Code Automorphisms and Permutation Decoding of Certain Reed–Solomon Binary Images

Fabian Lim, *Student Member, IEEE*, Marc Fossorier, *Fellow, IEEE*, and Aleksandar Kavčić, *Senior Member, IEEE*

***Abstract*—We consider primitive Reed–Solomon (RS) codes over the field $\mathbb{F}_{2^m}$ of length $n = 2^m - 1$. Building on Lacan *et al.*'s results for the case of binary extension fields, we show that the binary images of certain two-parity symbol RS $[n, n-2, 3]$ code, have a code automorphism subgroup related to the general linear group $GL(m,2)$. For these codes, we obtain a code automorphism subgroup of order $m! \cdot |GL(m,2)|$. An explicit algorithm is given to compute a code automorphism (if it exists), that sends a particular choice of m binary positions, into binary positions that correspond to a single symbol of the RS code. If one such automorphism exists for a particular choice of m binary symbol positions, we show that there are at least $m!$ of them. Computationally efficient permutation decoders are designed for the two-parity symbol RS $[n, n-2, 3]$ codes. Simulation results are shown for the additive white Gaussian noise (AWGN) channel. For the finite fields $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$, we go on to derive subgroups of code automorphisms, belonging to binary images of certain RS codes that have three-parity symbols. A table of code automorphism subgroup orders, computed using the Groups, Algorithms, and Programming (GAP) software, is tabulated for the fields $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$, and $\mathbb{F}_{2^5}$.**

***Index Terms*—Automorphism group, binary images, finite fields, permutation decoding, Reed–Solomon (RS) codes.**

I. Introduction

The q-ary images of q^m-ary cylic codes over $\mathbb{F}_{q^m}$ have been widely studied. It is well-known that q-ary images of q^m-ary cyclic codes can be viewed as generalized concatenated codes (see, for example, [1]), where this connection can be exploited to obtain lower bounds on their minimum distance [2]. Furthermore, *code automorphism* subgroups of q-ary images belonging to q^m-ary cyclic codes have been obtained. Séguin has showed that certain classes of q^m-ary cyclic codes have q-ary images that are also cyclic [3]. Lacan *et al.* have derived code automorphisms belonging to the q-ary images of certain q^m-ary cyclic codes [4].

As pointed out by Lacan *et al.* in [4], code automorphism subgroups have interesting applications in the soft decoding of the q-ary images. Since most practical channels only accept binary inputs, we are particularly interested in binary images of cyclic codes over binary extension fields $\mathbb{F}_{2^m}$. One particular class of cyclic codes, namely Reed–Solomon (RS) codes, has been extremely popular in applications because they are maximum distance separable (MDS) codes. Furthermore, RS codes have efficient *hard decoders*. Other various methods for decoding RS codes have been proposed [5]–[8]. Of particular interest to us are soft decision reliability-based decoders, such as the generalized minimum distance (GMD) decoder [9], or the *Chase* decoder [10]. Reliability-based decoders utilize soft information from the channel, by assigning *symbol reliabilities* to the received symbols, in accordance to their probability of correct reception. Symbols with high reliability have high probability of being correct, and are thus taken in favor of less reliable ones. Reliability-based decoders such as the GMD and Chase decoder, have been shown to obtain significant gains over hard decision at modest complexities [9]–[11]. Their simplicity make them very attractive for hardware implementation.

On the other hand for nonbinary codes, the main drawback of directly applying GMD/Chase approaches to binary input channels, is that binary symbol reliabilities are not efficiently utilized. Nonbinary codes (such as RS codes) have to be first expanded into binary form before transmission across binary input channels. The received binary symbols are then mapped to nonbinary ones (before being decoded by the decoder) and the assignment of nonbinary symbol reliabilities has to be done by converting (the natural) binary symbol reliabilities. Information is invariably lost during this said conversion, which results in performance loss. This issue has been pointed out by many researchers, including Berlekemp *et al.* [12] and Jiang *et al.* [7]. Possibly, this issue may be addressed by permutation decoders. In [4], it was shown how to utilize code automorphisms of certain RS binary images, to build reliability-based decoders that directly use binary symbol reliabilities. The idea of permutation decoding is best explained using the example given in Section II.

II. Permutation Decoding and Automorphism Groups

In this section, the basic ideas behind the permutation decoder given in [4] is explained. The following notation is used throughout this paper. Let $\mathbb{F}_{2^m}$ be the Galois field of size 2^m. Primitive field elements are specifically denoted by α. In this paper, only RS codes of length $n = 2^m - 1$ are considered. Code parameters are written out as $[n, k, d_{\min}]$, where k and $d_{\min}$ denote the dimension, and minimum distance, respectively. Because n and 2 are relatively prime, the $[n, k, d_{\min}]$ RS code is uniquely defined by its set of nonzeros $\mathcal{N}$ (or by its set of zeros $\mathbb{F}_{2^m} \setminus (\mathcal{N} \cup \{0\})$, see [13]). Note that $|\mathcal{N}| = k$, and $\mathcal{N}$ contains k elements in $\mathbb{F}_{2^m} \setminus \{0\}$ that are *consecutive powers* of some primitive element $\alpha \in \mathbb{F}_{2^m}$. For any element $\beta \in \mathbb{F}_{2^m}$, let $\mathcal{C}(\beta)$ denote its set of cyclotomic conjugates (i.e., $\mathcal{C}(\beta) =$

Manuscript received August 11, 2009; revised May 03, 2010. Date of current version September 15, 2010. This work was supported by the National Science Foundation under Grant ECCS07-25649. The material in this paper has been presented in part at the ISIT 2008 and 2010 conferences.

F. Lim and A. Kavčić are with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu HI 96822 USA (e-mail: flim@hawaii.edu; alek@hawaii.edu).

M. Fossorier is with ETIS ENSEA/UCP/CNRS UMR-8051, 6 avenue du Ponceau, 95014, Cergy Pontoise, France (e-mail: mfossorier@ieee.org).

Communicated by Mario Blaum, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2010.2059633

$\{\beta, \beta^2, \ldots, \beta^{2^{r-1}}\}$ for some non-negative constant $r \leq m$). Let $\mathbb{F}_{2^m}^n$ denote the n-dimensional vector space over $\mathbb{F}_{2^m}$. The codewords of a RS code lie in some fixed k-dimensional vector subspace of $\mathbb{F}_{2^m}^n$.

To transmit a nonbinary $[n, k, d_{\min}]$ RS codeword across a binary input channel, we first need to map its nonbinary symbols (in $\mathbb{F}_2^m$) into binary symbols. We use $\mathbf{c}$ to denote an arbitrary codeword in a $[n, k, d_{\min}]$ RS code. A RS code is naturally viewed as an *ideal* in the ring $\mathcal{R}_{2^m} = \mathbb{F}_{2^m}[x]/(x^n - 1)$, and whenever we require the ideal representation of a RS code, we denote its codeword $\mathbf{c} \in \mathbb{F}_{2^m}^n$ as a polynomial $c(x) \in \mathcal{R}_{2^m}$. To be consistent with the coefficient indices of both the vector codeword $\mathbf{c}$ and the polynomial $c(x)$, the indices of a codeword vector $\mathbf{c}$ always start from 0 (i.e., $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]^T$). In all other instances, vector indices always start from 1. Note that all vectors are by default column vectors. We refer to both vector $\mathbf{c}$ and polynomial $c(x)$ as codewords. We denote $\mathbb{Z}_n$ to be the ring of integers modulo n. Finally, elements of a $m \times n$ matrix $\mathbf{A}$ will be indexed using the notation $A_{[i,j]}$, where $i \in \{1, 2, \ldots, m\}$ and $j \in \mathbb{Z}_n$ (i.e., matrix column indices are also fixed to start at 0).

Definition 1: Let $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_m]^T$ be a basis of $\mathbb{F}_{2^m}$ over the binary field $\mathbb{F}_2$. The **binary image** of a $[n, k, d_{\min}]$ RS code, is obtained by representing every vector codeword $\mathbf{c}$ as an $m \times n$ matrix

$$\mathcal{B}_M(\mathbf{c}) = \begin{bmatrix} c_{[1,0]} & c_{[1,1]} & c_{[1,2]} & \cdots & c_{[1,n-1]} \\ c_{[2,0]} & c_{[2,1]} & c_{[2,2]} & \cdots & c_{[2,n-1]} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{[m,0]} & c_{[m,1]} & c_{[m,2]} & \cdots & c_{[m,n-1]} \end{bmatrix}$$

where every column of $\mathcal{B}_M(\mathbf{c})$ satisfies

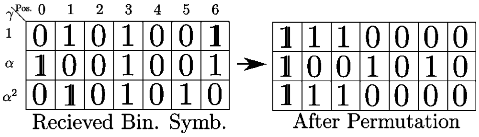
$$c_j = c_{[1,j]} \cdot \gamma_1 + c_{[2,j]} \cdot \gamma_2 + \cdots + c_{[m,j]} \cdot \gamma_m \ for\ all\ j \in \mathbb{Z}_n,$$

Alternatively, we represent every polynomial form codeword $c(x)$ as a vector

$$\mathcal{B}_P(\mathbf{c}) = \left[c^{(1)}(x), c^{(2)}(x), \ldots, c^{(m)}(x)\right]^T$$

of binary polynomials $c^{(i)}(x) = \sum_{j \in \mathbb{Z}_n} c_{[i,j]} x^j \in \mathcal{R}_2$ for all $i \in \{1, 2, \ldots, m\}$. Note that $c(x)$ can also be written as $c(x) = \sum_{i=1}^{m} c^{(i)}(x)\gamma_i$.

A permutation that acts on the set $\mathbb{Z}_n$, is a bijective mapping from $\mathbb{Z}_n$ to $\mathbb{Z}_n$. We write $\rho(j)$ to denote the element that $j \in \mathbb{Z}_n$ is mapped to. Let $b(x) = \sum_{j \in \mathbb{Z}_n} b_j x^j$ denote a binary polynomial in $\mathcal{R}_2$. We say that $b(x)$ is indexed by the set $\mathbb{Z}_n$. Let Φ_n denote the symmetric group[1] on the set $\mathbb{Z}_n$. A permutation $\rho \in \Phi_n$ acts on $b(x)$ by acting on its index set $\mathbb{Z}_n$, i.e., the monomial $x^j \in \mathcal{R}_2$ is acted on by ρ and permuted to $x^{\rho(j)}$. Similarly, permutations act on binary vectors $\mathbf{b} = [b_0, b_1, \ldots, b_{n-1}]^T$ that are indexed by $\mathbb{Z}_n$. Permutations are explicitly denoted using *cycle notation* (e.g., the permutation $\rho = (2, 5, 3)$ means that $\rho(2) = 5$, $\rho(5) = 3$, and $\rho(3) = 2$).



γ \ Pos.	0	1	2	3	4	5	6
1	0	1	0	1	0	0	$\mathbb{1}$
α	$\mathbb{1}$	0	0	1	0	0	1
α^2	0	$\mathbb{1}$	0	1	0	1	0

Recieved Bin. Symb.

$\mathbb{1}$	1	1	0	0	0	0
$\mathbb{1}$	0	0	1	0	1	0
$\mathbb{1}$	1	1	0	0	0	0

After Permutation

Fig. 1. Example of permutation decoding. The locations marked by $\mathbb{1}$ denote erroneous binary symbols.

We present the idea of permutation decoding [4] using the example depicted in Fig. 1. We choose a primitive element $\alpha \in \mathbb{F}_{2^3}$ that satisfies $\alpha^3 = \alpha + 1$, and we choose the basis $\gamma = [1, \alpha, \alpha^2]^T$. Consider a (double-parity) 1-symbol correcting RS code over $\mathbb{F}_{2^3}$, with zeros $\{1, \alpha\}$ (thus, $\mathcal{N} = \{\alpha^2, \alpha^3, \ldots, \alpha^6\}$). The codeword $\mathbf{c} = [0, 1, 0, \alpha^5, 0, \alpha^2, \alpha]^T$ is transmitted across a binary input channel by first obtaining its binary image $\mathcal{B}_M(\mathbf{c})$ (see Definition 1), and then sending the binary symbols in $\mathcal{B}_M(\mathbf{c})$ across the channel. The binary image $\mathcal{B}_M(\mathbf{c})$ written explicitly as a vector of polynomials $\mathcal{B}_P(\mathbf{c})$ is given as $\mathcal{B}_P(\mathbf{c}) = [c^{(1)}(x), c^{(2)}(x), c^{(3)}(x)]^T = [x + x^3, x^3 + x^6, x^3 + x^5]^T$. As shown in Fig. 1, the received binary symbols are arranged in a $m \times n$ grid, such that each column corresponds to a sent symbol in $\mathcal{B}_M(\mathbf{c})$. Assume that 3 received symbols are in error (marked in bold in Fig. 1). Consider what happens if we permute the first row of the received symbol grid by (0, 4, 6)(2, 5, 3), the second by (1, 4, 2)(3, 5, 6), and the third by (0, 3, 1)(2, 4, 5). Observe that all the erroneous binary symbols get permuted to the 0th symbol location. Furthermore, we see that the binary image $\mathcal{B}_P(\mathbf{c}) = [x + x^3, x^3 + x^6, x^3 + x^5]^T$ is permuted to $\mathcal{B}_P(\mathbf{c}') = [x + x^2, x^3 + x^5, x + x^2]^T$, where $\mathbf{c}' = [0, \alpha^6, \alpha^6, \alpha, 0, \alpha, 0]^T$ is also a codeword. Hence, if we decode the permuted grid of received symbols, we would decode to $\mathbf{c}'$ and correct all 3 errors, which now appear within the same symbol. A decoder that adopts this strategy of permuting received binary symbols, with the aim of sending erronous symbols into a single symbol location, is known as a *permutation decoder*.

From the previous example, note that the permutation applied to the received symbols must not destroy the code structure, in the sense that the transmitted binary image $\mathcal{B}_M(\mathbf{c})$ must be sent to a binary image $\mathcal{B}_M(\mathbf{c}')$ of a (possibly different) codeword $\mathbf{c}'$. Since the transmitted $\mathcal{B}_M(\mathbf{c})$ is unknown at the receiver end, the applied permutation must be capable of sending any binary image codeword to some binary image codeword.

Note that the RS binary image is a binary linear block code [11], [13]. Define the matrix index set $\mathcal{I} \triangleq \{[i, j] : 1 \leq i \leq m, j \in \mathbb{Z}_n\}$.

Definition 2: The **automorphism group** of a RS binary image, is the group of all permutations acting on the index set $\mathcal{I} \triangleq \{[i, j] : 1 \leq i \leq m, j \in \mathbb{Z}_n\}$, which preserve the additive group structure of the RS binary image. An element of the automorphism group is called a **code automorphism**.

Code automorphisms for q-ary images of q^m-ary cyclic codes have been obtained by Séguin [3] and Lacan *et al.* [4]. In particular, Lacan *et al.* obtained code automorphisms for the binary images of two-parity $[n, n-2, 3]$ RS codes, as well as certain triple-parity RS $[n, n-3, 4]$ binary images over $\mathbb{F}_{2^3}$. We recap

[1] The symmetric group defined on a set A contains all possible permutations of elements in the set A.

and build on Lacan *et al.*'s results (for binary extension fields) in Sections III–VI of this paper. In Section III, we review the algebraic structure of certain double-parity RS $[n, n-2, 3]$ binary images (from which these code automorphisms will be derived in the later sections). In Section IV, we first show that the binary images of certain double-parity RS $[n, n-2, 3]$ codes have a code automorphism subgroup of order $m! \cdot \prod_{r=0}^{m-1}(2^m - 2^r)$. Our automorphism subgroup contains the code automorphisms derived in [4] for binary extension fields. In Section IV-A, we develop theory that leads to efficient RS $[n, n-2, 3]$ permutation decoders. In Sections IV-B–C, we describe the implementation of two RS $[n, n-2, 3]$ permutation decoders (termed Algorithms A, and B, respectively) and show their performances. In Section IV-D, we provide methods to further reduce the permutation decoder complexities. In Section V, we then go on to consider the binary images of certain triple-parity RS codes over $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$. We show that RS $[7, 4, 4]$ codes over $\mathbb{F}_{2^3}$, have binary image code automorphism subgroups of order $3! \cdot 7 \cdot 6 \cdot 4 = 1008$, that are similar to automorphism subgroups obtained for the double-parity case. Also, we show that certain RS $[15, 12, 4]$ codes over $\mathbb{F}_{2^4}$, have a subgroup of 4 automorphisms. Section VI concludes the paper, where we present a table of code automorphism group orders, computed using the Groups, Algorithms, and Programming (GAP) software for the fields $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^5}$.

Note that this paper is organized in such a way that the sections on the decoder designs are self-contained. Readers who are only interested in implementations of permutation decoding Algorithms A and B, may focus only on Subsections III-B through III-D.

III. Structure of Certain Double-Parity Symbol RS Binary Images

In this section, we recap Lacan *et al.*'s presentation on the algebraic structure of the two-parity symbol $[n, n-2, 3]$ RS binary images. The considered RS codes have zeros $\{1, \alpha\}$ (or $\mathcal{N} = \{\alpha^2, \alpha^3, \ldots, \alpha^{n-1}\}$), where α is primitive in $\mathbb{F}_{2^m}$. Other excellent resources on the algebraic structure of RS binary images can be found in [1], [3], and [14]. The main aim of this paper is to obtain code automorphisms, and the material in this section is condensed from [4] to suit our purposes. It was pointed out in [4] that the code automorphisms of RS binary images, can be obtained from its dual code, which in our case is the binary image of the RS$[n, 2, n-1]$ with $\mathcal{N} = \{1, \alpha^{-1}\}$. This is because both primal and dual codes have the same automorphism group [3], [4]. Note that if the primal code is imaged under the basis γ, the dual has to be imaged under the trace-dual basis $\gamma^{\perp}$ ([3 Lemma 6]).

Note that because the binary images are taken with respect to a linear basis γ (see Definition 1), the binary images themselves are binary linear block codes [1], [3], [4], [13]. Thus, the binary images are represented using $\mathbb{F}_2$-generator matrices. In [4], the $\mathbb{F}_2$-generator matrix of the RS binary image is shown to be related to binary *irreducible* cyclic codes. Recall that a binary cyclic code is a principal ideal in the ring $\mathcal{R}_2$ which may be generated by a polynomial $\theta(x) \in \mathcal{R}_2$ (termed the *generator polynomial*). We denote the cyclic code generated by $\theta(x)$ as $\langle\theta(x)\rangle$. We recall the following well-known facts (see [13] for proofs), stated in the following two propositions.

Proposition 1: Recall that the set of conjugates of an element $\beta \in \mathbb{F}_{2^m}$ is denoted as $\mathcal{C}(\beta)$. A binary **irreducible cyclic code**, of length n and dimension $|\mathcal{C}(\beta)|$, is generated by a polynomial $\theta_\beta(x) \in \mathcal{R}_2$ which satisfies

$$\begin{aligned} \theta_\beta(\beta') &= 1 \; for \; all \; \beta' \in \mathcal{C}(\beta) \\ \theta_\beta(\beta') &= 0 \; for \; all \; \beta' \notin \mathcal{C}(\beta). \end{aligned} \tag{1}$$

The polynomial $\theta_\beta(x)$ which satisfies (1) is known as the **idempotent** [13] related to the set of conjugates $\mathcal{C}(\beta)$. We see from (1) that $\theta_\beta(x)^2 = \theta_\beta(x)$; thus, the idempotent is invariant under the squaring operation.

Proposition 2: Let $\alpha \in \mathbb{F}_{2^m}$ be primitive. The irreducible cyclic code $\langle\theta_\alpha(x)\rangle$ generated by the idempotent $\theta_\alpha(x)$ consists only of cyclic shifts of $\theta_\alpha(x)$ and the polynomial 0.

For notational convenience, we define $\breve{\alpha} \triangleq \alpha^{-1}$. Recall that $\mathbb{Z}_n$ denotes the ring of integers modulo n. We adopt the polynomial view of the binary image (see Definition 1). It was shown in [4] that the generator matrix of the RS $[n, 2, n-1]$ binary image with $\mathcal{N} = \{1, \alpha^{-1}\} = \{1, \breve{\alpha}\}$ (which is dual to the binary image of the RS $[n, n-2, 3]$ with zeros $\{1, \alpha\}$) is given by the following $2m \times m$ matrix with entries in $\mathcal{R}_2 = \mathbb{F}_2[x]/(x^n - 1)$

$$\begin{bmatrix} \theta_1(x) & 0 & \cdots & 0 \\ 0 & \theta_1(x) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \theta_1(x) \\ \theta_{\breve{\alpha}}(x)x^{u_1} & \theta_{\breve{\alpha}}(x)x^{u_2} & \cdots & \theta_{\breve{\alpha}}(x)x^{u_m} \\ \vdots & \vdots & & \vdots \\ \theta_{\breve{\alpha}}(x)x^{u_1+m-1} & \theta_{\breve{\alpha}}(x)x^{u_2+m-1} & \cdots & \theta_{\breve{\alpha}}(x)x^{u_m+m-1} \end{bmatrix} \tag{2}$$

where $u_i \in \mathbb{Z}_n$. Any binary image codeword $\mathcal{B}_P(\mathbf{c})$ (or $\mathcal{B}_M(\mathbf{c})$, see Definition 1) may be formed, by taking $\mathbb{F}_2$-linear combinations of the rows of its $\mathbb{F}_2$-generator matrix. To determine the values $\mathbf{u} \triangleq [u_1, u_2, \ldots, u_m]^T$, we require the next proposition by Lacan *et al.*

Proposition 3: (Lacan *et al.* [4]). Let $\alpha \in \mathbb{F}_{2^m}$ be the primitive element, and consider a dimension-1 RS code with $\mathcal{N} = \{\alpha\}$. Define the matrix $\mathcal{M}_\alpha$ as

$$\mathcal{M}_\alpha = \begin{bmatrix} \theta_\alpha(x)x^{u_1} & \theta_\alpha(x)x^{u_2} & \cdots & \theta_\alpha(x)x^{u_m} \\ \vdots & \vdots & & \vdots \\ \theta_\alpha(x)x^{u_1+m-1} & \theta_\alpha(x)x^{u_2+m-1} & \cdots & \theta_\alpha(x)x^{u_m+m-1} \end{bmatrix}. \tag{3}$$

If the vector $\mathbf{u} = [u_1, u_2, \ldots, u_m]^T$ is determined by:

- taking the binary image (with respect to basis γ) of the polynomial

$$\theta_\alpha(x) \prod_{\beta \in \mathcal{C}(\alpha) \setminus \alpha} x - \beta$$

to obtain

$$\theta_\alpha(x)\left(b_0(x)\gamma_1 + \cdots + b_{m-1}(x)\gamma_m\right)$$

TABLE I
u VECTORS COMPUTED FOR DIMENSION-1 RS CODES WITH $\mathcal{N} = \breve{\alpha}$ AND $\gamma = \{1, \alpha, \dots, \alpha^{m-1}\}$

$\mathbb{F}_{2^m}$	$\mathbf{u}$	prim. elem. α	trace-dual basis $\gamma^{\perp}$
$\mathbb{F}_{2^3}$	$[2, 1, 0]^T$	$\alpha^3 = \alpha + 1$	$[1, \alpha^2, \alpha]^T$
$\mathbb{F}_{2^4}$	$[2, 1, 0, 14]^T$	$\alpha^4 = \alpha + 1$	$[\alpha^{-1}, \alpha^2, \alpha, 1]^T$
$\mathbb{F}_{2^5}$	$[30, 29, 28, 27, 26]^T$	$\alpha^5 = \alpha^2 + 1$	$[\alpha^{-5}, \alpha^{-6}, \alpha^{-2}, \alpha^{-3}, \alpha^{-4}]^T$
$\mathbb{F}_{2^6}$	$[4, 3, 2, 1, 0, 62]^T$	$\alpha^6 = \alpha + 1$	$[\alpha^{-1}, \alpha^4, \alpha^3, \alpha^2, \alpha^1, 1]^T$

where the polynomials $b_i(x) \neq 0$ lie in the ring $\mathcal{R}_2$;

- obtaining $u_i \in \mathbb{Z}_n$ that satisfies $\theta_\alpha(x)x^{u_i} = \theta_\alpha(x)b_i(x)$ for all $i \in \{1, 2, \dots, m\}$.

Then $\mathcal{M}_\alpha$ is the $\mathbb{F}_2$-generator matrix for the RS code $\mathcal{N} = \{\alpha\}$ imaged under the basis γ.

See [4] for the proof of Proposition 3. Note that for primitive $\alpha \in \mathbb{F}_{2^m}$, the nonzero elements of the irreducible cyclic code $\langle\theta_\alpha(x)\rangle$ are the n cyclic shifts of $\theta_\alpha(x)$ (see Proposition 2). This implies $\theta_\alpha(x)b(x) = \theta_\alpha(x)x^u$ for some nonzero $b(x) \in \mathcal{R}_2$ and $u \in \mathbb{Z}_n$. In [4], it was pointed out that because the dimension-1 RS code with $\mathcal{N} = \{\breve{\alpha}\}$ is a *subcode* of the dimension-2 RS code with $\mathcal{N} = \{1, \breve{\alpha}\}$, the $\mathbf{u}$ vector in (2) is determined as in the statement of Proposition 3, by using the primitive element $\breve{\alpha}$ and the trace-dual basis $\gamma^{\perp}$. Table I shows the $\mathbf{u}$ vectors computed for RS $[n, 2, n-1]$ codes with $\mathcal{N} = \{1, \breve{\alpha}\}$ over various fields $\mathbb{F}_{2^m}$, imaged under the trace-dual of the canonical basis $\gamma = [1, \alpha, \dots, \alpha^{m-1}]^T$.

IV. CODE AUTOMORPHISMS AND PERMUTATION DECODING OF CERTAIN DOUBLE-PARITY RS BINARY IMAGES

Building on the results of [4], we now show how to obtain $[n, n-2, 3]$ RS binary image automorphisms, from the $\mathbb{F}_2$-generator matrix of its dual code. The code automorphism subgroup found in [4], is a *direct product* of two groups with orders $m!$ and $m \cdot n$, respectively. Define the symmetric group on the sets $\{1, 2, \dots, m\}$ and $\mathbb{Z}_n$, as Ω_m and Φ_n, respectively. Define $\varsigma \in \Phi_n$ as an elementary cyclic shift to the right by one (i.e., $\varsigma(i) = i + 1$). A composition of two permutations ρ_1, ρ_2, whereby ρ_1 acts before ρ_2, is written as $\rho_2\rho_1$. We sometimes write $A \xrightarrow{\rho} A'$, to indicate that A' results from the action of a permutation ρ on A.

Recall the index set $\mathcal{I} \triangleq \{[i, j] : 1 \leq i \leq m, j \in \mathbb{Z}_n\}$. Recall that $\mathcal{B}_M(\mathbf{c})$ is indexed by the set $\mathcal{I}$ (see Definition 1). Let $\mathbf{a} = [a^{(1)}(x), a^{(2)}(x), \dots, a^{(m)}(x)]^T$ be an arbitrary length-m vector with entries in $\mathcal{R}_2$. The set $\mathcal{I}$ indexes $\mathbf{a}$ as follows; an element $[i, j] \in \mathcal{I}$ points to the jth coefficient of the ith entry $a^{(i)}(x)$. The set $\mathcal{I}$ indexes the rows of the $\mathbb{F}_2$-matrix (2) similarly. Recall that both $\sigma(i)$ and $\rho(j)$ denote the elements that $i \in \{1, 2, \dots, m\}$ and $j \in \mathbb{Z}_n$ are mapped to. Finally, we define the action of $\sigma \in \Omega_m$ (and $\rho \in \Phi_n$) on $\mathcal{I}$, such that $[i, j] \xrightarrow{\sigma} [\sigma(i), j]$ (and $[i, j] \xrightarrow{\sigma} [i, \rho(j)]$).

Proposition 4: (Lacan *et al.* [4]).Let $\mathcal{P}^{(1)}$ be a permutation group acting on the index set $\mathcal{I}$. Let every element $g \in \mathcal{P}^{(1)}$ act on $\mathcal{B}_P(\mathbf{c}) = [c^{(1)}(x), c^{(2)}(x), \dots, c^{(m)}(x)]^T$ by sending the index $[i, j] \in \mathcal{I}$ to $[\sigma(i), j - u_i + u_{\sigma(i)}]$, for some $\sigma \in \Omega_m$. Then $\mathcal{P}^{(1)}$ is a subgroup of code automorphisms of the RS $[n, n-2, 3]$ binary image with zeros $\{1, \alpha\}$ and $|\mathcal{P}^{(1)}| = m!$.

Proof: Since the $\mathbb{F}_2$-matrix (2) is equivalent to a parity-check matrix of the RS $[n, n-2, 3]$ binary image, it suffices to show that its $\mathbb{F}_2$-rowspace is invariant under these permutations. We verity that $\theta_1(x) = \sum_{i=0}^{n-1} x^i$ from (1); thus, clearly the $\mathbb{F}_2$-rowspace of the first m rows of (2), are invariant under these permutations. For the last m rows of (2), pick any $\sigma \in \Omega_m$ and row $[\theta_{\breve{\alpha}}(x)x^{u_1+r}, \theta_{\breve{\alpha}}(x)x^{u_2+r}, \dots, \theta_{\breve{\alpha}}(x)x^{u_m+r}]$ for some $r \in \{0, 1, \dots, m-1\}$. It can be verified that the chosen row is permuted as follows. Its ith entry is first cyclically shifted as

$$\theta_{\breve{\alpha}}(x)x^{u_i+r} \overset{\varsigma^{-u_i}}{\rightarrow} \theta_{\breve{\alpha}}(x)x^{r} \overset{\varsigma^{u_{\sigma(i)}}}{\rightarrow} \theta_{\breve{\alpha}}(x)x^{u_{\sigma(i)}+r}.$$

Next, the entry $\theta_{\breve{\alpha}}(x)x^{u_{\sigma(i)}+r}$ is sent to the $\sigma(i)$th position. Thus, the last m rows of (2) are invariant under the permutations in $\mathcal{P}^{(1)}$; hence, we conclude that $\mathcal{P}^{(1)}$ is indeed a group of code automorphisms of the RS $[n, n-2, 3]$ binary image. The size $|\mathcal{P}^{(1)}| = m!$ follows from the fact that there exists $m!$ unique choices for $\sigma \in \Omega_m$. ■

For the next proposition, recall that for any primitive $\alpha \in \mathbb{F}_{2^m}$, the ideal $\langle\theta_{\breve{\alpha}}(x)\rangle$ is generated by $\theta_{\breve{\alpha}}(x)$, and that the nonzero codewords in $\langle\theta_{\breve{\alpha}}(x)\rangle$ are cyclic shifts of $\theta_{\breve{\alpha}}(x)$. The following result is taken from [15]. Denote the automorphism group of the code $\langle\theta_{\breve{\alpha}}(x)\rangle$ as $Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$.

Proposition 5: (Lim *et al.* [15]). Let $\mathcal{P}^{(2)}$ be a permutation group acting on the index set $\mathcal{I}$. Let every $h \in \mathcal{P}^{(2)}$ act on $\mathcal{B}_P(\mathbf{c}) = [c^{(1)}(x), c^{(2)}(x), \dots, c^{(m)}(x)]^T$ by sending the index $[i, j] \in \mathcal{I}$ to $[i, \rho(j - u_i) + u_i]$, for some corresponding $\rho \in Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$. Then $\mathcal{P}^{(2)}$ is a subgroup of code automorphisms of the $[n, n-2, 3]$ RS binary image with zeros $\{1, \alpha\}$, and $|\mathcal{P}^{(2)}| = |Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)|$.

Proof: Note that the first element in the index tuple $[i, j]$ is invariant under the action of permutations in $\mathcal{P}^{(2)}$. The first m rows of (2) are invariant under the action of any element in $\mathcal{P}^{(2)}$. For the last m rows of (2), pick any $\rho \in Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$ and any row $[\theta_{\breve{\alpha}}(x)x^{u_1+r}, \theta_{\breve{\alpha}}(x)x^{u_2+r}, \dots, \theta_{\breve{\alpha}}(x)x^{u_m+r}]$ for some $r \in \{0, 1, \dots, m-1\}$. We see that the ith entry of the chosen row gets permuted as

$$\theta_{\breve{\alpha}}(x)x^{u_i+r} \overset{\varsigma^{-u_i}}{\rightarrow} \theta_{\breve{\alpha}}(x)x^{r} \xrightarrow{\rho} \theta_{\breve{\alpha}}(x)x^{r'} \overset{\varsigma^{u_i}}{\rightarrow} \theta_{\breve{\alpha}}(x)x^{u_i+r'}$$

where the second arrow follows from Proposition 2 and $r' \in \mathbb{Z}_n$. Thus, the chosen row is permuted to

$$[\theta_{\breve{\alpha}}(x)x^{u_1+r'}, \theta_{\breve{\alpha}}(x)x^{u_2+r'}, \dots, \theta_{\breve{\alpha}}(x)x^{u_m+r'}]^T.$$

Because the m last rows of (2) were linearly independent over $\mathbb{F}_2$, then so are they after applying permutations in $\mathcal{P}^{(2)}$. Thus, we conclude that all elements in $\mathcal{P}^{(2)}$ leave the row-space of (2) invariant. To show $|\mathcal{P}^{(2)}| = |Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)|$, it suffices to show that the permutations $\varsigma^{u_1}\rho\varsigma^{-u_1}$ are unique[2] for any $\rho \in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$. Clearly, this is true because if there exists $\rho, \rho' \in Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$ such that $\varsigma^{u_1}\rho\varsigma^{-u_1} = \varsigma^{u_1}\rho'\varsigma^{-u_1}$, then it must be that $\rho = \rho'$. ■

Remark 1: The group of code automorphisms $\mathcal{P}^{(1)}$ includes the code automorphisms previously found in [4] for the special case of binary extension fields. The code automorphisms found in [4] are obtained as in Proposition 5, by limiting $\rho \in Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$ to be an element of a cyclic subgroup. This cyclic

[2] For any group A and element $a \in A$, there exists an automorphism of A onto aAa^{-1}.

subgroup of size m, is generated by an element that sends $i \rightarrow 2 \cdot i$ for all $i \in \mathbb{Z}_n$.

We next show that permutations in the group $\mathcal{P}^{(1)}$ commute with permutations in the group $\mathcal{P}^{(2)}$ (and vice-versa). To clearly distinguish permutations belonging to the two different groups $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$, we use the notation g and h, to indicate $g \in \mathcal{P}^{(1)}$ and $h \in \mathcal{P}^{(2)}$.

Proposition 6: Any permutation $g \in \mathcal{P}^{(1)}$ and any permutation $h \in \mathcal{P}^{(2)}$ commute.

Proof: The proof is a straight-forward verification of the claim. Recall from Propositions 4 and 5 that there exist a one-to-one correspondence to the permutations g and h, with some permutations $\sigma \in \Omega_m$ and $\rho \in Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$, respectively. If we compose $g \in \mathcal{P}^{(1)}$ and $h \in \mathcal{P}^{(2)}$ as hg, then we see that the image of any index $[i,j] \in \mathcal{I}$ is given as

$$[i,j] \xrightarrow{g} \left[\sigma(i), j - u_i + u_{\sigma(i)}\right] \xrightarrow{h} \left[\sigma(i), \rho(j-u_i) + u_{\sigma(i)}\right]$$

and if we compose $g \in \mathcal{P}^{(1)}$ and $h \in \mathcal{P}^{(2)}$ as gh, then

$$[i,j] \xrightarrow{h} [i, \rho(j-u_i) + u_i] \xrightarrow{g} \left[\sigma(i), \rho(j-u_i) + u_{\sigma(i)}\right].$$

Thus, we see that both images of any $[i,j] \in \mathcal{I}$ are equal for both compositions gh and hg; therefore, $gh = hg$. ■

Because elements in both $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ commute (see Proposition 6), then any composition of permutations, selected arbitrarily from the two groups $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$, can be written in the form gh, for some $g \in \mathcal{P}^{(1)}$ and $h \in \mathcal{P}^{(2)}$. Therefore, any permutation that is obtained by composition from $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ must lie in the product group $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$, defined as $\mathcal{P}^{(1)}\mathcal{P}^{(2)} \triangleq \{gh : g \in \mathcal{P}^{(1)}, h \in \mathcal{P}^{(2)}\}$. The final proposition of this subsection shows that the product group $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$, has size $m! \cdot |Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)|$, and is isomorphic to the direct product $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$. The proof of the following proposition requires the fact that $\mathcal{P}^{(1)} \cap \mathcal{P}^{(2)} = \{i_d\}$, where i_d is the *identity element* of both groups $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$.

Proposition 7: The automorphism subgroup $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ of the double-parity symbol RS $[n, n-2, 3]$ binary image (with zeros $\{1, \alpha\}$), is isomorphic to the direct product $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$ and has size $m! \cdot |Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)|$

Proof: Because $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ commute, it suffices to prove that the group $\mathcal{P}^{(1)}\mathcal{P}^{(2)} = \{gh : g \in \mathcal{P}^{(1)}, h \in \mathcal{P}^{(2)}\}$ has order $|\mathcal{P}^{(1)}| \cdot |\mathcal{P}^{(2)}|$. Consider some g, $g' \in \mathcal{P}^{(1)}$ and some h, $h' \in \mathcal{P}^{(2)}$, such that $gh = g'h'$. This implies that $gg'^{-1} = h'h^{-1}$, where $gg'^{-1} \in \mathcal{P}^{(1)}$ and $h'h^{-1} \in \mathcal{P}^{(2)}$. But because $gg'^{-1} = h'h^{-1}$ and since $\mathcal{P}^{(1)} \cap \mathcal{P}^{(2)} = i_d$, then $gg'^{-1} = h'h^{-1} = i_d$. By the uniqueness of inverses, we conclude that $g = g'$ and $h = h'$. Thus, we conclude that $|\mathcal{P}^{(1)}\mathcal{P}^{(2)}| = |\mathcal{P}^{(1)}| \cdot |\mathcal{P}^{(2)}|$. ■

Corollary 1: The order of the automorphism group of the double-parity symbol RS $[n, n-2, 3]$ binary image is at least $m! \cdot |Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)|$.

Proof: Follows from Proposition 7. ■

A. Finding Code Automorphisms in $\mathcal{P}^{(2)}$ That Send Chosen m Binary Locations Into a Single Symbol Location

In this subsection, we expose techniques to efficiently determine code automorphisms in the group $\mathcal{P}^{(2)}$, that send a chosen set of m binary locations into a single symbol. Our strategy is to first show that $\mathcal{P}^{(2)}$ is isomorphic to the general linear group denoted $GL(m, 2)$ (i.e., the matrix group that contains all $m \times m$ invertible binary matrices). We make use of many previously known results found in [13].

Proposition 8: For any primitive $\alpha \in \mathbb{F}_{2^m}$, the cyclic code $\langle\theta_\alpha(x)\rangle$ is **equivalent** to the binary **simplex** code [13].

Proof: The polynomial $1 + \theta_\alpha(x)$ can be verified to be invariant under squaring; therefore, it is an idempotent and generates an ideal $\langle 1 + \theta_\alpha(x)\rangle$. The only zeros of $1 + \theta_\alpha(x)$ are elements in the set of conjugates $\mathcal{C}(\alpha)$; hence, the ideal $\langle 1 + \theta_\alpha(x)\rangle$ is equivalent to the binary Hamming code. Thus, the ideal $\langle\theta_\alpha(x)\rangle$ is equivalent to the simplex code (i.e., the Hamming code dual) because $\theta_\alpha(x)(1 + \theta_\alpha(x)) = \theta_\alpha(x) + \theta_\alpha(x) = 0$. ■

Since $\breve{\alpha}$ is primitive, then Proposition 8 holds and $\langle\theta_{\breve{\alpha}}(x)\rangle$ is equivalent to the simplex code. Let $\mathbf{S}$ denote an $m \times n$ matrix that generates $\langle\theta_{\breve{\alpha}}(x)\rangle$, and we take $\mathbf{S}$ to be the *circulant* matrix

$$\mathbf{S} = \begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_{n-m+1} & \cdots & \theta_{n-1} \\ \theta_{n-1} & \theta_0 & \cdots & \theta_{n-m-2} & \cdots & \theta_{n-2} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \theta_{n-m+1} & \theta_{n-m+2} & \cdots & \theta_0 & \cdots & \theta_{n-m} \end{bmatrix} \tag{4}$$

whose first row corresponds to $\theta_{\breve{\alpha}}(x)$ (i.e., $\sum_{j \in \mathbb{Z}_n} \theta_j x^j = \theta_{\breve{\alpha}}(x)$). Since $\mathbf{S}$ generates a code that is equivalent to the simplex code, then $\mathbf{S}$ contains all possible nonzero length-m vectors as its columns. Recall that $GL(m, 2)$ denotes the general linear group over $\mathbb{F}_2$, and that $Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$ denotes the automorphism group of the simplex code $\langle\theta_{\breve{\alpha}}(x)\rangle$.

Proposition 9: There exists a **one-to-one correspondence** between an automorphism $\rho \in Aut\,(\langle\theta_{\breve{\alpha}}(x)\rangle)$, and an element $\mathbf{K}$ in the general linear group $GL(m, 2)$ [13].

Proof: Take any $\mathbf{K} \in GL(m, 2)$. Recall that $\mathbf{S}$ generates $\langle\theta_{\breve{\alpha}}(x)\rangle$. Note that the left action of $\mathbf{K}$ on the columns of $\mathbf{S}$, results in a bijective mapping $\mathbb{F}_2^m \backslash \{\mathbf{0}\} \mapsto \mathbb{F}_2^m \backslash \{\mathbf{0}\}$. Thus, the matrix $\mathbf{KS}$ can be obtained by some column permutation $\rho \in \Phi_n$ on the matrix $\mathbf{S}$. Since $\mathbf{KS}$ generates the same simplex code $\langle\theta_{\breve{\alpha}}(x)\rangle$, then $\rho \in Aut\,(\langle\theta_{\breve{\alpha}}(x)\rangle)$. Finally, $\mathbf{KS}$ is unique for any $\mathbf{K} \in GL(m, 2)$. Thus, we are done. ■

In other words, the order $|Aut\,(\langle\theta_{\breve{\alpha}}(x)\rangle)| = |GL(m, 2)| = (2^m - 1)(2^m - 2) \cdots (2^m - 2^{m-1})$ [13] (follows from counting the number of ways to construct invertible $m \times m$ binary matrices). Also, because $Aut\,(\langle\theta_{\breve{\alpha}}(x)\rangle)$ is isomorphic to $GL(m, 2)$, the group $Aut\,(\langle\theta_{\breve{\alpha}}(x)\rangle)$ is *2-transitive* [13], i.e., it contains a permutation that sends any two indices (in $\mathbb{Z}_n$) to any two indices (in $\mathbb{Z}_n$).

Definition 3: We define a **location vector** $\mathbf{j} = [j_1, j_2, \ldots, j_m]^T \in \mathbb{Z}_n^m$, where each component $j_i \in \mathbb{Z}_n$ points to the binary symbol index $[i, j_i] \in \mathcal{I}$.

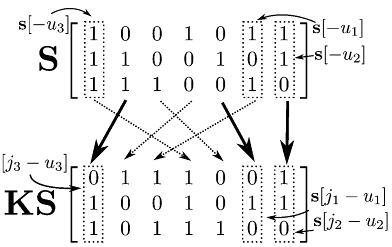


Fig. 2. Simplex automorphism that sends $\mathbf{j}=[6,0,1]^T$ to the first symbol.

For example, a location vector $\mathbf{j}\in\mathbb{Z}_7^3$ points to *one* binary location in each row of the (3×7) grid shown in Fig. 1. We would like to consider which automorphisms (if they exist) in $\mathcal{P}^{(2)}$ send the locations specified in some $\mathbf{j}\in\mathbb{Z}_n^m$, to a single nonbinary symbol position. As motivated in Section II, if we can identify such automorphisms, then we can attempt to correct any binary symbol errors in the locations $\mathbf{j}$. Since the RS code is cyclic, it suffices to only consider automorphisms $h\in\mathcal{P}^{(2)}$ that send $[i,j_i]\xrightarrow{h}[i,0]$ for all $i\in\{1,2,\ldots,m\}$ (i.e., the locations $\mathbf{j}\in\mathbb{Z}_n^m$ are sent to the 0th symbol position). Recall from Proposition 5 that every $h\in\mathcal{P}^{(2)}$ corresponds to some $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ (written as $h\leftrightarrow\rho$).

Proposition 10: An automorphism $h\in\mathcal{P}^{(2)}$ sends the locations $\mathbf{j}\in\mathbb{Z}_n^m$ to the 0th symbol position (i.e., $[i,j_i]\xrightarrow{h}[i,0]$ for all $i\in\{1,2,\ldots,m\}$), if the corresponding element $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$, satisfies

$$\rho(j_i-u_i)=-u_i \tag{5}$$

for all $i\in\{1,2,\ldots,m\}$.

Proof: If there exists an element $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ that satisfies (5), then from Proposition 5, we have

$$[i,j_i]\xrightarrow{h}[i,\rho(j_i-u_i)+u_i]=[i,0]$$

for all $i\in\{1,2,\ldots,m\}$. ■

Example 1: Let $\alpha^3=\alpha+1$, and we have $\theta_{\breve{\alpha}}(x)=1+x^3+x^5+x^6$ (see Proposition 1). In Fig. 2, the matrix $\mathbf{S}$ [see (4)] is shown. Consider the location vector $\mathbf{j}=[6,0,1]^T$ and $\mathbf{u}=[2,1,0]^T$ (see Proposition 3 and Table I). From Proposition 10, in order to send the locations in $\mathbf{j}$ to the 0th symbol, we require an element $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ that sends $\rho(4)=5$, $\rho(6)=6$, and $\rho(1)=0$, as depicted in Fig. 2 using bold arrows.

For all $j\in\mathbb{Z}_n$, denote $\mathbf{s}[j]$ to be the jth column of the generator matrix $\mathbf{S}$ (see (4)) belonging to the simplex code $\langle\theta_{\breve{\alpha}}(x)\rangle$.

Proposition 11: Pick any arbitrary $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$. Let an element $\mathbf{K}\in GL(m,2)$ correspond to $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$, i.e., $\mathbf{K}\leftrightarrow\rho$, see Proposition 9. Then, both $\mathbf{K}$ and ρ must satisfy

$$\mathbf{K}\mathbf{s}[\rho(j)]=\mathbf{s}[j] \tag{6}$$

for all $j\in\mathbb{Z}_n$. Furthermore, the action of ρ on the matrix $\mathbf{S}$, is completely determined by the left action of $\mathbf{K}$ on any set of m linearly independent columns of $\mathbf{S}$.

TABLE II
SETS $\{\mathbf{s}[-u_i]:1\le i\le m\}$ COMPUTED FOR SIMPLEX CODE $\langle\theta_{\breve{\alpha}}(x)\rangle$ AND TRACE DUAL-BASIS $\gamma^\perp$

$\mathbb{F}_{2^m}$	Cols. $\{\mathbf{s}[-u_i]:1\le i\le m\}$ of $\mathbf{S}$
$\mathbb{F}_{2^3}$	$\{5,3,7\}$
$\mathbb{F}_{2^4}$	$\{15,7,14,12\}$
$\mathbb{F}_{2^5}$	$\{18,4,17,10,5\}$
$\mathbb{F}_{2^6}$	$\{63,39,31,47,62,12\}$

Proof: The relationship (6) can be verified from Proposition 9 (also see Fig. 2 for a depicted example in $\mathbb{F}_{2^3}$). From (6), we have the relationship $\mathbf{K}(\mathbf{s}[\rho(j)]+\mathbf{s}[\rho(j')])=\mathbf{K}\mathbf{s}[\rho(j)]+\mathbf{K}\mathbf{s}[\rho(j')]=\mathbf{s}[j]+\mathbf{s}[j']$ for all $j\neq j'$ in $\mathbb{Z}_n$. Thus, the left action of $\mathbf{K}$ on the columns of $\mathbf{S}$, is determined by how $\mathbf{K}$ acts on sets of m linearly independent columns of $\mathbf{S}$. ■

It follows from Proposition 11 that an automorphism $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ satisfying (5), can be easily found when the conditions for applying the next proposition are satisfied.

Proposition 12: There exists an element $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ that satisfies Proposition 10, if both sets $\{\mathbf{s}[-u_i]:1\le i\le m\}$ and $\{\mathbf{s}[j_i-u_i]:1\le i\le m\}$ each contain m **linearly independent** vectors.

Proof: Let an element $\mathbf{K}\in GL(m,2)$ correspond to $\rho\in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ (i.e., $\mathbf{K}\leftrightarrow\rho$). Then we have $\mathbf{K}\mathbf{s}[\rho(j_i-u_i)]=\mathbf{s}[j_i-u_i]$ for all $i\in\{1,2,\ldots,m\}$, see Proposition 11. If ρ satisfies (5), then we require

$$\mathbf{K}\mathbf{s}[\rho(j_i-u_i)]=\mathbf{K}\mathbf{s}[-u_i]=\mathbf{s}[j_i-u_i]$$

for all $i\in\{1,2,\ldots,m\}$. We can always find some $\mathbf{K}\in GL(m,2)$ to satisfy the second equality whenever both sets $\{\mathbf{s}[-u_i]:1\le i\le m\}$ and $\{\mathbf{s}[j_i-u_i]:1\le i\le m\}$ each contain m linearly independent vectors. ■

Recall from Proposition 3 that $\mathbf{u}$ is determined by $\breve{\alpha}$ and $\gamma^\perp$. Table II shows the sets $\{\mathbf{s}[-u_i]:1\le i\le m\}$ computed for various fields $\mathbb{F}_{2^m}$ and fixed choices of $\alpha=\breve{\alpha}^{-1}$ and $\gamma^\perp$. We next show that the m columns in the set $\{\mathbf{s}[-u_i]:1\le i\le m\}$ are guaranteed to be linearly independent. Note from (4) that the rows of $\mathbf{S}$ are obtained using cyclic shifts of the idempotent $\theta_{\breve{\alpha}}(x)$. From [13, Lemma 10 pg. 225], the jth coefficient of $\theta_{\breve{\alpha}}(x)$ is given as $\mathrm{Tr}\left(\breve{\alpha}^{-j}\right)$, where $\mathrm{Tr}:\mathbb{F}_{2^m}\mapsto\mathbb{F}_2$ is the trace function defined as $\mathrm{Tr}\left(\beta\right)=\beta+\beta^2+\cdots+\beta^{2^{(m-1)}}$ for all $\beta\in\mathbb{F}_{2^m}$. Thus, the column $\mathbf{s}[j]$ can be written using the trace function $\mathrm{Tr}\left(\cdot\right)$ as

$$\mathbf{s}[j]=\begin{bmatrix}\mathrm{Tr}\left(\breve{\alpha}^{-j}\right)\\ \mathrm{Tr}\left(\breve{\alpha}^{-(j-1)}\right)\\ \vdots\\ \mathrm{Tr}\left(\breve{\alpha}^{-(j-m+1)}\right)\end{bmatrix}=\begin{bmatrix}\mathrm{Tr}\left(\breve{\alpha}^{-j}\right)\\ \mathrm{Tr}\left(\breve{\alpha}^{-j+1}\right)\\ \vdots\\ \mathrm{Tr}\left(\breve{\alpha}^{-j+m-1}\right)\end{bmatrix}.$$

The following proof utilizes the property $\mathrm{Tr}\left(\beta+\beta'\right)=\mathrm{Tr}\left(\beta\right)+\mathrm{Tr}\left(\beta'\right)$ for any $\beta,\beta'\in\mathbb{F}_{2^m}$, see [13, pg 115].

Proposition 13: Let $\mathbf{u}$ be determined as in the statement of Proposition 3, using both $\breve{\alpha}$ and $\gamma^\perp$. Then the m columns in $\{\mathbf{s}[-u_i]:1\le i\le m\}$ must be linearly independent.

Proof: Assume the contrary, namely the m columns are not linearly independent. Then there exists some nonzero binary vector $\mathbf{b} \in \mathbb{F}_2^m$, such that

$$\sum_{j=1}^{m} b_j \mathbf{s}[-u_j] = \begin{bmatrix} \sum_{j=1}^{m} b_j \mathrm{Tr}\left(\breve{\alpha}^{u_j}\right) \\ \sum_{j=1}^{m} b_j \mathrm{Tr}\left(\breve{\alpha}^{u_j+1}\right) \\ \vdots \\ \sum_{j=1}^{m} b_j \mathrm{Tr}\left(\breve{\alpha}^{u_j+m-1}\right) \end{bmatrix} = \begin{bmatrix} \mathrm{Tr}\left(\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right) \\ \mathrm{Tr}\left(\breve{\alpha}\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right) \\ \vdots \\ \mathrm{Tr}\left(\breve{\alpha}^{m-1}\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right) \end{bmatrix} = \mathbf{0} \quad (7)$$

where the 2nd equality follows from the property of the trace function. If follows from the last equality in (7) that we can write

$$\sum_{i=1}^{m} a_i \mathrm{Tr}\left(\breve{\alpha}^{i-1}\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right) = \mathrm{Tr}\left(\left(\sum_{i=1}^{m} a_i \breve{\alpha}^{i-1}\right)\left(\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right)\right) = \mathrm{Tr}\left(\beta \sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right) = 0 \quad (8)$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_m]^T$ is a length-m binary vector and $\beta = \sum_{i=1}^{m} a_i \breve{\alpha}^{i-1} \in \mathbb{F}_{2^m}$. Note that since $\breve{\alpha}$ is primitive, then $1, \breve{\alpha}, \ldots, \breve{\alpha}^{m-1}$ constitute a basis of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$, thus any element in $\beta \in \mathbb{F}_{2^m}$ can be written as some linear combination $\beta = \sum_{i=1}^{m} a_i \breve{\alpha}^{i-1}$ where $\mathbf{a} \in \mathbb{F}_2^m$. Now, if the term $\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}$ in (8) were to be nonzero, this implies that $\mathrm{Tr}\left(\beta \sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}\right) = 0$ for all $\beta \in \mathbb{F}_{2^m}$ (i.e., it implies that the kernel of the trace function $\mathrm{Tr}(\cdot)$ equals the field $\mathbb{F}_{2^m}$). This cannot be true, see [13 pg 116]. Thus, $\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j}$ must equal zero. By our initial assumption that $\mathbf{b} \neq \mathbf{0}$, this implies that the elements $\breve{\alpha}^{u_1}, \breve{\alpha}^{u_2}, \ldots, \breve{\alpha}^{u_m}$ are linearly dependent over $\mathbb{F}_2$ (i.e., $\sum_{j=1}^{m} b_j \breve{\alpha}^{u_j} = 0$ for some nonzero $\mathbf{b} \in \mathbb{F}_2^m$).

On the other hand the ideal $\langle \theta_{\breve{\alpha}}(x) \rangle$ is isomorphic to the field $\mathbb{F}_{2^m}$, see [13 Lemma 10 pg 225]. Consider the isomorphism obtained by sending $\theta_{\breve{\alpha}}(x)x \to \breve{\alpha}$. If the elements $\breve{\alpha}^{u_1}$, $\breve{\alpha}^{u_2}, \ldots, \breve{\alpha}^{u_m}$ are linearly dependent over $\mathbb{F}_2$, then this implies that there exists polynomials in the set $\{\theta_{\breve{\alpha}}(x)x^{u_i} : 1 \leq i \leq m\}$ that sum to zero, i.e., there exists some $\mathbf{b} \neq \mathbf{0}$ such that

$$\sum_{i=1}^{m} b_i \theta_{\breve{\alpha}}(x) x^{u_i} = 0. \quad (9)$$

But (9) implies that

$$\sum_{i=1}^{m} \theta_{\breve{\alpha}}(x) x^{u_i} \gamma_i^{\perp} = \sum_{i=1}^{m} \theta_{\breve{\alpha}}(x) x^{u_i} \gamma_i^{\perp} + \left(\sum_{i=1}^{m} b_i \theta_{\breve{\alpha}}(x) x^{u_i}\right) \gamma_{i_0}^{\perp} = \sum_{\substack{i=1 \\ i \neq i_0}}^{m} \theta_{\breve{\alpha}}(x) x^{u_i} \left(\gamma_i^{\perp} + b_i \gamma_{i_0}^{\perp}\right) \quad (10)$$

for some $i_0 \in \{1, 2, \ldots, m\}$ where $b_{i_0} = 1$. We draw the following line-of-thought from (10). We know that there exists a basis $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_m]^T$ (of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$) that satisfies

$$\gamma_i = \begin{cases} \gamma_i^{\perp} + b_i \cdot \gamma_{i_0}^{\perp}, & \text{for } i \neq i_0 \\ \gamma_{i_0}^{\perp}, & \text{for } i = i_0. \end{cases}$$

However, (10) implies that the binary image of $\theta_{\breve{\alpha}}(x) \prod_{\beta \in \mathcal{C}(\breve{\alpha}) \backslash \breve{\alpha}} x - \beta$ under γ gives

$$\theta_{\breve{\alpha}}(x) \left(\cdots + x^{u_{i_0-1}} \gamma_{i_0-1} + 0 + x^{u_{i_0+1}} \gamma_{i_0+1} + \cdots\right). \quad (11)$$

This contradicts Proposition 3, as it is clearly stated that there can be no 0 terms in (11) regardless of the basis γ that we image under. Thus, our earlier assumption that the set $\{\mathbf{s}[-u_i] : 1 \leq i \leq m\}$ contains linearly dependent vectors is notwithstanding, and we are done. ■

Corollary 2: There exists an element $\rho \in \mathit{Aut}\left(\langle \theta_{\breve{\alpha}}(x) \rangle\right)$ that satisfies Proposition 10, if and only if the set $\{\mathbf{s}[j_i - u_i] : 1 \leq i \leq m\}$ contains m **linearly independent** vectors.

Proof: Follows because the (invertible) $\mathbf{K} \in GL(m, 2)$ corresponding to ρ, must satisfy $\mathbf{K}\mathbf{s}[-u_i] = \mathbf{s}[j_i - u_i]$ for all $i \in \{1, 2, \ldots, m\}$ (see Proposition 12), and the set $\{\mathbf{s}[-u_i] : 1 \leq i \leq m\}$ is linearly independent by Proposition 13. ■

The algorithm to find a simplex automorphism $\rho \in \mathit{Aut}\left(\langle \theta_{\breve{\alpha}}(x) \rangle\right)$ that satisfies Proposition 10 is given in Appendix A, and requires only $\mathbb{F}_2^m$ additions (exactly n of them). By Corollary 2, the algorithm will succeed if and only if the set $\{\mathbf{s}[j_i - u_i] : 1 \leq i \leq m\}$ contains linearly independent vectors (recall also Propositions 12 and 13). There is no need to check that $\{\mathbf{s}[j_i - u_i] : 1 \leq i \leq m\}$ contains linearly independent vectors prior to running the algorithm; in the event they are dependent, the algorithm simply quits prematurely.

Example 2: Continuing from Example 1, we consider the case where $\mathbf{j} = [6, 0, 1]^T$ and $\mathbf{u} = [2, 1, 0]^T$. We can verify that the vectors in the set $\{\mathbf{s}[j_i - u_i] : 1 \leq i \leq 3\}$ (see Fig. 2) are linearly independent. The simplex automorphism $\rho \in \mathit{Aut}\left(\langle \theta_{\breve{\alpha}}(x) \rangle\right)$ that satisfies Proposition 10, is determined using the algorithm given in Appendix A to be $\rho = (0, 3, 1)(2, 4, 5)$. The automorphism ρ is depicted in Fig. 2 using the arrows.

Proposition 14: There exists exactly $\prod_{r=0}^{m-1}(2^m - 2^r)$ location vectors $\mathbf{j}$, where each binary symbol position (pointed to by j_i) can be sent to the 0th symbol position, using permutations in $\mathcal{P}^{(2)}$

Proof: The number of such location vectors $\mathbf{j}$, is equal to the number of unique sequences $\mathbf{s}[j_1 - u_1], \mathbf{s}[j_2 - u_2], \ldots, \mathbf{s}[j_m - u_m]$ that consist of m linearly independent vectors (see Corollary 2 and Proposition 10). This is equal to the number of ways to choose $m \times m$ invertible matrices, which is given by $|GL(m, 2)| = (2^m - 1)(2^m - 2) \cdots (2^m - 2^{m-1})$. ■

From Definition 3, we note that there exists a total of $n^m = (2^m - 1)^m$ possible location vectors $\mathbf{j}$. From Proposition 14 the formula $\prod_{r=0}^{m-1}(2^m - 2^r) = (n+1)^m \cdot \prod_{r=1}^{m}(1 - 2^{-r})$. Thus, the fraction of location choices $\mathbf{j} \in \mathbb{Z}_n^m$ that can be sent to the 0th symbol, is approximately $\prod_{r=1}^{m}(1 - 2^{-r})$. For large m, this fraction converges roughly to 0.289.

B. Permutation Decoding Algorithm A

Thus far, we considered binary images of double-parity RS $[n, n-2, 3]$ codes with zeros $\{1, \alpha\}$, where α is primitive. We showed that these binary images have at least $m! \cdot \prod_{r=0}^{m-1}(2^m - 2^r)$ automorphisms. In this subsection, we describe a permutation decoder (inspired by [4]) for the AWGN channel. The decoder described here will only utilize $\prod_{r=0}^{m-1}(2^m - 2^r)$ code automorphisms; these automorphisms belong to the permutation group $\mathcal{P}^{(2)}$ (see Proposition 5 for details).

Before going into the details of the decoding algorithm, we quickly recap the structure of the permutations in the group $h \in \mathcal{P}^{(2)}$. Recall the matrix index set $\mathcal{I} \triangleq \{[i,j] : 1 \le i \le m, j \in \mathbb{Z}_n\}$, where $\mathbb{Z}_n$ is the ring of integers modulo n. The set $\mathcal{I}$ indexes each (binary) symbol of the binary image codeword $\mathcal{B}_M(\mathbf{c})$ (see Definition 1). The permutation group $\mathcal{P}^{(2)}$ acts on the index set $\mathcal{I}$ (and, thus, on $\mathcal{B}_M(\mathbf{c})$). Each permutation $h \in \mathcal{P}^{(2)}$ corresponds to a permutation ρ, where ρ acts (not on $\mathcal{I}$ but) on $\mathbb{Z}_n$. This one-to-one correspondence is denoted as $h \leftrightarrow \rho$. The permutation ρ belongs to the group $Aut(\langle \theta_{\tilde{\alpha}}(x) \rangle)$ (see Proposition 5 for details), and we termed each element of $Aut(\langle \theta_{\tilde{\alpha}}(x) \rangle)$ a *simplex automorphism.*

Definition 4: Define a $m \times n$ **index grid** $\mathcal{G}$ that contains entries $\mathcal{G}_{[i,j]} = j + u_i \in \mathbb{Z}_n$ for all $[i,j] \in \mathcal{I}$.

Proposition 15: Let $h \in \mathcal{P}^{(2)}$ and $\rho \in Aut(\langle \theta_{\tilde{\alpha}}(x) \rangle)$ correspond $h \leftrightarrow \rho$. Let $\mathcal{G}$ be the index grid. Then h permutes the indices in $\mathcal{I}$ as

$$\left[i, \mathcal{G}_{[i,j]}\right] \xrightarrow{h} \left[i, \mathcal{G}_{[i,\rho(j)]}\right] \tag{12}$$

for all $i \in \{1, 2, \ldots, m\}$ and $j \in \mathbb{Z}_n$.

Proof: Shown using Definition 4 and Proposition 5. ■

Equation (12) gives a useful representation of the action of $h \in \mathcal{P}^{(2)}$ on the matrix index set $\mathcal{I}$. A location vector $\mathbf{j} \in \mathbb{Z}_n^m$ points to the binary symbols that we would like to permute into a RS code symbol; each element j_i in $\mathbf{j}$ points to one symbol in each row of the binary image codeword $\mathcal{B}_M(\mathbf{c})$ (see Definition 3 for details). A permutation $h \in \mathcal{P}^{(2)}$ that sends m chosen binary symbol locations $\mathbf{j}$ to a single nonbinary symbol position (namely the 0th symbol position), may be obtained by first executing Algorithm 1 (given in Appendix A) to obtain a specific $\rho \in Aut(\langle \theta_{\tilde{\alpha}}(x) \rangle)$ that satisfies Proposition 10, and then using (12) to construct the permutation $h \in \mathcal{P}^{(2)}$. We provide further clarification using the following example.

Example 3: We consider an example in $\mathbb{F}_{2^3}$, i.e., the grid $\mathcal{G}$ will now be a 3×7 matrix, and we choose $\mathbf{u} = [2, 1, 0]^T$. As shown in Fig. 3, we replicated $\mathcal{G}$, and placed the two copies one on top of each other. Let us consider the automorphism $h \in \mathcal{P}^{(2)}$ which corresponds to the permutation $\rho = (0, 3, 1)(2, 4, 5)$. As shown in Fig. 3, connect the jth column of the top replica of $\mathcal{G}$, to the $\rho(j)$th column of the bottom replica. Using (12) we obtain the images of the indices in $\mathcal{I}$ as follows. For any $i \in \{1, 2, \ldots, m\}$ and $j \in \mathbb{Z}_n$, look into the entry $\mathcal{G}_{i,j}$ in the top replica of $\mathcal{G}$ (eg. $\mathcal{G}_{[2,3]} = 4$). Follow the arrow going to the $\rho(j)$th column of the bottom replica. Look into $\mathcal{G}_{i,\rho(j)}$ and the image of $\left[i, \mathcal{G}_{[i,j]}\right]$ is obtained as $\left[i, \mathcal{G}_{[i,\rho(j)]}\right]$ (eg. the image of $\mathcal{G}_{[2,3]} = 4$ is $\mathcal{G}_{[2,\rho(3)]} = \mathcal{G}_{[2,1]} = 2$)).

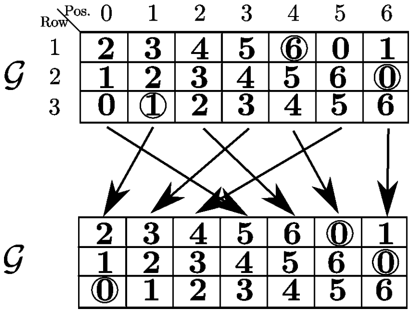


Fig. 3. Automorphism diagram of $h \in \mathcal{P}^{(2)}$ that sends $\mathbf{j} = [6, 0, 1]^T$ to the 0th symbol position.

In particular, note from Fig. 3 that the indices $[1, 6]$, $[2, 0]$ and $[3, 1]$ are all sent to the 0th symbol position (as marked by the circles). This confirms that the simplex automorphism $\rho = (0, 3, 1)(2, 4, 5)$ (which is exactly the same as the one computed in Example 2) indeed corresponds to a permutation $h \in \mathcal{P}^{(2)}$, that sends the locations $\mathbf{j} = [6, 0, 1]^T$ to the 0th symbol position.

Next, we start to describe our decoding algorithm. We begin by laying down the relevant framework. Let E_b and N_0 denote the energy per information symbol, and (single-sided) AWGN noise variance, respectively. A double-parity RS $[n, n-2, 3]$ codeword $\mathbf{c}$ (with zeros $\{1, \alpha\}$) is transmitted by first taking the binary image $\mathcal{B}_M(\mathbf{c})$ (see Definition 1), and mapping each binary symbol $c_{[i,j]}$ to an element in the bipolar signaling set $\{-1, 1\}$. Each (bipolar version of the) binary symbol $c_{[i,j]}$ is then passed through the AWGN channel. Let $\mathbf{y}$ denote a $m \times n$ matrix of channel observations (corresponding to the sent codeword $\mathbf{c}$), and let the element $y_{[i,j]}$ be the observation corresponding to the binary symbol $c_{[i,j]}$.

We take the *reliability* of the channel observation $y_{[i,j]}$ to be $|y_{[i,j]}|$ (see [11], [16]). Let $\eta \in \{1, 2, \ldots, n-1\}$ denote a chosen parameter of the permutation decoder.

Definition 5: Let $|y_1|, |y_2|, \ldots, |y_r|$ be a sequence of $r \ge 1$ channel reliabilities. Let $|y'_1|, |y'_2|, \ldots, |y'_r|$ be the sequence obtained by ordering $|y_1|, |y_2|, \ldots, |y_r|$ in increasing order, i.e.,

$$|y'_1| \le |y'_2| \le \cdots \le |y'_r|. \tag{13}$$

For an integer $\eta \ge 1$, we define the η-minimum **function** $\min^{(\eta)}$, that returns (a vector of) the first η values of the ordering (13), i.e.,

$$\begin{aligned}\min_{1 \le \ell \le r}{}^{(\eta)} |y_\ell| &\triangleq \left[|y'_1|, |y'_2|, \ldots, |y'_\eta|\right]^T \\ &= \left[|y_{\ell_1}|, |y_{\ell_2}|, \ldots, |y_{\ell_\eta}|\right]^T \end{aligned} \tag{14}$$

where the vector of indices $\{\ell_1, \ell_2, \ldots, \ell_\eta\}$ in (14) are returned as

$$\arg \min_{1 \le \ell \le r}{}^{(\eta)} |y_\ell| \triangleq [\ell_1, \ell_2, \ldots, \ell_\eta]^T.$$

Note from (13) and (14) that the indexing scheme on the reliability sequence $|y_1|, |y_2|, \ldots, |y_r|$, will always be clearly indicated in the subscripts of the η-minimum function $\min^{(\eta)}$.

```
Algorithm A: Permutation Decoder
 Input: Observations $\mathbf{y}$. Parameter $\eta$. Matrix $\mathbf{S}$ and vector
        $\mathbf{u}$. Basis $\gamma$;
 Initialize: Construct location vector set $\mathcal{J}(\mathbf{y},\eta)$, index
        grid $\mathcal{G}$, and codeword list $\mathcal{L} := \emptyset$;
 Output: $\hat{\mathbf{c}} := \arg\min_{\mathbf{c}\in\mathcal{L}} f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y},\mathbf{c})$;
 1  Perform hard decision decoding (HDD) on $z(\mathbf{y})$ (see
    Definition 7);
 2  if HDD decoded to some codeword $\mathbf{c}$ then store $\mathbf{c}$ in $\mathcal{L}$;
 3  forall $\mathbf{j} \in \mathcal{J}(\mathbf{y},\eta)$ do
 4      Run Alg. 1 with inputs $\mathbf{S}$, $\mathbf{j}-\mathbf{u}$ and $-\mathbf{u}$;
 5      if Alg. 1 returned a permutation $\rho \in Aut(\langle\theta_{\tilde{\alpha}}(x)\rangle)$
        then
 6          Construct $\mathbf{y}^{(h)}$ by setting
                $y^{(h)}_{[i,\mathcal{G}_{[i,j]}]} := y^{(h)}_{[i,\mathcal{G}_{[i,\rho(j)]}]}$,
            see (12);
 7          Compute
                $[i_0,\tau_0] = \arg\min_{[i,j]\in\mathcal{I}:\, j>0} |y^{(h)}_{i,j}|$;
 8          Erase the 0-th and $\tau_0$-th symbol and decode
            $z(\mathbf{y}^{(h)})$ to obtain codeword $\mathbf{c}^{(h)}$;
 9          Permute $\mathbf{c}^{(h)}$ with $h^{-1}$ and store in $\mathcal{L}$;
10      end
11  end
```

Definition 6: Define the **set** $\mathcal{J}(\mathbf{y},\eta) \subseteq \mathbb{Z}_n^m$ of location vectors $\mathbf{j}$ (see Definition 3), as the cartesian product $\mathcal{J}(\mathbf{y},\eta) \triangleq \mathcal{L}_1(\eta)\times\mathcal{L}_2(\eta)\times\cdots\times\mathcal{L}_m(\eta)$, where the set $\mathcal{L}_i(\eta)$ is defined as

$$\mathcal{L}_i(\eta) \triangleq \arg\min_{j\in\mathbb{Z}_n}{}^{(\eta)} |y_{[i,j]}|.$$

Note that the size of the location vector set $\mathcal{J}(\mathbf{y},\eta)$ increases as the algorithm parameter η increases. The intuition behind our permutation decoder is as follows. Note that for every location vector $\mathbf{j} \in \mathcal{J}(\mathbf{y},\eta)$, the column index j_i points to the binary symbol position $[i,j_i]$, whose reliability $|y_{[i,j_i]}|$ is ranked as one of the η-lowest reliabilities amongst the reliabilities in the ith row (i.e., $|y_{[i,1]}|, |y_{[i,2]}|, \ldots, |y_{[i,n-1]}|$). Our aim is to permute (the binary symbols pointed to by) each $\mathbf{j} \in \mathcal{J}(\mathbf{y},\eta)$ into a single nonbinary symbol position. The permuted symbols are then decoded using a RS erasure decoder, which has the capability to correct all binary symbol errors that lie within a nonbinary symbol.

Definition 7: Let γ denote the basis used to image the transmitted codewords (see Definition 1). We define the **vector of symbol decisions** $z(\mathbf{y}) = [z_0, z_1, \ldots, z_{n-1}]^T$ belonging to the observation matrix $\mathbf{y}$, where

$$z_j = b_{[1,j]}\cdot\gamma_1 + b_{[2,j]}\cdot\gamma_2 + \cdots + b_{[m,j]}\cdot\gamma_m$$

for all $j \in \mathbb{Z}_n$, and $b_{[1,j]}, b_{[2,j]} \ldots, b_{[m,j]}$ correspond to binary decisions of the observations $y_{[1,j]}, y_{[2,j]} \cdots, y_{[m,j]}$.

We denote $\mathbf{y}^{(h)}$ to be the $m \times n$ matrix of observations obtained under the action of $h \in \mathcal{P}^{(2)}$; if $[i,j] \xrightarrow{h} [i',j']$, then we have $y^{(h)}_{[i',j']} = y_{[i,j]}$. Denote the conditional probability that $\mathbf{y}$ is observed, given that a codeword $\mathbf{c}$ is sent, as $f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y},\mathbf{c})$ (also known as the likelihood of $\mathbf{c}$). Recall that a RS $[n, n-2, 3]$ code can decode any 2 symbol erasures. The permutation decoding algorithm is given in Algorithm A. Algorithm A constructs a list $\mathcal{L}$ of codewords, where $\mathcal{L}$ has size at most $|\mathcal{J}(\mathbf{y},\eta)| + 1 = \eta^m + 1$, and selects the codeword with the highest likelihood in the list $\mathcal{L}$. In Line 4, Algorithm 1 (given in the appendix) is executed to obtain a simplex automorphism $\rho \in Aut(\langle\theta_{\tilde{\alpha}}(x)\rangle)$, whereby ρ corresponds to the permutation $h \in \mathcal{P}^{(2)}$ that sends (the binary positions pointed to by) each $\mathbf{j} \in \mathcal{J}(\mathbf{y},\eta)$ to the 0th symbol position. Note that Algorithm 1 is not always guaranteed to succeed, as mentioned in Proposition 10. If Algorithm 1 succeeds, we proceed to erase both the 0th and τ_0th (see Line 7) symbols (see Line 8). The reason for choosing these two symbols to erase is as follows. Firstly, as argued before, the low reliability symbols pointed to by $\mathbf{j}$ now reside in the 0th symbol position after the permutation h. Secondly, the permuted observation $y^{(h)}_{[i_0,\tau_0]}$ possesses the lowest reliability $\left|y^{(h)}_{[i_0,\tau_0]}\right|$, amongst permuted observations residing in nonbinary symbol positions $j > 1$.

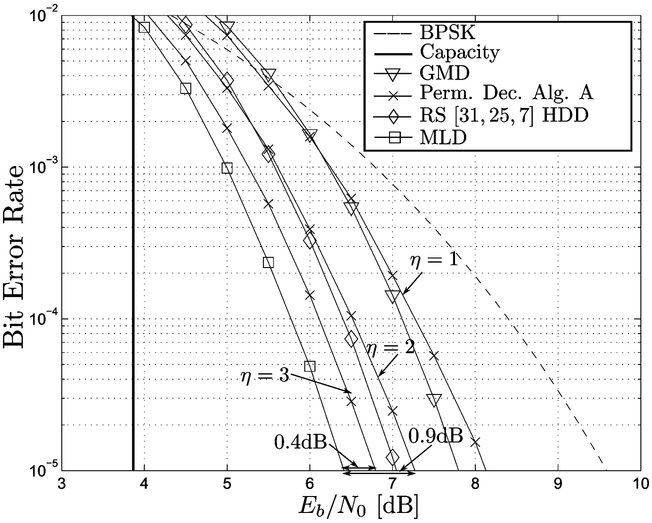


Fig. 4. Performance of permutation decoding Alg. A for the RS $[31, 29, 3]$ binary image.

Fig. 4 shows the bit error rate (BER) performance of our permutation decoder for the RS $[31, 29, 3]$ binary image, with the parameters $\eta = 1$,2 and 3. We see from Fig. 4 that when we choose $\eta = 1$, Algorithm A performs very close to the GMD. In fact when $\eta = 1$ note that $|\mathcal{J}(\mathbf{y},\eta)| = 1$, and we that see that Algorithm A looks very similar to the GMD algorithm. However, we do not perform as well as the GMD when we choose $\eta = 1$, because if Algorithm 1 fails (see Line 4), then the erasure decoding in Line 8 is not performed. We observe significant gains when we choose $\eta = 2$ and 3, with the latter case coming within 0.4 dB on the maximum likelihood decoder (MLD). The maximal list sizes $|\mathcal{L}| = \eta^m$ for $\eta = 1$,2 and 3 are 2,33 and 244 respectively. We also compare with the HDD of the NASA standard RS $[31, 25, 7]$ code. We see that our permutation decoder performs very close to the RS $[31, 25, 7]$ HDD when we choose $\eta = 2$, and we outperform it when we choose $\eta = 3$. As a final comment, we would like to point out that the erasure decoding performed in Line 8, can be performed with extremely low complexity, because only 2 symbols are erased.

Remark 2: We would like to emphasize that the results presented so far, improve over those in [4] in two ways. Both improvements are obtained here, by simply considering **all** simplex automorphisms in $Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$; in [4] only a cyclic subgroup was considered (see Remark 1). Firstly, for a chosen location vector $\mathbf{j}$, we have an efficient Alg. 1 (in the Appendix) to compute the required binary image automorphism h. Secondly, we have more "correctable" location vectors $\mathbf{j}$ (we view positions $\mathbf{j}$ as "correctable" if they can be sent to the 0th position to be erased, recall Alg. A Line 8). Because there is a one-to-one correspondence between a "correctable" $\mathbf{j}$ and a simplex automorphism (recall Proposition 10), it makes sense to consider all the simplex automorphisms in $Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$.

C. Permutation Decoding Algorithm B

In the previous subsection, we presented permutation decoding Algorithm A, which selects permutations from the group $\mathcal{P}^{(2)}$ (of order $\prod_{r=0}^{m-1}(2^m - 2^r)$) during decoding. In this subsection, we present permutation decoding Algorithm B, which is an improvement over Algorithm A. Algorithm B selects permutations from the product group $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ (see Proposition 7 for details). The size $|\mathcal{P}^{(1)}\mathcal{P}^{(2)}|$ is $m!$ times larger than $|\mathcal{P}^{(2)}|$; therefore, Algorithm B selects from a wider selection of permutations, resulting in performance gain over Algorithm A. We will show that in the product group $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$, there exist $m!$ automorphisms that send (the binary symbol positions pointed to by) $\mathbf{j}$ to 0th symbol position.

Any permutation in the product group $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ is a composition gh of 2 permutations $g \in \mathcal{P}^{(1)}$ and $h \in \mathcal{P}^{(2)}$ (see Proposition 4 for details on the group $\mathcal{P}^{(1)}$). Similar to the one-to-one correspondence between a permutation $h \in \mathcal{P}^{(2)}$ and a permutation $\rho \in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$, we previously established (in Proposition 4) that there is also a one-to-one correspondence between an element $g \in \mathcal{P}^{(1)}$ and a permutation σ of letters $\{1, 2, \ldots, m\}$ (i.e., σ is in the symmetric group Ω_m). The action of a permutation in $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ can be described using the index grid $\mathcal{G}$ (see Definition 4), similar to the discussions in the previous subsection.

Remark 3: Readers who are only interested in implementation of permutation decoding Algorithm B, may skip the proof of the following Proposition 16, and skip Proposition 17.

Proposition 16: Let $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ and $\sigma \in \Omega_m$ and $\rho \in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ correspond as $g \leftrightarrow \sigma$ and $h \leftrightarrow \rho$. Let $\mathcal{G}$ denote the index grid. Then gh permutes the indices in $\mathcal{I}$ as

$$\left[i, \mathcal{G}_{[i,j]}\right] \xrightarrow{gh} \left[\sigma(i), \mathcal{G}_{[\sigma(i),\rho(j)]}\right]. \qquad (15)$$

Proof: We see from (12) and Proposition 4 that

$$\begin{aligned}\left[i, \mathcal{G}_{[i,j]}\right] &\xrightarrow{h} \left[i, \mathcal{G}_{[i,\rho(j)]}\right] = \left[i, \rho(j) + u_i\right] \\ &\xrightarrow{g} \left[\sigma(i), \rho(j) + u_{\sigma(i)}\right] \\ &= \left[\sigma(i), \mathcal{G}_{[\sigma(i),\rho(j)]}\right].\end{aligned}$$

■

Proposition 17: Assume there exists some $h' \in \mathcal{P}^{(2)}$ that sends the positions $\mathbf{j}$ to the 0th symbol position. Then for any $g \in \mathcal{P}^{(1)}$, there exist some $h \in \mathcal{P}^{(2)}$, such that gh sends the positions $\mathbf{j}$ to the 0th symbol position.

```
Algorithm B: Permutation Decoder
Input: Observations y. Parameters $\eta$ and $\kappa$. Matrix S
       and vector u. Basis $\gamma$;
Initialize: Construct location vector set $\mathcal{J}(\mathbf{y},\eta)$, index
            grid $\mathcal{G}$, and codeword list $\mathcal{L} := \emptyset$;
Initialize: Collection of sets $\mathcal{T} := \emptyset$;
Output: $\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{L}} f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y},\mathbf{c})$;
Perfom Lines 1-2 of Algorithm A;
forall $\mathbf{j} \in \mathcal{J}(\mathbf{y},\eta)$ do
    forall $\sigma \in \Omega_m$ do
        Run Alg. 1 with inputs S, and j − u, and
        $-[u_{\sigma(1)}, u_{\sigma(2)}, \cdots, u_{\sigma(m)}]^T$;
        if $\sigma = i_d$ and Alg. 1 did not return a permutation
        $\rho \in Aut(\langle\theta_{\breve{\alpha}}(x)\rangle)$ then break;
        Construct $\mathbf{y}^{(gh)}$ by setting
            $y^{(gh)}_{[\sigma(i),\mathcal{G}_{\sigma(i),\rho(j)}]} := y_{[i,\mathcal{G}_{i,j}]}$,
        see (15);
        Compute
            $[[i_0,\tau_0],[i_1,\tau_1],\cdots,[i_\kappa,\tau_\kappa]]^T$
              $= \arg\min^{(\kappa+1)}_{[i,j]\in\mathcal{I}:\, j>0} |y^{(gh)}_{[i,j]}|$;
        if $\{i : 1 \le i \le \kappa, \tau_i = \tau_0\} \notin \mathcal{T}$ then
            Erase 0-th and $\tau_0$-th symbol and decode
            $z(\mathbf{y}^{(gh)})$ to obtain codeword $\mathbf{c}^{(gh)}$;
            Permute $\mathbf{c}^{(gh)}$ with $h^{-1}g^{-1}$ and store in $\mathcal{L}$;
            Store $\{i : 1 \le i \le \kappa, \tau_i = \tau_0\}$ in $\mathcal{T}$;
        end
        if $|\mathcal{T}| = 2^\kappa$ then break;
    end
end
```

Proof: From (15), we see that the image of the index $[i, j_i] = \left[\sigma(i), \mathcal{G}_{[i,j_i-u_i]}\right]$ under some $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ equals $\left[i, \mathcal{G}_{[\sigma(i),\rho(j_i-u_i)]}\right]$, where σ and ρ correspond to $g \in \mathcal{P}^{(1)}$ and $h \in \mathcal{P}^{(1)}$, respectively. To send positions $\mathbf{j}$ to the 0th symbol position, we require $\mathcal{G}_{[\sigma(i),\rho(j_i-u_i)]} = 0$ for all $i \in \{1, 2, \ldots, m\}$, or equivalently

$$\rho(j_i - u_i) = -u_{\sigma(i)}$$

for all $i \in \{1, 2, \ldots, m\}$ (see Definition 4). We require a $\rho \in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ and a corresponding $\mathbf{K} \in GL(m,2)$ that satisfies

$$\mathbf{K}\mathbf{s}[\rho(j_i - u_i)] = \mathbf{K}\mathbf{s}[-u_{\sigma(i)}] = \mathbf{s}[j_i - u_i] \qquad (16)$$

(recall that if $\rho \leftrightarrow \mathbf{K}$, then $\mathbf{K}\mathbf{s}[\rho(j)] = \mathbf{s}[j]$ for all $j \in \mathbb{Z}_n$, see Proposition 11). By our assumption there exists some $h' \in \mathcal{P}^{(2)}$ and corresponding $\rho' \in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ and $\mathbf{K}' \in GL(m,2)$ (i.e., $h' \leftrightarrow \rho' \leftrightarrow \mathbf{K}'$) that satisfies Proposition 12, i.e.,

$$\mathbf{K}'\mathbf{s}[\rho'(j_i - u_i)] = \mathbf{K}'\mathbf{s}[-u_i] = \mathbf{s}[j_i - u_i]$$

; therefore, the m columns in the set $\{\mathbf{s}[j_i - u_i] : 1 \le i \le m\}$ must be linearly independent (recall arguments in Corollary 2). Therefore there must exist a $\mathbf{K}$ that satisfies (16). ■

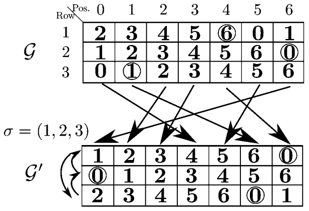


Fig. 5. Automorphism diagram of $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that sends $\mathbf{j} = [6,0,1]^T$ to the 0th symbol position.

Corollary 3: Assume there exists some $h' \in \mathcal{P}^{(2)}$ that sends the positions $\mathbf{j}$ to the 0th symbol position. Then there exists $m!$ unique automorphisms $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that send the positions $\mathbf{j}$ to the 0th symbol position.

Similar to Example 3, we can simply use (15) to determine the image of any index $\left[i, \mathcal{G}_{[i,j]}\right]$ under a permutation $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$. The only difference is that the row index i in $\left[i, \mathcal{G}_{[i,j]}\right]$ will be sent to $\sigma(i)$. If we simply construct a new index grid $\mathcal{G}'$, with entries $\mathcal{G}'_{[i,j]} = \mathcal{G}_{[\sigma(i),j]}$, then we see that

$$\left[i, \mathcal{G}_{[i,j]}\right] \xrightarrow{gh} \left[\sigma(i), \mathcal{G}_{[\sigma(i),\rho(j)]}\right] = \left[\sigma(i), \mathcal{G}'_{[i,\rho(j)]}\right].$$

The new index grid $\mathcal{G}'$ gives us a more systematic way to view the permutation gh, as shown in the next example.

Example 4: Continuing from Example 3, Fig. 5 shows a different automorphism from the group $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that sends $\mathbf{j} = [6,0,1]^T$ to the 0th symbol position. Here, the permutation σ corresponding to the chosen $g \in \mathcal{P}^{(1)}$ is given as $\sigma = (1,2,3)$. The top index grid remains as $\mathcal{G}$, but now the bottom grid is the row permuted version $\mathcal{G}'$ of $\mathcal{G}$. Specifically, we obtain $\mathcal{G}'$ by permuting row 1 to row 3, row 2 to row 1 and row 3 to row 2, exactly the permutation $(1,3,2) = \sigma^{-1}$. Note that the locations $\mathbf{j} = [6,0,1]^T$ (marked by circles) still correspond to the 0 locations (marked by circles) in each respective row.

The new permutation decoding algorithm is given in Algorithm B. In contrast with Algorithm A, Algorithm B has a new parameter κ in addition to η. We see that both parameters $\eta \geq 1$ and $\kappa \geq 0$ jointly determine the list size $|\mathcal{L}|$ (and the number of erasure decodings performed in Line 9. Though not immediately obvious, it can be verified that Algorithm B specializes to Algorithm A when we choose $\kappa = 0$. Note from Lines 7 and 9 that, the erased symbol positions (0 and τ_0) are exactly the same as those in Algorithm A (see Alg. A Lines 7–8). However, in contrast with Alg. A (Line 7), we obtain κ additional symbol locations $\tau_1, \tau_2, \ldots, \tau_\kappa$ that correspond to least reliable observations in the (permuted) observation matrix $\mathbf{y}^{(gh)}$ (see Alg. B Line 7). In Lines 3–14, we search through the $m!$ permutations in $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that send the locations $\mathbf{j}$ to the 0th symbol position. However, in Line 11, we record the set $\{i : 1 \leq i \leq \kappa, \tau_i = \tau_0\}$ in $\mathcal{T}$, which indicates which of the κ least reliable positions $[i_1, \tau_1], [i_2, \tau_2], \ldots, [i_\kappa, \tau_\kappa]$ get sent to the τ_0th symbol. The reason for this is to prevent repetitive (erasure) decoding of the same low reliability binary symbols. The maximal number of erasure decodings within the loop specified by Lines 3–14, is 2^κ (see Line 13), and the maximal size of $|\mathcal{L}|$ is $2^\kappa \cdot \eta^m + 1$.

Figs. 6 and 7 shows the performance of Algorithm B for the RS $[31,29,3]$ and $[63,61,3]$ codes respectively. In Fig. 6, we also show the performance for $\eta = 2, \kappa = 0$ and $\eta = 3, \kappa = 0$, which correspond to Algorithm A when we choose $\eta = 2$ and $\eta = 3$, respectively. Comparing the cases $\eta = 2$, $\kappa = 2$ and (Alg. A) $\eta = 2$, $\kappa = 0$, we see a gain of approximately 0.15 dB at $\mathrm{BER} = 10^{-5}$. Also comparing the cases $\eta = 2$, $\kappa = 2$ and (Alg. A) $\eta = 2$, $\kappa = 0$, we see a gain of approximately 0.1 dB at $\mathrm{BER} = 10^{-5}$. For the RS $[31,29,3]$ code, the maximum number of erasure decodings for $\eta = 2$, $\kappa = 2$ and $\eta = 3$, $\kappa = 4$ are $2^2 \cdot 2^5 = 128$ and $2^4 \cdot 3^5 = 3888$, respectively. The maximum number of erasure decodings for Alg. A $\eta = 2$ and $\eta = 3$ are $2^5 = 32$ and $3^5 = 243$. Thus, we see that the complexity of Alg. B $\eta = 2$, $\kappa = 2$, is in between that of Alg. A $\eta = 2$ and $\eta = 3$. From Fig. 6, we conclude similarly about the BER performance of Alg. A $\eta = 2$ and $\eta = 3$ and Alg. B $\eta = 2$, $\kappa = 2$. Interestingly, at low SNRs (≤ 4.5 dB) Alg. B with $\eta = 3$, $\kappa = 4$ performs extremely close to MLD. We also show the performance of the RS [255,239,17] HDD. Note that $239/255 \approx 29/31$. We see from Fig. 6 that for SNRs ≤ 6.5 dB, both permutation decoders with $\eta = 2$, $\kappa = 2$ and $\eta = 3$, $\kappa = 4$ perform better than the HDD of the longer code (i.e the RS [255,239,17]).

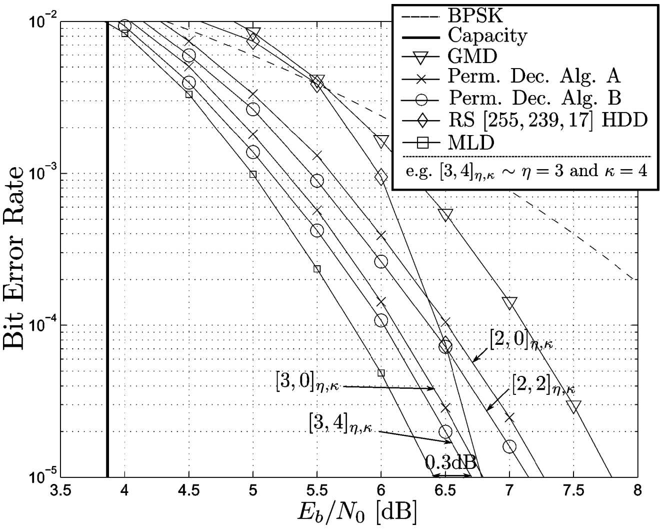


Fig. 6. Performance of permutation decoding Alg. B (and also Alg. A) for the RS $[31,29,3]$ binary image.

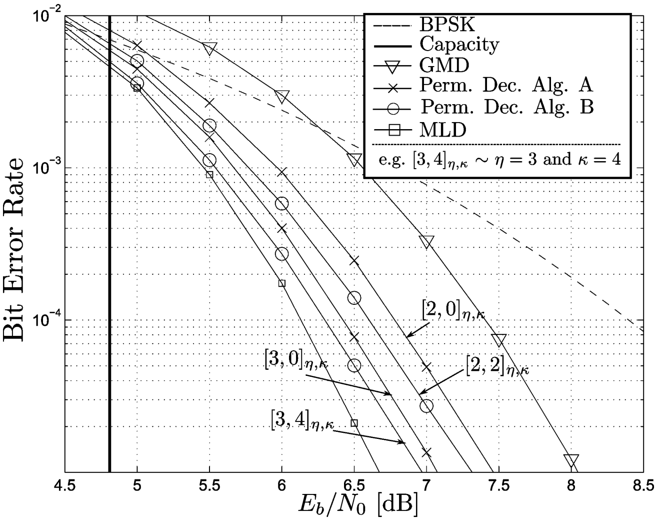


Fig. 7. Performance of permutation decoding Alg. B (and also Alg. A) for the RS $[63,61,3]$ binary image.

Similar observations are made in Fig. 7 on the permutation decoder performance for the RS $[63, 61, 3]$ code. Similar to the RS $[31, 29, 3]$ case, the gain of $\eta = 2$, $\kappa = 2$ over (Alg. A) $\eta = 2$, $\kappa = 0$ is also approximately 0.15 dB at $\mathrm{BER} = 10^{-5}$, and the gain of $\eta = 2$, $\kappa = 2$ over (Alg. A) $\eta = 2$, $\kappa = 0$ is also approximately 0.1 dB at $\mathrm{BER} = 10^{-5}$.

D. Reducing the Complexity of the Permutation Decoders

In this subsection, we address two ways to reduce the complexity of both permutation decoding Algorithms A and B. This subsection consists of two parts. The first part addresses a more efficient implementation of Algorithm B. In Algorithm B, the loop in Lines 4–13 may run up to $m!$ times in the worst case. This implies that Algorithm 1 (see Line 4) may potentially run up to $m!$ times, computing a total of $m!$ simplex automorphisms ρ. We will show that this is unnecessary. The second part deals with reducing the average complexities[3] of the permutation decoders. Note that both Algorithms A and B are *list decoders*, i.e., they both construct a codeword list $\mathcal{L}$ from which the codeword with the highest likelihood (in the list) is chosen. In the literature, there exist well-known methods to reduce the average complexities of list decoders (see [11], [16]–[18]). The general idea behind these methods are as follows. During the construction of the list $\mathcal{L}$, we can determine if the most-likely codeword [4] resides in the semi-constructed list, when a candidate codeword passes the *sufficient condition* for optimality [11], [16]–[18]. In this case there is no need to further populate the list $\mathcal{L}$. In other scenarios, prior to the actual computation of a candidate codeword (to be placed into the list $\mathcal{L}$), it is sometimes possible to determine that the new candidate codeword must have a *lower likelihood* than the best codeword in the semi-constructed list $\mathcal{L}$. This is known as the *neccessary condition* for optimality [11], [16]. In this case we skip the computation of the candidate codeword.

We first address the efficient implementation for Algorithm B. Note that for every $\sigma \in \Omega_m$ in the loop spanning Lines 3–12, we need to run Algorithm 1 (to obtain a simplex automorphism ρ) in order to perform the check on Line 8. If the check fails, then the simplex automorphism ρ is discarded and Algorithm B is run again to obtain another simplex automorphism. In the following, we will show that for purposes of conducting the check on Line 8, repeated executions of Algorithm 1 (for every $\sigma \in \Omega_m$) is unnecessary. We will show there exists a more efficient method to perform the check on Line 8. The following exposition gives us further insight into the structure of the permutations described in Proposition 17.

For the next proposition, define the $m \times m$ square matrix $\mathbf{K}_\mathbf{u} \triangleq [\mathbf{s}[-u_1], \mathbf{s}[-u_2], \ldots, \mathbf{s}[-u_m]]$ (i.e., the vectors $\mathbf{s}[-u_i]$ make up the columns of $\mathbf{K}_\mathbf{u}$). By Proposition 13, the matrix $\mathbf{K}_\mathbf{u}$ is nonsingular and has an inverse. Define $\boldsymbol{\mathcal{S}} \triangleq \mathbf{K}_\mathbf{u}^{-1}\mathbf{S}$ and let $\boldsymbol{\mathcal{S}}[j]$ denote the jth column of $\boldsymbol{\mathcal{S}}$. Let $\mathbf{b}^{(\sigma)}$ be a permutation of the binary vector $\mathbf{b} \in \mathbb{F}_2^m$, under the action of an element $\sigma \in \Omega_m$, defined as $\mathbf{b}^{(\sigma)} = [b_{\sigma^{-1}(1)}, b_{\sigma^{-1}(2)}, \ldots, b_{\sigma^{-1}(m)}]^T$ (e.g., if $\sigma = (1, 2, 3)$ and $\mathbf{b} = [1, 1, 0]^T$, then $\mathbf{b}^{(\sigma)} = [0, 1, 1]^T$).

Readers who are only interested in the efficient implementation of permutation decoding B, may skip the following Proposition 18.

Proposition 18: Assume that the automorphism $h \in \mathcal{P}^{(2)}$ that sends some chosen positions $\mathbf{j}$ to the 0th symbol position exists. Let $\rho \leftrightarrow h$. Recall Proposition 16 that the image of an index $\left[i, \mathcal{G}_{[i,j]}\right]$ under any $gh' \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ is $\left[\sigma(i), \mathcal{G}_{[\sigma(i),\rho'(j)]}\right]$, where $\sigma \leftrightarrow g$ and $h' \leftrightarrow \rho'$. Let $\boldsymbol{\mathcal{S}} = \mathbf{K}_\mathbf{u}^{-1} \cdot \mathbf{S}$. If gh' also sends locations $\mathbf{j}$ to the 0th symbol position, then we must have

$$\boldsymbol{\mathcal{S}}[\rho'(j)] = \left(\boldsymbol{\mathcal{S}}[\rho(j)]\right)^{(\sigma)}.$$

for all $j \in \mathbb{Z}_n$.

Proof: Note that $\boldsymbol{\mathcal{S}}_{[i,j]}$ is the $[i, j]$th element of $\boldsymbol{\mathcal{S}}$. Note that we can express the column $\mathbf{s}[\rho(j)]$ as

$$\mathbf{s}[\rho(j)] = \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho(j)]} \cdot \mathbf{s}[-u_i]. \tag{17}$$

To verify (17), multiply both sides of (17) by $\mathbf{K}_\mathbf{u}^{-1}$, and note that $\mathbf{K}_\mathbf{u}^{-1}$ sends each $\mathbf{s}[-u_i]$ to the unit vector with a one in the ith position. From the correspondence $\rho \leftrightarrow \mathbf{K} \in GL(m, 2)$ (see Proposition 11), the column $\mathbf{s}[\rho(j)]$ is sent to $\mathbf{s}[j]$ under the left action of $\mathbf{K}$ (i.e., $\mathbf{K}\mathbf{s}[\rho(j)] = \mathbf{s}[j]$). Therefore continuing from (17), we can write the left action of $\mathbf{K}$ on $\mathbf{s}[\rho(j)]$ as

$$\begin{aligned} \mathbf{s}[\rho(j)] &= \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho(j)]} \cdot \mathbf{s}[-u_i] \\ &\xrightarrow{\mathbf{K}} \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho(j)]} \cdot \mathbf{s}[j_i - u_i] = \mathbf{s}[j] \end{aligned} \tag{18}$$

where we note from Proposition 12 that $\mathbf{s}[-u_i] \xrightarrow{\mathbf{K}} \mathbf{s}[j_i - u_i]$ for all $i \in \{1, 2, \ldots, m\}$.

Now consider some automorphism $gh' \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$, where $g \leftrightarrow \sigma$, where gh' also sends $\mathbf{j}$ to the 0th symbol position. Denote the corresponding $\mathbf{K}' \leftrightarrow \rho' \leftrightarrow h'$, and the left action of $\mathbf{K}'$ on $\mathbf{s}[\rho'(j)]$ can be written as

$$\begin{aligned} \mathbf{s}[\rho'(j)] &= \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho(j)]} \cdot \mathbf{s}[-u_{\sigma(i)}] \\ &\xrightarrow{\mathbf{K}'} \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho(j)]} \cdot \mathbf{s}[j_i - u_i] = \mathbf{s}[j] \end{aligned} \tag{19}$$

where the last equality follows from the last equality of (18), the pre-image of $\mathbf{K}'$ follows from (16) and the choice of gh', and the first equality follows because $\mathbf{K}'\mathbf{s}[\rho'(j)] = \mathbf{s}[j]$.

Finally, we express $\mathbf{s}[\rho'(j)]$ as

$$\begin{aligned} \mathbf{s}[\rho'(j)] &= \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho'(j)]} \cdot \mathbf{s}[-u_i] = \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[i,\rho(j)]} \cdot \mathbf{s}[-u_{\sigma(i)}] \\ &= \sum_{i=1}^{m} \boldsymbol{\mathcal{S}}_{[\sigma^{-1}(i),\rho(j)]} \cdot \mathbf{s}[-u_i] \end{aligned} \tag{20}$$

[3] In our case, the average complexity is indicated by the average size of the list $\mathcal{L}$.

[4] If the set A contains all possible codewords, then the most-likely codeword equals $\arg\min_{\mathbf{c}\in A} f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y}, \mathbf{c})$.

where the first equality follows similarly to (17), the second equality follows from (19) and the third equality follows from an index change. We draw the following conclusion from the equality between the 2nd and 4th terms of (20). Since the set $\{\mathbf{s}[-u_i] : 1 \leq i \leq m\}$ is a basis of $\mathbb{F}_2^m$ over $\mathbb{F}_2$ (see Proposition 13), then we must have $\boldsymbol{\mathcal{S}}[\rho'(j)] = (\boldsymbol{\mathcal{S}}[\rho(j)])^{(\sigma)}$. Thus, we are done. ■

Recall that Algorithm B selects from all $m!$ automorphisms in $\mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that send locations $\mathbf{j}$ to the 0th symbol. This selection is done by performing the check on Alg. B Line 8, and this said check requires knowledge on where the κ least reliable binary positions are sent. In other words, if $\left[i, \mathcal{G}_{[i,j]}\right]$ points to one of the κ least reliable positions, then we need to know its image $\left[\sigma(i), \mathcal{G}_{[\sigma(i),\rho'(j)]}\right]$ under the automorphism $gh' \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ (as inferred from Alg. B Lines 6–7). It appeared before that computing the image $\left[\sigma(i), \mathcal{G}_{[\sigma(i),\rho'(j)]}\right]$ requires computation of ρ' that corresponds to h'. However, Proposition 18 states that, if both $\boldsymbol{\mathcal{S}}$ and $\rho \leftrightarrow h$ (as defined in the statement of Proposition 18) are known, then we only need to perform the following procedure.

Procedure 1: Obtaining image of $[i, \mathcal{G}[i, j]]$.

Input: Matrix $\mathbf{S}$ and permutation ρ;

Output: Image $\left[\sigma(i), \mathcal{G}_{[\sigma(i),j'(\sigma)]}\right]$;

1) Permute the length-m vector $\mathbf{S}[\rho(j)]$ by $\sigma \in \Omega_m$ to obtain $(\mathbf{S}[\rho(j)])^{(\sigma)}$, and find $j'(\sigma) \in \mathbb{Z}_n$ that satisfies

$$\mathbf{S}[j'(\sigma)] = (\mathbf{S}[\rho(i)])^{(\sigma)};$$

2) Obtain the image of $\left[i, \mathcal{G}_{[i,j]}\right]$ under gh' as $\left[\sigma(i), \mathcal{G}_{[\sigma(i),j'(\sigma)]}\right]$.

Therefore, by Proposition 18, computation of $\rho' \leftrightarrow h'$ is completely unnecessary. We further clarify Proposition 18 by the following example in $\mathbb{F}_{2^3}$.

Example 5: Continuing from Examples 3 and 4, we consider the group of automorphisms $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that send the locations $\mathbf{j} = [6, 0, 1]^T$ to the 0th symbol location. We showed two different automorphisms corresponding to $\sigma = i_d$ and $\sigma = (1, 2, 3)$ in Examples 3 and 4, respectively. Fig. 8 shows the matrix $\mathbf{S}$ obtained from $\mathbf{S}$ (and as observed each vector $\mathbf{s}[-u_i]$ is sent to a unit vector). For the index $\left[i, \mathcal{G}_{[i,j]}\right] = [3, 2]$ (boxed in the top replica of $\mathcal{G}$), we determine its images under all $m!$ permutations using Procedure 1. First note that $j = 2$, and $\rho(j) = \rho(2) = 4$, and we have $\mathbf{S}[\rho(j)] = \mathbf{S}[4] = [1, 0, 1]^T$. We see that both $\sigma = i_d$ and $\sigma = (1, 3)$ stabilize $\mathbf{S}[4] = [1, 0, 1]^T$ (i.e., $\mathbf{S}[4]^{(i_d)} = \mathbf{S}[4]^{((1,3))} = \mathbf{S}[4]$), and they both give the same $j'(\sigma) = 4$ and 2 different images $\left[\sigma(3), \mathcal{G}_{[\sigma(3),j'(\sigma)]}\right] = [3, 4]$, and $\left[\sigma(3), \mathcal{G}_{[\sigma(3),j'(\sigma)]}\right] = [1, 6]$, respectively. Similarly, both $\sigma = (1, 2)$ and $\sigma = (1, 3, 2)$ send $\mathbf{S}[4] = [1, 0, 1]^T$ to $\mathbf{S}[2] = [0, 1, 1]^T$ (i.e., both give $j'(\sigma) = 2$), and give 2 images $[3, 2]$ and $[2, 3]$, respectively. Finally we conclude both $\sigma = (2, 3)$ and $\sigma = (1, 2, 3)$ give images $[2, 2]$ and $[1, 3]$, respectively. All images of $[3, 2]$ are boxed in the bottom replica of $\mathcal{G}$ in Fig. 8.

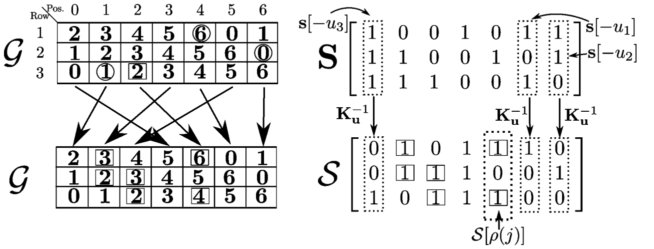


Fig. 8. Images of the index $[3, 2]$ (boxed in the upper replica of $\mathcal{G}$) under the action of the $m!$ automorphisms $gh' \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$ that send $\mathbf{j} = [6, 0, 1]^T$ to the 0th symbol position.

Remark 4: Recall $\mathbf{S}_{[i',j']}$ denotes the $[i, j]$th element of the matrix $\mathbf{S}$. It may have been noticed from Fig. 8, that all images of the index $\left[i, \mathcal{G}_{[i,j]}\right] = [3, 2]$ (boxed in the bottom replica of $\mathcal{G}$), correspond to all the elements $\boldsymbol{\mathcal{S}}_{[\sigma(i),j'(\sigma)]}$ that have the value 1 (boxed in the matrix $\boldsymbol{\mathcal{S}}$). This is by no means a coincidence. Indeed we can verify from Procedure 1 that for a given $i \in \{1, 2, \ldots, m\}$ and $j \in \mathbb{Z}_n$, we have

$$\boldsymbol{\mathcal{S}}_{[\sigma(i),j'(\sigma)]} = \boldsymbol{\mathcal{S}}_{[\sigma^{-1}(\sigma(i)),\rho(j)]} = \boldsymbol{\mathcal{S}}_{[i,\rho(j)]}.$$

Thus, for a fixed $j'(\sigma)$, the image $\left[\sigma(i), \mathcal{G}_{[\sigma(i),j'(\sigma)]}\right]$ of $\left[i, \mathcal{G}_{[i,j]}\right]$ can be quickly identified, by searching for the elements in the column $\boldsymbol{\mathcal{S}}[j'(\sigma)]$ that equal $\boldsymbol{\mathcal{S}}_{[i,\rho(j)]}$.

Algorithm B * is the efficient implementation of Algorithm B. Note that $\boldsymbol{\mathcal{S}}$ only depends on $\mathbf{u}$; thus, it can be precomputed and passed to Algorithm B * as an input. We see that in Algorithm B * Line 3, we first run Algorithm 1 to obtain a permutation $\rho \in Aut\,(\langle\theta_{\tilde{\alpha}}(x)\rangle)$ (if it exists). Then in Line 5, we find the $\kappa + 1$ least reliable positions[5] $[i_0, \tau_0], [i_1, \tau_1], \ldots, [i_\kappa, \tau_\kappa]$. In Line 7, we utilize Procedure 1 to quickly determine their images $[i'_0, \tau'_0], [i'_1, \tau'_1], \ldots, [i'_\kappa, \tau'_\kappa]$. Then if the check in Line 8 fails, we move on to the next $\sigma \in \Omega_m$. Note in Line 9 we elect to run Algorithm 1 again, rather than utilize Procedure 1. This is because Procedure 1 is only efficient in computing the images of a reasonably small number of indices (i.e., in practice κ is typically small).

In the second part of this subsection, we derive the necessary condition used to reduce the average complexity of our permutation decoders. A derivation of a generic sufficient condition, which can be used for our permutation decoding scheme, can be found in [11] (see Chapter 10). Also see [16]–[18] for other good references to sufficient/necessary conditions. Using computer simulations, we evaluate the average complexities[6] (given in terms of the average size of the list $|\mathcal{L}|$) of both Algorithms A and B, when both sufficient and necessary conditions are used.

Definition 8: Let $\mathbf{b}$ be the $m \times n$ matrix of binary decisions taken on $\mathcal{B}_M(\mathbf{c})$. The discrepancy $\mathcal{D}(\mathbf{c})$ of a codeword $\mathbf{c}$ is

[5] Note that in both Alg. B Line 7 and Alg. B * Line 5, the $\kappa + 1$ least reliable binary symbols are the same. The only difference is the way they are referenced. In Alg. B, they are referenced after permutation by gh, whereas they are referenced in transmitted order in Alg. B *.

[6] Note that in the second part of the subsection, there is no need to identify Algorithms B and B * separately, as they both have the same average list size $|\mathcal{L}|$.

Algorithm B*: Permutation Decoder

```
Input: Observations $\mathbf{y}$. Parameters $\eta$ and $\kappa$. Matrices $\mathbf{S}$
        and $\boldsymbol{\mathcal{S}}$ and vector $\mathbf{u}$. Basis $\gamma$;
Initialize: Construct location vector set $\mathcal{J}(\mathbf{y},\eta)$, index
        grid $\mathcal{G}$, and codeword list $\mathcal{L} := \emptyset$;
Initialize: Collection of sets $\mathcal{T} := \emptyset$;
Output: $\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{L}} f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y},\mathbf{c})$;
Perfom Lines 1-2 of Algorithm A;
forall $\mathbf{j} \in \mathcal{J}(\mathbf{y},\eta)$ do
    Run Alg. 1 with inputs $\mathbf{S}$, and $\mathbf{j}-\mathbf{u}$, and $-\mathbf{u}$;
    if Alg. 1 did not return a permutation
    $\rho \in Aut\left(\langle\theta_{\breve{\alpha}}(x)\rangle\right)$ then continue;
    Compute
        $\Delta \triangleq [[i_0,\tau_0],[i_1,\tau_1],\cdots,[i_\kappa,\tau_\kappa]]^T$
          $= \arg \min^{(\kappa+1)}_{[i,j]\in\mathcal{I}'(\mathbf{j})} |y_{[i,j]}|,$
    where the set $\mathcal{I}'(\mathbf{j}) = \mathcal{I}\setminus\{[i,j_i] : 1 \le i \le m\}$ ;
    forall $\sigma \in \Omega_m$ do
        Use $\boldsymbol{\mathcal{S}}$, and $\rho$, and $\sigma$ and Procedure 1 to obtain
        the images of $\Delta$ as
            $[[i'_0,\tau'_0],[i'_1,\tau'_1],\cdots,[i'_\kappa,\tau'_\kappa]]^T$;
        if $\{i : 1 \le i \le \kappa, \tau'_i = \tau'_0\} \notin \mathcal{T}$ then
            Run Alg. 1 with inputs $\mathbf{S}$, and $\mathbf{j}-\mathbf{u}$, and
            $-[u_{\sigma(1)},u_{\sigma(2)},\cdots,u_{\sigma(m)}]^T$;
            Construct $\mathbf{y}^{(gh)}$ by setting
                $y^{(gh)}_{[\sigma(i),\mathcal{G}_{\sigma(i),\rho(j)}]} := y_{[i,\mathcal{G}_{i,j}]},$
            see (12);
            Erase 0-th and $\tau'_0$-th symbol and decode
            $z\left(\mathbf{y}^{(gh)}\right)$ to obtain codeword $\mathbf{c}^{(gh)}$;
            Permute $\mathbf{c}^{(gh)}$ with $h^{-1}g^{-1}$ and store in $\mathcal{L}$;
            Store $\{i : 1 \le i \le \kappa, \tau'_i = \tau'_0\}$ in $\mathcal{T}$;
        end
        if $|\mathcal{T}| = 2^\kappa$ then break;
    end
end
```

defined to be

$$\mathcal{D}(\mathbf{c}) \triangleq \sum_{[i,j]\in\mathcal{I}:\ c_{[i,j]}\neq b_{[i,j]}} |y_{[i,j]}|.$$

Note it can be shown (see for example [16]) that the codeword $\mathbf{c}$ that maximizes the likelihood $f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y},\mathbf{c})$ is equivalent to the one that minimizes the discrepancy $\mathcal{D}(\mathbf{c})$. Recall that in both permutation decoding algorithms A and B, we employ erasure decoders to obtain codewords which fill up the list $\mathcal{L}$. Assume that $\mathcal{L}$ is partially filled (in the sense that the permutation decoder has yet to output the decoded codeword). Let $\hat{\mathbf{c}}$ be the codeword with the lowest discrepancy $\mathcal{D}(\hat{\mathbf{c}})$ (or the highest likelihood $f_{\mathbf{Y}|\mathbf{C}}(\mathbf{y},\hat{\mathbf{c}})$) in the partially filled list $\mathcal{L}$.

Definition 9: Let $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$. Let τ_0 denote the erased symbol location (other than 0, see Algorithm A Line 8 or Algorithm B Line 9). We define the **erased set** $\mathcal{E}(gh) \subset \mathcal{I}$ as

$$\mathcal{E}(gh) \triangleq \left\{[i,j]\in\mathcal{I} : [i,j] \overset{gh}{\to} [i',0]\right\} \cup \left\{[i,j]\in\mathcal{I} : [i,j] \overset{gh}{\to} [i',\tau_0]\right\}.$$

Thus, the indices in the erased set $\mathcal{E}(gh)$, correspond to the indices $[i,j]$ that are permuted into the erased symbol locations 0 and τ_0. Recall that in both[7] Algorithms A and B, we erase the 0th and τ_0th symbol of $z\left(\mathbf{y}^{(gh)}\right)$ (see Definition 7) to obtain a permuted codeword $\mathbf{c}^{(gh)}$. We then reverse the permutation gh to obtain a codeword $\mathbf{c}$, which is then stored in the codeword list $\mathcal{L}$. Effectively, this means that the codeword $\mathbf{c}$ is obtained from $z\left(\mathbf{y}^{(gh)}\right)$ by erasing all binary symbol positions pointed to by the erased set $\mathcal{E}(gh)$. In this respect, we will say that $\mathbf{c}$ is the codeword obtained by erasing the set $\mathcal{E}(gh)$.

Proposition 19: Let $\mathbf{b}$ be the matrix of binary decisions taken on the observations $\mathbf{y}$. Let $gh \in \mathcal{P}^{(1)}\mathcal{P}^{(2)}$. Let $\hat{\mathbf{c}}$ be the codeword with the lowest discrepancy $\mathcal{D}(\hat{\mathbf{c}})$ in the partially filled list $\mathcal{L}$. Assume that the integer

$$r(\hat{\mathbf{c}}) = 4 - \left|\{[i,j]\in\mathcal{I} : \hat{c}_{[i,j]} \neq b_{[i,j]}\}\right|$$

is greater than 0. Then the discrepancy $\mathcal{D}(\mathbf{c})$ of the codeword $\mathbf{c}$ obtained by erasing $\mathcal{E}(gh)$, it at least the sum of all reliability values in the set

$$\min^{(r(\hat{\mathbf{c}}))}_{[i,j]\in\mathcal{E}(gh):\hat{c}_{[i,j]}\neq b_{[i,j]}} |y_{[i,j]}|. \tag{21}$$

Proof: Note that the codeword $\mathbf{c}$ obtained by erasing positions in $\mathcal{E}(gh)$, will agree with $\mathbf{b}$ in all positions $\mathcal{I}\setminus\mathcal{E}(gh)$, therefore we have from Definition 8 that $\mathcal{D}(\mathbf{c})$ must satisfy

$$\mathcal{D}(\mathbf{c}) = \sum_{[i,j]\in\mathcal{I}:c_{[i,j]}\neq b_{[i,j]}} |y_{[i,j]}| = \sum_{[i,j]\in\mathcal{E}(gh):c_{[i,j]}\neq b_{[i,j]}} |y_{[i,j]}|. \tag{22}$$

We want to show that the sum of all values in the set (21), is a lower bound to (22).

To see this, we consider the set $\{[i,j]\in\mathcal{I} : c_{[i,j]} \neq \hat{c}_{[i,j]}\}$ (i.e., the set of indices in which $\mathbf{c}$ and $\hat{\mathbf{c}}$) in which we partition as follows:

$$\{[i,j]\in\mathcal{I} : c_{[i,j]} \neq \hat{c}_{[i,j]}, \hat{c}_{[i,j]} = b_{[i,j]}\} \cap \{[i,j]\in\mathcal{I} : c_{[i,j]} \neq \hat{c}_{[i,j]}, \hat{c}_{[i,j]} \neq b_{[i,j]}\}.$$

It is clear that the first partition is nothing but

$$\{[i,j]\in\mathcal{I} : c_{[i,j]} \neq b_{[i,j]}, \hat{c}_{[i,j]} = b_{[i,j]}\} = \{[i,j]\in\mathcal{E}(gh) : c_{[i,j]} \neq b_{[i,j]}, \hat{c}_{[i,j]} = b_{[i,j]}\}$$

the previous set equality follows from similarly as (22).

Then, we note that the size of $\{[i,j]\in\mathcal{I} : c_{[i,j]} \neq \hat{c}_{[i,j]}\}$ is at least 4, because the minimum distance of the RS $[n, n-2, 3]$ binary image is 4 [this can be easily seen from the parity check matrix (2)]. By upper bounding the size of the second partition as $|\{[i,j]\in\mathcal{I} : \hat{c}_{[i,j]} \neq b_{[i,j]}\}|$, we see that $\mathbf{c}$ must differ from $\hat{\mathbf{c}}$ in at least $r(\hat{\mathbf{c}}) = 4 - |\{[i,j]\in\mathcal{I} : \hat{c}_{[i,j]} \neq b_{[i,j]}\}|$ positions in

[7] The permutation g simply equals the identity i_d in Algorithm A.

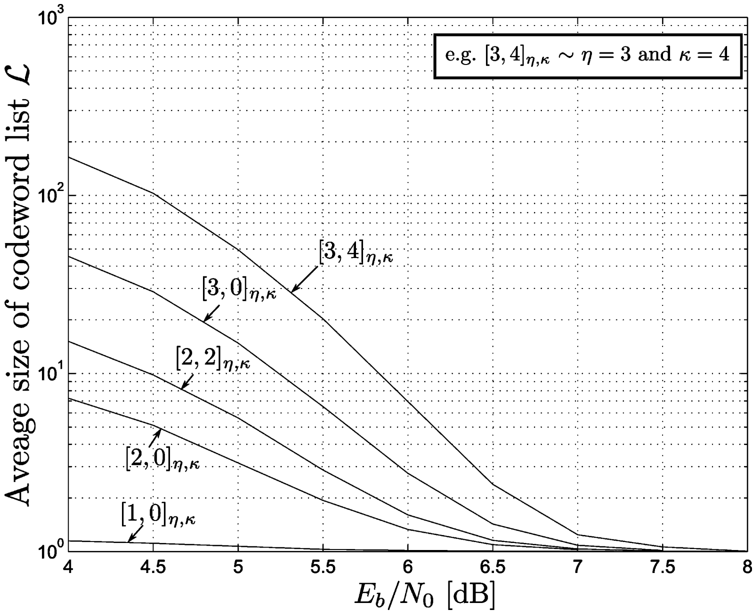


Fig. 9. Average complexity of permutation decoding Alg. A and B for the RS $[31, 29, 3]$.

$\mathcal{E}(gh)$. Thus, $\mathcal{D}(\mathbf{c})$ is lower bounded by the sum of all values in the set (21). ■

Let $\hat{\mathbf{c}}$ be the codeword with the lowest discrepancy $\mathcal{D}(\hat{\mathbf{c}})$ in the partially filled list $\mathcal{L}$. If the discrepancy $\mathcal{D}(\hat{\mathbf{c}})$ is lower than (21), then it is impossible that the codeword $\mathbf{c}$ obtained by erasing the set $\mathcal{E}(gh)$ will satisfy $\mathcal{D}(\mathbf{c}) < \mathcal{D}(\hat{\mathbf{c}})$. Thus, the necessary condition for erasing the set $\mathcal{E}(gh)$, is that (21) must be larger than $\mathcal{D}(\hat{\mathbf{c}})$.

We show the average complexities of both permutation decoding Algorithms A and B, when both sufficient and necessary conditions are utilized. We remind the reader that the average complexity is measured by the average list size $|\mathcal{L}|$. Note that the list need not contain unique codewords; each erasure decoding counts towards the complexity (list size). Results are shown for both RS codes $[31, 29, 3]$ and $[63, 61, 3]$, in Figs. 9 and 10, respectively. In both figures, the average complexities are seen to decrease monotonically as SNR increases. Also, we see that for a given value for parameter η, the average complexity of Algorithm B is in between that of Algorithm A for parameters η and $\eta + 1$ (a similar trend is observed as well for its bit error performance, see Figs. 6 and 7). The results in Figs. 9 and 10 testify to the effectiveness of the sufficient and necessary conditions in reducing the average complexity. For example, note that for the RS $[31, 29, 3]$ code and $\eta = 3$ and $\kappa = 4$, we see from Fig. 9 that the average size $|\mathcal{L}| \approx 2$ at 6.5 dB SNR, in contrast to 3889 (which is the fixed size of $\mathcal{L}$ if optimality conditions are not used). Further note that for the RS $[31, 29, 3]$ code and for this particular choice of parameters η, κ and SNR, we perform approximately 0.3 dB away from MLD at BER 10^{-5}. Thus, we claim that our permutation decoders are attractive alternatives for MLD, at the SNR's where the average complexity is seen to be very low. For the same 10^{-5} BER and 0.3 dB gain from MLD, the longer RS $[63, 61, 3]$ code has a slightly larger average size $|\mathcal{L}| \approx 8$ (as seen from Fig. 10) than that of the shorter RS $[31, 29, 3]$ code.

V. Code Automorphisms of Certain Triple-Parity RS Binary Images

In this section, we derive code automorphism subgroups of certain $[n, n-3, 4]$ RS codes over $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$ with zeros $\{1, \alpha, \alpha^2\}$, where α is a primitive element. Note that both elements α and α^2 belong to the same cyclotomic coset (i.e., $\mathcal{C}(\alpha) = \mathcal{C}(\alpha^2)$). Our results are summarized as follows. For the above mentioned $[7, 4, 4]$ RS binary images over $\mathbb{F}_{2^3}$, we show how to find an automorphism subgroup of order $3! \cdot 7 \cdot 6 \cdot 4 = 1008$. Our result improves that of Lacan *et al.* [4], in which they had previously reported an automorphism subgroup of order $3! \cdot 3 \cdot 7 = 126$. Also, for the above mentioned $[15, 12, 4]$ RS images over $\mathbb{F}_{2^4}$, we show how to construct a set of (at least 4 unique) code automorphisms, which together with the subgroup of code automorphisms that send $[i, j] \to [i, j + a]$ for all $a \in \mathbb{Z}_{15}$, generate a code automorphism subgroup of order at least 18. This result extends that of Lacan *et al.* [4], in which they did not consider RS codes over fields larger than $\mathbb{F}_{2^3}$ with zeros $\{1, \alpha, \alpha^2\}$.

A. *RS* $[7, 4, 4]$ *Binary Images Over* $\mathbb{F}_{2^3}$ *With Zeros* $\{1, \alpha, \alpha^2\}$

In the case $\mathbb{F}_{2^3}$, because the primal code has only dimension 4 (over $\mathbb{F}_{2^3}$), it is easier to derive the code automorphism subgroups directly from the primal code. First, we begin by describing the structure of code by looking at its $\mathbb{F}_2$-generator matrix. Note that since the primal code has zeros $\{1, \alpha, \alpha^2\}$ and $\mathcal{C}(\alpha) = \{\alpha, \alpha^2, \alpha^4\}$; thus, it has the set of nonzeros $\mathcal{N}$ of the form $\mathcal{N} = \mathbb{F}_{2^3} \setminus \{0, 1, \alpha, \alpha^2\} = \mathcal{C}(\alpha^3) \cup \{\alpha^4\}$. Recall that in $\mathbb{F}_{2^3}$, all elements except 0 and 1 are primitive, therefore α^3 is also primitive. Let γ denote the basis we image the primal code under.

Recall the irreducible cyclic code $\langle \theta_{\alpha^3}(x) \rangle$ of length $n = 7$ and dimension $|\mathcal{C}(\alpha^3)| = 3$ (see Propositions 1 and 2). Define $\mathcal{T}$ to be the $\mathbb{F}_2$-generator matrix (with entries in $\mathcal{R}_2 =$

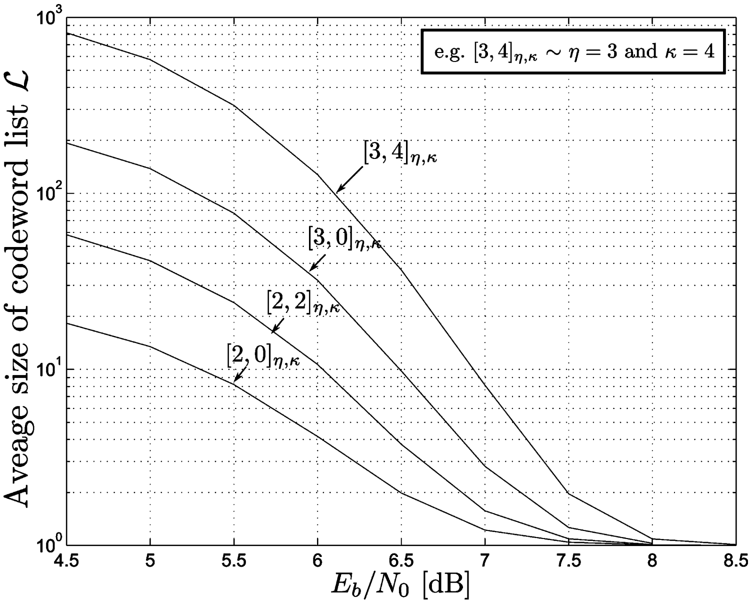


Fig. 10. Average complexity of permutation decoding Alg. A and B for the RS $[63, 61, 3]$.

$\mathbb{F}_2[x]/(x^7 - 1)$) of the irreducible cyclic code $\langle \theta_{\alpha^3}(x) \rangle$, given as

$$\mathcal{T} \triangleq \begin{bmatrix} \theta_{\alpha^3}(x) \\ \theta_{\alpha^3}(x)x \\ \theta_{\alpha^3}(x)x^2 \end{bmatrix}.$$

Recall the definition of the matrix $\mathcal{M}_{\alpha^4}$ (see (3)) with entries in $\mathcal{R}_2$, where in the case of $\mathbb{F}_{2^3}$ the matrix $\mathcal{M}_{\alpha^4}$ will have the size 3×3. In [4], it was shown that the binary image of a $[7, 4, 4]$ RS code over $\mathbb{F}_{2^3}$, with zeros $\{1, \alpha, \alpha^2\}$ (or nonzeros $\mathcal{N} = \mathcal{C}(\alpha^3) \cup \{\alpha^4\}$) has the following 12×3 $\mathbb{F}_2$-generator matrix with entries in $\mathcal{R}_2$

$$\begin{bmatrix} \mathcal{T} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{T} \\ & \mathcal{M}_{\alpha^4} & \end{bmatrix}. \tag{23}$$

Also recall from (3) that the matrix $\mathcal{M}_{\alpha^4}$ corresponds to a vector $\mathbf{u} = [u_1, u_2, u_3]^T$ of constants in $\mathbb{Z}_7$. The vector $\mathbf{u}$ can be determined (using the element α^4 and the basis γ) as in the statement of Proposition 3 (see [4]).

Recall the definitions of the symmetric groups Ω_3 and Φ_7 and the elementary cyclic shift $\varsigma \in \Phi_7$ (see Section IV). Recall the index set $\mathcal{I} \triangleq \{[i, j] : 1 \leq i \leq 3, j \in \mathbb{Z}_7\}$. In [4], it was shown that the permutation group $\mathcal{P}^{(1)}$ (see Proposition 4) is a code automorphism subgroup of the $[7, 4, 4]$ RS binary image with the $\mathbb{F}_2$-generator matrix (23). We briefly recap their result here.

Proposition 20: (Lacan *et al.* [4]). Let $\mathcal{P}^{(1)}$ be a permutation group acting on the index set $\mathcal{I}$. Let every element $g \in \mathcal{P}^{(1)}$ act on $\mathcal{B}_P(\mathbf{c}) = [c^{(1)}(x), c^{(2)}(x), c^{(3)}(x)]^T$ by sending the index $[i, j] \in \mathcal{I}$ to $[\sigma(i), j - u_i + u_{\sigma(i)}]$, for some $\sigma \in \Omega_3$. Then $\mathcal{P}^{(1)}$ is a subgroup of code automorphisms of the RS $[7, 4, 4]$ binary image with nonzeros $\mathcal{N} = \mathcal{C}(\alpha^3) \cup \{\alpha^4\}$ and $|\mathcal{P}^{(1)}| = 3! = 6$.

Proof: We can show the result by showing that the permutation group $\mathcal{P}^{(1)}$ leaves invariant the $\mathbb{F}_2$ row-spaces of both the first 9 rows of (23) and the submatrix $\mathcal{M}_{\alpha^4}$. Clearly, the action of any $g \in \mathcal{P}^{(1)}$, leaves invariant the $\mathbb{F}_2$ row-space of the first 9 rows of (23). Furthermore, the arguments in Proposition 4 show that the $\mathbb{F}_2$ row-space of $\mathcal{M}_{\alpha^4}$ is invariant under the action of $\mathcal{P}^{(1)}$. Thus, we are done. ■

Next we want to show that $\mathcal{P}^{(2)}$ (where here we require a slightly different definition for $\mathcal{P}^{(2)}$ from that of Proposition 5) is also a subgroup of code automorphisms belonging to the $[7, 4, 4]$ RS binary image with zeros $\{1, \alpha, \alpha^2\}$. We require the following result from [13].

Proposition 21: The polynomial $\theta_\alpha(x) + \theta_{\alpha^3}(x)$ is an idempotent and generates the $[7, 6, 2]$ binary single parity check code (in cyclic form) with a single zero $\{1\}$

Proof: It is clear the polynomial $\theta_\alpha(x) + \theta_{\alpha^3}(x)$ evaluates to 1 at elements $\mathbb{F}_{2^3} \backslash \{0, 1\}$, and 0 otherwise; thus, it is an idempotent of a cyclic code with a single zero $\{1\}$ (see [13 pg. 218, Lemma 2]). A binary cyclic code with a single zero $\{1\}$ is equivalent to a binary single parity check code. Finally by [13, pg. 217, Theorem 1], the idempotent of a cyclic code generates the code. ■

Recall that $Aut(\langle \theta_\alpha(x) \rangle)$ denotes the automorphism group of the code $\langle \theta_\alpha(x) \rangle$ (see Section IV). The following result is taken from [15]. Recall Proposition 2.

Proposition 22: (Lim *et al.* [15]). Let $\mathcal{P}^{(2)}$ be a permutation group acting on the index set $\mathcal{I}$. Let every $h \in \mathcal{P}^{(2)}$ act on $\mathcal{B}_P(\mathbf{c}) = [c^{(1)}(x), c^{(2)}(x), c^{(3)}(x)]^T$ by sending the index $[i, j] \in \mathcal{I}$ to $[i, \rho(j - u_i) + u_i]$, for some $\rho \in Aut(\langle \theta_{\alpha^3}(x) \rangle)$. Then $\mathcal{P}^{(2)}$ is a subgroup of code automorphisms of the RS $[7, 4, 4]$ binary image with nonzeros $\mathcal{N} = \mathcal{C}(\alpha^3) \cup \{\alpha^4\}$, and $|\mathcal{P}^{(2)}| = |Aut(\langle \theta_{\alpha^3}(x) \rangle)|$.

Proof: First, we show that any permutation in $\mathcal{P}^{(2)}$ leave the $\mathbb{F}_2$ row-space of the first 9 rows of (23) invariant. Consider the first row of (23), and we see that its first entry $\theta_{\alpha^3}(x)$ gets permuted as

$$\theta_{\alpha^3}(x) \xrightarrow{\varsigma^{-u_1}} \theta_{\alpha^3}(x)x^{-u_1} \xrightarrow{\rho} \theta_{\alpha^3}(x)x^{r} \xrightarrow{\varsigma^{u_1}} \theta_{\alpha^3}(x)x^{u_1+r}$$

where the second step holds because ρ belongs to $Aut\,(\langle\theta_{\alpha^3}(x)\rangle)$ and r is some element in $\mathbb{Z}_7$. Similar arguments apply to rows 2–9 of (23). Next, we consider the submatrix $\mathcal{M}_{\alpha^4}$. First note that if $\rho \in Aut\,(\langle\theta_{\alpha}(x)\rangle) \cap Aut\,(\langle\theta_{\alpha^3}(x)\rangle)$, then the $\mathbb{F}_2$ row-space of $\mathcal{M}_{\alpha^4}$ can be shown to be invariant under ρ using the arguments in Proposition 5 (note that $\theta_{\alpha^4}(x) = \theta_{\alpha}(x)$). We claim that if $\rho \in Aut\,(\langle\theta_{\alpha^3}(x)\rangle) \setminus Aut\,(\langle\theta_{\alpha}(x)\rangle)$, we only have the following two outcomes:

1) ρ permutes a codeword from $\langle\theta_{\alpha}(x)\rangle$ into $\langle\theta_{\alpha^3}(x)\rangle$.
2) ρ permutes a codeword from $\langle\theta_{\alpha}(x)\rangle$ into a cyclic shift of the form $\theta_{\alpha}(x) + \theta_{\alpha^3}(x)x^a$, where $a \in \mathbb{Z}_7$.

The argument is as follows. From Proposition 21, note that $\langle\theta_{\alpha}(x) + \theta_{\alpha^3}(x)\rangle$ is equivalent to the $[7, 6, 2]$ binary single parity check code, and has the symmetric group Φ_7 as its automorphism group. Therefore, we must have $\rho \in Aut\,(\langle\theta_{\alpha}(x) + \theta_{\alpha^3}(x)\rangle) = \Phi_7$. Also, we note that $\langle\theta_{\alpha}(x)\rangle$ is a subcode of $\langle\theta_{\alpha}(x) + \theta_{\alpha^3}(x)\rangle$. Therefore, ρ must permute a codeword from $\langle\theta_{\alpha}(x)\rangle$ to a codeword from $\langle\theta_{\alpha}(x) + \theta_{\alpha^3}(x)\rangle$, which implies that either one of the cases 1) and 2) must occur (note that $\langle\theta_{\alpha}(x) + \theta_{\alpha^3}(x)\rangle$ is the direct sum of $\langle\theta_{\alpha}(x)\rangle$ and $\langle\theta_{\alpha^3}(x)\rangle$).

Choose any row $[\theta_{\alpha}(x)x^{u_1+r}, \theta_{\alpha}(x)x^{u_2+r}, \theta_{\alpha}(x)x^{u_3+r}]^T$ in the submatrix $\mathcal{M}_{\alpha^4}$ (for some $r \in \{0, 1, 2\}$, see (3)). If 1) occurs, then the chosen row permutes to a row vector $[\theta_{\alpha^3}(x)x^{u_1+r'}, \theta_{\alpha^3}(x)x^{u_2+r'}, \theta_{\alpha^3}(x)x^{u_3+r'}]^T$ (for some $r' \in \mathbb{Z}_7$) in the rowspace of (23). If 2) occurs, the the ith entry $\theta_{\alpha}(x)x^{u_i+r}$ of the chosen row gets permuted as

$$\begin{aligned}\theta_{\alpha}(x)x^{u_i+r} &\xrightarrow{\varsigma^{-u_i}} \theta_{\alpha}(x)x^{r} \xrightarrow{\rho} \theta_{\alpha}(x)x^{r'} + \theta_{\alpha^3}(x)x^{r'+a}\\ &\xrightarrow{\varsigma^{u_i}} \theta_{\alpha}(x)x^{u_i+r'} + \theta_{\alpha^3}(x)x^{u_i+r'+a}\end{aligned}$$

for some $a,\ r' \in \mathbb{Z}_7$. We see that both row vectors $[\theta_{\alpha}(x)x^{u_1+r'}, \theta_{\alpha}(x)x^{u_2+r'}, \theta_{\alpha}(x)x^{u_3+r'}]^T$ and $[\theta_{\alpha^3}(x)x^{u_1+r'+a}, \theta_{\alpha^3}(x)x^{u_2+r'+a}, \theta_{\alpha^3}(x)x^{u_3+r'+a}]^T$ lie in the $\mathbb{F}_2$ row-space of (23); thus, so must their sum. Hence, we are done. ■

Example 6: Let α be a fixed primitive element that satisfies $\alpha^3 = \alpha + 1$. Consider the $[7, 4, 4]$ RS code with zeros $\{1, \alpha^6, (\alpha^6)^2\} = \{1, \alpha^6, \alpha^5\}$. Its set of nonzeros are given as $\mathcal{N} = \mathcal{C}((\alpha^6)^3) \cup \{(\alpha^6)^4\} = \mathcal{C}(\alpha^4) \cup \{\alpha^3\} = \{\alpha, \alpha^2, \alpha^3, \alpha^4\}$. As in the statement of Proposition 3, we determine that $\mathbf{u} = [2, 5, 0]^T$ (using α^3 and the basis $\gamma = [1, \alpha, \alpha^2]^T$). Proposition 20 holds for this particular code. Also, using Proposition 22, we determine that the binary image of this particular code is left invariant by the action of $\mathcal{P}^{(2)}$, which sends $[i, j] \in \mathcal{I}$ to $[i, \rho(j - u_i) + u_i]$, where $\rho \in Aut\,(\langle\theta_{\alpha^4}(x)\rangle)$.

Remark 5: We wish to point out that an alternative proof of Proposition 20 is given in [15].

Some final remarks on the code automorphism subgroups derived for the triple-parity $[7, 4, 4]$ RS binary image with zeros $\{1, \alpha, \alpha^2\}$. Because Propositions 20 and 22 are so similar with Propositions 4 and 5 (which derive the code automorphism subgroups for the double-parity $[7, 5, 3]$ RS binary image with zeros $\{1, \alpha\}$) in Section IV, Propositions 6 and 7 apply and the total number of code automorphisms found are $|\mathcal{P}^{(1)}||\mathcal{P}^{(2)}| = |\mathcal{P}^{(1)}\mathcal{P}^{(2)}| = 3! \cdot 7 \cdot 6 \cdot 4 = 1008$. Furthermore in a similar manner, all the results in Sections IV-A–IV-D apply to the triple-parity $[7, 4, 4]$ RS binary image over $\mathbb{F}_{2^3}$.

B. The Case $\mathbb{F}_{2^4}$

For the case of $\mathbb{F}_{2^4}$, we derive code automorphism subgroups of $[15, 12, 4]$ RS binary images with zeros $\{1, \alpha, \alpha^2\}$, from their code dual binary images (similar to Section IV). Recall that α is always fixed to be primitive. We begin by describing the structure of the $[15, 3, 13]$ code duals with non zeros $\mathcal{N} = \{1, \breve{\alpha}, \breve{\alpha}^2\} = \{1, \alpha^{-1}, \alpha^{-2}\}$. In [4], it was shown that the $\mathbb{F}_2$-generator matrix of the code dual (binary image) is given by the following 12×4 matrix

$$\begin{bmatrix} \theta_1(x) & 0 & \cdots & 0 \\ 0 & \theta_1(x) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \theta_1(x) \\ & \mathcal{M}_{\breve{\alpha}} & & \\ & \mathcal{M}_{\breve{\alpha}^2} & & \end{bmatrix} \tag{24}$$

where both $\mathcal{M}_{\breve{\alpha}}$ and $\mathcal{M}_{\breve{\alpha}^2}$ are defined in (3). We denote $\mathbf{u} = [u_1, u_2, u_3, u_4]^T$ and $\mathbf{u'} = [u'_1, u'_2, u'_3, u'_4]^T$ to be the vectors that correspond to $\mathcal{M}_{\breve{\alpha}}$ and $\mathcal{M}_{\breve{\alpha}^2}$, respectively (see (3)). It was shown in [4] that the vectors $\mathbf{u}$ and $\mathbf{u'}$ can be determined (using $\breve{\alpha}$ and $\breve{\alpha}^2$ and the trace-dual basis $\gamma^{\perp}$) as in the statement of Proposition 3.

The approach we have taken so far to derive code automorphism subgroups for the various RS binary images, does not apply here. This is because (24) has a dense submatrix $[\mathcal{M}_{\breve{\alpha}}^T, \mathcal{M}_{\breve{\alpha}^2}^T]^T$ which corresponds to two vectors $\mathbf{u}$ and $\mathbf{u'}$, while previously we only had a dense submatrix of the form $\mathcal{M}_{\alpha}^T$ (for some $\alpha \in \mathbb{F}_{2^m}$) and a single vector $\mathbf{u}$. We follow the strategy in [15] to derive permutations that leave some $\mathbb{F}_2$-basis of $[\mathcal{M}_{\breve{\alpha}}^T, \mathcal{M}_{\breve{\alpha}^2}^T]^T$ invariant under their action. We require the following result by Sakakibara *et al.*, where we restate it in a slightly different way for the specific case that we are interested in.

Proposition 23: (Sakakibara *et al.* [2, Theorem 3]). Recall that $\breve{\alpha}$ is also primitive. Then the only vector with more than 1 nonzero entry, in the $\mathbb{F}_2$-rowspace of the submatrix $[\mathcal{M}_{\breve{\alpha}}^T, \mathcal{M}_{\breve{\alpha}^2}^T]^T$, is the all zero vector $[0, 0, 0, 0]^T$.

Proof: Shown by noting that both $\breve{\alpha}$ and $\breve{\alpha}^2$ are consecutive powers of a primitive element $\breve{\alpha}$, which allows us to apply Theorem 3 of [2] to obtain the result. ■

The following two propositions serve to find a set of $\mathbb{F}_2$-bases of $[\mathcal{M}_{\breve{\alpha}}^T, \mathcal{M}_{\breve{\alpha}^2}^T]^T$, from which code automorphisms may be identified.

Proposition 24: Consider the submatrix $[\mathcal{M}_{\breve{\alpha}}^T, \mathcal{M}_{\breve{\alpha}^2}^T]^T$, of the $\mathbb{F}_2$-generator matrix belonging to the $[15, 4, 12]$ RS binary image with $\mathcal{N} = \{1, \check{\alpha}, \check{\alpha}^2\}$. There exists a set of constants $\{a_{i,j} : 1 \leq i, j \leq 4, i \neq j\}$ in $\mathbb{Z}_{15}$, whereby the rows of the following 4×4 matrix:

$$\begin{bmatrix} 0 & \theta_{\check{\alpha}}(x)x^{a_{1,2}} & \theta_{\check{\alpha}}(x)x^{a_{1,3}} & \theta_{\check{\alpha}}(x)x^{a_{1,4}} \\ \theta_{\check{\alpha}}(x)x^{a_{2,1}} & 0 & \theta_{\check{\alpha}}(x)x^{a_{2,3}} & \theta_{\check{\alpha}}(x)x^{a_{2,4}} \\ \theta_{\check{\alpha}}(x)x^{a_{3,1}} & \theta_{\check{\alpha}}(x)x^{a_{3,2}} & 0 & \theta_{\check{\alpha}}(x)x^{a_{3,4}} \\ \theta_{\check{\alpha}}(x)x^{a_{4,1}} & \theta_{\check{\alpha}}(x)x^{a_{4,2}} & \theta_{\check{\alpha}}(x)x^{a_{4,3}} & 0 \end{bmatrix} \quad (25)$$

lie in the $\mathbb{F}_2$-rowspace of $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$.

Proof: Recall that the $\mathbb{F}_2$-rowspace of $\mathcal{M}_{\check{\alpha}}$ contains all row vectors of the form $[\theta_{\check{\alpha}}(x)x^{u_1+r}, \ldots, \theta_{\check{\alpha}}(x)x^{u_4+r}]$ for all $r \in \mathbb{Z}_{15}$. Also because $\theta_{\check{\alpha}}(x) = \theta_{\check{\alpha}^2}(x)$, the $\mathbb{F}_2$-rowspace of $\mathcal{M}_{\check{\alpha}'}$ contains all vectors of the form $[\theta_{\check{\alpha}}(x)x^{u'_1+r}, \ldots, \theta_{\check{\alpha}}(x)x^{u'_4+r}]$ for all $r \in \mathbb{Z}_n$. Thus, we can select one vector each from the $\mathbb{F}_2$-rowspaces of $\mathcal{M}_{\check{\alpha}}$ and $\mathcal{M}_{\check{\alpha}^2}$, such that at least one of their entries are the same. It follows from Proposition 23 and the fact that $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$ is full-rank, that the selected vectors cannot agree in more than one entry. For the two chosen vectors that agree in the ith entry, we add them and obtain the set of constants $\{a_{i,j} : 1 \leq j \leq 4, i \neq j\}$. Thus, we are done. ■

Proposition 25: Choose any 2 rows from (25), say $[b_1^{(1)}(x), \ldots, b_4^{(1)}(x)]$ and $[b_1^{(2)}(x), \ldots, b_4^{(2)}(x)]$, where $b_i^{(1)}(x), b_i^{(2)}(x) \in \mathcal{R}_2$ for all $i \in \{1, 2, 3, 4\}$. Then the following two sets of vectors $\{[b_1^{(1)}(x)x^r, \ldots, b_4^{(1)}(x)x^r] : 0 \leq r < 4\}$ and $\{[b_1^{(2)}(x)x^r, \ldots, b_4^{(2)}(x)x^r] : 0 \leq r < 4\}$ constitute an $\mathbb{F}_2$-basis of the submatrix $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$.

Proof: It suffices to simply show that both sets

$$\{[b_1^{(1)}(x)x^r, \ldots, b_4^{(1)}(x)x^r]^T : 0 \leq r < 4\}$$
$$\{[b_1^{(2)}(x)x^r, \ldots, b_4^{(2)}(x)x^r]^T : 0 \leq r < 4\}$$

span a space of dimension 8 over $\mathbb{F}_2$. First note that both sets each have dimension 4 over $\mathbb{F}_2$, because there exists two nonzero polynomials $b_{i_1}^{(1)}(x) \neq 0$ and $b_{i_2}^{(2)}(x) \neq 0$ such that both sets $\{b_{i_1}^{(1)}(x)x^r : 0 \leq r < 4\}$ and $\{b_{i_2}^{(2)}(x)x^r : 0 \leq r < 4\}$ individually span the irreducible cyclic code $\langle \theta_{\check{\alpha}}(x) \rangle$ (see (25)). Furthermore, we claim that since both $[b_1^{(1)}(x), \ldots, b_4^{(1)}(x)]$ and $[b_1^{(2)}(x), \ldots, b_4^{(2)}(x)]$ have 0 elements in different entry locations, the spaces spanned by each set must be disjoint. This follows from Proposition 23, as a vector residing in both spaces must have more than one 0 entry, which happens to be all-zero vector $[0, 0, 0, 0]^T$. ■

We are now ready to use the two preceding Propositions 24 and 25 to construct a set of code automorphisms that leaves invariant the $[15, 12, 4]$ RS binary image over $\mathbb{F}_{2^4}$ with zeros $\{1, \alpha, \alpha^2\}$. The following proposition is taken from [15]. To facilitate the exposition, with regards to the constants $\{a_{i,j} : 1 \leq i, j \leq 4, i \neq j\}$ in (25), we notate the (permuted) constant $a_{i,j}^{(\sigma)} \triangleq a_{\sigma(i),\sigma(j)}$ with respect to an element $\sigma \in \Omega_4$.

Proposition 26: (Lim *et al.* [15]). Let any $\sigma \in \Omega_4$ refer to a bijective mapping from $\{1, 2, 3, 4\} \mapsto \{1, 2, 3, 4\}$. Define the constant $\delta(\sigma) \in \mathbb{Z}_{15}$ as

$$\begin{aligned} \delta(\sigma) &\triangleq a_{2,3}^{(\sigma)} - a_{1,3}^{(\sigma)} + a_{2,4}^{(\sigma)} - a_{1,4}^{(\sigma)} \\ &= a_{\sigma(2),\sigma(3)} - a_{\sigma(1),\sigma(3)} \\ &\quad + a_{\sigma(2),\sigma(4)} - a_{\sigma(1),\sigma(4)} \end{aligned} \quad (26)$$

where the set of constants $\{a_{i,j} : 1 \leq i, j \leq 4, i \neq j\}$ are defined in Proposition 24. Let $\{g(\sigma) : \sigma \in \Omega_4\}$ be a set of permutations[8] acting on the index set $\mathcal{I}$. Let each $g(\sigma)$ act on $\mathcal{B}_P(\mathbf{c}) = [c^{(1)}(x), c^{(2)}(x), c^{(3)}(x), c^{(4)}(x)]^T$ by sending the indices as follows:

$$\begin{aligned} [\sigma(1), j] &\overset{g(\sigma)}{\rightarrow} \left[\sigma(2), j - a_{2,1}^{(\sigma)} + a_{1,2}^{(\sigma)} + \delta(\sigma)\right] \\ [\sigma(2), j] &\overset{g(\sigma)}{\rightarrow} \left[\sigma(1), j - a_{1,2}^{(\sigma)} + a_{2,1}^{(\sigma)}\right] \\ [\sigma(3), j] &\overset{g(\sigma)}{\rightarrow} \left[\sigma(4), j - a_{1,3}^{(\sigma)} + a_{2,4}^{(\sigma)}\right] \\ [\sigma(4), j] &\overset{g(\sigma)}{\rightarrow} \left[\sigma(3), j - a_{1,4}^{(\sigma)} + a_{2,3}^{(\sigma)}\right] \end{aligned} \quad (27)$$

for all $j \in \mathbb{Z}_{15}$ and $\delta(\sigma)$ defined in (26). Then all permutations in the set $\{g(\sigma) : \sigma \in \Omega_4\}$ are code automorphisms for the $[15, 12, 4]$ RS binary image over $\mathbb{F}_{2^4}$ with zeros $\{1, \alpha, \alpha^2\}$.

Proof: First, note that the $\mathbb{F}_2$-rowspace of the first 3 rows of the $\mathbb{F}_2$-generator matrix (24) is clearly left invariant under the action of all permutations in $\{g(\sigma) : \sigma \in \Omega_4\}$. We are then left to show that the $\mathbb{F}_2$-rowspace of the submatrix $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$ is left invariant. Consider some permutation $g(\sigma)$, and recall that for all constants in the set $\{a_{i,j} : 1 \leq i, j \leq 4, i \neq j\}$ and some $\sigma \in \Omega_4$, we denote $a_{i,j}^{(\sigma)} \triangleq a_{\sigma(i),\sigma(j)}$. By Proposition 25, the following matrix:

$$\begin{bmatrix} 0 & \theta_{\check{\alpha}}(x)x^{a_{1,2}^{(\sigma)}} & \theta_{\check{\alpha}}(x)x^{a_{1,3}^{(\sigma)}} & \theta_{\check{\alpha}}(x)x^{a_{1,4}^{(\sigma)}} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \theta_{\check{\alpha}}(x)x^{a_{1,2}^{(\sigma)}+3} & \theta_{\check{\alpha}}(x)x^{a_{1,3}^{(\sigma)}+3} & \theta_{\check{\alpha}}(x)x^{a_{1,4}^{(\sigma)}+3} \\ \theta_{\check{\alpha}}(x)x^{a_{2,1}^{(\sigma)}} & 0 & \theta_{\check{\alpha}}(x)x^{a_{2,3}^{(\sigma)}} & \theta_{\check{\alpha}}(x)x^{a_{2,4}^{(\sigma)}} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{\check{\alpha}}(x)x^{a_{2,1}^{(\sigma)}+3} & 0 & \theta_{\check{\alpha}}(x)x^{a_{2,3}^{(\sigma)}+3} & \theta_{\check{\alpha}}(x)x^{a_{2,4}^{(\sigma)}+3} \end{bmatrix} \quad (28)$$

has an equivalent $\mathbb{F}_2$-rowspace to the submatrix $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$ (up to a column permutation of $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$ by σ^{-1}). It follows then that $g(\sigma)$ invariants the $\mathbb{F}_2$-rowspace of $[\mathcal{M}_{\check{\alpha}}^T, \mathcal{M}_{\check{\alpha}^2}^T]^T$, if and only if the permutation

$$\begin{aligned} [1, j] &\rightarrow \left[2, j - a_{2,1}^{(\sigma)} + a_{1,2}^{(\sigma)} + \delta(\sigma)\right] \\ [2, j] &\rightarrow \left[1, j - a_{1,2}^{(\sigma)} + a_{2,1}^{(\sigma)}\right] \\ [3, j] &\rightarrow \left[4, j - a_{1,3}^{(\sigma)} + a_{2,4}^{(\sigma)}\right] \\ [4, j] &\rightarrow \left[3, j - a_{1,4}^{(\sigma)} + a_{2,3}^{(\sigma)}\right] \end{aligned} \quad (29)$$

invariants the $\mathbb{F}_2$-rowspace of the matrix (28). Consider $[0, \theta_{\check{\alpha}}(x)x^{a_{1,2}^{(\sigma)}+r}, \theta_{\check{\alpha}}(x)x^{a_{1,3}^{(\sigma)}+r}, \theta_{\check{\alpha}}(x)x^{a_{1,4}^{(\sigma)}+r}]$ (for some $r \in \mathbb{Z}_{15}$) from the $\mathbb{F}_2$-rowspace of the first 3 rows of (28).

[8] Note that we repeated the notation g to indicate a permutation $g(\sigma) \in \mathcal{I}$, where g previously indicated a permutation in the group $\mathcal{P}^{(1)}$ (see Propositions 4 and 20). While $g(\sigma)$ does not possess the exact same structure as permutations in $\mathcal{P}^{(1)}$, we re-used the notation to indicate their similarities. They are similar in the sense that both $g(\sigma)$ and permutations in $\mathcal{P}^{(1)}$, are the only code automorphisms described in this paper, that permute the first element i in the index tuple $[i, j] \in \mathcal{I}$.

TABLE III
CODE AUTOMORPHISM SUBGROUP ORDERS, OBTAINED USING BOTH THEORY AND THE GROUPS, ALGORITHMS, AND PROGRAMMING (GAP) [19] SOFTWARE

Field	No. of Parity	Starting power of zeros s	Zeros	Characterized by Theory (any basis)	GAP Comp. (can. basis)
$\mathbb{F}_{2^3}$	1	$1 \sim 6$	α^s	$2^4{\times}3^2{\times}7$	$2^{10}{\times}3^8{\times}7$
	2	$0,\ 6$	$1, \alpha^s$	$2^4{\times}3^2{\times}7$	$2^4{\times}3^2{\times}7$
	2	1	α, α^2	$2^4{\times}3^2{\times}7$	$2^4{\times}3^2{\times}7$
	2	$2 \sim 4$	α^s, α^{s+1}	7	$2{\times}7$
	2	5	α^5, α^6	$2^4{\times}3^2{\times}7$	$2^4{\times}3^2{\times}7$
	3	0	$1, \alpha, \alpha^2$	$2^4{\times}3^2{\times}7$	$2^4{\times}3^2{\times}7$
	3	$1 \sim 4, 6$	$\alpha^s, \alpha^{s+1}, \alpha^{s+2}$	7	$7 \sim 2{\times}7$
	3	5	$1, \alpha^5, \alpha^6$	$2^4{\times}3^2{\times}7$	$2^4{\times}3^2{\times}7$
$\mathbb{F}_{2^4}$	1	*s.t.* $\gcd(15, s) = 1$	α^s	$2^9{\times}3^3{\times}5{\times}7$	$2^{51}{\times}3^{17}{\times}5{\times}7$
	2	$0, 14$	$1, \alpha$	$2^9{\times}3^3{\times}5{\times}7$	$2^9{\times}3^3{\times}5{\times}7$
	2	$1, 13$	α^s, α^{s+1}	≥ 18	$2^3{\times}3{\times}5$
	3	$0, 13$	$1, \alpha, \alpha^2$	≥ 18	$2^3{\times}3{\times}5$
	3	$1 \sim 12, 14$	$\alpha^s, \alpha^{s+1}, \alpha^{s+2}$	$3{\times}5$	$3{\times}5 \sim 2{\times}3{\times}5$
$\mathbb{F}_{2^5}$	1	$1 \sim 30$	α^s	$2^{13}{\times}3^3{\times}5^2{\times}7{\times}31$	-
	2	$0, 30$	$1, \alpha$	$2^{13}{\times}3^3{\times}5^2{\times}7{\times}31$	$2^{13}{\times}3^3{\times}5^2{\times}7{\times}31$
	2	$1 \sim 29$	α^s, α^{s+1}	31	$31 \sim 2^2{\times}31$
	3	$1 \sim 30$	$\alpha^s, \alpha^{s+1}, \alpha^{s+2}$	31	$31 \sim 2^2{\times}31$

Under the action of the permutation (29), this row permutes as follows:

$$\begin{aligned}&[0, \theta_{\breve{\alpha}}(x)x^{a_{1,2}^{(\sigma)}+r}, \theta_{\breve{\alpha}}(x)x^{a_{1,3}^{(\sigma)}+r}, \theta_{\breve{\alpha}}(x)x^{a_{1,4}^{(\sigma)}+r}]^T\\ &\qquad \rightarrow [\theta_{\breve{\alpha}}(x)x^{a_{2,1}^{(\sigma)}+r}, 0, \theta_{\breve{\alpha}}(x)x^{a_{2,3}^{(\sigma)}+r}, \theta_{\breve{\alpha}}(x)x^{a_{2,4}^{(\sigma)}+r}]^T\end{aligned}$$

where the result lies in the $\mathbb{F}_2$-rowspace of the last 4 rows of (28). Next consider an element from the $\mathbb{F}_2$-rowspace of the last 4 rows of (28). Under the action of the permutation (29), we have

$$\begin{aligned}&[\theta_{\breve{\alpha}}(x)x^{a_{2,1}^{(\sigma)}+r}, 0, \theta_{\breve{\alpha}}(x)x^{a_{2,3}^{(\sigma)}+r}, \theta_{\breve{\alpha}}(x)x^{a_{2,4}^{(\sigma)}+r}]^T\\ &\rightarrow [0, \theta_{\breve{\alpha}}(x)x^{a_{1,2}^{(\sigma)}+\delta(\sigma)+r}, \theta_{\breve{\alpha}}(x)x^{a_{2,4}^{(\sigma)}-a_{1,4}^{(\sigma)}+a_{2,3}^{(\sigma)}+r}\\ &\qquad \theta_{\breve{\alpha}}(x)x^{a_{2,3}^{(\sigma)}-a_{1,3}^{(\sigma)}+a_{2,4}^{(\sigma)}+r}]^T\\ &= [0, \theta_{\breve{\alpha}}(x)x^{a_{1,2}^{(\sigma)}+\delta(\sigma)+r}, \theta_{\breve{\alpha}}(x)x^{a_{1,3}^{(\sigma)}+\delta(\sigma)+r}\\ &\qquad \theta_{\breve{\alpha}}(x)x^{a_{1,4}^{(\sigma)}+\delta(\sigma)+r}]^T\end{aligned}$$

which is in the $\mathbb{F}_2$-rowspace of the first 4 rows of (28). Thus, we are done. ■

Note that it is not guaranteed that the size $|\{g(\sigma) : \sigma \in \Omega_4\}| = |\Omega_4| = 24$. This is because, depending on the set of constants $\{a_{i,j} : 1 \leq i, j \leq 4, i \neq j\}$, the permutation (27) may be equal for two different $\sigma, \sigma' \in \Omega_4$.

Example 7: Let α be a fixed primitive element that satisfies $\alpha^4 = \alpha + 1$. Consider the $[15, 12, 4]$ RS binary image with zeros $\{1, \alpha, \alpha^2\}$, imaged under the canonical basis. The dual code is the binary image of the $[15, 3, 13]$ RS code with $\mathcal{N} = \{1, \alpha^{14}, \alpha^{13}\}$, imaged under the trace-dual basis $[\alpha^{14}, \alpha^2, \alpha, 1]^T$. It can be verified that matrix (25) is given as

$$\begin{bmatrix} 0 & \theta_{\alpha^{14}}(x)x^7 & \theta_{\alpha^{14}}(x)x^{12} & \theta_{\alpha^{14}}(x)x \\ \theta_{\alpha^{14}}(x)x^8 & 0 & \theta_{\alpha^{14}}(x)x^{14} & \theta_{\alpha^{14}}(x)x^4 \\ \theta_{\alpha^{14}}(x)x^{14} & \theta_{\alpha^{14}}(x) & 0 & \theta_{\alpha^{14}}(x)x^6 \\ \theta_{\alpha^{14}}(x)x^4 & \theta_{\alpha^{14}}(x)x^6 & \theta_{\alpha^{14}}(x)x^7 & 0 \end{bmatrix}.$$

Using Proposition 26, we obtain 12 unique code automorphisms $g(\sigma)$, corresponding to the mappings σ in the set (here denoted in permutation form) $\{i_d, (3,4), (2,3), (2,4,3), (1,2)(3,4), (1,2), (1,2,3,4), (1,2,4,3), (1,3,4,2), (1,3)(2,4), (1,3,2,4), (1,4)(2,3)\}$.

Note that there must be at least 3 unique permutations $g(\sigma)$, formed by taking σ to be in the set $\{(1,2)(3,4), (1,3)(2,4), (1,4)(2,3)\}$ (the set known as the conjugacy class[9] of (1,2)(3,4)). From the structure of $g(\sigma)$ (given in (27)), it is not clear what the order of the group generated by the set $\{g(\sigma) : \sigma \in \Omega_4\}$ is. However, we know that there exists at least $15 + 3 = 18$ unique code automorphisms of the $[15, 12, 4]$ RS binary image with zeros $\{1, \alpha, \alpha^2\}$; there are 15 of them that send $[i, j] \rightarrow [i, j + a]$ for all $a \in \mathbb{Z}_{15}$, and 3 unique permutations in the set $\{g(\sigma) : \sigma \in \Omega_4\}$.

VI. CONCLUSION

In the conclusion, we present a table of code automorphism (subgroup) orders computed using the Groups, Algorithms, and Programming (GAP) software [19]. Table III shows the automorphism subgroup orders computed for various RS codes over $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^5}$, imaged under the canonical basis. Due to computational limitations, only codes with 1∼3 parity symbols are listed in Table III, but recall that these results may be extrapolated to their dual codes and subcodes. For comparison purposes, we also include the orders of the automorphism subgroups that have been characterized by theory. The shaded entries of Table III, indicate wherever the subgroup orders, obtained by both theory and computations, match. Finally, we also point out that the primitive element α indicated in Table III, satisfies the relationships shown in Table I.

From Table III, we see that for most of the RS codes over $\mathbb{F}_{2^3}$, the subgroup orders obtained using theory, match[10] the subgroup orders computed by GAP. The limitation of the current theory is obvious for triple-parity codes over fields $\mathbb{F}_{2^4}$ and larger. We observe a drastic decrease in automorphism group sizes as the dimension increases, which incidently correlates to the increase in difficulty in obtaining theoretical results. Finally, we would like to point out that the single-parity codes

[9] For a group A and an element $a \in A$, the conjugacy class of a is its orbit, under the conjugation of the elements of A.

[10] Note that two groups of the same order are not necessarily equal.

Algorithm 1: Computing a simplex automorphism

```
Input: Matrix $\mathbf{S}$ and vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_n^m$
Output: The permutation $\rho$ if valid.
Initialize: Set $\rho(j) := -1$ for all $j \in \mathbb{Z}_n$
forall $i = 1$ to $m$ do
    Make a copy $\rho'$ of $\rho$;
    forall $j \in \{j \in \mathbb{Z}_n | \rho'(j) \neq -1\}$ do
        if $\mathbf{s}[b_i] + \mathbf{s}[\rho'(j)] = \mathbf{0}$ then quit;
        if $\mathbf{s}[a_i] + \mathbf{s}[j] = \mathbf{0}$ then quit;
        Set $j' := \varphi(\mathbf{s}[a_i] + \mathbf{s}[j])$;
        Set $\rho(j') := \varphi(\mathbf{s}[b_i] + \mathbf{s}[\rho'(j)])$;
    end
    Set $\rho(a_i) := b_i$;
end
```

have a much larger automorphism subgroup found by GAP, than what is obtained using theory. This is due to a limitation in our theory, as we only considered code automorphisms that collectively permute each row of the binary image $\mathcal{B}_M(\mathbf{c})$ (see Definition 1). That is, the groups $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ (see Propositions 4, 20 and 22) had the structure, whereby if the index $[i, j] \in \mathcal{I}$ permuted to some index which belongs to row i', then all other indices $[i, j']$ (where $j' \neq j$) will also permute to row i'.

In this paper, we extended Lacan *et al.*'s results in [4] for the special case of binary extension fields $\mathbb{F}_2$. We derived the automorphism subgroups of the double-parity $[n, n-2, 3]$ RS binary images with zeros $\{1, \alpha\}$. For the fields $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$, we also derived automorphism subgroups of triple-parity $[n, n-2, 3]$ RS binary images with zeros $\{1, \alpha, \alpha^2\}$. We showed that the derived code automorphisms of the double-parity $[n, n-2, 3]$ binary image, have specific structure which can be exploited for designing low-complexity permutation decoders. We showed that our permutation decoders for the $[31, 29, 3]$ and $[63, 61, 3]$ binary images, perform close to MLD, and have extremely low average complexity at moderate SNRs.

Appendix

1) Efficient Computation of a Simplex Automorphism $\rho \in Aut(\langle\theta_{\tilde{\alpha}}(x)\rangle)$: Algorithm 1 accepts as input $\mathbf{S}$ (see (4)) and vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_n^m$, and computes an element $\rho \in Aut(\langle\theta_{\tilde{\alpha}}(x)\rangle)$ that sends $\rho(a_i) = b_i$ for all $i \in \{1, 2, \ldots, m\}$. For sake of clearer presentation, we slightly abuse notation and allow a permutation $\rho \in \Phi_n$ to map to -1. That is, we may have $\rho(j) = -1$ for some $j \in \mathbb{Z}_n$, and in this case we say that ρ is an *invalid* permutation. If Algorithm 1 returns an invalid permutation, then it is unable to find a simple automorphism that sends $\rho(a_i) = b_i$ for all $i \in \{1, 2, \ldots, m\}$. We also define a map $\varphi : \mathbb{F}_2^m \setminus \{\mathbf{0}\} \mapsto \mathbb{Z}_n$, that maps the (unique) columns of $\mathbf{S}$ to their column indexes (i.e., if we have $\mathbf{s}[j] = \nu$ for some $\nu \in \mathbb{F}_2^m$ and $j \in \mathbb{Z}_n$, then we have $\varphi(\nu) = j$).

To obtain a permutation $\rho \in Aut(\langle\theta_{\tilde{\alpha}}(x)\rangle)$ that satisfies Proposition 10 (the conditions where this is guaranteed to be possible is given in Proposition 12), run Algorithm 1 with inputs $\mathbf{S}, \mathbf{j} - \mathbf{u}$ and $-\mathbf{u}$.

Acknowledgment

The authors would like to thank GAP software and J. S. Leon (ftp://ftp.math.uic.edu/pub/leon/partn) for the Partition Backtrack algorithm [20], which were used to perform group computations in this work. The authors would also like to thank the reviewers for their constructive comments which greatly improved the draft.

References

[1] K. Sakakibara, K. Tokiwa, and M. Kasahara, "Notes on q-ary expanded Reed–Solomon codes over $\mathrm{GF}(q^m)$," *Electron. Commun. Jpn. (Part III: Fundam. Electron. Sci.)*, vol. 72, no. 2, pp. 14–23, 1989.

[2] K. Sakakibara and M. Kasahara, "On the minimum distance of a q-ary image of a q^m-ary cylic code," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1631–1635, Sep. 1996.

[3] G. E. Seguin, "The q-ary image of a q^m-ary cyclic code," *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 387–399, Mar. 1995.

[4] J. Lacan and E. Delpeyroux, "The q -ary image of some q^m-ary cyclic codes: Permutation group and soft-decision decoding," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 2069–2078, Jul. 2002.

[5] V. Guruswami and M. Sudan, "Improved decoding of Reed–Solomon codes and algebraic geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999.

[6] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[7] J. Jiang and K. R. Narayanan, "Algebraic soft-decision decoding of Reed–Solomon codes using bit-level soft information," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3907–3928, Sep. 2008.

[8] J. Bellorado and A. Kavčić, "A low complexity method for Chase-type decoding of Reed–Solomon codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, Jul. 2006, pp. 2037–2041.

[9] G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Inf. Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966.

[10] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.

[11] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2000.

[12] E. R. Berlekamp, R. E. Peile, and S. P. Pope, "The application of error control coding to communications," *IEEE Commun. Mag.*, vol. 25, pp. 44–57, 1987.

[13] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, 2nd ed. Amsterdam, The Netherlands: North-Holland, 1983.

[14] A. Vardy and Y. Be'ery, "Bit level soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 440–444, 1991.

[15] F. Lim, M. P. Fossorier, and A. Kavčić, "Notes on the automorphism groups of Reed–Solomon binary images," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, ON, Canada, Jul. 2008, pp. 1813–1817.

[16] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995.

[17] D. J. Taipale and M. B. Pursley, "An improvement to generalized minimum distance decoding," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 167–172, Jan. 1991.

[18] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum liklihood decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.

[19] GAP—Groups, Algorithms, and Programming, Version 4.4.10 The GAP Group, 2007 [Online]. Available: http://www.gap-system.org

[20] J. S. Leon, "Computing automorphism groups of error-correcting codes," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 3, pp. 496–511, May 1982.

Fabian Lim (S'10) received the B.Eng and M.Eng degrees from the National University of Singapore in 2003 and 2006, respectively, and the Ph.D. degree from the University of Hawaii, Manoa, in 2010, all in electrical engineering. Currently, he is a postdoctoral fellow at the University of Hawaii.

Dr. Lim held short-term visiting research positions at Harvard University, Cambridge, MA, in 2004 and 2005. From October 2005 to May 2006, he was a staff member in the Data Storage Institute in Singapore. From May 2008 to July 2008, he was an intern at Hitachi Global Storage Technologies, San Jose, CA. In March 2009, he was a visitor at the Research Center for Information Security, Japan. His research interests include error-control coding and signal processing, mainly for data storage applications.

Marc Fossorier (F'06) received the B.E. degree from the National Institute of Applied Sciences (I.N.S.A.) Lyon, France, in 1987, and the M.S. and Ph.D. degrees in 1991 and 1994, all in electrical engineering.

His research interests include decoding techniques for linear codes, communication algorithms, and statistics.

Dr. Fossorier was a recipient of a 1998 NSF Career Development award and became IEEE Fellow in 2006. He has served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2003 to 2006, as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS from 1996 to 2003, as an Associate Editor for the IEEE COMMUNICATIONS LETTERS from 1999 to 2007, and as Treasurer of the IEEE Information Theory Society from 1999 to 2003. Since 2002, he has also been an elected member of the Board of Governors of the IEEE Information Theory Society for which he served as Second and First Vice-President. He was Program Co-Chairman for the 2007 International Symposium on Information Theory (ISIT), the 2000 International Symposium on Information Theory and Its Applications (ISITA), and Editor for the Proceedings of the 2006, 2003, and 1999 Symposium on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC).

Aleksandar Kavčić (S'93–M'98–SM'04) received the Dipl. Ing. degree in electrical engineering from Ruhr-University, Bochum, Germany, in 1993, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1998.

Since 2007, he has been with the University of Hawaii, Honolulu, where he is presently an Associate Professor of electrical engineering. Prior to 2007, he was in the Division of Engineering and Applied Sciences at Harvard University, Cambridge, MA, as Assistant Professor of electrical engineering from 1998 to 2002, and as the John L. Loeb Associate Professor of Natural Sciences from 2002 to 2006. While on leave from Harvard University, he served as a Visiting Associate Professor at the City University of Hong Kong in the Fall of 2005 and as Visiting Scholar at the Chinese University of Hong Kong in the Spring of 2006.

Prof. Kavčić received the IBM Partnership Award in 1999 and the NSF CAREER Award in 2000. He is a co-recipient, with X. Ma and N. Varnica, of the 2005 IEEE Best Paper Award in Signal Processing and Coding for Data Storage. He served on the Editorial Board of the IEEE TRANSACTIONS ON INFORMATION THEORY as an Associate Editor for Detection and Estimation from 2001 to 2004, as a Guest Editor of the *IEEE Signal Processing Magazine* in 2003–2004, and as a Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in 2008–2009. From 2005 until 2007, he was the Chair of the Data Storage Technical Committee of the IEEE Communications Society.