https://doi.org/10.5755/j02.eie.43974 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, EARLY ACCESS

Four Times Extended Reed Solomon Codes, their Decoding and Application in EII Codes for Data Repair

Peter Farkas[*,1,2], Martin Rakus[1], Katarina Farkasova[1], Frederik Pavelka[1], Ladislav Divinec[1]
[1]*Institute of Multimedia ICT, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava,*
Ilkovicova 3, 812 19 Bratislava, Slovakia
[2]*Institute of Applied Informatics, Faculty of Informatics, Pan - European University,*
Tomasikova, 20, 821 02, Bratislava, Slovakia
[*]*p.farkas@ieee.org; martin.rakus@stuba.sk; katarina.farkasova@stuba.sk; frederik.pavelka@stuba.sk; ladislav.divinec@stuba.sk*

***Abstract*—Data recovery in distributed and general storage systems requires a broad range of regenerating codes with different properties such as locality, availability, scalability, etc. The main goal of this paper is to enrich this palette with four times extended Reed Solomon (RS) codes. Recently, it was shown that RS codes can be extended five times, when constructed over finite fields $GF(q)$, where $q = 2^{\xi}$ and $\xi \geq 3$ is an odd integer. These codes are almost maximum distance separable (AMDS), and they are reaching upper bounds on code distance in known tables on the best linear block codes. However, such codes have limitations for practical applications since ξ has to be an odd integer. To overcome this limitation, four times extended RS codes are presented in this paper. These $[q+3, q-1, 4]$ AMDS codes can be constructed over finite field $GF(2^{\mu})$, where the integer $\mu \geq 3$ is arbitrary. They can be used in numerous different constructions of product codes, Extended Product codes, Integrated Interleaved codes, Extended Integrated Interleaved codes, Staircase codes, and others, which are suitable for data recovery in storage systems.**

***Index Terms*—Digital storage; Error correction codes; Product codes; Reed Solomon codes.**

I. INTRODUCTION

Distributed data storage and other data storage in different forms will be essential to support the needs of diverse applications in 6G networks. For example, edge- and fog-type clouds can enhance autonomous transport with autonomous cars and drones. In data centres and storage systems, as a rule, devices and elements sometimes experience failures. In such situations, regenerating codes can be used to protect against information loss. There are numerous publications dealing with data recovery in storage systems [1]–[7]. Actual systems need codes with locality, availability, scalability, and other properties. There are different constructions that try to adapt the classical Error Correcting codes (ECCs) to these needs. For example, Locally Recoverable codes (LRC), product codes (PC), Extended Product codes (EPC), Integrated Interleaved codes (IIC), Extended Integrated Interleaved codes (EIIC), Staircase codes (SC), etc. These families of codes are related in most cases, and their nomenclature is not yet consolidated in all cases. In [8] and [9], an excellent overview of these relationships is given. Therefore, it will not be repeated here. However, it is worth mentioning that the roots of the ideas that led to these actual constructions can be traced back to publications in [10], [11].

Reed Solomon (RS) codes were discovered at the M. I. T. Lincoln Laboratory in 1958 [12], [13]. Since then, they have been applied in countless practical applications, which also include information recovery in storage systems. The theoretical interest in these codes continues also today more than sixty years after their first publication, which is documented by recent publications, for example, in [14]–[22]. With increasing demands on data rates, the encoding and decoding techniques of RS codes are also improving [23]–[29]. RS codes are used not only for error control but also in other fields, for example, in security and privacy applications [30]–[33].

The applications of RS codes relevant to the new knowledge presented in this manuscript are in data storage systems (clouds, Redundant Arrays of Independent Disks, Solid State Devices, DNA storage, etc.) [34]–[42]. In these systems, the codes are used for data recovery in case some subsystem (node) is damaged. The subsystem could be, for example, a server or sector. In Fig. 1, it is illustrated how such a subsystem could be interpreted as a codeword symbol. Damage to one node in this case corresponds to the erasure of one symbol in a codeword.

Particularly with distributed storage systems behind the overall redundancy, the volumes of data which have to be communicated during data recovery also have to be considered when the expenditures are calculated. Therefore, the so-called codes with locality are the focus of research in this area at present. In a code with locality, the symbols of

This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00051.4.

its codewords could be recovered locally by contacting only a restricted number of other symbols. If all symbols have this property, the code is denoted as a locally recoverable code (LRC).

However, if the local capability to correct the erasures is unable to provide data recovery, global redundancy can help. Therefore, many different families of codes were proposed that combine local with global redundancy, for example, PC, EPC, II, EII, SC, and numerous others [8], [9].

Recently, in [43], it was discovered that RS codes could be extended not only three times but even five times over certain finite fields. In [43], it was also proven that these codes belong to the class of almost MDS (AMDS) codes. These codes are optimal with regard to their code distance for the corresponding values of the codeword lengths and the number of information symbols in the codewords. As was mentioned in the abstract, the codes from [43] can be constructed only over $GF(2^{\xi})$, where $\xi \geq 3$ is an odd integer. To overcome this limitation, a family of four times extended RS codes over $GF(2^{\mu})$ is proposed, where $\mu \geq 3$ can be even or odd integer. Such codes have a much easier practical implementation than the codes described in [43].

This paper is organized as follows. In Section II, we provide the theoretical background on linear block codes and RS codes. In Section III, the four times extended RS codes are proposed and analysed. In Section IV, the erasure decoding method is described for the four time extended RS codes. In Section V, the application of the four times extended RS codes in constructions of EII codes is illustrated using three examples, and the obtained codes are analysed. In conclusion, together with a review of the new results obtained, the optimality of the codes proposed in this paper is discussed.

II. THEORETICAL BACKGROUND

RS codes are linear block codes and are among the most frequently applied error correcting codes even to the present.

Linear Block Codes (LBC)

LBC C is a k - dimensional subspace of an n - dimensional vector space over finite field $GF(q)$, where q is the number of elements in the finite field.

LBC can be defined using a generator matrix $\mathbf{G}$ or a control matrix $\mathbf{H}$, which are in the following relationship:

$$\mathbf{G} \times \mathbf{H}^T = \mathbf{0} \tag{1}$$

The generator matrix contains k linearly independent vectors of length n. Therefore, C is often described via the following triple $[n, \ k, \ d]$, or sometimes also specifying the finite field over which the code is defined as follows $[n, \ k, \ d]_{GF(q)}$. Besides n and k, which could be also interpreted as codeword length and number of information bearing symbols, respectively, it contains the third basic parameter, which is the code distance d. It is the minimum Hamming distance of any two of the codewords $c_i, c_j \in \mathbf{C}$; $i \neq j$. Hamming distance $d(\mathbf{c}_i, \mathbf{c}_j)$ is the number of coordinates by which $\mathbf{c}_i$, $\mathbf{c}_j$ differ. It is well known that in linear block codes the number of correctable erasures z in a codeword (in case of bounded distance decoding) is in the following relationship with code distance:

$$z + 1 \leq d. \tag{2}$$

The codeword of a linear block code is a vector with coordinates (codeword symbols) from a finite field $GF(q)$,

$$\mathbf{c} = (c_0, \ c_1, \ \ldots, \ c_{n-2}, \ c_{n-1}). \tag{3}$$

Codes with locality

Codes with locality allow for correction of erased symbols using only a small number of other (local) symbols. More precisely, the i^{th} symbol $c_i \in GF(2)$; $i = 0, 1, \ldots, n-1$ of a codeword $\mathbf{c} \in C$, where $\mathbf{c} \in C$ is a $[n, \ k, \ d]$ linear block code, has locality r if c_i can be recovered by accessing at most r other symbols from $\mathbf{c}$.

A $[n, \ k, \ d]$ code C has locality r and is denoted as r - LRC if and only if all codeword symbols of all its codewords have locality r.

Reed Solomon (RS) codes

The most practical RS codes are defined over a finite field with characteristics two [44]. The original RS codes could be described, for example, using the control matrix, which is a Vandermonde matrix

$$\mathbf{H} = \begin{bmatrix} \alpha^{(q-2)\vartheta} & \cdots & \alpha^{2\vartheta} & \alpha^{\vartheta} & \alpha^0 \\ \alpha^{(q-2)(\vartheta+1)} & \cdots & \alpha^{2(\vartheta+1)} & \alpha^{\vartheta+1} & \alpha^0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha^{(q-2)(\vartheta+2t-1)} & \cdots & \alpha^{2(\vartheta+2t-1)} & \alpha^{\vartheta+2t-1} & \alpha^0 \end{bmatrix}, \tag{4}$$

where α is a primitive element of the finite field over which the RS code is defined and ϑ is a positive integer for which the following holds:

$$\vartheta \leq \left\lfloor \frac{q-1}{2} \right\rfloor. \tag{5}$$

The RS codes could also be defined as cyclic codes using a generator polynomial

$$g(x) = (x + \alpha^{\vartheta}) \times (x + \alpha^{\vartheta+1}) \times (x + \alpha^{\vartheta+2}) \times \ldots \times (x + \alpha^{\vartheta+2t-1}). \tag{6}$$

The interested reader can find more information on error control codes, for example, in the classical monographs mentioned in the abstract in [44]–[48].

Five times extended RS codes

In [43], a new family of codes was proposed over $GF(2^{\xi})$, where $\xi \geq 3$ is an odd integer, using the following $\mathbf{H}$ matrix:

$$\mathbf{H} = \begin{bmatrix} \alpha^0 & \dots & \alpha^0 & \alpha^0 & \alpha^0 & 1 & 0 & 0 & 0 & 0 \\ \alpha^{(q-2)} & \dots & \alpha^2 & \alpha^1 & \alpha^0 & 0 & 1 & 0 & 0 & 0 \\ \alpha^{2(q-2)} & \dots & \alpha^4 & \alpha^2 & \alpha^0 & 0 & 0 & 1 & 0 & 0 \\ \alpha^{3(q-2)} & \dots & \alpha^6 & \alpha^3 & \alpha^0 & 0 & 0 & 0 & 1 & 0 \\ \alpha^{4(q-3)} & \dots & \alpha^8 & \alpha^4 & \alpha^0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

The basic parameters of this family of block codes can be characterized by a triple $[q-4, \quad q-1, \quad 5]$.

III. FOUR TIMES EXTENDED RS CODES

The proposed four times extended RS codes defined over $GF(2^\mu)$, where $\mu \geq 3$ is an arbitrary integer, are defined by the following control matrix:

$$\mathbf{H} = \begin{bmatrix} \alpha^{(q-2)\vartheta} & \cdots & \alpha^{2\vartheta} & \alpha^{\vartheta} & \alpha^0 & 1 & 0 & 0 & 0 \\ \alpha^{(q-2)(\vartheta+1)} & \cdots & \alpha^{2(\vartheta+1)} & \alpha^{\vartheta+1} & \alpha^0 & 0 & 1 & 0 & 0 \\ \alpha^{(q-2)(\vartheta+2)} & \cdots & \alpha^{2(\vartheta+2)} & \alpha^{\vartheta+2} & \alpha^0 & 0 & 0 & 1 & 0 \\ \alpha^{(q-2)(\vartheta+3)} & \cdots & \alpha^{2(\vartheta+3)} & \alpha^{\vartheta+3} & \alpha^0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

where ϑ is an integer and α is a primitive element of the finite field with characteristic 2, over which the family of codes is defined. The basic parameters of the four times extended RS codes are $[q+3, q-1, 4]$. The code distance will be analysed in the following paragraphs.

Without loss of generality, we can let $\vartheta = 0$ and consequently the control matrix of the four times extended RS codes is

$$\mathbf{H} = \begin{bmatrix} \alpha^0 & \cdots & \alpha^0 & \alpha^0 & \alpha^0 & 1 & 0 & 0 & 0 \\ \alpha^{(q-2)} & \cdots & \alpha^2 & \alpha^1 & \alpha^0 & 0 & 1 & 0 & 0 \\ \alpha^{2(q-2)} & \cdots & \alpha^4 & \alpha^2 & \alpha^0 & 0 & 0 & 1 & 0 \\ \alpha^{3(q-2)} & \cdots & \alpha^6 & \alpha^3 & \alpha^0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Theorem 1

The four times extended RS codes constructed over $GF(2^\mu)$, where $\mu \geq 3$ is an arbitrary integer, defined by (9), have a code distance $d \geq 4$.

Proof

The columns in the parity check matrix in (9) are of two types. In part $\mathbf{P}^T$, they are of the form

$$\chi_i = (\alpha^0, \quad \alpha^i, \quad \alpha^{2i}, \quad \alpha^{3i}); \qquad 0 \leq \quad i \leq \quad q-2. \quad (10)$$

In part $\mathbf{I}$, they are unit vectors $\mathbf{e}_i$; $1 \leq \quad i \leq \quad 4$. To prove this, $d = 4$, it is necessary to show that any three columns in matrix (9) are linearly independent.

The following proof is similar to the one in [43], which is a variation of Theorem 8 in [45]: "An $[n, \quad k, \quad d]$ code $\mathbf{C}$ with generator matrix $\mathbf{G} = [\mathbf{I} \quad \mathbf{A}]$, where $\mathbf{A}$ is a $k \times (n-k)$ matrix is MDS if every square submatrix (formed from any i rows and any i columns for any $i = \quad 1, \quad 2, \quad \dots, \quad \min\{k, n-k\}$) of $\mathbf{A}$ is not singular".

For the case where one column of $\mathbf{H}$ is a unit vector and the other two are of type χ_i, it is necessary to show that any three rows of matrix (9) contain two independent rows. There are four cases. If the fourth row is excluded, then vectors χ_x and χ_y are linearly independent in any two positions, i.e., for $x \neq y$, all 2×2 minors in matrix (11) are of full rank. If the first, second, or third row is excluded, then the two remaining rows which do not contain a row with element α^{3x} will contain the 2×2 minor in matrix (11), which is full rank

$$\begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^x & \alpha^y \\ \alpha^{2x} & \alpha^{2y} \\ \alpha^{3x} & \alpha^{3y} \end{bmatrix}. \quad (11)$$

This follows from the fact that the map $x \mapsto x^i$ is injective for $i = 1, 2$ on a finite field of size 2^μ. QED

For $\mu = 3, 4, 5, 6, 7, 8$, it was possible to check the code distance by computer calculating the weight spectra of the codes defined by (9) using their duals. The following shortened weight enumerators of the four times extended Reed Solomon codes defined by (9) were obtained using Krawtchouk equations as follows.

$[11, \quad 7, \quad 4]_{GF(8)}$:

$1 + 98y^4 + 2548y^5 + 11760y^6 + 68180y^7 + 230965y^8$
$+542332y^9 + 758520y^{10} + 482748y^{11}$

$[19, \quad 15, \quad 4]_{GF(16)}$:

$1 + 3195y^4 + 132255y^5 + 4735455y^6 + 131289015y^7 + \cdots$

$[35, \quad 31, \quad 4]_{GF(32)}$:

$1 + 38440y^4 + 8871952y^5 + 1376459520y^6 + \cdots$

$[67, \quad 63, \quad 4]_{GF(64)}$:

$1 + 740880y^4 + 563111136y^5 + 372303546048y^6 + \cdots$

$[131, \quad 127, \quad 4]_{GF(128)}$:

$1 + 10838688y^4 + 36419023936y^5 + 97712593219584y^6 + \cdots$

$[259, \quad 255, \quad 4]_{GF(256)}$:

$1 + 181723200y^4 + 2336289502080y^5$
$+25319042157669120y^6 + \cdots$

Exponents of y denote the particular weights.

The difference between MDS codes and AMDS codes can be expressed using the "Singleton defect", which for a $[n, \quad k, \quad d]$ code $\mathbf{C}$ is defined as the quantity

$$s(C) = n - k + 1 - d. \quad (12)$$

For MDS codes, it holds that $s(C)=0.$ For AMDS codes $s(C)=1$ [49]. The four times extended RS codes are AMDS codes because they are $\left[q+3, \quad q-1, \quad 4\right]$ codes.

After consulting the tables that contain upper and lower bounds on the code distance for linear block codes [26], it can be seen that in finite fields for which these tables exist, the presented codes reach an upper bound on the code distance for a given *n* and *k*.

IV. Erasure Decoding of Four Times Extended RS Codes

RS codes are attractive for data repair in data centre storage systems. The information mapping if RS codes are used against data loss could be illustrated in a simplified manner by Fig. 1. In data centres, when a server is damaged, its position is known. In this scheme, each codeword protects only one symbol from an RS codeword. Consequently, in this case, an erasure decoding algorithm can be used for data repair. Compared to error decoding, erasure decoding is not only simpler, but also allows for repairing twice as many symbols in each codeword.

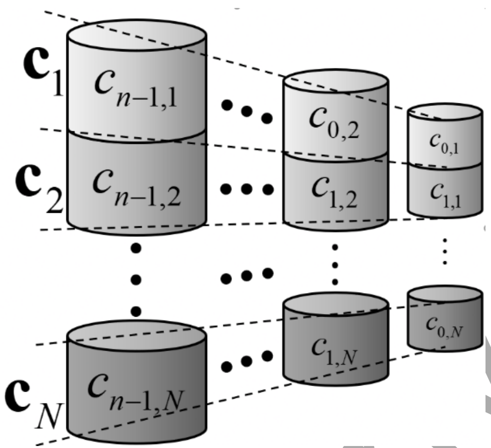

Fig. 1. Illustration of how the nodes in the datacentre could be mapped on codeword symbols and represented by them in regenerating codes.

The occurrence of erasures in a codeword could be modelled using an erasure channel, as shown in Fig. 2.

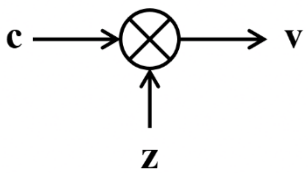

Fig. 2. Erasure channel model: $\otimes$ denotes a multiplication of two vectors over $GF(2^u)$, $\mathbf{c}$ is the transmitted codeword, $\mathbf{z}$ is an erasure vector, and $\mathbf{v}$ is the received vector. In the erasure vector, all coordinates are neutral elements from $GF(2^u)$, denoted as α^0 except the coordinates corresponding to the erased positions, which are zeros denoted as 0.

An erasure is defined only by its value since its position within the received vector is known to the decoder prior to decoding. The *j*th erasure is denoted as Z_j. To correct one erasure, the decoder has to calculate the erasure value Z_j.

The received vector will be denoted as

$$\mathbf{v}=\left(v_{n-1}, \quad v_{n-2}, \quad \ldots, \quad v_1, \quad v_0\right). \tag{13}$$

In the known up to three erased positions the values of the coordinates v_i will be zero(s). The following erasure decoding algorithm is syndrome-based. Therefore, the values of syndromes S_0, S_1, S_2, S_3 must to be calculated

$$\left(S_0, \quad S_1, \quad S_2, \quad S_3\right)=\mathbf{v}\mathbf{H}^T. \tag{14}$$

For each codeword $\mathbf{c}$ from a linear block code based on (1),

$$\mathbf{c}\mathbf{H}^T=\mathbf{0}. \tag{15}$$

Therefore, if there are z erasures in the received vector, then

$$\left(S_0, \quad S_1, \quad S_2, \quad S_3\right)=\mathbf{z}\mathbf{H}_z^T, \tag{16}$$

where $\mathbf{z}$ is a vector, with coordinates corresponding to the unknown values of the erasure(s) and $\mathbf{H}_z$ is $4\times z$ matrix with z column(s) from $\mathbf{H}$ corresponding to the position(s) where the erasure(s) occurred in the same order. The unknown values can be obtained from (16).

It must be noted that if the number of erasures does not exceed three, then (16) can always be solved. It is obvious because the control matrix (8) is of the form

$$\mathbf{H}=\left[\mathbf{H}_P\mathbf{I}\right], \tag{17}$$

where $\mathbf{H}_p$ is the Vandermonde matrix with linearly independent rows. It is well known that adding columns from $\mathbf{I}$ to it cannot destroy this linear independence. Consequently, any matrix formed from a subset of columns from $\mathbf{H}$, as for example $\mathbf{H}_z$, also has linearly independent rows.

The decoding algorithm can be described as follows.

Input: received vector $\mathbf{v}$, number of erasures z, and positions of erasures in $\mathbf{v}$ (indexes of corresponding coordinates).

Decoding steps:

1. If $z=0$, then the received vector is outputted as the estimation of the transmitted codeword, and the decoding is ended. Else go to 2;
2. If $z>3$, a decoding failure is outputted, and the decoding is ended. Else go to 3;
3. Syndromes calculation using (14);
4. Test if there is no discrepancy between z and syndrome values. If a discrepancy is detected (for example if $S_0=S_1=S_2=S_3=0$), then decoding failure is outputted and the decoding is ended. Else go to 5;
5. The values of erasure(s) are calculated by solving (16);
6. Inserting the erasure(s) value(s) calculated in step 5 into appropriate coordinates in $\mathbf{v}$ and outputting the resulting vector $\hat{\mathbf{v}}$ as an estimation of the transmitted codeword $\mathbf{c}$; End of decoding.

Example 1

In this example, the procedure is described in detail for a four times extended RS code over $GF(2^3)$ defined by the primitive polynomial $p(x)=x^3+x+1$ supposing that the transmitted message vector is

$$\mathbf{m}=\left(\alpha^4,\alpha^6,\alpha^2,\alpha^4,\alpha^5,\alpha^2,\alpha^3\right). \tag{18}$$

This code has the following $\mathbf{H}$ matrix:

$$\mathbf{H}=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 1 & 0 & 0 \\ \alpha^5 & \alpha^3 & \alpha & \alpha^6 & \alpha^4 & \alpha^2 & 1 & 0 & 0 & 1 & 0 \\ \alpha^4 & \alpha & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

and the following $\mathbf{G}$ matrix:

$$\mathbf{G}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha^6 & \alpha^5 & \alpha^4 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha^5 & \alpha^3 & \alpha \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \alpha^4 & \alpha & \alpha^5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \alpha^3 & \alpha^6 & \alpha^2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \alpha & \alpha^2 & \alpha^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{20}$$

The codewords of this code will be denoted

$$\mathbf{c}=\left(c_{10},c_9,c_8,c_7,c_6,c_5,c_4,c_3,c_2,c_1,c_0\right), \tag{21}$$

where

$$\mathbf{c}=\mathbf{mG}. \tag{22}$$

From (18) and (22),

$$\mathbf{c}=\left(\alpha^4,\alpha^6,\alpha^2,\alpha^4,\alpha^5,\alpha^2,\alpha^3,\alpha^0,0,\alpha^0,\alpha^6\right). \tag{23}$$

Decoding

Input: vector $\mathbf{v}$ given by (23), number of erasures $z=3$ and the positions of these erasures $i=7=I$, $i=4=II$, and $i=0=III$.

In the first step, the decoder will test if the number of erasures is zero. If yes, the decoding process will stop and the senseword is declared as the codeword. In the opposite case, it is tested if the number of erasures is greater than 3. If the answer is positive, then the decoding will be stopped, and a decoding failure will be declared.

In the opposite case, the calculation of erasures will follow.

Syndromes will be calculated using (15)

$$\left(S_0,\quad S_1,\quad S_2,\quad S_3\right)=\mathbf{v}\begin{bmatrix} 1 & \alpha^6 & \alpha^5 & \alpha^4 \\ 1 & \alpha^5 & \alpha^3 & \alpha \\ 1 & \alpha^4 & \alpha & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

In this example, we get the following vector containing syndromes as its coordinates

$$\left(S_0,\quad S_1,\quad S_2,\quad S_3\right)=\left(\alpha^2,\quad \alpha^1,\quad 0,\quad \alpha^3\right). \tag{25}$$

In the next step, it is tested if there is a discrepancy between $z=3$ and the values of the syndromes. In case all are zero, the decoding failure is declared and the decoding ends.

If the discrepancy is not detected, in the following step, the $\mathbf{H}_\zeta^T$ matrix will be formed

$$\mathbf{H}_z^T=\begin{bmatrix} 1 & \alpha^3 & \alpha^6 & \alpha^2 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{26}$$

And then, the following equation will be solved

$$\left(Z_I,\quad Z_{II},\quad Z_{III}\right)\begin{bmatrix} 1 & \alpha^3 & \alpha^6 & \alpha^2 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}=\left(S_0,\quad S_1,\quad S_2,\quad S_3\right)^T. \tag{27}$$

Equation (27) can also be expressed as follows:

$$Z_I+Z_{II}=\alpha^2, \tag{28}$$

$$Z_I\alpha^3+Z_{II}=\alpha, \tag{29}$$

$$Z_I\alpha^6+Z_{II}=0, \tag{30}$$

$$Z_I\alpha^2+Z_{II}+Z_{III}=\alpha^3. \tag{31}$$

Solving (27) or (28)–(31), the following erasure values are obtained: $Z_I=\alpha^4$, $Z_{II}=\alpha^3$, and $Z_{III}=\alpha^6$.

These values are substituted into the erasure coordinates, and the resulting vector is outputted from the decoder as the estimation of the transmitted codeword.

V. EXAMPLES OF EII CODES OBTAINED FROM FOUR TIMES EXTENDED RS CODES

In this section, some examples are given on how the four times extended RS codes can be used in order to obtain 1-level EII codes.

Definition

1-level Extended Integrated Interleaved (EII) codes [8]

C is a one-level EII denoted as $C(n, \ \mathbf{u})$ if and only if its codewords are $m \times n$ arrays $C = c_{i,j}; \quad 0 \le i \le m-1, \quad 0 \le j \le n-1$ that satisfy:

1. $c_{i,j} = 0$ for $0 \le i < m-1$ and $0 \le j \le n - u_1 - 1$;
2. $c_{i,j}$ for each i; $0 \le i < m-1$ are symbols of a codeword $\mathbf{c} = (c_{i,n-u_1}, \ c_{i,n-u_1+1}, \ \ldots, \ c_{i,n-u_0-1})$ in a vertical $\left[m, \ s_0, \ d_0^{(V)}\right]$-code V_0 defined over $\left(GF(q)\right)^{u_1-u_0}$ that is linear over $GF(q)$;
3. for each i; $0 \le i < m-1$ the row vector $\mathbf{c} = (c_{i,0}, \ c_{i,1}, \ \ldots, \ c_{i,n-1})$ in each array is an horizontal $\left[n, \ n-u_0, \ d_0^H\right]_{GF(q)}$-code C_0 that admits systematic encoder on its $n-u_0$ symbols, where $\mathbf{u} = (\underbrace{u_0, u_0, ..., u_0}_{s_0}, \underbrace{u_1, u_1, ..., u_1}_{s_1})$ is a vector of length $m = s_0 + s_1$, where for the integers s_0, s_1, u_0, and u_1 the following inequalities are valid $s_0 \ge 1, \quad s_1 \ge 0,$ $0 \le u_0 < u_1 \le n.$

The quality of the EII codes obtained was tested by the Monte Carlo method as follows. In the codewords, erasures were generated - one at the time until the number of erasures z_{fail} caused decoding failure. In such a case, the number of correctable erasures in this trial was $\zeta = z_{fail} - 1$. Repeating this trial many times, the average number of correctable erasures in a codeword denoted as ζ_{av} was estimated. This performance measure for EII codes was proposed in [8] and was argued that it is a useful comparison for the performance of EII codes because it is closely related to the mean time to data loss (MTTDL) for practical memory systems. It is interesting that for some of the codes, the value ζ_{av} is higher than the number of theoretically correctable erasures z, deduced from (2) and the code distance.

Monte Carlo simulations were realized in MATLAB for EII one-level codes. After generating the following erasure in the original codeword (stored array), it was duplicated, and we attempted to decode this copy. The decoding was simulated in such a way that in the first iteration, all erasures were corrected in rows in which the number of erasures was $z \le d_H - 1$, where d_H denotes the code distance of the horizontal code. In the next iteration, the erasures correctable by the vertical code were corrected. This cycle composed from two described iterations was repeated until all erasures were corrected or until after 20 cycles some erasures still remained in the array. In the first case, one additional erasure was generated in the stored original array and the decoding was started from the beginning for another copy of it. In the second case, the situation was considered as the decoding failure and the number of erasures minus one was included into the calculation of ζ_{av} for that simulation run, and a new run began with new original array. Altogether, for estimation of ζ_{av}, 100 000 runs were used.

Example 2

The binary one-level EII code $C(4 \quad (1,1,1,1,1,1,4,4,4,4))$ with C_0, the $\left[4, \ 3, \ 2\right]$ single parity check code over $GF(2)$ and V_0, the $\left[11, \ 7, \ 4\right]$ four times extended RS code over $GF(8)$, is illustrated in Fig. 3.

The resulting EII code over $GF(2)$ has overall codeword length $N = 44$ and dimension $K = 21$. By the Monte Carlo simulation the estimation of $\zeta_{av} \simeq 12$ was obtained.

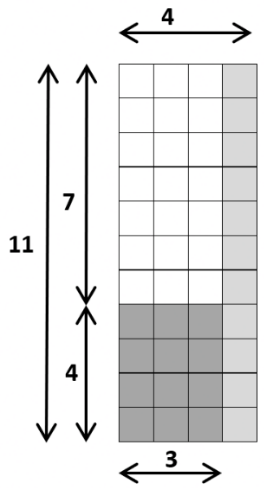

Fig. 3. Illustration of the $C(4 \quad (1,1,1,1,1,1,4,4,4,4))$ EII code.

Example 3

The binary one-level EII code $C(5 \quad (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,5,5,5,5))$ with C_0, the $\left[5, \ 4, \ 2\right]$ single parity check code over $GF(2)$, and V_0, the $\left[19, \ 15, \ 4\right]$ four times extended RS code over $GF(16)$, is illustrated in Fig. 4.

The resulting EII code over $GF(2)$ has an overall codeword length $N = 95$ and dimension $K = 60$. By the Monte Carlo simulation, the estimation of $\zeta_{av} \simeq 11.33$ was obtained. It should be noted that the dimensionality of the codes presented in these examples can be increased using the construction proposed in [10] by adding a code with code distance 8 into the last column of the codes that correspond to the parity symbols of the single-parity check codes.

For example, it is possible to add $\left[19, \ 7, \ 8\right]$ code to the last column of the $C(5 \quad (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,5,5,5,5))$ EII code, as illustrated in Fig. 5. This way a code with an overall codeword length $N = 95$ and dimension $K = 67$ is obtained. By the Monte Carlo simulation the estimation of $\zeta_{av} \simeq 11.33$ was obtained, which is the same as in the code in Example 2.

For this code, the simulation software was slightly modified. After 20 cycles of iterative decoding of the EII codes from the previous example, it was tested if there are more than 7 remaining erasures in the $\left[19, \ 7, \ 8\right]$ code. If the answer was negative, the erasures were corrected, and the experiment could continue. Otherwise, the decoding failure was declared and the appropriate number of erasures was included in the ζ_{av} calculation. It can be conjectured that because the values of ζ_{av} are the same as in Example 2, the event that more than 7 erasures will remain in the $[19,\quad 7,\quad 8]$ code after 20 cycles of iteration is very rare.

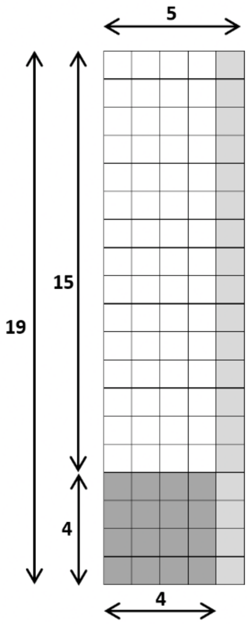

Fig. 4. Illustration of the $C(5\quad(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,5,5,5,5))$ EII code.

In practice, the number of elements q in the finite field $GF(q)$ over which the code is defined is also very important. This is because the complexity of the operations and the volume of memory needed during encoding and decoding could be prohibitively large for large values of q. Therefore, for the optimality evaluation of the EII codes in [8] the following upper bound on the code distance from [50] was used:

$$d \le \min\left\{ d_{opt}^{(q)}\left[(m-j)n,\quad k-jk^*\right]; 0 \le j \le \left\lceil \frac{k}{k^*} \right\rceil - 1 \right\}, \tag{32}$$

where $d_{opt}^{(q)}[\nu,\quad \kappa]$ denotes the largest code distance of a linear block code with codeword length ν and dimension κ over $GF(q)$ and

$$k^* = k_{opt}^{(q)}[n,\quad d_0], \tag{33}$$

where $k_{opt}^{(q)}[n,\quad d_0]$ denotes the largest possible dimension of a linear block code with codeword length n and code distance d_0.

After obtaining the parameters $d_{opt}^{(q)}[\nu,\quad \kappa]$ and $k^* = k_{opt}^{(q)}[n,\quad d_0]$ using the query forms in [51] one can see that the EII code $[95,\quad 67,\quad 8]$ is optimal with respect to the bound in (32).

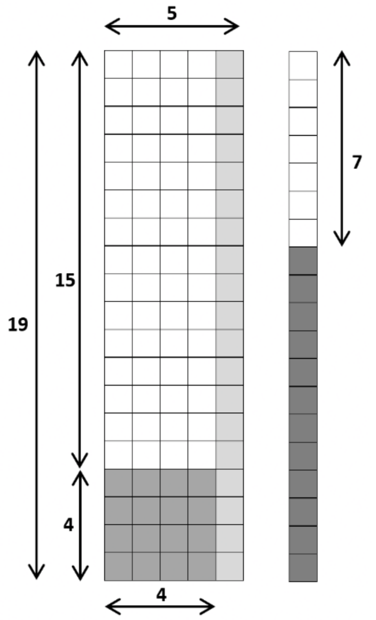

Fig. 5. Illustration of the $[95,\quad 67,\quad 8]$ code obtained from the EII code $C(5\quad(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,5,5,5,5))$ by adding $[19,\quad 7,\quad 8]$ code to the last column of this EII code in accordance with the construction proposed in [8].

For the reader's convenience, the values needed in (28) are presented in Table I.

TABLE I. PARAMETERS VALUES NEEDED IN (28) FOR THE CODE DEPICTED IN FIG. 5 OBTAINED FROM [51].

j	$(m-j)n$	$k-jk^*$	$d_{opt}^{(q)}$
0	95	67	10–12
1	90	63	9/12
2	85	59	8–11
3	80	55	8–10
4	75	51	8–10
5	70	47	8–10
6	65	43	8–10
7	60	39	8–10
8	55	35	8–9
9	50	31	8
10	45	27	8
11	40	23	8
12	35	19	8
13	30	15	8
14	25	11	8
15	20	7	8
16	15	3	8

VI. CONCLUSIONS

In this paper, the family of four times extended Reed Solomon codes defined over $GF(2^{\mu})$, where $\mu \ge 3$ is an arbitrary integer, where analysed. It was proven that these codes have code distance equal to four and also that they are almost maximum distance separable codes. The main result of this communication is that in contrast to five times extended RS codes it was proven that these codes can be constructed over any finite field $GF(2^{\mu})$, where $\mu \ge 3$ is an arbitrary integer. It can be conjectured that the codes presented are optimal in the sense of reaching the upper bounds on the code distance for a given n and k. This assumption is based on the fact that the $[11,\quad 7,\quad 4]_{GF(8)}$,

$[19,\ 15,\ 4]_{GF(16)}$, $[35,\ 31,\ 4]_{GF(32)}$, $[67,\ 63,\ 4]_{GF(64)}$, and $[131,\ 127,\ 4]_{GF(128)}$ codes, for which the entries in table in [51] exist, have an optimal code distance. The presented weight spectra confirm that the code distance is four. The four times extended RS codes allow the expansion of the number of information symbols by 3, when compared to ordinary RS codes with the same code distance.

The paper also contains a brief explanation of the erasure decoding of these codes, which is needed in data repair applications. Examples are presented, in which the four times extended RS codes are used for construction of EII codes suitable for data repair in memory systems. It is also noted that the optimal LRC codes with respect to the bound of [51] for the code distance could be found using the four times extended RS codes. An interesting fact is that the Monte Carlo simulation revealed that the average values of mean time to data loss for the EII codes presented in the examples are higher than the theoretical number of erasures deduced from their code distances.

In future research, other constructions of EII codes obtained from extended RS codes in combination with other linear block codes could be analysed. It is obvious that there are huge numbers of combinations possible, which can be advantageous for adapting the EII codes to concrete practical demands in different future memory systems.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] H. Li, H. Hou, and H. Tan, "Codes for distributed storage", in *Distributed Storage in Practice*. Wiley-IEEE Press, 2026, pp. 19–92. DOI: 10.1002/9781394334780.ch2.

[2] C. Pamnani, "Optimizing data availability and performance in cloud storage with hybrid network-coding and object-based storage systems", in *Proc. of 2025 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 2025, pp. 7–13. DOI: 10.1109/DISCOVER66922.2025.11258954.

[3] Z. Huang, Z. Yu, L. Wang, H. Zhou, E. Yang, and B. Guo, "Erasure coding-based cost-optimized and latency-aware data storage in UAV-enabled edge systems", in *IEEE Transactions on Mobile Computing*, vol. 25, no. 1, pp. 691–705, 2026. DOI: 10.1109/TMC.2025.3594283.

[4] G. K. Verma, A. Singh, and A. K. Singh, "Function-correcting b-symbol codes for locally (λ, ρ, b)-functions", *IEEE Transactions on Information Theory*, vol. 72, no. 1, pp. 331–341, 2026. DOI: 10.1109/TIT.2025.3633259.

[5] H. Zhou and D. Feng, "Cross-rack aware recycle technique in erasure-coded data centers", *IEEE Transactions on Parallel and Distributed Systems*, vol. 37, no. 2, pp. 365–379, 2026. DOI: 10.1109/TPDS.2025.3639066.

[6] Y. Lei, X. Wang, B. Chen, M. Cao, and Z. Guo, "A weakly hybrid decoding algorithm for staircase codes via multi-level bit marking", *IEEE Transactions on Communications*, vol. 74, pp. 2580–2594, 2026. DOI: 10.1109/TCOMM.2025.3647757.

[7] D. Kim, J. Lee, W. Jung, M. B. Sullivan, and J. Kim, "Unity ECC: Unified memory protection against bit and chip errors", in *Proc. of SC23: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023, pp. 1–16. DOI: 10.1145/3581784.3607081.

[8] M. Blaum, "Extended integrated interleaved codes over any field with applications to locally recoverable codes", *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 936–956, 2020. DOI: 10.1109/TIT.2019.2934134.

[9] M. Blaum and S. R. Hetzler, "Extended product and integrated interleaved codes", *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1497–1513, 2018. DOI: 10.1109/TIT.2017.2789299.

[10] M. Kasahara, Y. Sugiyama, S. Hirasawa, and T. Namekawa, "New classes of binary codes constructed on the basis of concatenated codes and product codes", *IEEE Transactions on Information Theory*, vol. 22, no. 4, pp. 462–468, 1976. DOI: 10.1109/TIT.1976.1055582.

[11] A. M. Patel, "Two-level coding for error control in magnetic disk storage products", *IBM Journal of Research and Development*, vol. 33, no. 4, pp. 470–484, 1989. DOI: 10.1147/rd.334.0470.

[12] I.-S. Reed and G. Solomon, "Polynomial codes over certain finite fields", *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960. DOI: 10.1137/0108018.

[13] I.-S. Reed and X. Chen, "Reed-Solomon codes", in *Error-Control Coding for Data Networks. The Springer International Series in Engineering and Computer Science*, vol. 508. Springer, Boston, MA, 1999, pp. 233–284. DOI: 10.1007/978-1-4615-5005-1_6.

[14] M. Kumar and K. Kumar, "Implementation of a 2.5 Gbps RS(360,326) encoder for automotive Ethernet", in *Proc. of 2025 International Conference on Electronics, AI and Computing (EAIC)*, 2025, pp. 1–5. DOI: 10.1109/EAIC66483.2025.11101314.

[15] G. Deng, Z. Jiang, B. Bai, G. Zhang, X. Zhang, and H. Hou, "New Reed-Solomon codes with all XOR operation for better en/decoding performance", in *Proc. of 2024 IEEE Information Theory Workshop (ITW)*, 2024, pp. 401–406. DOI: 10.1109/ITW61385.2024.10807002.

[16] R. Wan and S. Zhu, "Constructions of l-MDS self-dual codes with flexible l via deleted generalized Reed-Solomon codes", *IEEE Transactions on Information Theory*, vol. 72, no. 2, pp. 961–984, 2026. DOI: 10.1109/TIT.2025.3645262.

[17] S. Madisetti, L. Tauz, and L. Dolecek, "Data availability attacks: Targeted sampling for 2-D Reed-Solomon codes in blockchain systems", in *Proc. of 2025 IEEE Information Theory Workshop (ITW)*, 2025, pp. 758–763. DOI: 10.1109/ITW62417.2025.11240529.

[18] S. Riasat and H. Mahdavifar, "Efficient covering using Reed-Solomon codes", in *Proc. of 2025 IEEE International Symposium on Information Theory (ISIT)*, 2025, pp. 1–6. DOI: 10.1109/ISIT63088.2025.11195316.

[19] E. Camps-Moreno, J. B. Lau, H. H. López, and W. Santos, "The permutation group of Reed-Solomon codes over arbitrary points", in *Proc. of 2025 IEEE International Symposium on Information Theory (ISIT)*, 2025, pp. 1–6. DOI: 10.1109/ISIT63088.2025.11195458.

[20] W. Cao, W. Zhang, and Y. Wang, "A efficient Reed-Solomon codes recognizer based on Galois field Fourier transform", in *Proc. of 2025 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSS)*, 2025, pp. 353–355. DOI: 10.1109/EDSSC64492.2025.11182709.

[21] K. Haymaker and E. McMillon, "A class of quasi-cyclic binary parity-check codes from Reed-Solomon codes", in *Proc. of 2025 13th International Symposium on Topics in Coding (ISTC)*, 2025, pp. 1–5. DOI: 10.1109/ISTC65386.2025.11154644.

[22] H. Liu, H. Wei, A. Wachter-Zeh, and M. Schwartz, "Linearized Reed-Solomon codes with support-constrained generator matrix and applications in multi-source network coding", *IEEE Transactions on Information Theory*, vol. 71, no. 2, pp. 895–913, 2025. DOI: 10.1109/TIT.2024.3503770.

[23] J. Wang *et al*., "A high-speed ASIC design of Reed-Solomon Erasure Code (RS-EC) decoders for fast data recovery in storage", *IEEE Embedded Systems Letters*, vol. 18, no. 1, pp. 11–14, 2026. DOI: 10.1109/LES.2025.3566686.

[24] G. Wang, H. Liu, and J. Luo, "Improved decoding algorithms for MDS and almost-MDS codes from twisted GRS codes", *IEEE Transactions on Information Theory*, vol. 71, no. 10, pp. 7688–7698, 2025. DOI: 10.1109/TIT.2025.3594174.

[25] H. Wang, Z. Wu, W. Zhang, Y. Wang, and Y. Liu, "Blind recognition of Reed-Solomon codes using efficient single-error correction", *IEEE Communications Letters*, vol. 30, pp. 642–646, 2026. DOI: 10.1109/LCOMM.2025.3647984.

[26] F. Hörmann and H. Bartz, "Syndrome-based error-erasure decoding of interleaved linearized Reed–Solomon codes", *IEEE Transactions on Information Theory*, vol. 72, no. 3, pp. 1590–1617, 2026. DOI: 10.1109/TIT.2025.3632981.

[27] D. Boucher and K. E. Nouetowa, "A decoding algorithm for skew cyclic generalized skew Reed-Solomon codes", in *Proc. of 2025 IEEE International Symposium on Information Theory (ISIT)*, 2025, pp. 1–6. DOI: 10.1109/ISIT63088.2025.11195485.

[28] A. Berman, Y. Shany, and I. Tamo, "Explicit subcodes of Reed-Solomon codes that efficiently achieve list decoding capacity", *IEEE Transactions on Information Theory*, vol. 71, no. 8, pp. 5898–5911, 2025. DOI: 10.1109/TIT.2025.3575899.

[29] T. M. Nisar and D. N.P., "Instruction decoding for dual-core soft-processors using Reed-Solomon codes", in *Proc. of 2024 Global

Conference on Communications and Information Technologies (GCCIT), 2024, pp. 1–7. DOI: 10.1109/GCCIT63234.2024.10862822.

[30] R. Con, "Anonymous Shamir's secret sharing via Reed-Solomon codes against permutations, insertions, and deletions", in *Proc. of 2025 IEEE International Symposium on Information Theory (ISIT)*, 2025, pp. 1–6. DOI: 10.1109/ISIT63088.2025.11195566.

[31] T. Wang, Z. Shi, J. Yang, and S. Liu, "Reed–Solomon-based private and secure distributed matrix multiplication from MDS-coded storage", *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 7243–7256, 2025. DOI: 10.1109/TIFS.2025.3587192.

[32] S. Kruglik, H. M. Kiah, S. H. Dau, and E. Yaakobi, "Recovering Reed–Solomon codes privately", *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2807–2821, 2025. DOI: 10.1109/TIFS.2025.3543123.

[33] N. Galkin and A. Levina, "Using Reed-Solomon codes to securely transmit data over radio frequencies using broadband communications", in *Proc. of 2024 International Conference Automatics and Informatics (ICAI)*, 2024, pp. 580–585. DOI: 10.1109/ICAI63388.2024.10851530.

[34] K. Wang and W. Ma, "Cooperative repair schemes of Reed–Solomon codes with three erasures", *IEEE Communications Letters*, vol. 29, no. 8, pp. 1829–1833, 2025. DOI: 10.1109/LCOMM.2025.3576872.

[35] B. Matuz, E. B. Yacoub, and S. Calabrò, "Serially concatenated codes for data center networks", in *Proc. of 2025 13th International Symposium on Topics in Coding (ISTC)*, 2025, pp. 1–5. DOI: 10.1109/ISTC65386.2025.11154486.

[36] Y. Yang, H. Cai, and X. Tang, "A multi-node repair scheme of Reed-Solomon codes", in *Proc. of 2025 IEEE International Symposium on Information Theory (ISIT)*, 2025, pp. 1–6. DOI: 10.1109/ISIT63088.2025.11195413.

[37] K. Wang, Z. Liu, R. Li, Y. Zhao, and Y. Zhang, "Asymptotically optimal repair of Reed-Solomon codes with small sub-packetization under rack-aware model", in *Proc. of 2025 IEEE Information Theory Workshop (ITW)*, 2025, pp. 674–679. DOI: 10.1109/ITW62417.2025.11240535.

[38] P. Beelen, R. Con, A. Gruica, M. Montanucci, and E. Yaakobi, "Reed-Solomon codes against insertions and deletions: Full-length and rate-1/2 codes", *IEEE Transactions on Information Theory*, vol. 72, no. 2, pp. 1149–1160, 2026. DOI: 10.1109/TIT.2025.3647216.

[39] Z. Liu, J. Xu, and Z. Zhang, "Calculating the I/O cost of linear repair schemes for RS codes evaluated on subspaces via exponential sums", *IEEE Transactions on Information Theory*, vol. 72, no. 2, pp. 994–1010, 2026. DOI: 10.1109/TIT.2025.3645309.

[40] R. T. Shwayyea, J. Bagherian, R. Sobhani, and A. Abdollahi, "A new lower bound for permutation codes using Reed–Solomon codes", *IEEE Access*, vol. 13, pp. 159412–159421, 2025. DOI: 10.1109/ACCESS.2025.3608654.

[41] J. Ren, J. Li, and T. Li, "Constructions of MSR code for optimal node repair in distributed storage", in *Proc. of 2024 International Conference on Meta Computing (ICMC)*, 2024, pp. 45–53. DOI: 10.1109/ICMC60390.2024.00012.

[42] J. Li, Z. Zhou, and J. Ding, "A reliability framework for proactive-tolerance Reed-Solomon storage systems", in *Proc. of 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2024, pp. 1484–1491. DOI: 10.1109/SMC54092.2024.10831258.

[43] M. Rakús, P. Farkaš, T. Páleník, and A. Daniš, "Five times extended Reed-Solomon codes applicable in memory storage systems", *IEEE Letters of the Computer Society*, vol. 2, no. 2, pp. 9–11, 2019. DOI: 10.1109/LOCS.2019.2911517.

[44] W.-C. Huffman and R. A. Brualdi, *Handbook of Coding Theory*, Amsterdam, Elsevier, 1998.

[45] F.-J. MacWillliams and N.-J. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1983.

[46] W.-W. Peterson and E.-J. Weldon Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, M.I.T. Press, 1972.

[47] M. Tomlinson, C.-J. Tjhai, M.-A. Ambroze, M. Ahmed, and M. Jibril, *Error Control Coding and Decoding*. Springer Open, 2017. DOI: 10.1007/978-3-319-51103-0.

[48] M. Rowshan and E. Viterbo, "Rate-compatible polar codes", in *Polar Codes: From Theory to Practice*. The Institute of Electrical and Electronics Engineers, Inc., 2026, pp. 281–292. DOI: 10.1002/9781119911760.ch11.

[49] R.-M. Roth and G. Seroussi, "On generator matrices of MDS codes", *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 826–830, 1985. DOI: 10.1109/TIT.1985.1057113.

[50] P. Huang, E. Yaakobi, and P. H. Siegel, "Multi-erasure locally recoverable codes over small fields", in *Proc. of 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017, pp. 1123–1130. DOI: 10.1109/ALLERTON.2017.8262863.

[51] M. Grassl, "Code Tables: Bounds on parameters of various types of code, December 2025. [Online]. Available: http:// https://www.codetables.de/