q = 2, \ell = 9, a = 3$ and $\mathbb{E} = \{0, 1, \alpha, \alpha^2, \dots, \alpha^6\}$. Let $A = \mathbb{E}^*$, $n = 7, k = 5$ so $r = n - k = 2$. Choose $s = \log_2 r = 1$ and $W = \{0, 1\}$ in Construction II. Then, we have $p_j(x) = \xi_j(x - \alpha^* + \xi_j)$. Let $\{\xi_1, \xi_2, \xi_3\}$ be $\{1, \alpha, \alpha^2\}$. It is easy to check that $\text{rank}_{\mathbb{B}}(\{p_1(\alpha^*), p_2(\alpha^*), p_3(\alpha^*)\}) = 3$ and $\text{rank}_{\mathbb{B}}(\{p_1(x), p_2(x), p_3(x)\}) = 2$ for $x \neq \alpha^*$. Therefore the repair bandwidth is $b = 36$ bits as suggested in Theorem 1. For the same (n, k, ℓ) , the repair bandwidth in [7] is 48 bits. For another example, consider $RS(14, 10)$ code used in Facebook [33], we have repair bandwidth of 52 bits for $\ell = 8$, while [7] requires 60 bits and the naive scheme requires 80 bits.

Remark 1. The schemes in [6] and [7] can also be used in an RS code over \mathbb{E} with repair bandwidth $(n-1)(a-s)$, and

with ℓ/a copies of the code. Thus, they can also reach the repair bandwidth of $\frac{\ell}{a}(n-1)(a-s)$. It should be noted that by doing so, the code is a vector code, however our scheme constructs a scalar code. To the best of our knowledge, this is the first example of such a scalar code in the literature.

B. Schemes in two cosets

Now we discuss our scheme when the evaluation points are chosen from two cosets. In this scheme, we choose the polynomials that have full rank when evaluated at the coset containing the failed node, and rank 1 when evaluated at the other coset.

Theorem 2. There exists an $RS(n, k)$ code over $\mathbb{F} = GF(q^\ell)$ with repair bandwidth $b < (n-1)\frac{\ell+a}{2}$ symbols over $\mathbb{B} = GF(q)$, where q is a prime number and a satisfies $n \leq 2(q^a - 1)$, $a|\ell$, $\frac{\ell}{a} \leq n - k$.

Proof: Assume a field $\mathbb{F} = GF(q^\ell)$ is extended from $\mathbb{E} = GF(q^a)$ and β is the primitive element of \mathbb{F} . We focus on the code $RS(A, k)$ over \mathbb{F} of dimension k with evaluation points A consisting of $n/2$ points from $\beta^{m_1}\mathbb{E}^*$ and $n/2$ points from $\beta^{m_2}\mathbb{E}^*$, $0 \leq m_1 < m_2 \leq \frac{q^\ell-1}{q^a-1}$ and $m_2 - m_1 = q^s$, $s \in \{0, 1, \dots, \frac{\ell}{a}\}$.

In this case we view \mathbb{E} as the base field and repair the failed node $f(\alpha^*)$ by

$$\begin{aligned} & \text{tr}_{\mathbb{F}/\mathbb{E}}(v_{\alpha^*} p_j(\alpha^*) f(\alpha^*)) \\ &= - \sum_{i=1, \alpha_i \neq \alpha^*}^n \text{tr}_{\mathbb{F}/\mathbb{E}}(v_i p_j(\alpha_i) f(\alpha_i)). \end{aligned} \quad (28)$$

Inspired by [6, Theorem 10], for $j \in [\frac{\ell}{a}]$, we choose

$$p_j(x) = \begin{cases} (\frac{x}{\beta^{m_2}})^{j-1}, & \text{if } \alpha^* \in \beta^{m_1}\mathbb{E}^*, \\ (\frac{x}{\beta^{m_1}})^{j-1}, & \text{if } \alpha^* \in \beta^{m_2}\mathbb{E}^*. \end{cases} \quad (29)$$

The degree of $p_j(x)$ is smaller than r when $\frac{\ell}{a} \leq r$. Then, we check the rank in each case.

When $\alpha^* \in \beta^{m_2}\mathbb{E}^*$, if $x = \beta^{m_1}\gamma \in \beta^{m_1}\mathbb{E}^*$, for some $\gamma \in \mathbb{E}^*$,

$$p_j(x) = \left(\frac{x}{\beta^{m_1}}\right)^{j-1} = \gamma^{j-1}, \quad (30)$$

so

$$\text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\frac{\ell}{a}}(x)\}) = 1. \quad (31)$$

If $x = \beta^{m_2}\gamma \in \beta^{m_2}\mathbb{E}^*$, for some $\gamma \in \mathbb{E}^*$,

$$p_j(x) = \left(\frac{x}{\beta^{m_1}}\right)^{j-1} = (\beta^{m_2-m_1})^{j-1} \gamma^{j-1}. \quad (32)$$

Since $m_2 - m_1 = q^s$ and $\{1, \beta, \beta^2, \dots, \beta^{\frac{\ell}{a}-1}\}$ is the polynomial basis for \mathbb{F} over \mathbb{E} , from Lemma 2 we know that

$$\text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\frac{\ell}{a}}(x)\}) = \frac{\ell}{a}. \quad (33)$$

When $\alpha^* \in \beta^{m_1}\mathbb{E}^*$, if $x = \beta^{m_1}\gamma \in \beta^{m_1}\mathbb{E}^*$, for some $\gamma \in \mathbb{E}^*$,

$$\begin{aligned} p_j(x) &= \left(\frac{x}{\beta^{m_2}}\right)^{j-1} \\ &= (\beta^{m_1-m_2})^{j-1}\gamma^{j-1} \\ &= (\beta^{m_2-m_1})^{1-\frac{\ell}{a}}(\beta^{m_2-m_1})^{\frac{\ell}{a}-j}\gamma^{j-1}. \end{aligned} \quad (34)$$

Since $(\beta^{m_2-m_1})^{1-\frac{\ell}{a}}$ is a constant, from Lemma 2 we know that

$$\text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\frac{\ell}{a}}(x)\}) = \frac{\ell}{a}. \quad (35)$$

If $x = \beta^{m_2}\gamma \in \beta^{m_2}\mathbb{E}^*$, for some $\gamma \in \mathbb{E}^*$,

$$p_j(x) = \left(\frac{x}{\beta^{m_2}}\right)^{j-1} = \gamma^{j-1}, \quad (36)$$

so

$$\text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\frac{\ell}{a}}(x)\}) = 1. \quad (37)$$

Therefore, $\{p_j(\alpha^*), j \in [\frac{\ell}{a}]\}$ has full rank over \mathbb{E} , for any evaluation point $\alpha^* \in A$. For x from the coset containing α^* , the polynomials have rank ℓ/a , and for x from the other coset, the polynomials have rank 1. Then, the repair bandwidth in symbols over \mathbb{B} can be calculated from (10) as

$$\begin{aligned} b &= \frac{\ell}{a}\left(\frac{n}{2} - 1\right) \log_q |\mathbb{E}| + \frac{n}{2} \log_q |\mathbb{E}| \\ &= (n-1) \frac{\ell+a}{2} - \frac{\ell-a}{2} \\ &< (n-1) \frac{\ell+a}{2}. \end{aligned} \quad (38)$$

Thus, the proof is completed. \blacksquare

Example 2. Take the $RS(14, 11)$ code over $\mathbb{F} = GF(2^{12})$ for example. Let β be the primitive element in \mathbb{F} , $a = 4$, $s = \ell/a = 3$ and $A = \mathbb{E}^* \cup \beta\mathbb{E}^*$. Assume $\alpha^* \in \beta\mathbb{E}^*$, then $\{p_j(x), j \in [3]\}$ is the set $\{1, x, x^2\}$. It is easy to check that when $x \in \beta\mathbb{E}^*$ the polynomials have full rank and when $x \in \mathbb{E}^*$ the polynomials have rank 1. The total repair bandwidth is 100 bits. For the same (n, k, ℓ) , the repair bandwidth of our scheme in one coset is 117 bits. For the scheme in [6], which only works for $\ell/a = 2$, we can only choose $a = 6$ and get the repair bandwidth of 114 bits for the same (n, k, ℓ) .

C. Schemes in multiple cosets

In the schemes in this subsection, we extend an original code to a new code over a larger field and the evaluation points are chosen from multiple cosets in Lemma 1 to increase the code length. The construction ensures that for fixed n , the sub-packetization size is smaller than the original code. If the original code satisfies several conditions to be discussed soon, the repair bandwidth in the new code is only slightly larger than that of the original code.

Particularly, if the original code is an MSR code, then we can get the new code in a much smaller sub-packetization level with a small extra repair bandwidth. Also, if the original code works for any number of helpers and multiple erasures, the new code works for any number of helpers and multiple erasures, too. We discuss multiple erasures in Section IV.

We first prove a lemma regarding the ranks over different base fields, and then describe the new code.

Lemma 3. Let $\mathbb{B} = GF(q)$, $\mathbb{F}' = GF(q^{\ell'})$, $\mathbb{E} = GF(q^a)$, $\mathbb{F} = GF(q^{\ell})$, $\ell = a\ell'$. a and ℓ' are relatively prime and q can be any power of a prime number. For any set of $\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\} \subseteq \mathbb{F}' \leq \mathbb{F}$, we have

$$\begin{aligned} &\text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) \\ &= \text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}). \end{aligned} \quad (39)$$

Proof: Assume $\text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) = c$ and without loss of generality, $\{\gamma_1, \gamma_2, \dots, \gamma_c\}$ are linearly independent over \mathbb{B} . Then, we can construct $\{\gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\} \subseteq \mathbb{F}'$ to make $\{\gamma_1, \gamma_2, \dots, \gamma_c, \gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\}$ form a basis for \mathbb{F}' over \mathbb{B} .

Assume we get \mathbb{F} by adjoining β to \mathbb{B} . Then, from [36, Theorem 1.86] we know that $\{1, \beta, \beta^2, \dots, \beta^{\ell'-1}\}$ is a basis for both \mathbb{F} over \mathbb{E} , and \mathbb{F}' over \mathbb{B} . So, any symbol $y \in \mathbb{F}$ can be presented as a linear combination of $\{1, \beta, \beta^2, \dots, \beta^{\ell'-1}\}$ with some coefficients in \mathbb{E} . Also, we know that there is an invertible linear transformation with coefficients in \mathbb{B} between $\{\gamma_1, \gamma_2, \dots, \gamma_c, \gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\}$ and $\{1, \beta, \beta^2, \dots, \beta^{\ell'-1}\}$, because they are a basis for \mathbb{F}' over \mathbb{B} . Combined with the fact that $\{1, \beta, \beta^2, \dots, \beta^{\ell'-1}\}$ is also a basis for \mathbb{F} over \mathbb{E} , we can conclude that any symbol $y \in \mathbb{F}$ can be represented as

$$y = x_1\gamma_1 + x_2\gamma_2 + \dots + x_c\gamma_c + x_{c+1}\gamma'_{c+1} + \dots + x_{\ell'}\gamma'_{\ell'} \quad (40)$$

with some coefficients $x_i \in \mathbb{E}$, which means that $\{\gamma_1, \gamma_2, \dots, \gamma_c, \gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\}$ is also a basis for \mathbb{F} over \mathbb{E} . Then, we have that $\{\gamma_1, \gamma_2, \dots, \gamma_c\}$ are linearly independent over \mathbb{E} ,

$$\begin{aligned} &\text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) \\ &\geq c \\ &= \text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}). \end{aligned} \quad (41)$$

Since $\mathbb{B} \leq \mathbb{E}$, we also have

$$\begin{aligned} &\text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) \\ &\leq \text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}). \end{aligned} \quad (42)$$

The proof is completed. \blacksquare

Theorem 3. Assume there exists a $RS(n', k')$ code \mathcal{E}' over $\mathbb{F}' = GF(q^{\ell'})$ with evaluation points set A' . The evaluation points are linearly independent over $\mathbb{B} = GF(q)$. The repair bandwidth is b' and the repair polynomials are

$p'_j(x)$. Then, we can construct a new $RS(n, k)$ code \mathcal{E} over $\mathbb{F} = GF(q^\ell)$, $\ell = a\ell'$ with $n = (q^a - 1)n'$, $k = n - n' + k'$ and repair bandwidth of $b = ab'(q^a - 1) + (q^a - 2)\ell$ symbols over $\mathbb{B} = GF(q)$ if we can find new repair polynomials $p_j(x) \in \mathbb{F}[x]$, $j \in [\ell']$, with degrees less than $n - k$ that satisfy

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\ell'}(x)\}) \\ &= \text{rank}_{\mathbb{B}}(\{p'_1(\alpha), p'_2(\alpha), \dots, p'_{\ell'}(\alpha)\}) \end{aligned} \quad (43)$$

for all $\alpha \in A'$, $x \in \alpha\mathbb{E}^*$, where $\mathbb{E} = GF(q^a)$.

Proof: We first prove the case when a and ℓ' are necessarily relatively prime using Lemma 3, the case when a and ℓ' are not relatively prime are proved in Appendix A. Assume the evaluation points of \mathcal{E}' are $A' = \{\alpha_1, \alpha_2, \dots, \alpha_{n'}\}$, then from Lemma 3 we know that they are also linearly independent over \mathbb{E} , so there does not exist $\gamma_i, \gamma_j \in \mathbb{E}^*$ that satisfy $\alpha_i \gamma_i = \alpha_j \gamma_j$, which implies that $\{\alpha_1 \mathbb{E}^*, \alpha_2 \mathbb{E}^*, \dots, \alpha_{n'} \mathbb{E}^*\}$ are distinct cosets. Then, we can extend the evaluation points to be

$$A = \{\alpha_1 \mathbb{E}^*, \alpha_2 \mathbb{E}^*, \dots, \alpha_{n'} \mathbb{E}^*\}. \quad (44)$$

and $n = (q^a - 1)n'$. We keep the same redundancy $r = n' - k'$ for the new code so $k = n - r$.

For the new code \mathcal{E} , we use $p_j(x) \in \mathbb{F}[x]$, $j \in [\ell']$ to repair the failed node $f(\alpha^*)$

$$\begin{aligned} & \text{tr}_{\mathbb{F}/\mathbb{E}}(v_{\alpha^*} p_j(\alpha^*) f(\alpha^*)) \\ &= - \sum_{\alpha \in A, \alpha \neq \alpha^*} \text{tr}_{\mathbb{F}/\mathbb{E}}(v_{\alpha} p_j(\alpha) f(\alpha)). \end{aligned} \quad (45)$$

Assume the failed node is $f(\alpha^*)$ and $\alpha^* \in \alpha_i \mathbb{E}^*$. Then, for the node $x \in \alpha_i \mathbb{E}^*$, because the original code satisfies the full rank condition, we have

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\ell'}(x)\}) \\ &= \text{rank}_{\mathbb{B}}(\{p'_1(\alpha_i), p'_2(\alpha_i), \dots, p'_{\ell'}(\alpha_i)\}) = \ell', \end{aligned} \quad (46)$$

then we can recover the failed node with $p_j(x)$, and each helper in the coset containing the failed node transmits ℓ' symbols over \mathbb{E} .

For a helper in the other cosets, $x \in \alpha_\epsilon \mathbb{E}^*$, $\epsilon \neq i$, by (43),

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{p_1(x), p_2(x), \dots, p_{\ell'}(x)\}) \\ &= \text{rank}_{\mathbb{B}}(\{p'_1(\alpha_\epsilon), p'_2(\alpha_\epsilon), \dots, p'_{\ell'}(\alpha_\epsilon)\}), \end{aligned} \quad (47)$$

then every helper in these cosets transmits $\frac{b'}{n'-1}$ symbols in \mathbb{E} on average.

The repair bandwidth of the new code can be calculated from the repair bandwidth condition (10) as

$$\begin{aligned} b &= \frac{b'}{n'-1} \cdot (n' - 1) |\mathbb{E}^*| \cdot a + (|\mathbb{E}^*| - 1) \ell' \cdot a \\ &= ab'(q^a - 1) + (q^a - 2)\ell \end{aligned} \quad (48)$$

which completes the proof. \blacksquare

Note that the calculation in (48) and (38) are similar in the sense that a helper in the coset containing the failure naively

transmits the entire stored information, and the other helpers use the bandwidth that is the same as the original code.

As a special case of Theorem 3, when $b' = \frac{\ell'}{r}(n' - 1)$ matching the MSR bound (1), we get

$$b = \frac{\ell}{r}(n - 1) + \frac{\ell}{r}(r - 1)(q^a - 2), \quad (49)$$

where the second term is the extra bandwidth compared to the MSR bound.

Next, we apply Theorem 3 to the near-MSR code [16] and the MSR code [17]. The first realization of the scheme in multiple cosets is inspired by [16].

Theorem 4. There exists an $RS(n, k)$ code over $\mathbb{F} = GF(q^\ell)$ of which $n = (q^a - 1) \log_r \frac{\ell}{a}$ and $a|\ell$, such that the repair bandwidth satisfies $b < \frac{\ell}{n-k} [n + 1 + (n - k - 1)(q^a - 2)]$, measured in symbols over $\mathbb{B} = GF(q)$ for some prime number q .

Proof: We first prove the case when a and ℓ' are relatively prime using Lemma 3, the case when a and ℓ' are not necessarily relatively prime are proved in Appendix A. We use the code in [16] as the original code. The original code is defined in $\mathbb{F}' = GF(q^{\ell'})$ and $\ell' = r^{n'}$. The evaluation points are $A' = \{\beta, \beta^r, \beta^{r^2}, \dots, \beta^{r^{n'-1}}\}$ where β is a primitive element of \mathbb{F}' .

In the original code, for $c = 0, 1, 2, \dots, \ell' - 1$, we write its r -ary expansion as $c = (c_{n'} c_{n'-1} \dots c_1)$, where $0 \leq c_i \leq r - 1$ is the i -th digit from the right. Assuming the failed node is $f(\beta^{r^{i-1}})$, the repair polynomials are chosen to be

$$p'_j(x) = \beta^c x^s, c_i = 0, s = 0, 1, 2, \dots, r - 1, x \in \mathbb{F}'. \quad (50)$$

Here c varies from 0 to $\ell' - 1$ given that $c_i = 0$, and s varies from 0 to $r - 1$. So, we have ℓ' polynomials in total. The subscript j is indexed by c and s , and by a small abuse of the notation, we write $j \in [\ell']$.

In the new code, let us define $\mathbb{E} = GF(q^a)$ of which a and ℓ' are relatively prime. Adjoining β to \mathbb{E} , we get $\mathbb{F} = GF(q^\ell)$, $\ell = a\ell'$. The new evaluation points are $A = \{\beta \mathbb{E}^*, \beta^r \mathbb{E}^*, \beta^{r^2} \mathbb{E}^*, \dots, \beta^{r^{n'-1}} \mathbb{E}^*\}$. Since A' is part of the polynomial basis for \mathbb{F}' over \mathbb{B} , we know that $\{\beta, \beta^r, \beta^{r^2}, \dots, \beta^{r^{n'-1}}\}$ are linearly independent over \mathbb{B} . Hence, we can apply Lemma 3 and the cosets are distinct, resulting in $n = |A| = (q^a - 1) \log_r \frac{\ell}{a}$.

In our new code, let us assume the failed node is $f(\alpha^*)$ and $\alpha^* \in \beta^{r^{i-1}} C$, and we choose the polynomial $p_j(x)$ with the same form as $p'_j(x)$,

$$p_j(x) = \beta^c x^s, c_i = 0, s = 0, 1, 2, \dots, r - 1, x \in \mathbb{F}. \quad (51)$$

For nodes corresponding to $x = \beta^{r^t} \gamma \in \beta^{r^t} \mathbb{E}^*$, for some $\gamma \in \mathbb{E}^*$, we know that

$$p_j(x) = \beta^c x^s = \beta^c (\gamma \beta^{r^t})^s = \gamma^s p'_j(\beta^{r^t}). \quad (52)$$

Since $p'_j(\beta^{r^t}) \in \mathbb{F}'$, from Lemma 3, we have

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{\gamma^s p'_1(\beta^{r^t}), \gamma^s p'_2(\beta^{r^t}), \dots, \gamma^s p'_{\ell'}(\beta^{r^t})\}) \\ &= \text{rank}_{\mathbb{E}}(\{p'_1(\beta^{r^t}), p'_2(\beta^{r^t}), \dots, p'_{\ell'}(\beta^{r^t})\}) \\ &= \text{rank}_{\mathbb{B}}(\{p'_1(\beta^{r^t}), p'_2(\beta^{r^t}), \dots, p'_{\ell'}(\beta^{r^t})\}), \end{aligned} \quad (53)$$

which satisfies (43). Since the repair bandwidth of the original code is $b' < (n' + 1)\frac{\ell'}{r}$, from (48) we can calculate the repair bandwidth as

$$\begin{aligned} b &= ab'(q^a - 1) + (q^a - 2)\ell \\ &< \frac{\ell}{r}[n + 1 + (r - 1)(q^a - 2)], \end{aligned} \quad (54)$$

where the second term is the extra bandwidth compared to the original code. ■

Example 3. We take an $RS(4, 2)$ code in $GF(2^{16})$ as the original code and extend it with $a = 3, |\mathbb{E}^*| = 7$ to an $RS(28, 26)$ code in $GF(2^{48})$ with normalized repair bandwidth of $\frac{b}{(n-1)\ell} < 0.65$. The $RS(28, 26)$ code in [16] achieves the normalized repair bandwidth of $\frac{b}{(n-1)\ell} < 0.54$, while it requires $\ell = 2.7 \times 10^8$. Our scheme has a much smaller ℓ compared to the scheme in [16] while the repair bandwidth is a bit larger.

In the above theorem, we extend [16] to a linearly larger sub-packetization and an exponentially larger code length, which means that for the same code length, we can have a much smaller sub-packetization level.

Next, we show our second realization of the scheme in multiple cosets, which is inspired by [17]. Different from the previous constructions, this one allows any number of helpers, $k \leq d \leq n - 1$. The sub-packetization size in the original code of [17] satisfies $\ell' \approx (n')^{n'}$ when n' grows to infinity, thus in our new code it satisfies $\ell \approx a(n')^{n'}$ for some integer a .

Theorem 5. Let q be a prime number. There exists an $RS(n, k)$ code over $\mathbb{F} = GF(q^\ell)$ of which $\ell = asq_1q_2 \dots q_{\frac{n}{a-1}}$, where q_i is the i -th prime number that satisfies $s|(q_i - 1)$, $s = d - k + 1$ and a is some integer. d is the number of helpers, $k \leq d \leq (n - 1)$. The average repair bandwidth is $b = \frac{d\ell}{(n-1)(d-k+1)}[n-1+(d-k)(q^a-2)]$ measured in symbols over $\mathbb{B} = GF(q)$.

Proof: We first prove the case when a and ℓ' are relatively prime using Lemma 3, the case when a and ℓ' are not necessarily relatively prime are proved in Appendix A. We use the code in [17] as the original code, where the number of helpers is d' . We set $n - k = n' - k'$ and calculate the repair bandwidth for d helpers from the original code when $d' = d - k + k'$. Let us define $\mathbb{F}_q(\alpha)$ to be the field obtained by adjoining α to the base field \mathbb{B} . Similarly, we define $\mathbb{F}_q(\alpha_1, \alpha_2, \dots, \alpha_n)$ for adjoining multiple elements. Let α_i be an element of order q_i over \mathbb{B} . The code is defined in the field $\mathbb{F}' = GF(q^{\ell'}) = GF(q^{sq_1q_2 \dots q_{n'}})$, which is the

degree- s extension of $\mathbb{F}_q(\alpha_1, \alpha_2, \dots, \alpha_{n'})$. The evaluation points are $A' = \{\alpha_1, \alpha_2, \dots, \alpha_{n'}\}$.

Assuming the failed node is $f(\alpha_i)$ and the helpers are chosen from the set R' , $|R'| = d'$, the base field for repair is \mathbb{F}'_i , defined as $\mathbb{F}'_i \triangleq \mathbb{F}_q(\alpha_j, j \in [n'], j \neq i)$. The repair polynomials are $\{\eta_t p'_j(\alpha_i), t \in [q_i], j \in [s]\}$, where

$$p'_j(x) = x^{j-1} g'(x), j \in [s], x \in \mathbb{F}', \quad (55)$$

$$g'(x) = \prod_{\alpha \in A'/(R' \cup \{\alpha_i\})} (x - \alpha), x \in \mathbb{F}'. \quad (56)$$

and $\eta_t \in \mathbb{F}'$, $t \in [q_i]$, are constructed in [17] such that $\{\eta_t p'_j(\alpha_i), t \in [q_i], j \in [s]\}$ forms the basis for \mathbb{F}' over \mathbb{F}'_i . The repair is done using

$$\begin{aligned} & \text{tr}_{\mathbb{F}'/\mathbb{F}'_i}(v_{\alpha_i} \eta_t p'_j(\alpha_i) f'(\alpha_i)) \\ &= - \sum_{\epsilon=1, \epsilon \neq i}^{n'} \text{tr}_{\mathbb{F}'/\mathbb{F}'_i}(v_{\alpha_\epsilon} \eta_t p'_j(\alpha_\epsilon) f'(\alpha_\epsilon)). \end{aligned} \quad (57)$$

For $x \notin R' \cup \{\alpha_i\}$, $p'_j(x) = 0$, so no information is transmitted. The original code reaches the MSR repair bandwidth

$$\begin{aligned} b' &= \sum_{\epsilon \in R'} \text{rank}_{\mathbb{F}'_i}(\{\eta_t p'_j(\alpha_\epsilon) : t \in [q_i], j \in [s]\}) \\ &= \frac{d' \ell'}{d' - k' + 1}. \end{aligned} \quad (58)$$

In our new code, we define $\mathbb{E} = GF(q^a) = \mathbb{F}_q(\alpha_{n+1})$ where a and ℓ' are relatively prime, and α_{n+1} is an element of order a over \mathbb{B} . Adjoining the primitive element of \mathbb{F}' to \mathbb{E} , we get $\mathbb{F} = GF(q^\ell)$, $\ell = a\ell'$. The new code is defined in \mathbb{F} . We extend the evaluation points to be $A = \{\alpha_1 \mathbb{E}^*, \alpha_2 \mathbb{E}^*, \dots, \alpha_{n'} \mathbb{E}^*\}$. Since $\{\alpha_1, \alpha_2, \dots, \alpha_{n'}\}$ are linearly independent over \mathbb{B} , we can apply Lemma 3 and the cosets are distinct. So, $n = |A| = (q^a - 1)n'$.

Assuming the failed node is $f(\alpha^*)$ and $\alpha^* \in \alpha_i \mathbb{E}^*$ and the helpers are chosen from the set R , $|R| = d$, the base field for repair is \mathbb{F}_i , which is defined by $\mathbb{F}_i \triangleq \mathbb{F}_q(\alpha_j, j \in [n+1], j \neq i)$, for $i \in [n]$. We define the repair polynomials $\{\eta_t p_j(x), t \in [q_i], j \in [s]\}$, where

$$p_j(x) = x^{j-1} g(x), j \in [s], x \in \mathbb{F}, \quad (59)$$

$$g(x) = \prod_{\alpha \in A/(R \cup \{\alpha^*\})} (x - \alpha), x \in \mathbb{F}, \quad (60)$$

and η_t is the same as that in the original code. Then, we repair the failed node by

$$\begin{aligned} & \text{tr}_{\mathbb{F}/\mathbb{F}_i}(v_{\alpha^*} \eta_t p_j(\alpha^*) f(\alpha^*)) \\ &= - \sum_{\alpha \in A, \alpha \neq \alpha^*} \text{tr}_{\mathbb{F}/\mathbb{F}_i}(v_{\alpha} \eta_t p_j(\alpha) f(\alpha)). \end{aligned} \quad (61)$$

For $x \in \alpha \mathbb{E}^*$, $\alpha \in A'$, we have

$$p_j(x) = \gamma^{j-1} \alpha^{j-1} g(x), j \in [s], \quad (62)$$

for some $\gamma \in \mathbb{E}^*$. If $x \notin R \cup \{\alpha^*\}$, since $g(x) = 0$, no information is transmitted from node x . Next, we consider all other nodes.

For $x = \alpha\gamma, \alpha \in A'$, since $g(x)$ is a constant independent of j , $\gamma \in \mathbb{E} \subseteq \mathbb{F}_i$ and $\eta_t, \alpha_i \in \mathbb{F}'$, from Lemma 3 we have

$$\begin{aligned} & \text{rank}_{\mathbb{F}_i}(\{\eta_t p_1(x), \eta_t p_2(x), \dots, \eta_t p_s(x) : t \in [q_i]\}) \\ &= \text{rank}_{\mathbb{F}_i}(\{\eta_t, \eta_t \gamma \alpha, \dots, \eta_t \gamma^{s-1} \alpha^{s-1} : t \in [q_i]\}) \\ &= \text{rank}_{\mathbb{F}_i}(\{\eta_t, \eta_t \alpha, \dots, \eta_t \alpha^{s-1} : t \in [q_i]\}) \\ &= \text{rank}_{\mathbb{F}'_i}(\{\eta_t, \eta_t \alpha, \dots, \eta_t \alpha^{s-1} : t \in [q_i]\}) \\ &= \text{rank}_{\mathbb{F}'_i}(\{\eta_t p'_1(\alpha), \eta_t p'_2(\alpha), \dots, \eta_t p'_s(\alpha) : t \in [q_i]\}), \end{aligned} \quad (63)$$

which satisfies (43).

When $k \leq d < n - 1$, assuming the helpers are randomly chosen from all the remaining nodes, the average repair bandwidth for different choices of the helpers can be calculated as

$$\begin{aligned} b &= d \left[\frac{b'a}{d'} \cdot \frac{n-1-(q^a-2)}{n-1} + \ell'a \cdot \frac{q^a-2}{n-1} \right] \quad (64) \\ &= \frac{d\ell}{d-k+1} + \frac{d}{n-1} \frac{\ell}{d-k+1} (d-k)(q^a-2). \end{aligned} \quad (65)$$

Here in (64) the second term corresponds to the helpers in the failed node coset, the first term corresponds to the helpers in the other cosets, and in (65) we used $d' - k' = d - k$. ■

In the case of $d = n - 1$, the repair bandwidth of the code in Theorem 5 can be directly calculated from (48) as

$$\begin{aligned} b &= ab'(q^a - 1) + (q^a - 2)\ell \\ &= \frac{\ell}{r}(n-1) + \frac{\ell}{r}(r-1)(q^a - 2). \end{aligned} \quad (66)$$

In (65) and (66), the second term is the extra repair bandwidth compared to the original code.

In Theorems 4 and 5, we constructed our schemes by extending previous schemes. However, it should be noted that since we only used the properties of the polynomials $p'_j(x)$, we have no restrictions on the dimensions k' of the original codes. So, in some special cases, even if k' is negative and the original codes do not exist, our theorems still hold. Thus, we can provide more feasible points of (n, k) using our schemes. This is illustrated in the example below.

Example 4. Let us take the $RS(12, 8)$ code as an example. We set $q = 2, s = 4, q_1 = 5, q_2 = 9, q_3 = 13$ and $a = 7$. Then, $\ell' = 2340$ and $\ell = 16380$. Assuming the failed node is $f(\alpha^*)$ and $\alpha^* \in \alpha_1 C$, then we repair it in \mathbb{F}_1 and set the polynomials in (59). We can easily check that when $x \in \alpha_1 C$, $\text{rank}_{\mathbb{F}_1}(\{\eta_t p_1(x), \eta_t p_2(x), \dots, \eta_t p_s(x) : t \in [5]\}) = 20$ and when x in other cosets, $\text{rank}_{\mathbb{F}_1}(\{\eta_t p_1(x), \eta_t p_2(x), \dots, \eta_t p_s(x) : t \in [5]\}) = 5$. Therefore, we transmit 100 symbols in \mathbb{F}_1 , which can be normalized to $\frac{b}{(n-1)\ell} = 0.4545$. Compared with the scheme in [17], which

need $\ell = 2.4 \times 10^{19}$ and $\frac{b}{(n-1)\ell} = 0.25$, we provide a tradeoff between ℓ and b .

It should be noted that in this example, the $RS(12, 8)$ code needs to be extended from an $RS(3, -1)$ code, which does not exist. However, since we only used the properties of the polynomials $p'_j(x)$ and $p_j(x)$, the new $RS(12, 8)$ code still works.

D. Numerical evaluations and discussions

In this subsection, we compare the existing and the proposed schemes. Table I shows the repair bandwidth and the code length of each scheme. For the comparison, we first show in Figures 1 and 2 the performance of each scheme when the sub-packetization changes, given $(n, k) = (12, 10)$ and $(12, 8)$, respectively. We only consider $n - 1$ helpers. Two single points $(\log_2(\ell) = 53.5, \frac{b}{(n-1)\ell} = 0.50)$ in $RS(12, 10)$ codes and $(\log_2(\ell) = 64.4, \frac{b}{(n-1)\ell} = 0.25)$ in $RS(12, 8)$ codes are not shown in the figures, they can be achieved by both our second realization in multiple cosets and [17]. We make the following observations.

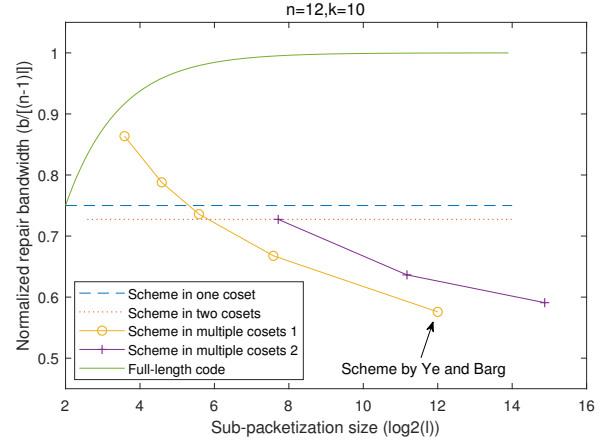


Fig. 1. Comparison of 3 schemes, $q = 2, n = 12, k = 10, r = 2$, x-axis is the log scale sub-packetization size, y-axis is the normalized repair bandwidth.

- 1) For a fixed (n, k) , we compare the normalized repair bandwidth $b/[(n-1)\ell]$ in different sub-packetization sizes. In our schemes in multiple cosets, we have a parameter a to adjust the sub-packetization size. From Theorems 4 and 5 we know that for the two schemes, $\ell = a \cdot r \frac{n}{q^{a-1}-1}$ and $\ell \approx a \cdot (\frac{n}{q^a-1}) (\frac{n}{q^{a-1}-1})$, respectively, which means that increasing a will decrease the sub-packetization ℓ . In our schemes in one coset and two cosets, the parameter a is determined by code length n , and will not be changed by increasing ℓ , neither will the normalized repair bandwidth. When $q = 2, a = 1$, the two schemes in multiple cosets coincides with [16] and [17], respectively.

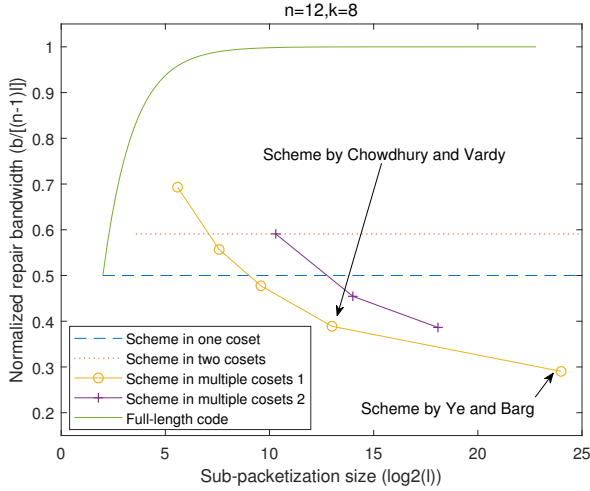


Fig. 2. Comparison of 3 schemes, $q = 2, n = 12, k = 8, r = 4$, x-axis is the log scale sub-packetization size, y-axis is the normalized repair bandwidth. The scheme by Chowdhury and Vardy is in [32], the scheme by Ye and Barg is in [16], and the full-length code is in [6] and [7].

- 2) The scheme in [32] also achieves one tradeoff point in Figure 2, which can be viewed as a special case of our scheme in multiple coset 1.
- 3) For fixed n, k , our schemes are better than the full-length code in [6] and [7] for all ℓ , except when $\ell = 4$, for which our scheme in one coset is identical to the full-length code.
- 4) While the repair bandwidth of the full-length code grows with ℓ , our schemes in one coset and two cosets have a constant normalized bandwidth, and our schemes in multiple cosets have a decreasing normalized bandwidth with ℓ .
- 5) For small ℓ : the schemes in one coset and two cosets are better than those in multiple cosets; when $n = 12, k = 10, 4 \leq \ell \leq 48$, the scheme in two cosets provides the lowest bandwidth; when $n = 12, k = 8, 4 \leq \ell \leq 768$, one can show that the scheme in one coset has the smallest bandwidth.
- 6) For large ℓ : the first realization in multiple cosets has better performance than the second realization in multiple cosets, but our second realization works for any number of helpers.

IV. REED-SOLOMON REPAIR SCHEMES FOR MULTIPLE ERASURES

In this section, we first present two definitions of the repair schemes for multiple erasures in a MDS code: linear repair scheme definition and dual code repair definition. We prove the equivalence of the two definitions. Then, we present two schemes for repairing multiple erasures in Reed-Solomon codes, where the evaluation points are in one coset and multiple cosets, respectively.

A. Definitions of the multiple-erasure repair

Let us assume a scalar MDS code \mathcal{E} over $\mathbb{F} = GF(q^\ell)$ has dimension k and code length n . Let a codeword be (C_1, C_2, \dots, C_n) . Without loss of generality, we assume $\{C_1, C_2, \dots, C_e\}$ are failed, $e \leq n - k$, and we repair them in the base field $\mathbb{B} = GF(q)$, where q can be any power of a prime number. We also assume that we use all the remaining $d = n - e$ nodes as helpers. The following definitions are inspired by [5] for single erasure.

Definition 1. A linear exact repair scheme for multiple erasures consists of the following.

- 1) A set of queries $Q_t \subseteq \mathbb{F}$ for each helper $C_t, e + 1 \leq t \leq n$. The helper C_t replies with $\{\gamma C_t, \gamma \in Q_t\}$.
- 2) For each failed node $C_i, i \in [e]$, a linear repair scheme that computes

$$C_i = \sum_{m=1}^{\ell} \lambda_{im} \mu_{im}, \quad (67)$$

where $\{\mu_{i1}, \mu_{i2}, \dots, \mu_{i\ell}\}$ is a basis for \mathbb{F} over \mathbb{B} and coefficients $\lambda_{im} \in \mathbb{B}$ are \mathbb{B} -linear combinations of the replies

$$\lambda_{im} = \sum_{t=e+1}^n \sum_{\gamma \in Q_t} \beta_{im\gamma t} \cdot tr_{\mathbb{F}/\mathbb{B}}(\gamma C_t), \quad (68)$$

with the coefficients $\beta_{im\gamma t} \in \mathbb{B}$. The repair bandwidth is

$$b = \sum_{t=e+1}^{\ell} rank_{\mathbb{B}}(Q_t). \quad (69)$$

In the following definition, we consider $e\ell$ dual codewords of \mathcal{E} , and index them by $i \in [e], j \in [\ell]$, denoted as $(C'_{ij1}, C'_{ij2}, \dots, C'_{ijn})$. Since they are dual codewords, we know that $\sum_{t=1}^n C_t C'_{ijt} = 0$.

Definition 2. A dual code scheme uses a set of dual codewords $\{(C'_{ij1}, C'_{ij2}, \dots, C'_{ijn}) : i \in [e], j \in [\ell]\}$ that satisfies:

- 1) The **full rank condition**: Vectors

$$V_{ij} = (C'_{ij1}, C'_{ij2}, \dots, C'_{ij_e}), i \in [e], j \in [\ell], \quad (70)$$

are linearly independent over \mathbb{B} .

- 2) The **repair bandwidth condition**:

$$b = \sum_{t=e+1}^n rank_{\mathbb{B}}(\{C'_{ijt} : i \in [e], j \in [\ell]\}). \quad (71)$$

We repair nodes $[e]$ from the linearly independent equations

$$\sum_{v=1}^e tr_{\mathbb{F}/\mathbb{B}}(C'_{ijv} C_v) = - \sum_{t=e+1}^n tr_{\mathbb{F}/\mathbb{B}}(C'_{ijt} C_t), i \in [e], j \in [\ell]. \quad (72)$$

Here we use the same condition names as the single erasure case, but in this section, they are defined for multiple erasures.

Theorem 6. Definitions 1 and 2 are equivalent.

The equivalence of Definitions 1 and 2 follows similarly as arguments in [6], except that we need to first solve e failed nodes simultaneously and then find out the form of each individual failure (67). The detailed proof of Theorem 6 is shown in Appendix B, part of which uses Lemma 4 in Section IV-B.

Remark 2. In this paper, we focus on repairing RS code and apply Theorem 6 to RS code. From [2, Thm. 4 in Ch. 10] we know that with the polynomial $p_{ij}(x) \in \mathbb{F}[x]$ for which the degrees are smaller than $n - k$, $(v_1 p_{ij}(\alpha_1), v_2 p_{ij}(\alpha_2), \dots, v_n p_{ij}(\alpha_n))$ is the dual codeword of $RS(n, k)$, where $v_i, i \in [n]$ are non-zero constants determined by the evaluation points set A . So, in RS code, Definition 2 reduces to finding polynomials $p_{ij}(x)$ with degrees smaller than $n - k$. In what follows we use $p_{ij}(\alpha_t)$ to replace the dual codeword symbol C'_{ijt} in Definition 2 for RS code. One can easily show that the constants $v_i, i \in [n]$ do not affect the ranks in the full rank condition and the repair bandwidth condition.

B. Multiple-erasure repair in one coset

There are several studies about the multiple erasures for full-length RS codes [18] and [19]. Inspired by these works, we propose our scheme for multiple erasures in one coset.

From Theorem 6, we know that finding the repair scheme for multiple erasures in RS code is equivalent to finding dual codewords (or polynomials) that satisfy the full rank condition and repair bandwidth condition. Given a basis $\{\xi_1, \xi_2, \dots, \xi_\ell\}$ for \mathbb{F} over \mathbb{B} , we define some matrices as below. They are used to help us check the two rank conditions according to Lemmas 4 and 5, whose proofs are shown in Appendices C and D, respectively. Let the evaluation points of an RS code over \mathbb{F} be $A = \{\alpha_1, \dots, \alpha_n\}$. Let $p_{ij}(x), i \in [e], j \in [\ell]$, be polynomials over \mathbb{F} , and \mathbb{B} a subfield of \mathbb{F} . Define

$$S_{it} = \begin{bmatrix} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 p_{i1}(\alpha_t)) & \cdots & \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_\ell p_{i1}(\alpha_t)) \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 p_{i2}(\alpha_t)) & \cdots & \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_\ell p_{i2}(\alpha_t)) \\ \vdots & \ddots & \vdots \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 p_{ie}(\alpha_t)) & \cdots & \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_\ell p_{ie}(\alpha_t)) \end{bmatrix}, \quad (73)$$

$$S \triangleq \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1e} \\ S_{21} & S_{22} & \cdots & S_{2e} \\ \vdots & \vdots & \ddots & \vdots \\ S_{e1} & S_{e2} & \cdots & S_{ee} \end{bmatrix}. \quad (74)$$

Lemma 4. The following two statements are equivalent:

- 1) Vectors $V_{ij} = (p_{ij}(\alpha_1), p_{ij}(\alpha_2), \dots, p_{ij}(\alpha_e)), i \in [e], j \in [\ell]$ are linearly independent over \mathbb{B} .

- 2) Matrix S in (74) has full rank.

Lemma 5. For $t \in [n]$, consider S_{it} in (73),

$$\text{rank} \begin{pmatrix} S_{1t} \\ S_{2t} \\ \vdots \\ S_{et} \end{pmatrix} = \text{rank}_{\mathbb{B}}(\{p_{ij}(\alpha_t) : i \in [e], j \in [\ell]\}). \quad (75)$$

Theorem 7. Let q be a prime number. There exists an $RS(n, k)$ code over $\mathbb{F} = GF(q^\ell)$ of which $n < q^a, q^s \leq r$ and $a|\ell$, such that the repair bandwidth for e erasures is $b \leq \frac{e\ell}{a}(n - e)(a - s)$ measured in symbols over \mathbb{B} , for e satisfying $a \geq \frac{e(e-1)}{2}(a - s)^2$.

Proof: We define the code over the field $\mathbb{F} = GF(q^\ell)$ extended by $\mathbb{E} = GF(q^a)$, where β is the primitive element of \mathbb{F} . The evaluation points are chosen to be $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\} \subseteq \mathbb{E}^*$, which is one of the cosets in Lemma 1. Without loss of generality, we assume the e failed nodes are $\{\alpha_1, \alpha_2, \dots, \alpha_e\}$. The base field is $\mathbb{B} = GF(q)$.

Construction III: We first consider the special case when $s = a - 1$. In this case, inspired by [19, Proposition 1], we choose the polynomials

$$p_{ij}(x) = \frac{\delta_i \text{tr}_{\mathbb{E}/\mathbb{B}}(\frac{\mu_j}{\delta_i}(x - \alpha_i))}{x - \alpha_i}, i \in [e], j \in [a], \quad (76)$$

where $\{\mu_1, \mu_2, \dots, \mu_a\}$ is the basis for \mathbb{E} over \mathbb{B} , and $\delta_i \in \mathbb{E}, i \in [e]$, are coefficients to be determined. From [19, Theorem 3], we know that for $a > \frac{e(e-1)}{2}$, there exists $\delta_i, i \in [e]$ such that $p_{ij}(x)$ satisfy the full rank condition: the vectors $V_{ij} = (p_{ij}(\alpha_1), p_{ij}(\alpha_2), \dots, p_{ij}(\alpha_e)), i \in [e], j \in [a]$ are linearly independent over \mathbb{B} and the repair bandwidth condition:

$$\begin{aligned} & \sum_{t=e+1}^n \text{rank}_{\mathbb{B}}(\{p_{ij}(\alpha_t) : i \in [e], j \in [a]\}) \\ & = (n - e)e - \frac{e(e-1)(q-1)}{2}. \end{aligned} \quad (77)$$

Then, let $\{\eta_1, \eta_2, \dots, \eta_{\ell/a}\}$ be a set of basis for \mathbb{F} over \mathbb{E} ; we have the $e\ell$ polynomials as $\{\eta_w p_{ij}(x) : w \in [\ell/a], i \in [e], j \in [a]\}$. Since $\{\eta_1, \eta_2, \dots, \eta_{\ell/a}\}$ are linearly independent over \mathbb{E} and for any $b_{ijw} \in \mathbb{B}, b_{ijw} p_{ij}(x) \in \mathbb{E}$, we have

$$\sum_{i,j,w} b_{ijw} \eta_w V_{ij} = 0 \iff \sum_{i,j} b_{ijw} V_{ij} = 0, \forall w \in [\frac{\ell}{a}]. \quad (78)$$

Also, we know that there does not exist nonzero $b_{ijw} \in \mathbb{B}$ that satisfies $\sum_{i,j} b_{ijw} V_{ij} = 0$, so we have that vectors $\{\eta_w V_{ij}, w \in [\ell/a], i \in [e], j \in [a]\}$ are also linearly

independent over \mathbb{B} . So, from Definition 2, we know that we can recover the failed nodes and the repair bandwidth is

$$\begin{aligned} b &= \text{rank}_{\mathbb{B}}(\{\eta_1 p_{ij}(x), \dots, \eta_{\ell/a} p_{ij}(x) : i \in [e], j \in [a]\}) \\ &= \frac{\ell}{a} \text{rank}_{\mathbb{B}}(\{p_{ij}(x), i \in [e], j \in [a]\}) \\ &= \frac{\ell}{a} \left[(n-e)e - \frac{e(e-1)(q-1)}{2} \right]. \end{aligned} \quad (79)$$

Construction IV: For $s \leq a-1$, consider the polynomials

$$p_{ij}(x) = \delta_i^{q^s-1} \mu_j \prod_{\varepsilon=1}^{q^s-1} \left(x - \left(\alpha_i - w_\varepsilon^{-1} \frac{\mu_j}{\delta_i} \right) \right), j \in [a], \quad (80)$$

where $\{\mu_1, \mu_2, \dots, \mu_a\}$ is the basis for \mathbb{E} over \mathbb{B} , $W = \{w_0 = 0, w_1, w_2, \dots, w_{q^s-1}\}$ is an s -dimensional subspace in \mathbb{E} , $s < a$, $q^s \leq r$, and $\delta_i \in \mathbb{E}$, $i \in [e]$, are coefficients to be determined.

When $x = \alpha_i$, we have

$$p_{ij}(\alpha_i) = \mu_j^{q^s} \prod_{\varepsilon=1}^{q^s-1} w_\varepsilon^{-1}. \quad (81)$$

Since $\prod_{\varepsilon=1}^{q^s-1} w_\varepsilon^{-1}$ is a constant, from Lemma 2 we have

$$\text{rank}_{\mathbb{B}}(\{p_{i1}(\alpha_i), p_{i2}(\alpha_i), \dots, p_{ia}(\alpha_i)\}) = a. \quad (82)$$

For $x \neq \alpha_i$, set $x' = \alpha_i - x$, we have

$$\begin{aligned} p_{ij}(x) &= \delta_i^{q^s-1} \mu_j \prod_{\varepsilon=1}^{q^s-1} \left(w_\varepsilon^{-1} \frac{\mu_j}{\delta_i} - x' \right) \\ &= \delta_i^{q^s-1} \mu_j \prod_{\varepsilon=1}^{q^s-1} (w_\varepsilon^{-1} x') \prod_{\varepsilon=1}^{q^s-1} \left(\frac{\mu_j}{\delta_i x'} - w_\varepsilon \right) \\ &= (\delta_i x')^{q^s} \prod_{\varepsilon=1}^{q^s-1} (w_\varepsilon^{-1}) \prod_{\varepsilon=0}^{q^s-1} \left(\frac{\mu_j}{\delta_i x'} - w_\varepsilon \right). \end{aligned} \quad (83)$$

By [35, p. 4], $g(y) = \prod_{\varepsilon=0}^{q^s-1} (y - w_\varepsilon)$ is a linear mapping from \mathbb{E} to itself with dimension $a-s$ over \mathbb{B} . Since $(\delta_i x')^{q^s} \prod_{\varepsilon=1}^{q^s-1} (w_\varepsilon^{-1})$ is a constant independent of j , we have

$$\text{rank}_{\mathbb{B}}(\{p_{i1}(x), p_{i2}(x), \dots, p_{ia}(x)\}) \leq a-s, \quad (84)$$

which means that $p_{ij}(x)$ can be written as

$$p_{ij}(x) = \delta_i^{q^s} \sum_{v=1}^{a-s} \rho_{jv} \lambda_v, \quad (85)$$

where $\{\lambda_1, \lambda_2, \dots, \lambda_{a-s}\}$ are linearly independent over \mathbb{B} , $\rho_{jv} \in \mathbb{B}$, and they are determined by δ_i, μ_j and $x - \alpha_i$.

From Lemma 4, we know that if the matrix S in (74) has full rank, then we can recover the e erasures. It is difficult to directly discuss the rank of the matrix, but assume that the polynomials above satisfy the following two conditions:

- 1) $S_{ii}, i \in [e]$ are identity matrices.
- 2) For any fixed $i \in [e]$,

$$S_{it} \cdot S_{ty} = \mathbf{0}_{\ell \times \ell}, i > t, y > t. \quad (86)$$

Then, it is easy to see that through Gaussian elimination, we can transform the matrix S^T to an upper triangular block matrix, which has identity matrices in the diagonal. Hence, S has full rank.

Here, we choose $\{\xi_1, \xi_2, \dots, \xi_\ell\}$ to be the dual basis of $\{\mu_1^{q^s} \prod_{\varepsilon=1}^{q^s-1} w_\varepsilon^{-1}, \mu_2^{q^s} \prod_{\varepsilon=1}^{q^s-1} w_\varepsilon^{-1}, \dots, \mu_\ell^{q^s} \prod_{\varepsilon=1}^{q^s-1} w_\varepsilon^{-1}\}$, so

$$\text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_i)) = \begin{cases} 0, m \neq j, \\ 1, m = j. \end{cases} \quad (87)$$

Therefore, $S_{ii}, i \in [e]$ are identity matrices. We set $\delta_1 = 1$, and recursively choose δ_i after choosing $\{\delta_1, \delta_2, \dots, \delta_{i-1}\}$ to satisfy (86). Define $\delta'_i = \delta_i^{q^s}$, and c_{mp} to be the (m, p) -th element in S_{ty} for $m, p \in [a]$. (86) can be written as

$$\begin{aligned} & \sum_{m=1}^a c_{mp} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_t)) \\ &= \sum_{m=1}^a c_{mp} \sum_{v=1}^{a-s} b_{jv} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m \delta'_i \lambda_v) \\ &= 0, \forall j \in [a], \end{aligned} \quad (88)$$

where $\lambda_v, v \in [a-s]$, are determined by δ_i, μ_j and $\alpha_t - \alpha_i$. Equation (88) is satisfied if

$$\sum_{m=1}^a c_{mp} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m \delta'_i \lambda_v) = 0, v \in [a-s], p \in [a]. \quad (89)$$

As a special case of Lemma 5, we have

$$\text{rank}(S_{ty}) = \text{rank}_{\mathbb{B}}(\{p_{tj}(\alpha_y), j \in [\ell]\}). \quad (90)$$

Then, from (84) we know that the rank of S_{ty} is at most $a-s$, which means in (89) we only need to consider p corresponding to the independent $a-s$ columns of S_{ty} . So, (89) is equivalent to $(a-s)^2$ linear requirements. For $\delta'_i \in \mathbb{E}$, we can view it as a unknowns over \mathbb{B} , and we have

$$\frac{(2e-i)(i-1)}{2} (a-s)^2 \leq \frac{e(e-1)}{2} (a-s)^2 \quad (91)$$

linear requirements over \mathbb{B} according to (86). Also knowing δ'_i , we can solve $\delta_i = \delta_i^{q^s} = \delta'_i^{q^{e-s}}$. Therefore, we can find appropriate $\{\delta_1, \delta_2, \dots, \delta_e\}$ to make matrix S full rank when

$$a \geq \frac{e(e-1)}{2} (a-s)^2. \quad (92)$$

Then, let $\{\eta_1, \eta_2, \dots, \eta_{\ell/a}\}$ be a basis for \mathbb{F} over \mathbb{E} , we have the $e\ell$ polynomials as $\{\eta_w p_{ij}(x), w \in [\ell/a], i \in [e], j \in [a]\}$. Similar to Construction III, we know that vectors $\{\eta_w V_{ij}, w \in [\ell/a], i \in [e], j \in [a]\}$ are linearly

independent over \mathbb{B} . Therefore, we can recover the failed nodes and the repair bandwidth is

$$\begin{aligned} b &= \text{rank}_{\mathbb{B}}(\{\eta_1 p_{ij}(x), \dots, \eta_{\ell/a} p_{ij}(x) : i \in [e], j \in [a]\}) \\ &= \frac{\ell}{a} \text{rank}_{\mathbb{B}}(\{p_{ij}(x) : i \in [e], j \in [a]\}) \\ &\leq \frac{e\ell}{a} (n-e)(a-s). \end{aligned} \quad (93)$$

Thus, the proof is completed. \blacksquare

In our scheme, we have constructions for arbitrary a, s , such that $a \mid \ell, s \leq a-1$, while the existing schemes in [18] and [19] mainly considered the special case $\ell = a$. It should be noted that the scheme in [19] can also be used in the case of $s = a-1$ over \mathbb{E} with repair bandwidth $(n-e)e - \frac{e(e-1)(q-1)}{2}$. And, with ℓ/a copies of the code, it can also reach the same repair bandwidth of our scheme. However, by doing so, the code is a vector code, but our scheme constructs a scalar code.

C. Multiple-erasure repair in multiple cosets

Recall that the scheme in Theorem 5 for a single erasure is a small sub-packetization code with small repair bandwidth for any number of helpers. When there are e erasures and d helpers, $e \leq n-k, k \leq d \leq n-e$, we can recover the erasures one by one using the d helpers. However, inspired by [22], the repaired nodes can be viewed as additional helpers and thus we can reduce the total repair bandwidth. Finally, for every helper, the transmitted information for different failed nodes has some overlap, resulting in a further bandwidth reduction.

The approach we take is similar to that of Section III-C. We take an original code and extend it to a new code with evaluation points as in (44). If a helper is in the same coset as any failed node, it transmits naively its entire data; otherwise, it transmits the same amount as the scheme in the original code. After the extension, the new construction decreases the sub-packetization size for fixed n , and the bandwidth is only slightly larger than the original code.

The location of the e erasures are described by $h_i, i \in [e]$, where $0 \leq h_i \leq e, h_1 \geq h_2 \geq \dots \geq h_e, \sum_{i=1}^e h_i = e$. We assume the erasures are located in h_1 cosets, and after removing one erasure in each coset, the remaining erasures are located in h_2 cosets. Then, for the remaining erasures, removing one in each coset, we get the rest of erasures in h_3 cosets, and so on. Figure 3 also shows the erasure locations described above.

In our scheme, we first repair h_1 failures, one from each of the h_1 cosets. Then, for $2 \leq i \leq e$, we repeat the following: After repairing h_1, h_2, \dots, h_{i-1} failures, we view these repaired nodes as helpers and repair next h_i failures, one from each of the h_i cosets.

The repair bandwidth of the scheme is showed in the following theorem.

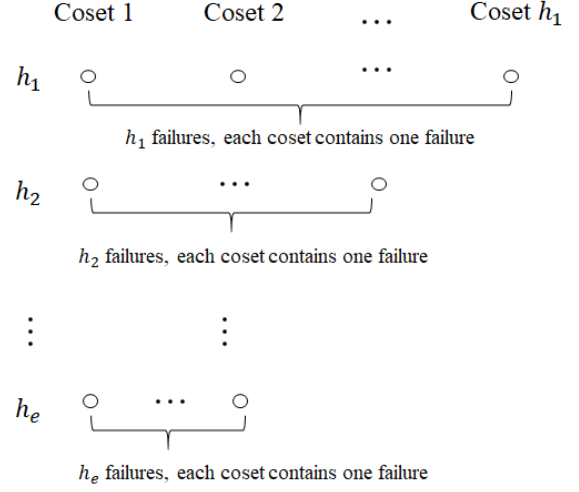


Fig. 3. Location of the erasures. e erasures are located in h_1 cosets. For $i \in [e]$, we set $0 \leq h_i \leq e, h_1 \geq h_2 \geq \dots \geq h_e$ and $\sum_{i=1}^e h_i = e$.

Theorem 8. Let q be a prime number. There exists an $RS(n, k)$ code over $\mathbb{F} = GF(q^\ell)$ for which $\ell = asq_1q_2 \dots q_{e-1}$, where q_i is the i -th prime number that satisfies $s \mid (q_i - 1), s = (n-k)!$ and a is an integer. For e erasures and d helpers, $e \leq n-k, k \leq d \leq n-e$, the average repair bandwidth measured in symbols over \mathbb{B} is

$$\begin{aligned} b &\leq \frac{d\ell}{(n-e)} \left[(h_1(q^a - 1) - e) \right. \\ &\quad \left. + (n - h_1(q^a - 1)) \sum_{i=1}^e \frac{h_i}{d - k + \sum_{v=1}^i h_v} \right], \end{aligned} \quad (94)$$

where $h_i, i \in [e]$ are the parameters that define the location of erasures in Fig. 3.

Proof: We first prove the case when a and ℓ' are relatively prime using Lemma 3, the case when a and ℓ' are not necessarily relatively prime are proved in Appendix A. We use the code in [22] as the original code. Let $\mathbb{F}_q(\alpha)$ be the field obtained by adjoining α to the base field $\mathbb{B} = GF(q)$. Similarly let $\mathbb{F}_q(\alpha_1, \alpha_2, \dots, \alpha_n)$ be the field for adjoining multiple elements. Let α_i be an element of order q_i over \mathbb{B} and h be the number of erasures in the original code. The original code is defined in the field $\mathbb{F}' = GF(q^{\ell'}) = GF(q^{sq_1q_2 \dots q_{n'}})$, which is the degree- s of extension of $\mathbb{F}_q(\alpha_1, \alpha_2, \dots, \alpha_{n'})$. The evaluation points are $A' = \{\alpha_1, \alpha_2, \dots, \alpha_{n'}\}$. The subfield $\mathbb{F}'_{[h]}$ is defined as $\mathbb{F}'_{[h]} = \mathbb{F}_q(\alpha_j, j = h+1, h+2, \dots, n')$, and \mathbb{F}'_i is defined as $\mathbb{F}_q(\alpha_j, j \neq i, j \in [n'])$.

In the original code, we assume without loss of generality that there are h failed nodes $f'(\alpha_1), f'(\alpha_2), \dots, f'(\alpha_h)$. Consider the polynomials for failed node $f'(\alpha_i), 1 \leq i \leq h$, as

$$p'_{ij}(x) = x^{j-1} g'_i(x), j \in [s_i], x \in \mathbb{F}', \quad (95)$$

where

$$g'_i(x) = \prod_{\alpha \in A' / (R' \cup \{\alpha_i, \alpha_{i+1}, \dots, \alpha_h\})} (x - \alpha), x \in \mathbb{F}', \quad (96)$$

for $R' \subseteq A', |R'| = d'$ being the set of helpers. The set of repair polynomials are $\{\eta_{it}p'_{ij}(x), i \in [h], j \in [s_i], t \in [\frac{sq_i}{s_i}]\}$, where $\eta_{it} \in \mathbb{F}'$ are constructed in [22] to ensure that $\{\eta_{it}p'_{i1}(\alpha_i), \eta_{it}p'_{i2}(\alpha_i), \dots, \eta_{it}p'_{is_i}(\alpha_i)\}$ forms the basis for \mathbb{F}' over \mathbb{F}'_i .

Then, the failed nodes are repaired one by one from

$$\begin{aligned} & tr_{\mathbb{F}'/\mathbb{F}'_i}(v_{\alpha_i}\eta_{it}p'_{ij}(\alpha_i)f'(\alpha_i)) \\ &= - \sum_{\epsilon=1, \epsilon \neq i}^n tr_{\mathbb{F}'/\mathbb{F}'_i}(v_{\epsilon}\eta_{it}p'_{ij}(\alpha_{\epsilon})f'(\alpha_{\epsilon})). \end{aligned} \quad (97)$$

For $x \notin R' \cup \{\alpha_i, \alpha_{i+1}, \dots, \alpha_h\}$, $p'_{ij}(x) = 0$ and no information is transmitted. Once $f'(\alpha_i)$ is recovered, it is viewed as a new helper for the failures $i+1, i+2, \dots, h$.

Since $\mathbb{F}'_{[h]} \leq \mathbb{F}'_i$, the information transmitted from the helper α_{ϵ} can be represented as

$$\begin{aligned} & tr_{\mathbb{F}'/\mathbb{F}'_i}(v_{\epsilon}\eta_{it}p'_{ij}(\alpha_{\epsilon})f'(\alpha_{\epsilon})) \\ &= tr_{\mathbb{F}'/\mathbb{F}'_i} \left(\xi'_{im} \sum_{m=1}^{q'_i} tr_{\mathbb{F}'_i/\mathbb{F}'_{[h]}}(v_{\epsilon}\eta_{it}\xi_{im}p'_{ij}(\alpha_{\epsilon})f'(\alpha_{\epsilon})) \right) \\ &= \sum_{m=1}^{q'_i} \xi'_{im} tr_{\mathbb{F}'/\mathbb{F}'_{[h]}}(v_{\epsilon}\eta_{it}\xi_{im}p'_{ij}(\alpha_{\epsilon})f'(\alpha_{\epsilon})), \end{aligned} \quad (98)$$

where $q'_i = \frac{q_1 q_2 \dots q_h}{q_i}$, $\{\xi_{i1}, \xi_{i2}, \dots, \xi_{iq'_i}\}$ and $\{\xi'_{i1}, \xi'_{i2}, \dots, \xi'_{iq'_i}\}$ are the dual basis for \mathbb{F}'_i over $\mathbb{F}'_{[h]}$. We used the fact that $tr_{\mathbb{F}'/\mathbb{F}'_i}(tr_{\mathbb{F}'_i/\mathbb{F}'_{[h]}}(\cdot)) = tr_{\mathbb{F}'/\mathbb{F}'_{[h]}}(\cdot)$, for $\mathbb{F}'_{[h]} \leq \mathbb{F}'_i \leq \mathbb{F}'$.

The original code satisfies the full rank condition for every $i \in [h]$, and each helper α_{ϵ} transmits [22]

$$\begin{aligned} & rank_{\mathbb{F}'_{[h]}} \left(\{\eta_{it}\xi_{im}p'_{ij}(\alpha_{\epsilon}) : \right. \\ & \left. i \in [h], j \in [s_i], t \in [\frac{sq_i}{s_i}], m \in [q'_i]\} \right) \\ &= rank_{\mathbb{F}'_{[h]}} \left(\{\eta_{it}\xi_{im} : i \in [h], t \in [\frac{sq_i}{s_i}], m \in [q'_i]\} \right) \\ &= \frac{h\ell'}{(d' - k' + h) \prod_{v=h+1}^{n'} p_v} \end{aligned} \quad (99)$$

symbols over $\mathbb{F}'_{[h]}$, which achieves the MSR bound.

In our new code, we extend the field to $\mathbb{F} = GF(q^{\ell})$, $\ell = a\ell'$, by adjoining an order- a element α_{n+1} to \mathbb{F} . We set $d - k = d' - k'$. The new evaluation points consist of $A = \{\alpha_1\mathbb{E}^*, \alpha_2\mathbb{E}^*, \dots, \alpha'_n\mathbb{E}^*\}$, $\mathbb{E} = GF(q^a) = \mathbb{F}_q(\alpha_{n+1})$. The subfield $\mathbb{F}_{[h]}$ is defined by adjoining α_{n+1} to $\mathbb{F}'_{[h]}$, and \mathbb{F}_i is defined as $\mathbb{F}_q(\alpha_j, j \neq i, j \in [n+1])$.

Assume first that each coset contains at most one failure, and there are h failures in total. We assume without loss of generality that the evaluation points of the h

failed nodes are in $\{\alpha_1\mathbb{E}^*, \alpha_2\mathbb{E}^*, \dots, \alpha_h\mathbb{E}^*\}$, and they are $\alpha_1\gamma_1, \alpha_2\gamma_2, \dots, \alpha_h\gamma_h$ for some $\gamma_w \in \mathbb{E}, w \in [h]$. Let the set of helpers be $R \subseteq A, |R| = d$. We define the polynomials

$$p_{ij}(x) = x^{j-1}g_i(x), j \in [s_i], x \in \mathbb{F}, \quad (100)$$

where

$$g_i(x) = \prod_{\alpha \in A / \{R \cup \{\alpha_i\gamma_i, \alpha_{i+1}\gamma_{i+1}, \dots, \alpha_h\gamma_h\}\}} (x - \alpha), x \in \mathbb{F}. \quad (101)$$

The set of repair polynomials are $\{\eta_{it}p_{ij}(x), i \in [h], j \in [s_i], t \in [\frac{sq_i}{s_i}]\}$, where $\eta_{it} \in \mathbb{F}'$ are the same as the original construction. We use field \mathbb{F}_i as the base field for the repair.

$$\begin{aligned} & tr_{\mathbb{F}/\mathbb{F}_i}(v_{\alpha_i\gamma_i}\eta_{it}p_{ij}(\alpha_i\gamma_i)f(\alpha_i\gamma_i)) \\ &= - \sum_{\alpha \in A, \alpha \neq \alpha_i\gamma_i} tr_{\mathbb{F}/\mathbb{F}_i}(v_{\alpha}\eta_{it}p_{ij}(\alpha)f(\alpha)). \end{aligned} \quad (102)$$

If $x \in R \cup \{\alpha_i\gamma_i, \alpha_{i+1}\gamma_{i+1}, \dots, \alpha_h\gamma_h\}$, $p_{ij}(x) = 0$ and no information is transmitted. Next, we consider all other nodes.

If $x = \alpha_i\gamma$ for some $\gamma \in \mathbb{E}^*$, we have

$$p_{ij}(x) = \gamma^{j-1}\alpha_i^{j-1}g_i(x). \quad (103)$$

Since $\eta_{it}, \alpha_i \in \mathbb{F}'$ and $g_i(x)$ is a constant independent of j , we have

$$\begin{aligned} & rank_{\mathbb{F}_i}(\{\eta_{it}p_{i1}(x), \dots, \eta_{it}p_{is_i}(x) : t \in [\frac{sq_i}{s_i}]\}) \\ &= rank_{\mathbb{F}_i}(\{\eta_{it}, \dots, \eta_{it}\alpha_i^{s_i-1} : t \in [\frac{sq_i}{s_i}]\}) \\ &= rank_{\mathbb{F}_i}(\{\eta_{it}p'_{i1}(\alpha_i), \dots, \eta_{it}p'_{is_i}(\alpha_i) : t \in [\frac{sq_i}{s_i}]\}) \end{aligned} \quad (104)$$

which indicates the full rank. Note that the last equation follows from Lemma 3. As a result we can recover the failed nodes and each helper in the cosets containing the failed nodes transmit ℓ symbols in \mathbb{B} .

For $x = \alpha_{\epsilon}\gamma, \epsilon > h$, since $\mathbb{F}_{[h]}$ is a subfield of \mathbb{F}_i and from Lemma 3 we know that $\{\xi_{i1}, \xi_{i2}, \dots, \xi_{iq'_i}\}$ and $\{\xi'_{i1}, \xi'_{i2}, \dots, \xi'_{iq'_i}\}$ are also the dual basis for \mathbb{F}_i over $\mathbb{F}_{[h]}$, then, similar to (98), we have

$$\begin{aligned} & tr_{\mathbb{F}/\mathbb{F}_i}(v_{\alpha}\eta_{it}p_{ij}(x)f(x)) \\ &= \sum_{m=1}^{q'_i} \xi'_{im} tr_{\mathbb{F}/\mathbb{F}_{[h]}}(v_{\epsilon}\eta_{it}\xi_{im}p_{ij}(x)f(x)). \end{aligned} \quad (105)$$

Using the fact that $g_i(x)$ is a constant independent of j , $x \in \mathbb{F}_{[h]}$ and $\eta_{it}\xi_{im} \in \mathbb{F}'$, from Lemma 3 we know that

$$\begin{aligned}
& \text{rank}_{\mathbb{F}_{[h]}} \left(\left\{ \eta_{it}\xi_{im} p_{ij}(x) : \right. \right. \\
& \quad \left. \left. i \in [h], j \in [s_i], t \in \left[\frac{sq_i}{s_i} \right], m \in [q'_i] \right\} \right) \\
&= \text{rank}_{\mathbb{F}_{[h]}} \left(\left\{ \eta_{it}\xi_{im} : i \in [h], t \in \left[\frac{sq_i}{s_i} \right], m \in [q'_i] \right\} \right) \\
&= \text{rank}_{\mathbb{F}'_{[h]}} \left(\left\{ \eta_{it}\xi_{im} : i \in [h], t \in \left[\frac{sq_i}{s_i} \right], m \in [q'_i] \right\} \right) \\
&= \text{rank}_{\mathbb{F}'_{[h]}} \left(\left\{ \eta_{it}\xi_{im} p'_{ij}(\alpha_\epsilon) : \right. \right. \\
& \quad \left. \left. i \in [h], j \in [s_i], t \in \left[\frac{sq_i}{s_i} \right], m \in [q'_i] \right\} \right) \\
&= \frac{h\ell'}{(d-k+h) \prod_{v=h+1}^{n'} q_v}, \tag{106}
\end{aligned}$$

where the last equality follows from (99) and $d' - k' = d - k$. So, each helper in the other cosets transmits $\frac{h\ell}{d-k+h}$ symbols over \mathbb{B} .

Using the above results, we calculate the repair bandwidth in two steps.

Step 1. We first repair h_1 failures, one from each of the h_1 cosets. From (104), we know that in the h_1 cosets containing the failed nodes, we transmit ℓ symbols over \mathbb{B} . By (106), for each helper in other cosets, we transmit $\frac{h_1\ell}{d-k+h_1}$ symbols over \mathbb{B} .

Step 2. For $2 \leq i \leq e$, repeat the following. After repairing h_1, h_2, \dots, h_{i-1} failures, these nodes can be viewed as helpers for repairing next h_i failures, one from each of the h_i cosets. So, we have $d + \sum_{v=1}^{i-1} h_v$ helpers for the h_i failures.

For the helpers in the h_1 cosets containing the failed nodes, we already transmit ℓ symbols over \mathbb{B} in Step 1 and no more information needs to be transmitted. For each helper in other cosets, we transmit $\frac{h_i\ell}{d-k+\sum_{v=1}^i h_v}$ symbols over \mathbb{B} .

Thus, we can repair all the failed nodes. The repair bandwidth can be calculated as (94). ■

Suppose that e failures are to be recovered. Compared to the naive strategy which always uses d helpers to repair the failures one by one, our scheme gets a smaller repair bandwidth since the recovered failures are viewed as new helpers and we take advantage of the overlapped symbols for repairing different failures similar to [22].

In the case when $n \gg e(q^a - 1)$, or when we arrange nodes with correlated failures in different cosets, we can assume that all the erasures are in different cosets, $h_1 = e, h_2 = h_3 = \dots = h_e = 0$. For example, if correlated failures tend to appear in the same rack in a data center, we can assign each node in the rack to a different coset. Under such conditions, we simplify the repair bandwidth as

$$b \leq \frac{d}{n-e} \frac{e\ell}{d-k+e} (n-e + (d-k)(q^a - 2)). \tag{107}$$

Indeed, one can examine the expression of (94). With the constraint that $\sum_{i=1}^e h_i = e$, the first term $h_1(q^a - 1) - e$ is an increasing function of h_1 and the second term $(n-h_1(q^a-1)) \sum_{i=1}^e \frac{h_i}{d-k+\sum_{v=1}^i h_v}$ is a decreasing function of h_1 . Under the assumption that n is large, the second term dominates, and increasing h_1 reduces the total repair bandwidth b . Namely, $h_1 = e$ corresponds to the lowest bandwidth for large code length.

In particular, when $d = n - e, h_1 = e$, we have

$$b = \frac{e\ell}{n-k} (n-e) + \frac{e\ell}{n-k} (n-k-e)(q^a - 2), \tag{108}$$

where the second term is the extra repair bandwidth compared with the MSR bound.

D. Numerical evaluations and discussions

In this subsection, we compare our schemes for multiple erasures with previous results, including separate repair and schemes in [19] and [22].

We first demonstrate that repairing multiple erasures simultaneously can save repair bandwidth compared to repairing erasures separately. Let us assume e failures happen one by one, and the rest of $n - 1$ nodes are available as helpers initially when the first failure occurs. We can either repair each failure separately using $n - 1$ helpers, or wait for e failures and repair all of them simultaneously with $n - e$ helpers. Table III shows the comparison. For our scheme in one coset, separate repair needs a repair bandwidth of $\frac{e\ell}{a}(n-1)(a-s)$ symbols in \mathbb{B} , simultaneous a repair requires bandwidth of $\frac{e\ell}{a}(n-e)(a-s)$. For our scheme in multiple cosets, we can repair the failures separately by $n - 1$ helpers with the bandwidth of $\frac{e\ell}{n-k}[n-1 + (n-k-1)(q^a - 2)]$, and with simultaneous repair we can achieve the bandwidth of $\frac{e\ell}{n-k}[n-e + (n-k-e)(q^a - 2)]$. One can see that in both constructions, simultaneous repair outperforms separate repair.

Nest we compare our scheme for multiple erasures with the existing schemes. Figure 4 shows the normalized repair bandwidth for different schemes when $n = 16, k = 8, e = 2, q = 2$. Table IV shows the comparison when $n = 64, k = 32, e = 2, q = 2$. We make the following observations:

- 1) For fixed (n, k) and our scheme with multiple cosets, we use the parameter a to adjust the sub-packetization size. From Theorem 8, we know that $\ell \approx a \cdot \left(\frac{n}{q^a-1}\right)^{\left(\frac{1}{q^a-1}\right)}$, which means that increasing a will decrease the sub-packetization ℓ . In our schemes with one coset and two cosets, the parameter a is determined by the code length n , so increasing ℓ will not change a or the normalized repair bandwidth. When $q = 2$, our code with $a = 1$ coincides with that of [22].

TABLE III
Repair bandwidth of different schemes for e erasures.

	repair bandwidth	number of helpers
Single-erasure repair in one coset (separate repair)	$\frac{e\ell}{a}(n-1)(a-s)$	$n-1$
Multiple-erasure repair in one coset (simultaneous repair)	$\frac{e\ell}{a}(n-e)(a-s)$	$n-e$
Single-erasure repair in multiple cosets (separate repair)	$\frac{e\ell}{n-k}[n-1+(n-k-1)(q^a-2)]$	$n-1$
Multiple-erasure repair in multiple cosets (simultaneous repair)	$\frac{e\ell}{n-k}[n-e+(n-k-e)(q^a-2)]$	$n-e$

TABLE IV
Normalized repair bandwidth($\frac{b}{(n-e)\ell}$) for different schemes when $n = 64, k = 32, e = 2, q = 2$. \circ can be also achieved by Scheme 1 in [19] and $*$ is also achieved by [22].

	$\ell = 6$	$\ell = 7$	$\ell = 8$	$\ell = 9$...	$\ell = 3.6 \times 10^6$	$\ell = 3.3 \times 10^{11}$	$\ell = 3.9 \times 10^{115}$
Normalized bandwidth for Scheme 1 in [19]	0.42	0.50	0.52	0.52	...	0.52	0.52	0.52
Normalized bandwidth for our scheme in one coset	0.49 \circ	0.49	0.49	0.49	...	0.49	0.49	0.49
Normalized bandwidth for our scheme in multiple cosets						0.52	0.48	0.0625*

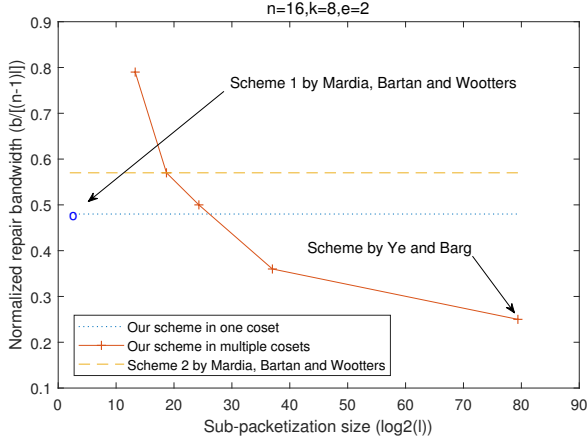


Fig. 4. Comparison of the schemes, $q = 2, n = 16, k = 8, e = 2$, x-axis is the log scale sub-packetization size, y-axis is the normalized repair bandwidth. The scheme by Mardia, Bartan and Wootters is in [19], and the scheme by Ye and Barg is in [22]

- 2) For small ℓ and full-length code ($\ell = \log_2 n$), the scheme in [19] has the smallest normalized repair bandwidth. (Our scheme in one coset also achieves the same point as Scheme 1 in [19] when $\ell = \log_2 n$.)
- 3) When ℓ grows larger ($4 < \ell < 2.1 \times 10^7$ in Figure 4, $6 < \ell < 3.3 \times 10^{11}$ in Table IV), our scheme in one coset has the smallest repair bandwidth.
- 4) For extremely large ℓ ($\ell \geq 2.1 \times 10^7$ in Figure 4, $\ell \geq 3.3 \times 10^{11}$ in Table IV), our scheme in multiple cosets has the smallest repair bandwidth.
- 5) The scheme in [22] also achieves one point in both Figure 4 and Table IV, which can be viewed as a special case of our scheme in multiple cosets.

V. CONCLUSION

In this paper, we designed three Reed-Solomon code repair schemes to provide a tradeoff between the sub-packetization size and the repair bandwidth. Our schemes choose the evaluation points of the Reed-Solomon code from one, two, or multiple cosets of the multiplicative group of the underlying finite field. For a single erasure, when the sub-packetization size is large, the scheme in multiple cosets has better performance, it approaches the MSR bound. When sub-packetization size is small, the scheme in one coset has advantages in repair bandwidth. The scheme in two cosets has smaller repair bandwidth with certain parameters in between the other two cases. For multiple erasures, our scheme in one coset has constructions for arbitrary redundancy $n - k$ and our scheme in multiple cosets reduced the sub-packetization size of an MSR code. The two schemes together provided a set of tradeoff points and we observe similar tradeoff characteristics as in the single erasure case. In spite of several tradeoff points we provided in this paper, the dependence of the sub-packetization size versus the repair bandwidth is still an open question.

APPENDIX A

PROOF OF SCHEMES IN MULTIPLE COSETS FOR THE CASE OF ARBITRARY a AND ℓ'

In this section, we first introduce a lemma similar to Lemma 3 that does not require a and ℓ' to be relatively prime. By applying this lemma, our constructions in multiple cosets for single and multiple erasures can be generalized when a and ℓ' are arbitrary integers.

We note that a finite field $\mathbb{F} = GF(q^\ell)$ is also a vector space over $GF(q)$. Let \mathbb{E} be a subspace of \mathbb{F} . Define the subspace spanned by a set of elements $\{\gamma_1, \gamma_2, \dots, \gamma_i\} \subseteq$

\mathbb{F} over \mathbb{E} as $\text{span}_{\mathbb{E}}\{\gamma_1, \gamma_2, \dots, \gamma_i\} \triangleq \{\sum_{j=1}^i b_j \gamma_j : b_j \in \mathbb{E}\}$. The rank $\text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_i\})$ is defined to be the cardinality of a maximal subset of $\{\gamma_1, \gamma_2, \dots, \gamma_i\}$ that is linearly independent over \mathbb{E} .

Lemma 6. Let $\mathbb{B} = GF(q), \mathbb{F}' = GF(q^{\ell'}), \mathbb{F} = GF(q^{\ell}), \ell = a\ell'$, and q be any power of a prime number. Define the subspace $\mathbb{E} = \text{span}_{\mathbb{B}}\{\beta_1, \beta_2, \dots, \beta_a\}$, where $\{\beta_1, \beta_2, \dots, \beta_a\}$ is a basis for \mathbb{F} over \mathbb{F}' . For any set of $\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\} \subseteq \mathbb{F}' \leq \mathbb{F}$, we have

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) \\ &= \text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}). \end{aligned} \quad (109)$$

Proof: Assume $\text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) = c$ and without loss of generality, $\{\gamma_1, \gamma_2, \dots, \gamma_c\}$ are linearly independent over \mathbb{B} . Then, we can construct $\{\gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\} \subseteq \mathbb{F}'$ to make $\{\gamma_1, \gamma_2, \dots, \gamma_c, \gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\}$ form a basis for \mathbb{F}' over \mathbb{B} .

Since $\{\beta_1, \beta_2, \dots, \beta_a\}$ is the basis for \mathbb{F} over \mathbb{F}' , we know that $\{\beta_i \gamma_1, \beta_i \gamma_2, \dots, \beta_i \gamma_c, \beta_i \gamma'_{c+1}, \beta_i \gamma'_{c+2}, \dots, \beta_i \gamma'_{\ell'} : i \in [a]\}$ is the basis for \mathbb{F} over \mathbb{B} . Then, we have $\mathbb{F} = \text{span}_{\mathbb{B}}\{\gamma_1, \gamma_2, \dots, \gamma_c, \gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\}$, namely, $\{\gamma_1, \gamma_2, \dots, \gamma_c, \gamma'_{c+1}, \gamma'_{c+2}, \dots, \gamma'_{\ell'}\}$ is a basis for \mathbb{F} over \mathbb{E} , hence $\{\gamma_1, \gamma_2, \dots, \gamma_c\}$ are linearly independent over \mathbb{E} ,

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) \\ & \geq c \\ & = \text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}). \end{aligned} \quad (110)$$

Since $\mathbb{B} \subseteq \mathbb{E}$, we also have

$$\begin{aligned} & \text{rank}_{\mathbb{E}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}) \\ & \leq \text{rank}_{\mathbb{B}}(\{\gamma_1, \gamma_2, \dots, \gamma_{\ell'}\}). \end{aligned} \quad (111)$$

The proof is completed. \blacksquare

For the schemes in multiple cosets when a and ℓ' are not relatively prime, we just use the subspace $\mathbb{E} = \text{span}_{\mathbb{B}}\{\beta_1, \beta_2, \dots, \beta_a\}$ to replace the subfield $GF(q^a)$. We denote by $\mathbb{E}^* = \mathbb{E} \setminus \{0\}$ for the subspace \mathbb{E} . The evaluation points of the new code are $\{\gamma \mathbb{E}^* : \gamma \in A'\}$ where $A' \subseteq \mathbb{F}'$ is the set of evaluation points for the original code. In the proofs, we use Lemma 6 instead of Lemma 3. For example, from Lemma 6 we know that the new evaluation points are all distinct if the elements in A' are linearly independent over \mathbb{B} .

APPENDIX B PROOF OF THEOREM 6

In this section, we prove the equivalence of Definitions 1 and 2. We first show that the dual code scheme in Definition 2 reduces to a linear repair scheme as in Definition 1 in Lemma 7. Then, we show that Definition 1 reduces to Definition 2 in Lemma 8 and Lemma 9.

Lemma 7. The dual code scheme can be reduced to the linear repair scheme in Definition 1.

Proof: In the dual code scheme, we repair nodes $[e]$ from the linearly independent equations

$$\sum_{v=1}^e \text{tr}_{\mathbb{F}/\mathbb{B}}(C'_{ijv} C_v) = - \sum_{t=e+1}^n \text{tr}_{\mathbb{F}/\mathbb{B}}(C'_{ijt} C_t), i \in [e], j \in [\ell]. \quad (112)$$

Here, C'_{ijt} can be written as

$$C'_{ijt} = \sum_{m=1}^{\ell} \xi'_m \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m C'_{ijt}), \quad (113)$$

where $\{\xi_1, \xi_2, \dots, \xi_{\ell}\}$ and $\{\xi'_1, \xi'_2, \dots, \xi'_{\ell}\}$ are the dual basis for \mathbb{F} over \mathbb{B} . Then, we can rewrite (112) in matrix form as

$$\begin{aligned} & \sum_{v=1}^e S_{iv} \begin{bmatrix} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_1 C_v) \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_2 C_v) \\ \vdots \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_{\ell} C_v) \end{bmatrix} \\ &= - \sum_{t=e+1}^n S_{it} \begin{bmatrix} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_1 C_t) \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_2 C_t) \\ \vdots \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_{\ell} C_t) \end{bmatrix}, i \in [e], \end{aligned} \quad (114)$$

where $S_{it} \in \mathbb{B}^{\ell \times \ell}$ is called the repair matrix defined as

$$S_{it} = \begin{bmatrix} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 C'_{i1t}) & \cdots & \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_{\ell} C'_{i1t}) \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 C'_{i2t}) & \cdots & \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_{\ell} C'_{i2t}) \\ \vdots & \ddots & \vdots \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 C'_{iet}) & \cdots & \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_{\ell} C'_{iet}) \end{bmatrix}, \quad (115)$$

Let

$$X_w \triangleq \begin{bmatrix} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_1 C_w) \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_2 C_w) \\ \vdots \\ \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_{\ell} C_w) \end{bmatrix}, w \in [n]. \quad (116)$$

We want to solve $X_i, i \in [e]$, which can be used to recover the e failed nodes C_1, C_2, \dots, C_e . Define matrix S as

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1e} \\ S_{21} & S_{22} & \cdots & S_{2e} \\ \vdots & \vdots & \ddots & \vdots \\ S_{e1} & S_{e2} & \cdots & S_{ee} \end{bmatrix}. \quad (117)$$

Then, (114) can be represented as

$$S \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_e \end{bmatrix} = - \sum_{t=e+1}^n \begin{bmatrix} S_{1t} \\ S_{2t} \\ \vdots \\ S_{et} \end{bmatrix} X_t. \quad (118)$$

Thus, from Lemma 4 we know that if the full rank condition satisfies¹, S has full rank so we can solve $X_i, i \in [e]$. Then, $C_i, i \in [e]$ can be repaired from

$$C_i = \sum_{m=1}^{\ell} \xi_m \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_m C_i), \quad (119)$$

Now, set $\mu_{im} = \xi_m$, we get $\lambda_{im} = \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi'_m C_i)$, which can be solved from (118). Note that the right side of (118) is equal to the right side of (112), we get the queries $Q_t = \{C'_{ijt}, i \in [e], j \in [\ell]\}$ and the coefficients $\beta_{im\gamma t}$ come from matrix S .

Then, we can get the repair bandwidth condition:

$$b = \sum_{t=e+1}^n \text{rank}_{\mathbb{B}}(\{C'_{ijt} : i \in [e], j \in [\ell]\}) = \sum_{t=e+1}^{\ell} \text{rank}_{\mathbb{B}}(Q_t). \quad (120)$$

Lemma 8. A linear repair scheme in Definition 1 can be represented in the form below:

$$\mu'_{ij} C_i = \sum_{t=e+1}^n \theta_{ijt} C_t, i \in [e], j \in [\ell], \quad (121)$$

where $\{\mu'_{i1}, \mu'_{i2}, \dots, \mu'_{i\ell}\}$ is the dual basis of $\{\mu_{i1}, \mu_{i2}, \dots, \mu_{i\ell}\}$, and $\theta_{ijt} \in \text{span}_{\mathbb{B}}(Q_t)$, $e+1 \leq t \leq n, i \in [e], j \in [\ell]$ are some coefficients in \mathbb{F} . The repair bandwidth is

$$b = \sum_{t=e+1}^n \text{rank}_{\mathbb{B}}(\{\theta_{ijt} : i \in [e], j \in [\ell]\}). \quad (122)$$

Proof: By (67) and (68) we have

$$\sum_{t=e+1}^n \sum_{\gamma \in Q_t} \text{tr}_{\mathbb{F}/\mathbb{B}}(\beta_{ij\gamma t} \cdot \gamma C_t) = \lambda_{ij} = \text{tr}_{\mathbb{F}/\mathbb{B}}(\mu'_{ij} C_i). \quad (123)$$

Set $\theta_{ijt} = \sum_{\gamma \in Q_t} \beta_{ij\gamma t} \cdot \gamma$. Then, we have $\theta_{ijt} \in \text{span}_{\mathbb{B}}(Q_t)$. Hence,

$$\begin{aligned} & \sum_{t=e+1}^n \text{tr}_{\mathbb{F}/\mathbb{B}}(\theta_{ijt} C_t) \\ &= \sum_{t=e+1}^n \text{tr}_{\mathbb{F}/\mathbb{B}}\left(\sum_{\gamma \in Q_t} \beta_{ij\gamma t} \cdot \gamma C_t\right) \\ &= \text{tr}_{\mathbb{F}/\mathbb{B}}(\mu'_{ij} C_i). \end{aligned} \quad (124)$$

Equations (123) and (124) hold for all $f \in \mathbb{F}[x]$. Since the RS code is a linear code, they also hold for $\delta_m \cdot f \in \mathbb{F}[x]$

¹We use Lemma 4 while in Lemma 4 we use the polynomials $p_{ij}(x)$ as part of the elements in the defined matrix S_{it} . However, in Lemma 4 we just view the polynomials $p_{ij}(x)$ as a symbol, change them to the dual codeword symbol C'_{ijt} will not have effects on the results of the lemma.

for all $\delta_m \in \mathbb{F}$. In particular, let $\delta_m, m \in [\ell]$, be a basis for \mathbb{F} over \mathbb{B} . Then,

$$\text{tr}_{\mathbb{F}/\mathbb{B}}(\delta_m \cdot \mu'_{ij} C_i) = \text{tr}_{\mathbb{F}/\mathbb{B}}(\delta_m \cdot \sum_{t=e+1}^n \theta_{ijt} C_t), \forall m \in [\ell], \quad (125)$$

which in turn implies that θ_{ijt} also satisfies (121). ■

Note that the repair bandwidth (122) also satisfies

$$\begin{aligned} b &= \sum_{t=e+1}^n \text{rank}_{\mathbb{B}}(\theta_{ijt} : i \in [e], j \in [\ell]) \\ &\leq \sum_{t=e+1}^n \text{rank}_{\mathbb{B}}(Q_t), \end{aligned} \quad (126)$$

since $\theta_{ijt} \in \text{span}_{\mathbb{B}}(Q_t)$. However, for any linear scheme L in Definition 1, if (126) holds with strict inequality, we can improve the linear scheme L by setting Q_t such that $\text{span}_{\mathbb{B}}(Q_t) = \text{span}_{\mathbb{B}}(\{\theta_{ijt}, i \in [e], j \in [\ell]\})$, for all $e+1 \leq t \leq n$. Hence, the linear scheme L and the scheme in Lemma 8 have identical bandwidth.

Lemma 9. The scheme in Lemma 8 can be represented by the dual code scheme in Definition 2.

Proof: By (121), $(0, \dots, \mu'_{ij}, \dots, 0, \theta_{ije+1}, \dots, \theta_{ijn})$ is a dual codeword, where μ'_{ij} is the i -th entry. Then, for $j \in [e]$, we set C'_{ijt} such that $C'_{iji} = -\mu'_{ij}$, $C'_{ijv} = 0, v \in [e], v \neq i$ and $C'_{ijt} = \theta_{ijt}, e+1 \leq t \leq n$. The full rank condition follows because $\{\mu'_{i1}, \mu'_{i2}, \dots, \mu'_{i\ell}\}$ is the basis for \mathbb{F} over \mathbb{B} , and the repair bandwidth condition follows from (126). Thus, we obtain the dual code scheme in Definition 2. ■

APPENDIX C PROOF OF LEMMA 4

Proof: Vectors $V_{ij}, i \in [e], j \in [\ell]$ are linearly independent over \mathbb{B} is equivalent to that there is no nonzero $b_{ij} \in \mathbb{B}, i \in [e], j \in [\ell]$ that satisfy

$$\sum_{i,j} b_{ij} p_{ij}(\alpha_v) = 0, \forall v \in [e]. \quad (127)$$

Here, $p_{ij}(x)$ can be written as

$$p_{ij}(x) = \sum_{m=1}^{\ell} \xi'_m \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(x)), \quad (128)$$

where $\{\xi_1, \xi_2, \dots, \xi_{\ell}\}$ and $\{\xi'_1, \xi'_2, \dots, \xi'_{\ell}\}$ are the dual basis for \mathbb{F} over \mathbb{B} . So, it is equivalent to that there is no nonzero $b_{ij} \in \mathbb{B}, i \in [e], j \in [\ell]$ that satisfy

$$\sum_{i,j} b_{ij} \sum_{m=1}^{\ell} \xi'_m \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_v)) = 0, \forall v \in [e]. \quad (129)$$

Since $\{\xi'_1, \xi'_2, \dots, \xi'_\ell\}$ are linearly independent over \mathbb{B} . Therefore, there is no nonzero $b_{ij} \in \mathbb{B}, i \in [e], j \in [\ell]$ that satisfy

$$\sum_{i,j} b_{ij} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_v)) = 0, \forall v \in [e], m \in [\ell], \quad (130)$$

which is equivalent to S has full rank. ■

APPENDIX D PROOF OF LEMMA 5

Proof: Assume $\text{rank}_{\mathbb{B}}(\{p_{ij}(\alpha_t), i \in [e], j \in [\ell]\}) = c$ and $\{p_{ij}(\alpha_t), (i, j) \in I\}$ are linearly independent over $\mathbb{B}, |I| = c$. Define $S_{it}(j)$ as the vector for the j -th row in S_{it} : $S_{it}(j) = (\text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_1 p_{ij}(\alpha_t)), \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_2 p_{ij}(\alpha_t)), \dots, \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_\ell p_{ij}(\alpha_t)))$. We first prove $\{S_{it}(j), (i, j) \in I\}$ are linearly independent and then prove $S_{i't}(j'), i' \in [e], j' \in [\ell], (i', j') \notin I$ can be represented as \mathbb{B} -linear combinations of $\{S_{it}(j), (i, j) \in I\}$.

If $\{S_{it}(j), (i, j) \in I\}$ are linearly dependent over \mathbb{B} , then there exists some nonzero $b_{ij} \in \mathbb{B}, (i, j) \in I$ that satisfies

$$\sum_{(i,j) \in I} b_{ij} S_{it}(j) = 0, \quad (131)$$

and we have

$$\sum_{(i,j) \in I} b_{ij} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_t)) = 0, \forall m \in [\ell]. \quad (132)$$

Multiplying the above equation by ξ'_m and summing over all $m \in [\ell]$ result in

$$\sum_{(i,j) \in I} \sum_{m=1}^{\ell} b_{ij} \xi'_m \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_t)) = 0. \quad (133)$$

Then, from (128) we know that b_{ij} satisfies

$$\sum_{(i,j) \in I} b_{ij} p_{ij}(\alpha_t) = 0, \quad (134)$$

which is contradictory to the statement that $\{p_{ij}(\alpha_t), (i, j) \in I\}$ are linearly independent over \mathbb{B} . Therefore, $\{S_{it}(j), (i, j) \in I\}$ are linearly independent over \mathbb{B} .

Let us assume $p_{i'j'}(\alpha_t), i' \in [e], j' \in [\ell], (i', j') \notin I$ can be represented as

$$p_{i'j'}(\alpha_t) = \sum_{(i,j) \in I} b'_{ij} p_{ij}(\alpha_t), \text{ for some } b'_{ij} \in \mathbb{B}. \quad (135)$$

Then, for $m \in [\ell]$,

$$\begin{aligned} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{i'j'}(\alpha_t)) &= \text{tr}_{\mathbb{F}/\mathbb{B}} \left(\xi_m \sum_{(i,j) \in I} b'_{ij} p_{ij}(\alpha_t) \right) \\ &= \sum_{(i,j) \in I} b'_{ij} \text{tr}_{\mathbb{F}/\mathbb{B}}(\xi_m p_{ij}(\alpha_t)), \end{aligned} \quad (136)$$

which means that for $i' \in [e], j' \in [\ell], (i', j') \notin I$,

$$S_{i't}(j') = \sum_{(i,j) \in I} b'_{ij} S_{it}(j) \quad (137)$$

is the \mathbb{B} -linear combination of $\{S_{it}(j), (i, j) \in I\}$. ■

REFERENCES

- [1] W. Li, Z. Wang, and H. Jafarkhani, "A tradeoff between the sub-packetization size and the repair bandwidth for Reed-Solomon code," in *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*. IEEE, 2017, pp. 942–949.
- [2] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977.
- [3] W. Ryan and S. Lin, *Channel codes: classical and modern*. Cambridge University Press, 2009.
- [4] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," in *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*. IEEE, 1998, pp. 28–37.
- [5] 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*, vol. 32, no. 5, pp. 998–1007, 2014.
- [6] V. Guruswami and M. Wootters, "Repairing Reed-Solomon codes," *IEEE Transactions on Information Theory*, 2017.
- [7] H. Dau and O. Milenkovic, "Optimal repair schemes for some families of full-length Reed-Solomon codes," in *Information Theory (ISIT), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 346–350.
- [8] 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*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [9] 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*, vol. 57, no. 8, pp. 5227–5239, 2011.
- [10] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through hadamard designs," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3021–3037, 2013.
- [11] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1597–1616, March 2013.
- [12] Z. Wang, I. Tamo, and J. Bruck, "Explicit minimum storage regenerating codes," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4466–4480, Aug 2016.
- [13] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Progress on high-rate MSR codes: Enabling arbitrary number of helper nodes," in *Information Theory and Applications Workshop (ITA), 2016*. IEEE, 2016, pp. 1–6.
- [14] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6318–6328, 2017.
- [15] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6307–6317, Oct 2017.
- [16] —, "Explicit constructions of MDS array codes and RS codes with optimal repair bandwidth," in *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1202–1206.
- [17] I. Tamo, M. Ye, and A. Barg, "Optimal repair of Reed-Solomon codes: achieving the cut-set bound," *arXiv preprint arXiv:1706.00112*, 2017.
- [18] H. Dau, I. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing reed-solomon codes with multiple erasures," *IEEE Transactions on Information Theory*, 2018.
- [19] J. Mardia, B. Bartan, and M. Wootters, "Repairing multiple failures for scalar MDS codes," *IEEE Transactions on Information Theory*, 2018.
- [20] M. Zorghi and Z. Wang, "Centralized multi-node repair regenerating codes," *IEEE Transactions on Information Theory*, 2019.

- [21] —, “On the achievability region of regenerating codes for multiple erasures,” in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2067–2071.
- [22] M. Ye and A. Barg, “Repairing Reed-Solomon codes: universally achieving the cut-set bound for any number of erasures,” *arXiv preprint arXiv:1710.07216*, 2017.
- [23] S. B. Gashkov and I. S. Sergeev, “Complexity of computation in finite fields,” *Journal of Mathematical Sciences*, vol. 191, no. 5, pp. 661–685, 2013.
- [24] K. M. Greenan, E. L. Miller, and S. T. J. Schwarz, “Optimizing galois field arithmetic for diverse processor architectures and applications,” in *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 1–10.
- [25] J. S. Plank, K. M. Greenan, and E. L. Miller, “Screaming fast galois field arithmetic using intel simd instructions.” in *FAST*, 2013, pp. 299–306.
- [26] J. Luo, K. D. Bowers, A. Oprea, and L. Xu, “Efficient software implementations of large finite fields $GF(2^n)$ for secure storage applications,” *ACM Transactions on Storage (TOS)*, vol. 8, no. 1, p. 2, 2012.
- [27] J. S. Plank, J. Luo, C. D. Schuman, L. Xu, Z. Wilcox-O’Hearn *et al.*, “A performance evaluation and examination of open-source erasure coding libraries for storage.” in *Fast*, vol. 9, 2009, pp. 253–265.
- [28] V. Guruswami and A. S. Rawat, “MDS code constructions with small sub-packetization and near-optimal repair bandwidth,” in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2017, pp. 2109–2122.
- [29] O. Khan, R. C. Burns, J. S. Plank, W. Pierce, and C. Huang, “Rethinking erasure codes for cloud file systems: minimizing I/O for recovery and degraded reads.” in *FAST*, 2012, p. 20.
- [30] J. Li and X. Tang, “A systematic construction of mds codes with small sub-packetization level and near optimal repair bandwidth,” *arXiv preprint arXiv:1901.08254*, 2019.
- [31] K. Kralevska, D. Gligoroski, R. E. Jensen, and H. Øverby, “HashTag erasure codes: from theory to practice,” *IEEE Transactions on Big Data*, vol. 4, no. 4, pp. 516–529, 2018.
- [32] A. Chowdhury and A. Vardy, “Improved schemes for asymptotically optimal repair of MDS codes,” *arXiv preprint arXiv:1710.01867*, 2017.
- [33] 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*, vol. 6, no. 5. VLDB Endowment, 2013, pp. 325–336.
- [34] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, “Asymptotic interference alignment for optimal repair of MDS codes in distributed storage,” *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.
- [35] D. Goss, *Basic structures of function field arithmetic*. Springer Science & Business Media, 2012.
- [36] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge university press, 1994.

Zhiying Wang (S’09–M’13) received the B.Sc. degree in Information Electronics and Engineering from Tsinghua University in 2007, M. Sc. and Ph.D degrees in Electrical Engineering from California Institute of Technology in 2009 and 2013, respectively. She was a postdoctoral fellow in Department of Electrical Engineering, Stanford University. She is currently Assistant Professor at Center for Pervasive Communications and Computing, University of California, Irvine. Dr. Wang is the recipient of NSF Center for Science of Information (CSoI) Postdoctoral Research Fellow, 2013. She received IEEE Communication Society Data Storage Best Paper Award. Her research focuses on information theory, coding theory, with an emphasis on coding for data storage. Department of Electrical Engineering and Computer Science, University of California, Irvine. His research interests includes coding theory, signal processing and communication network.

Hamid Jafarkhani (M’89–SM’01–F’06) is a Chancellor’s Professor at the Department of Electrical Engineering and Computer Science, University of California, Irvine, where he is also the Director of Center for Pervasive Communications and Computing and the Conexant-Broadcom Endowed Chair. He was a Visiting Scholar at Harvard University in 2015 and a Visiting Professor at California Institute of Technology in 2018. Among his awards are the IEEE Marconi Prize Paper Award in Wireless Communications, the IEEE Communications Society Award for Advances in Communication, and the IEEE Eric E. Sumner Award.

Dr. Jafarkhani is listed as a highly cited researcher in <http://www.isihighlycited.com>. According to the Thomson Scientific, he is one of the top 10 most-cited researchers in the field of “computer science” during 1997-2007. He is the 2017 Innovation Hall of Fame Inductee at the University of Maryland’s School of Engineering. He is a Fellow of AAAS, an IEEE Fellow, and the author of the book “Space-Time Coding: Theory and Practice.”

Weiqli Li received his B.S. and M.S. degree from Xian Jiaotong University, China in 2013 and 2016, respectively. Currently, he is pursuing a Ph.D in the Department of Electrical Engineering and Computer Science, University of California, Irvine. His research interests includes coding theory, signal processing and communication network.