Available online at http://scik.org
J. Math. Comput. Sci. 10 (2020), No. 5, 1510-1528
https://doi.org/10.28919/jmcs/4664
ISSN: 1927-5307

s-PD SETS, RANK AND KERNEL OF HADAMARD CODES AND CONSTRUCTION OF HADAMARD CODES USING MAGMA

BRIJI J. CHATHELY*, RAJENDRA P. DEORE

Department of Mathematics, University of Mumbai, Santacruz East, Mumbai 400098, Maharashtra, India

Abstract. This article presents the recursive construction of s-PD sets, a special subset of the permutation automorphism group of a code, which enables the correction of s errors of binary linear Hadamard codes over the field F_4. We discuss the rank and kernel of these Hadamard codes. We develop new MAGMA functions to generate Hadamard codes over the field F_4. We also give MAGMA functions to find the minimum distance and generator matrix of a Hadamard code over F_4.

Keywords: Hadamard; fields; codes; weights; gray map; information sets; PD-sets.

2010 AMS Subject Classification: 94B05, 94B60, 11T71.

1. Introduction

A Hadamard matrix is an n- square matrix with entries from $\{1, -1\}$, for which the inner product of any pair of distinct rows is 0. Among the various types of matrices having distinct properties, Hadamard matrices with orthogonal properties is widely studied because of its practical use in the fields of Coding Theory, Signal Processing and Cryptography. The first example of Hadamard matrix was published by James Joseph Sylvester in *Thoughts on inverse orthogonal matrices, simultaneous sign successions and tesselated pavements in two or more*

*Corresponding author

E-mail address: brijijacob@gmail.com

Received April 27, 2020

colours, with applications to Newton's rule, ornamental tile work and the theory of numbers, Philosophical Magazine 34 (1867) (pp. 461–475). Jacques Hadamard further studied this matrix in *Resolution d'une question relative aux determinants*, Bulletin Sciences Mathematique 17 (1893) (pp. 240–246). Hadamard gave the famous conjecture on existence of Hadamard matrix, which still remains an open problem and has excited interest among researchers. The conjecture states $-$ *A Hadamard matrix of order n (for $n > 2$) exists if and only if* $n \equiv 0 \bmod 4$. Various methods of constructing Hadamard matrices were developed and various results equivalent to the existence of Hadamard matrix were proposed. Till date the smallest order unknown Hadamard matrix is for $n = 668$.

Hadamard code is an error-correcting code named after Jacques Hadamard, which is used for error detection and correction when transmitting messages over noisy or unreliable channels. Hadamard codes are usually based on Sylvester's construction of Hadamard matrices. However Hadamard codes using arbitary Hadamard matrix not necessarily of Sylvester type are also constructed. Such codes were first constructed by R. C. Bose and S. S. Shrikhande in 1959 in *A Note on a Result in the Theory of Code Construction*, Information and Control, Volume 2 (pp. 183–194).

Finding an efficient decoding algorithm is one of the fundamental problems in coding theory. Permutation decoding, a technique which uses a subset of the automorphism group of the code called s-PD set was developed by MacWilliams in [12] and Prange in [14]. The efficiency of the permutation decoding technique depends on the size of the s-PD set, thus making it an interesting problem of research. A new permutation decoding method for $\mathbb{Z}_4$-linear codes was introduced in [3]. This technique is strongly based on existence of special subsets of permutation automorphism group $\mathrm{PAut}(C)$ of a code C. The s-PD sets of minimum size $s+1$ for partial permutation decoding was developed for families of $\mathbb{Z}_4$-linear codes in [2].

MAGMA is a software package designed to solve computationally hard problems in algebra, number theory, geometry and combinatorics. MAGMA currently supports the basic computations in coding theory for linear codes over integer residue rings and Galois rings. Available functions in MAGMA can be referred in [9, 15]. Functions to construct families of codes over $\mathbb{Z}_4$ is discussed in [1]. The basic introduction to MAGMA can be referred from [7, 5]. MAGMA functions for codes can be referred from [4].

In this paper, given an s-PD set of length l with $l \geq s+1$ for H_α we construct an s-PD set of same length for $H_{\alpha+k}$. We discuss the rank and kernel of the Hadamard codes. We also develop MAGMA functions to generate Hadamard codes over the fields F_4. All the computational work is done through MAGMA calculator available at `http://magma.maths.usyd.edu.au/calc/`.

2. PRELIMINARIES

Definition 2.1. *[10] An n- square matrix H with entries* $+1$ *and* -1 *such that the set of its row vectors (or column vectors) forms an orthogonal set is called a* ***Hadamard Matrix****.*

For example,

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

Definition 2.2. *Any subset C of F_q^n is called a* ***linear code*** *of length n over the field F_q, if it is an additive subgroup of F_q^n.*

Definition 2.3. *An element of C is called the* ***codeword*** *in C. The number of codewords in C is called the size of C.*

Definition 2.4. *The* ***Hamming weight*** *of any codeword $x = (x_1, x_2, \cdots, x_n)$ in F_q^n denoted as $\omega_H(x)$ is the number of non zero coordinates in x.*

Definition 2.5. *The* ***Hamming distance*** *between any two codewords x and y in F_q^n is the number of places at which x and y differ.*

Definition 2.6. *The* ***Lee weight*** *of any codeword $x = (x_1, x_2, \cdots, x_n)$ in F_q^n is defined as*

$$\omega_L(x) = \sum_{i=1}^{n} \omega_L^*(x_i),$$

where ω_L^* is the Lee weight of each x_i.

Definition 2.7. *Let $C \subseteq F_q^n$ be any code. The **minimum Lee weight** (also known as minimum distance) of C is defined as*

$$d_L(C) = min\ \{\omega_L(x-y) : x, y \in C, x \neq y\}$$

Note: Any code of length n with k codewords and minimum distance d is denoted as (n,k,d)-code.

Definition 2.8. *A binary linear code with parameters $(n, 2n, \frac{n}{2})$ is called a **Hadamard code**.*

Definition 2.9. *The permutation automorphism group of a code C is the set of permutations that maps C to itself and is denoted by **PAut**(C).*

Note: If C is a code of length n, then PAut(C) is a subgroup of the symmetric group S_n.

2.1. Field F_4.

The field $F_4 = F_2[w]/ < w^2+w+1 > = \{0, 1, w, w+1\}$ is a field with the condition $w^2 = w+1$ and characteristic 2. Binary operations on the ring F_4 are defined as follows

$+$	0	1	w	$w+1$
0	0	1	w	$w+1$
1	1	0	$1+w$	w
w	w	$1+w$	0	1
$w+1$	$w+1$	w	1	0

$\cdot$	0	1	w	$w+1$
0	0	0	0	0
1	0	1	w	$w+1$
w	0	w	$w+1$	1
$w+1$	0	$w+1$	1	w

For any $x = (x_1, x_2, \cdots, x_n)$ in F_4^n, let $n_o(x)$ be the number of zeros in x and $n_2(x)$ be the number of ones in x. Let $n_1(x) = n - n_0(x) - n_2(x)$. Lee weight $\omega_L(x)$ is given in [8] by the formula $n_1(x) + 2n_2(x)$.

When $n = 1$, we get the Lee weights of elements of F_4 as follows:

$$\omega_L(0) = 0, \quad \omega_L(1) = 2, \quad \omega_L(w) = 1, \quad \omega_L(w+1) = 1.$$

3. Recursive Construction of s-PD Sets

Let $\mathscr{H}_\alpha$ be the quaternary linear Hadamard code constructed over the field F_4 of length $\delta = 2^{m-1}$ where $m = 2\alpha+1$ and $H_\alpha = \phi(\mathscr{H}_\alpha)$ be the corresponding F_4-linear code of length $2\delta = 2^m$ as discussed in [6]. A generator matrix N_α for the code $\mathscr{H}_\alpha$ can be constructed by the recursive construction

$$N_{\alpha+1} = \begin{bmatrix} N_\alpha & N_\alpha & N_\alpha & N_\alpha \\ 0 & 1 & w & w+1 \end{bmatrix} \tag{1}$$

where $N_0 = [1]$.

A set $\mathscr{I} = \{i_1, i_2, \dots, i_\alpha\} \subseteq \{1, 2, \dots, \delta\}$ is said to be a quaternary information set of a quaternary linear code C if $|C_\mathscr{I}| = \delta$. The set $\phi(\mathscr{I}) = \{2i_1 - 1, 2i_1, \dots, 2i_\alpha - 1, 2i_\alpha\}$ forms an information set for the code $\phi(C)$. The construction of s-PD sets for Hadamard codes over the field F_4 is discussed in [6].

In this section, given an s-PD set of length l with $l \geq s+1$ for H_α we construct an s-PD set of same length for $H_{\alpha+k}$, a Hadamard code of length 2^{m+2k}.

Proposition 3.1 (Proposition 1, [6]). *Let $\mathscr{I}$ be a quaternary information set for the quaternary linear Hadamard code $\mathscr{H}_\alpha$ of length $\delta = 2^{m-1}$. Then $\mathscr{I} \cup \{\delta+1\}$ is a quaternary information set for the code $\mathscr{H}_{\alpha+1}$ which are obtained from $\mathscr{H}_\alpha$ by applying* (1).

Let C be a quaternary linear code of length δ and $\phi(C)$ be the corresponding F_4-linear code of length 2δ. Let S_δ and $S_{2\delta}$ be the symmetric groups of order δ and 2δ respectively. Define $\phi : S_\delta \to S_{2\delta}$ as

$$\phi(\sigma)(i) = \begin{cases} 2\sigma(i/2), & \text{if } i \text{ is even }, \\ 2\sigma((i+1)/2) - 1, & \text{if } i \text{ is odd }, \end{cases}$$

for all $\sigma \in S_\delta$ and $i \in \{1, 2, \dots, 2\delta\}$. For any $S \subseteq S_\delta$, define $\phi(S) = \{\phi(\sigma) : \sigma \in S\} \subseteq S_{2\delta}$. Then if $S \subseteq \mathrm{PAut}(C) \subseteq S_\delta$, then $\phi(S) \subseteq \mathrm{PAut}(\phi(C)) \subseteq S_{2\delta}$. From [13], it is known that the permutation automorphism group PAut $(\mathscr{H}_\alpha)$ of $\mathscr{H}_\alpha$ is isomorphic to the general affine group $\mathrm{AGL}(\alpha, 2)$. Let $\mathrm{GL}(\alpha, 2)$ be the general linear group over F_2. Then $\mathrm{AGL}(\alpha, 2)$ consists of all mappings $\eta : F_2^\alpha \to F_2^\alpha$ such that $\eta(x) = Ax + b$ where $A \in$ GL $(\alpha, 2)$ and $b \in F_2^\alpha$.

The map $\varphi :$ AGL $(\alpha,2) \rightarrow$ GL $(\alpha+1,2)$ defined as

$$\varphi(b,A) = \begin{bmatrix} 1 & b \\ \mathbf{0} & A \end{bmatrix}$$

gives an isomorphism between AGL $(\alpha,2)$ and the subgroup of GL $(\alpha+1,2)$ consisting of all non-singular matrices with first column as e_1. Thus we can consider PAut $(\mathscr{H}_\alpha)$ as this subgroup of GL $(\alpha+1,2)$. Any matrix $P \in$ PAut $(\mathscr{H}_\alpha)$ can be seen as a permutation of coordinate positions $\sigma \in S_\delta$ such that $\sigma(i) = j$ whenever $y_j = y_i P$ where y_k is the k^{th} column vector of the generator matrix N_α and $i,j \in \{1,2,\ldots,\alpha\}$. Define $\phi(P) = \phi(\sigma) \in S_{2\delta}$ and $\phi(\mathscr{P}) = \{\phi(P) : P \in \mathscr{P}\} \subseteq S_{2\delta}$ for any $\mathscr{P} \subseteq$ PAut $(\mathscr{H}_\alpha)$. Define P^* as the matrix where the first row is x_1 and i^{th} row is $x_1 + x_i$ where x_i is the i^{th} row of P for $i \in \{2,3,\ldots,\alpha\}$.

Let

$$M = \begin{bmatrix} 1 & b \\ \mathbf{0} & A \end{bmatrix} \in \text{PAut}(\mathscr{H}_\alpha)$$

where $A \in$ GL $(\alpha,2)$ and $b \in F_2^\alpha$. For an integer $k \geq 1$, we define

$$M[k] = \begin{bmatrix} 1 & b & \mathbf{0} \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_k \end{bmatrix}$$

Proposition 3.2. *Let $\mathscr{P}_s = \{M_0, M_1, \cdots, M_s\} \subseteq PAut(\mathscr{H}_\alpha)$ such that $\phi(\mathscr{P}_s)$ is an s-PD set of size $s+1$ for H_α with information set $\phi(\mathscr{I}_\alpha)$. Then $\mathscr{W}_s = \{(M_0^{-1}[k])^{-1}, (M_1^{-1}[k])^{-1}, \cdots, (M_s^{-1}[k])^{-1}\} \subseteq PAut(\mathscr{H}_{\alpha+k})$ and $\phi(\mathscr{W}_s)$ is an s-PD set of size $s+1$ for $H_{\alpha+k}$ with information set $\phi(\mathscr{I}_{\alpha+k})$ for any $k \geq 1$.*

Proof. Clearly from the definition of $M[k]$, $M \in \text{PAut}(\mathscr{H}_\alpha)$ implies $M[k] \in \text{GL}(\alpha+k+1,2)$. Hence $M^{-1}[k] \in \text{PAut}(\mathscr{H}_{\alpha+k})$ and its inverse that is $(M^{-1}[k])^{-1}$ also is in $\text{PAut}(\mathscr{H}_{\alpha+k})$. Thus $\mathscr{W}_s \subseteq \text{PAut}(\mathscr{H}_{\alpha+k})$. From [Theorem 4, [6]], $\phi(\mathscr{P}_s)$ is an s-PD for H_α implies no two matrices $(M_i^{-1})^*$ and $(M_j^{-1})^*$ for $i \neq j$ have rows in common. Hence no two matrices $(M_i^{-1}[k])^*$ and $(M_j^{-1}[k])^*$ for $i \neq j$ have rows in common and [Theorem 4, [6]] implies $\phi(\mathscr{W}_s)$ is an s-PD set of size $s+1$ for $H_{\alpha+k}$ with information set $\phi(\mathscr{I}_{\alpha+k})$ for any $k \geq 1$. □

Example 3.1. *Let $\mathscr{P}_4 = \{\mathscr{Q}_0^{-1}, \mathscr{Q}_1^{-1}, \mathscr{Q}_2^{-1}, \mathscr{Q}_3^{-1}, \mathscr{Q}_4^{-1}\} \subseteq PAut(\mathscr{H}_2)$ be the set given in [Example 2, [6]] such that $\phi(\mathscr{P}_4)$ is a 4-PD set of size 5 for H_2. Then from Proposition3.2, for $k = 1$, $\mathscr{W}_4 = \{(\mathscr{Q}_i[1])^{-1} : 0 \leq i \leq 4\} \subseteq PAut(\mathscr{H}_3)$ and $\phi(\mathscr{W}_4)$ is a 4-PD set of size 5 for H_3. Here $\mathscr{Q}_0[1] = Id_4$,*

$$\mathscr{Q}_1[1] = \begin{bmatrix} 1 & w & 1 & 0 \\ 0 & 0 & w & 0 \\ 0 & w+1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathscr{Q}_2[1] = \begin{bmatrix} 1 & w & 0 & 0 \\ 0 & w & w & 0 \\ 0 & 1 & w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathscr{Q}_3[1] = \begin{bmatrix} 1 & w+1 & 1 & 0 \\ 0 & 1 & w+1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathscr{Q}_4[1] = \begin{bmatrix} 1 & 0 & w+1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

and $(\mathscr{Q}_0[1])^* = Id_4^*$,

$$(\mathscr{Q}_1[1])^* = \begin{bmatrix} 1 & w & 1 & 0 \\ 1 & w & w+1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & w & 1 & 1 \end{bmatrix}, (\mathscr{Q}_2[1])^* = \begin{bmatrix} 1 & w & 0 & 0 \\ 1 & 0 & w & 0 \\ 1 & w+1 & w & 0 \\ 1 & w & 0 & 1 \end{bmatrix},$$

$$(\mathscr{Q}_3[1])^* = \begin{bmatrix} 1 & w+1 & 1 & 0 \\ 1 & w & w & 0 \\ 1 & w+1 & 0 & 0 \\ 1 & w+1 & 1 & 1 \end{bmatrix}, (\mathscr{Q}_4[1])^* = \begin{bmatrix} 1 & 0 & w+1 & 0 \\ 1 & 1 & w+1 & 0 \\ 1 & 1 & w & 0 \\ 1 & 0 & w+1 & 1 \end{bmatrix}.$$

Clearly $(\mathscr{Q}_i[1])^*$ and $(\mathscr{Q}_j[1])^*$ has no rows in common for $i \neq j$.

Now, we show a second recursive construction for which we consider the elements of PAut(H_α) as permutation of coordinate positions, that is as elements of S_{2^m}.

Given permutations $\sigma_1 \in S_{n_1}$, $\sigma_2 \in S_{n_1+n_2}$, $\sigma_3 \in S_{n_1+n_2+n_3}$ and $\sigma_4 \in S_{n_1+n_2+n_3+n_4}$ we define $(\sigma_1|\sigma_2|\sigma_3|\sigma_4) \in S_{n_1+n_2+n_3+n_4}$ such that σ_1 acts on the first n_1 coordinates, σ_2 acts on the coordinates $\{n_1+1, \cdots, n_1+n_2\}$, σ_3 acts on the coordinates $\{n_1+n_2+1, \cdots, n_1+n_2+n_3\}$ and σ_4 acts on the coordinates $\{n_1+n_2+n_3+1, \cdots, n_1+n_2+n_3+n_4\}$.

Proposition 3.3. *Let $\mathscr{S} \subseteq PAut(\mathscr{H}_\alpha)$ such that $\phi(\mathscr{S})$ is an s-PD set of size l for H_α of length $n = 2^m = 2(4^\alpha)$ where $m = 2\alpha + 1$ with information set I. Then $\phi((\mathscr{S}|\mathscr{S}|\mathscr{S}|\mathscr{S})) = \{\phi((\sigma|\sigma|\sigma|\sigma)) : \sigma \in \mathscr{S}\}$ is an s-PD set of size l for $H_{\alpha+1} = \phi(\mathscr{H}_{\alpha+1})$ of length $4n = 2^{m+2} = 2(4^{\alpha+1})$, where $\mathscr{H}_{\alpha+1}$ is constructed from (1) and $I' = I \cup \{i+n, j+n : i, j \in I\}$ is an information set $H_{\alpha+1}$.*

Proof. Since $\mathscr{H}_{\alpha+1}$ is constructed from (1), $\mathscr{H}_{\alpha+1} = \{(x,x,x,x),(x,x+1,x+w,x+w+1),(x,x+w,x+w+1,x+1),(x,x+w+1,x+1,x+w) : x \in \mathscr{H}_\alpha\}$. If $\sigma \in \mathrm{PAut}(\mathscr{H}_\alpha)$, then $\sigma \in S_{2^{m-1}}$ which implies $(\sigma|\sigma|\sigma|\sigma) \in S_{2^{m+1}}$. That is $(\sigma|\sigma|\sigma|\sigma) \in \mathrm{PAut}(\mathscr{H}_{\alpha+1})$.

Let $\tau = \phi(\sigma)$. We show that for every $e \in F_2^{4n}$ with $\omega_H(e) \leq s$, there exists $(\tau|\tau|\tau|\tau) \in \phi((\mathscr{S}|\mathscr{S}|\mathscr{S}|\mathscr{S}))$ such that $(\tau|\tau|\tau|\tau)(e)_{I'} = 0$ where $I' \subseteq \{1,2,\cdots,4n\}$ is an information set with $2(\alpha+1)$ coordinate points for $H_{\alpha+1}$. Let $e = (a,b,c,d) \in F_2^{4n}$ where $a = (a_1,a_2,\cdots,a_n)$, $b = (b_1,b_2,\cdots,b_n)$, $c = (c_1,c_2,\cdots,c_n)$ and $d = (d_1,d_2,\cdots,d_n)$. We take a binary vector $p = (p_1,p_2,\cdots,p_n)$ such that $p_i = 1$ if and only if $a_i = 1$ or $b_i = 1$ or $c_i = 1$ or $d_i = 1$ for all $i \in \{1,2,\cdots,n\}$. Since $\omega_H(e) \leq s$, we have $\omega_H(c) \leq s$. As $\phi(\mathscr{S})$ is an s-PD set for H_α with information set I, there exists $\tau \in \phi(\mathscr{S})$ such that $\tau(p)_I = \mathbf{0}$. Therefore we have $(\tau|\tau|\tau|\tau) \in \phi((\mathscr{S}|\mathscr{S}|\mathscr{S}|\mathscr{S}))$ such that $(\tau|\tau|\tau|\tau)(a,b,c,d)_{I\cup J} = \mathbf{0}$, where $J = \{i+n, j+n : i, j \in I\}$. Hence $\phi((\mathscr{S}|\mathscr{S}|\mathscr{S}|\mathscr{S}))$ is an s-PD set for $H_{\alpha+1}$. From Proposition3.1 it follows that $I' = I \cup \{i+n, j+n : i, j \in I\}$ is an information set $H_{\alpha+1}$. □

Corollary 3.1. *Let $\mathscr{S} \subseteq PAut(\mathscr{H}_\alpha)$ such that $\phi(\mathscr{S})$ is an s-PD set of size l for H_α of length 2^m with information set I. Then $\phi(4^k\mathscr{S})$ is an s-PD set of size l for $H_{\alpha+1}$ of length 2^{m+2k} with information set obtained by recursively applying Proposition3.1, for all $i, j \geq 0$.*

Proof. The proof comes trivially by applying Proposition3.1 and Proposition3.3. □

4. Rank and Kernel of Hadamard Codes over F_4

Definition 4.1. *Two codes are said to be equivalent if one can be obtained from the other by permuting the coordinates.*

Definition 4.2. *[11] Let E_n be the set of all binary words of length n. Two* binary codes *C and C' of length n are said to be equivalent if there exists a word x in E_n and a permutation π in S_n such that $C = \pi(C'+x)$.*

Definition 4.3. *[11] Let E_n be the set of all binary words of length n and H_n be a binary code of length n.* ***Kernel*** *of H_n is defined as*

$$ker(H_n) := \{x \in E_n : x + H_n = H_n\}.$$

Definition 4.4. *[11] Let C be a binary code.* ***Rank*** *of C is defined as the number of linearly independent vectors of C.*

Proposition 4.1. *If two binary codes H and H' of length n are equivalent then $|ker(H)| = |ker(H')|$.*

Proof. Let $x \in E_n$ be such that $H = \pi(H'+x)$. Let $k \in \ker(H)$, then $k+H = H$. That is for h_1 in H we have h_2 in H such $k+h_1 = h_2$. Also H and H' are equivalent implies there exists $h_1^{'}$ and $h_2^{'}$ in H' such that $h_1 = \pi(h_1^{'}+x)$ and $h_2 = \pi(h_2^{'}+x)$. Thus

$$k + \pi(h_1^{'}+x) = \pi(h_2^{'}+x)$$

Premultiplying with π^{-1}, we get

$$\pi^{-1}(k) + h_1^{'} = h_2^{'}$$

Thus for every k in $\ker(H)$, we have $k' = \pi^{-1}(k)$ in $\ker(H')$. Similarly we can show that for every p' in $\ker(H')$, we have some p in $\ker(H)$. Hence $|\ker(H)| = |\ker(H')|$. □

Theorem 4.1. *Let $H_\alpha = \phi(\mathscr{H}_\alpha)$ be a Hadamard code over F_4 of length $2^{2\alpha+1}$. Then $|ker(H_\alpha)| = 2^{2\alpha+2}$.*

Proof. Since H_α is a binary linear code, for any two codewords x and y in H_α, $x+y$ is in H_α. Thus every codeword in H_α lies in $\ker(H_\alpha)$. Now, let z be any word in $\ker(H_\alpha)$, then $z+H_\alpha = H_\alpha$ and linearity of H_α implies $z \in H_\alpha$. Thus $\ker(H_\alpha) = H_\alpha$ and $|\ker(H_\alpha)| = |H_\alpha| = 2^{2\alpha+2}$. □

Proposition 4.2. *Let C and C' be two equivalent binary codes of length n. Then $rank(C)$=$rank(C')$.*

Proof. As C and C' are equivalent, we have $C = \pi(C' + x)$ for some x in E_n. Hence rank(C)=rank$(\pi(C'+x)) = \text{rank}(C'+x) = \text{rank}(C')$. □

Theorem 4.2. *Let $H_\alpha = \phi(\mathscr{H}_\alpha)$ be a Hadamard code over F_4 of length $2^{2\alpha+1}$. Then $rank(H_\alpha) = 2(rank(\mathscr{H}_\alpha))$.*

From the generator matrix of $\mathscr{H}_\alpha$, it can be observed that rank$(\mathscr{H}_\alpha)$ is the number of rows of the matrix that is $\alpha+1$. Hence rank(H_α) is $2(\alpha+1)$.

5. Computational Results using MAGMA

In this section we present the construction of new MAGMA functions to generate Hadamard codes over the field F_4. Functions to find the generator matrix, rank and minimum distance of a Hadamard code is also discussed in this section.

5.1. Construction of N_α for Hadamard Code. We give the construction of the generator matrix N_α as discussed in (1).

```
> G<w> := GF(2, 2);//Finite Field of order 4
> AssertAttribute(G, "PowerPrinting", false);
> function A(alpha)// Vector with all entries zero
> if(alpha le 0) then return "Not Defined"; end if;
> if(alpha eq 1) then return Matrix(G,1,[0]); end if;
> return ZeroMatrix(G, 1,4^(alpha-1));
> end function;
> A(0);
> A(1);
> A(3);
OUTPUT:
Not Defined
[0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function B(alpha)//Vector with all entries one
> if(alpha le 0) then return "Not Defined"; end if;
> if(alpha eq 1) then return Matrix(G, 1,[1]); end if;
> return Matrix(G, 1,4^(alpha-1),[i^0: i in [1..4^(alpha-1)]]);
> end function;
> B(0);
> B(1);
> B(3);
OUTPUT:
Not Defined
[1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function C(alpha)//Vector with all entries w
> if(alpha le 0) then return "Not Defined"; end if;
> if(alpha eq 1) then return Matrix(G, 1,[G.1]); end if;
> return Matrix(G, 1,4^(alpha-1),[i^0*G.1: i in
   [1..4^(alpha-1)]]);
> end function;
> C(0);
> C(1);
> C(3);
OUTPUT:
Not Defined
```

```
[w]
[w w w w w w w w w w w w w w w w]
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function D(alpha)//Vector with all entries w+1
> if(alpha le 0) then return "Not Defined"; end if;
> if(alpha eq 1) then return Matrix(G, 1,[G.1^2]); end if;
> return Matrix(G, 1,4^(alpha-1),[i^0*G.1^2: i in
   [1..4^(alpha-1)]]);
> end function;
> D(0);
> D(1);
> D(2);
OUTPUT:
Not Defined
[w+1]
[w + 1 w + 1 w + 1 w + 1]
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function N(alpha)
> if (alpha eq 0) then return Matrix(G,1,[1]); end if;
> if (alpha eq 1) then
  return Matrix(G, 2, 4, [1,1,1,1, 0,1,G.1,G.1^2]); end if;
> Row1 := HorizontalJoin(N(alpha-1), N(alpha-1));
> Row2 := HorizontalJoin(N(alpha-1), N(alpha-1));
> Row := HorizontalJoin(Row1, Row2);
> Col1 := HorizontalJoin(A(alpha), B(alpha));
```

```
> Col2 := HorizontalJoin(C(alpha), D(alpha));
> Col := HorizontalJoin(Col1, Col2);
> return VerticalJoin(Row, Col);
> end function;
> N(0);
> N(1);
> N(2);
OUTPUT:
[1]

[    1     1     1     1]
[    0     1     w w + 1]

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 1 w w + 1 0 1 w w + 1 0 1 w w + 1 0 1 w w + 1]
[0 0 0 0 1 1 1 1 w w w w w + 1 w + 1 w + 1 w + 1]
```

5.2. Construction of Hadamard codes over F_4. We give the function to construct Hadamard codes over F_4 and its generator matrix. We also discuss the functions to find the minimum distance of the code.

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function HCF4(alpha,m)//List of Hadamard codes
> if (m ne 2*alpha+1) then
  return "m-1 should be a multiple of 2"; end if;
> return [c*N(alpha) : c in VectorSpace(G,alpha+1)];
> end function;
> HCF4(1,3);
OUTPUT:
```

```
[
(0 0 0 0),
(1 1 1 1),
(w w w w),
(w + 1 w + 1 w + 1 w + 1),
(    0     1     w w + 1),
(    1     0 w + 1     w),
(    w w + 1     0     1),
(w + 1     w     1     0),
(    0     w w + 1     1),
(    1 w + 1     w     0),
(    w     0     1 w + 1),
(w + 1     1     0     w),
(    0 w + 1     1     w),
(    1     w     0 w + 1),
(    w     1 w + 1     0),
(w + 1     0     w     1)
]
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function W(alpha, m)// Gives basis of HCF4(alpha,m)
> return sub<VectorSpace(G,2^(m-1)) | HCF4(alpha,m)>;
> end function;
> function LengthHCF4(alpha,m)//Length of the code
> return 4^(alpha);
> end function;
> function GeneratorMatrixHCF4(alpha,m)//Gives the generator
   matrix of a code
```

```
> if (m ne 2*alpha+1) then
  return "m-1 should be a multiple of 2";
  end if;
> return N(alpha);
> end function;
> Dimension(W(4,9));//Gives rank of HCF4(4,9)
> LengthHCF4(1,3);
> GeneratorMatrixHCF4(1,3);
OUTPUT:
5
4
[   1    1    1    1]
[   0    1    w  w + 1]
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function N0(x,m)//Number of zeros in a code
> while x in VectorSpace(G,2^(m-1)) do;
> return 2^(m-1)-Weight(x);
> end while;
> end function;
> function v(m)
> return Vector(2^(m-1),[i^0: i in [1..2^(m-1)]]);
> end function;
> function N2(x,m)//Number of ones in a code
> while x in VectorSpace(G,2^(m-1)) do;
> return 2^(m-1)-Weight(x+v(m));
> end while;
> end function;
```

```
> function N1(x,m)
> while x in VectorSpace(G,2^(m-1)) do;
> return 2^(m-1)-N0(x,m)-N2(x,m);
> end while;
> end function;
> x:=VectorSpace(G,4)![0,1,w,w+1];
> N0(x,3);
> v(3);
> N2(x,3);
> N1(x,3);
OUTPUT:
1
(1 1 1 1)
1
2
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function LeeWeight(x,m, alpha)//Calculates Lee weight of a
   codeword
> while x in HCF4(alpha,m) do;
> return N1(x,m)+2*N2(x,m);
> end while;
> end function;
> function LeeDistance(x,y,m, alpha)//Calculates Lee distance
   between two codewords
> return LeeWeight(x-y,m,alpha);
> end function;
```

```
> function MinimumDistance(alpha, m)//Gives minimum weight of
   a code
> return Minimum({Integers()|LeeDistance(x,y,m, alpha):
  x in HCF4(alpha,m), y in HCF4(alpha,m)}diff {0}) ;
> end function;
> y:=VectorSpace(G,4)![0,1,w,w+1];
> x:=VectorSpace(G,4)![1,1,1,1];
> LeeWeight(x,3, 1);
> LeeDistance(x,y,3, 1);
> MinimumDistance(1,3);
OUTPUT:
8
4
4
```

```
> G<w> := GF(2, 2);
> AssertAttribute(G, "PowerPrinting", false);
> function HadamardCodeF4(alpha, m)
> if (m ne 2*alpha+1) then return "m-1 should be a multiple of
   2";
  end if;
> print "For m = 2*alpha+1, the function HadamardCodeF4(alpha,
   m) gives the generator matrix for Hadamard Code with
   parameters (Length of a Code, Cardinality, Minimum
   Distance) = (2^(m-1),2^(m+1),2^(m-1)) over the field F4. ";
> return Vector(3,[LengthHCF4(alpha,m), #HCF4(alpha, m),
   MinimumDistance(alpha, m)]), GeneratorMatrixHCF4(alpha,m);
> end function;
> HadamardCodeF4(2,5);
```

```
OUTPUT:
For m = 2*alpha+1, the function HadamardCodeF4(alpha, m) gives
   the generator matrix for Hadamard Code with parameters
   (Length of a Code,Cardinality,Minimum Distance) =
   (2^(m-1),2^(m+1),2^(m-1)) over the field F4.

(16 64 16)

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 1 w w + 1 0 1 w w + 1 0 1 w w + 1 0 1 w w + 1]
[0 0 0 0 1 1 1 1 w w w w w + 1 w + 1 w + 1 w + 1]
```

Conflict of Interests

The author(s) declare that there is no conflict of interests.

References

[1] R. Barrolleta, J. Pernas, J. Pujol, and M. Villanueva, Codes over $\mathbb{Z}_4$ and Permutation Decoding of Linear Codes: A MAGMA Package, (2017), pp. 1–52. https://ddd.uab.cat/pub/caplli/2017/188175/lincodove_a2017.pdf

[2] R.D. Barrolleta, M. Villanueva, Partial permutation decoding for binary linear and Z_4-linear Hadamard codes, Des. Codes Cryptogr. 86 (2018), 569–586.

[3] J.J. Bernal, J. Borges, C. Fernández-Córdoba, M. Villanueva, Permutation decoding of $\mathbb{Z}_2\mathbb{Z}_4$-linear codes, Des. Codes Cryptogr. 76 (2015), 269–277.

[4] J. Canon, W. Bosma, C. Fieker, A. Steel, Handbook of Magma Functions, Sydney, 2008, version 2.14.

[5] J. Canon, W. Bosma, C. Fieker, A. Steel, Handbook of Magma Functions, Sydney, 2013, version 2.19.

[6] B. J. Chathely, R. P. Deore, Construction of Binary Hadamard Codes and their s-PD sets, Communicated.

[7] D. A. Craven, Computing with Magma, (2008), pp. 1–116. `http://web.mat.bham.ac.uk/D.A.Craven/magma.html`.

[8] P. Gaborit, V. Pless, P. Solé, O. Atkin, Type II Codes over F4, Finite Fields Appl. 8 (2002), 171–183.

[9] A.R. Hammons, P.V. Kumar, A.R. Calderbank, N.J.A. Sloane, P. Solé, The Z_4-linearity of Kerdock, Preparata, Goethals, and related codes, IEEE Trans. Inform. Theory. 40 (1994) 301–319.

[10] K.J. Horadam, Hadamard matrices and their applications, Princeton University Press, Princeton, N.J, 2007.

[11] D.S. Krotov, Z4-linear Hadamard and extended perfect codes, Electronic Notes in Discrete Mathematics. 6 (2001), 107–112.

[12] J. Macwilliams, Permutation Decoding of Systematic Codes, Bell Syst. Techn. J. 43 (1964), 485–505.

[13] F.J. MacWilliams, N.J.A. Sloane, The theory of error correcting codes, North-Holland Pub. Co.; sole distributors for the U.S.A. and Canada, Elsevier/North-Holland, Amsterdam; New York: New York, 1977.

[14] E. Prange, The use of information sets in decoding cyclic codes, IEEE Trans. Inform. Theory. 8 (1962), 5–9.

[15] Z. Wan, Quaternary codes, World Scientific Pub. Co, Singapore, 1997.