Classification of Extremal and s-Extremal Binary Self-Dual Codes of Length 38

Carlos Aguilar Melchor, Philippe Gaborit, Jon-Lark Kim, Lin Sok, Patrick Solé

Classification of Extremal and s-Extremal Binary Self-Dual Codes of Length 38

Carlos Aguilar-Melchor, Philippe Gaborit, Jon-Lark Kim, *Associate Member, IEEE*, Lin Sok, and Patrick Solé

***Abstract*—In this paper we classify all extremal and s-extremal binary self-dual codes of length 38. There are exactly 2744 extremal $[38, 19, 8]$ self-dual codes, two s-extremal $[38, 19, 6]$ codes, and 1730 s-extremal $[38, 19, 8]$ codes. We obtain our results from the use of a recursive algorithm used in the recent classification of all extremal self-dual codes of length 36, and from a generalization of this recursive algorithm for the shadow. The classification of s-extremal $[38, 19, 6]$ codes permits to achieve the classification of all s-extremal codes with $d = 6$.**

***Index Terms*—Classification, extremal, recursive construction, self-dual codes, s-extremal, shadow.**

I. Introduction

Self-dual codes are one of the most interesting classes of linear codes. They have close connections with group theory, lattice theory, design theory, and modular forms. It is well known that self-dual codes are asymptotically good [22]. There has been an active research on the classification of self-dual codes over finite fields and over rings in general (see [25], [23] for details). In particular, the classification of binary self-dual codes was started by Pless [24] and has been actively studied by many authors (see [19] for a survey of optimal self-dual codes over small alphabets).

Recently, using a recursive method, Aguilar and Gaborit classified all 41 extremal $[36, 18, 8]$ binary self-dual codes. These results were pushed further by Harada and Munemasa [17] who, besides the 41 extremal codes of [1], also give a complete classification of all self-dual codes of length 36.

A natural question is hence to consider the case of length 38. A simple computation on the mass formula shows that there are at least 13,644,433 inequivalent binary self-dual $[38, 19]$ codes [17]. It is hence natural to consider the case of special subclasses of self-dual codes. The most interesting such subclass is the class of extremal codes. Given the classification of all $[36, 18, 6]$ self-dual codes of [17], we apply an optimized recursive algorithm as in [1] to derive the classification of all 2744 extremal self-dual $[38, 19, 8]$ codes.

Another subclass of interesting self-dual codes with combinatorial properties is the class of s-extremal codes: these codes are self-dual codes whose weight enumerator is uniquely determined, depending on the condition on a high weight of the shadow. The notion of codes (and lattices) with long shadows was first developed by Elkies [11]. This notion was generalized by Bachoc and Gaborit in [2] who introduced the notion of s-extremal codes. These codes exist depending on conditions on their length and their minimum distance. The classification of s-extremal codes with $d = 4$ was done by Elkies. The case of $d = 6$ was mainly considered in [2], but two lengths remained to be classified. One is length 36, which was classified in [1], and the other is length 38, which is what we classify in this paper. Our classification is based on a generalization of the subtraction algorithm in the case of the shadow. It permits us to use the recursive algorithm by showing that in certain cases for n even, the subtraction of (11) from a $[2n+2, n+1, d+2]$ self-dual code with shadow weight $s+1$ leads to a $[2n, n, d]$ self-dual code with shadow weight s. This result is interesting in itself.

The paper is organized as follows: Section II gives preliminaries and background for self-dual codes, Section III compares the different method to extend a self-dual code in a purpose of classification. In Section IV we show that there are exactly 2744 extremal $[38, 19, 8]$ binary self-dual codes. In Section V we prove that there are only two s-extremal $[38, 19, 6]$ codes and 1730 s-extremal $[38, 19, 8]$ codes. The last section describes the covering radii of self-dual codes of length 38.

II. Preliminaries

We refer to [20] for basic definitions and results related to self-dual codes. All codes in this paper are binary. A *linear* $[n, k]$ code C of length n is a k-dimensional subspace of $GF(2)^n$. An element of C is called a *codeword*. The (Hamming) weight $\mathrm{wt}(\mathbf{x})$ of a vector $\mathbf{x} = (x_1, \dots, x_n)$ is the number of non-zero coordinates in it. The *minimum distance* (or *minimum weight*) $d(C)$ of C is $d(C) := \min\{\mathrm{wt}(\mathbf{x}) \mid \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}$. The Euclidean inner product of $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ in $GF(2)^n$ is $\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i$. The *dual* of C, denoted by $C^{\perp}$ is the set of vectors orthogonal to every codeword of C under the Euclidean inner product. If $C = C^{\perp}$, C is called *self-dual*. A self-dual code is called Type II (or doubly-even) if every codeword has weight divisible by 4, and Type I (or singly-even) if there exists a codeword whose weight is congruent to 2 $\pmod 4$.

Manuscript received April 04, 2011; accepted September 20, 2011. Date of publication November 30, 2011; date of current version March 13, 2012.

C. Aguilar-Melchor and P. Gaborit are with the XLIM-DMI, UMR 6172, Université de Limoges, 87000 Limoges, France (e-mail: carlos.aguilar@xlim.fr; gaborit@unilim.fr).

J.-L. Kim is with the Department of Mathematics, University of Louisville, Louisville, KY 40292 USA (e-mail: jl.kim@louisville.edu).

L. Sok is with the Department of Comelec, Telecom ParisTech, 75013 Paris, France (e-mail: lin.sok@telecom-paristech.fr).

P. Solé is with the CNRS/LTCI, UMR 5141, Telecom ParisTech, 75 634 Paris cedex 13, France and also with the MECAA, Math Department of King Abdulaziz University, Jeddah, Saudi Arabia (e-mail: patricksole@telecom-paristech.fr).

Communicated by A. Ashikhmin, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2011.2177809

Two codes over $GF(2)$ are said to be *equivalent* if they differ only by a permutation of the coordinates. Let C be a binary self-dual code of length n and minimum distance $d(C)$. Then $d(C)$ satisfies the following (see [25]):

$$d(C) \leq \begin{cases} 4\left[\frac{n}{24}\right]+4, & \text{if } n \neq 22 \pmod{24}, \\ 4\left[\frac{n}{24}\right]+6, & \text{if } n = 22 \pmod{24}. \end{cases}$$

A self-dual code meeting one of the above bounds is called *extremal*. A code is called *optimal* if it has the highest possible minimum distance for its length and dimension.

By the well known Gleason's theorem, the weight enumerator $W_C(x,y)$ of a Type I code can be written as follows (for rational coefficients c_i):

$$W_C(x,y) = \sum_{i=0}^{[n/8]} c_i (x^2+y^2)^{\frac{n}{2}-4i}\{x^2y^2(x^2-y^2)^2\}^i.$$

An important notion associated to a Type I code is the *shadow* S of a code C, defined by $S = C_0^{\perp}\backslash C$, where C_0 is the doubly-even subcode of C. In [9], Conway and Sloane show that for a weight enumerator $W_C(x,y)$ given above, the weight enumerator W_S of S satisfies

$$W_S(x,y) = \sum_{i=0}^{[n/8]} c_i (-1)^i 2^{\frac{n}{2}-6i}(xy)^{\frac{n}{2}-4i}(x^4-y^4)^{2i}.$$

This notion of shadow permits to give more information on potential weight enumerators of self-dual codes, and is also used to define s-extremal codes (see [2] or Section V).

The main tool to classify self-dual codes is based on the so-called mass formula. It is known from [24] that self-dual binary codes (Type I or Type II) of length n satisfy a formula (a mass formula)

$$N(n) = \sum_j \frac{n!}{|Aut(C_j)|}$$

where the sum is made over all inequivalent self-dual codes (Type I or Type II) of length n, $|Aut(C)|$ denotes the order of the automorphism group of a code C, and $N(n)$ is the number of Type I or Type II codes. In particular, for Type I codes, $N(n) = \prod_{i=1}^{\frac{n}{2}-1}(2^i+1)$ and for Type II codes $N(n) = \prod_{i=0}^{\frac{n}{2}-2}(2^i+1)$.
Therefore, for $n = 38$,

$$N(38) = \prod_{i=1}^{18}(2^i+1) = \sum_j \frac{38!}{|Aut(C_j)|}.$$

Hence

$$\begin{aligned} 13644432.20346 &< \frac{\prod_{i=1}^{18}(2^i+1)}{38!} \\ &= \sum\nolimits_j \frac{1}{|Aut(C_j)|} \\ &\leq \#(\text{all inequivalent self-dual codes}). \end{aligned}$$

Moreover, as there is no mass formula for extremal self-dual codes, it might be also difficult to classify all extremal binary $[38,19,8]$ codes. However, using the recursive construction [1] which was used in classifying all extremal binary $[36,18,8]$ codes, we are successful in classifying all extremal binary $[38,19,8]$ codes.

A very interesting tool for self-dual codes is the subtraction procedure of (11) on two coordinates of a code. This procedure permits to construct a $[2n,n,d' \geq d]$ self-dual code from a $[2n+2,n+1,d+2]$ self-dual code. It works as follows: suppose one starts from a $[2n+2,n+1,d+2]$ self-dual code C for $d \geq 2$. Let i and j be two different coordinates of the columns of C. Since $d+2 \geq 3$ and C is self-dual, any two columns of C are independent (if not, there should be a codeword of weight 2 in C, a contradiction). This implies that the coordinates of the two columns of the codewords of C contain (00),(10),(01) and (00). For the subtraction procedure of (11) on columns i and j, one first keeps all codewords which are either (00) or (11) on columns i and j, and then deletes columns i and j for these codewords. Let C' be the obtained code. Since $d+2 > 2$ and by an argument similar to the shortening of a code, the dimension of C' is n. Moreover since the scalar product of any two codewords of C is 0, the scalar product of any two codewords of C' is also 0. Now as the minimum distance of C is $d+2$, the minimum distance d' of C' is either d either $d+2$ (depending on the fact that columns i and j intersect or not with codewords of C of weight $d+2$). Overall C' is a $[2n,n,d' \geq d]$ self-dual code.

III. Construction Methods

There exist several methods to construct self-dual codes of length $n+2$ from self-dual codes of length n. In this section we recall these methods; the recursive construction, the building-up construction and the Harada-Munemasa construction. We eventually compare them.

A. The Recursive Construction

In [1], Aguilar and Gaborit give a recursive construction of binary self-dual codes. This algorithm can be seen as the reverse operation of the subtraction procedure of (11) given above. We recall that a subtraction procedure produces a self-dual $[2n,n,d' \geq d]$ code C' from a self-dual $[2n+2,n+1,d+2]$ code C. The recursive algorithm starts from a self-dual $[2n,n,d]$ code C' and constructs (up to permutation) all self-dual $[2n+2,n+1,d+2]$ codes which by subtraction of (11) on certain two columns give the code C'. The idea of the recursive algorithm is very simple and consists of extending the code C' with 11 for all codewords of weight d, then constructing all possibilities with (00) or (11) for a basis of remaining codewords, and eventually checking for addition of a vector strictly contained in the shadow of the extended code. This approach is very useful in classifying extremal self-dual $[2n+2,n+1,d+2]$ codes because it is sufficient to know (up to permutation) a classification of $[2n,n,\geq d]$ self-dual codes. Indeed, any $[2n+2,n+1,d+2]$ code gives a $[2n,n,d]$ code by subtraction of (11) on adequate columns, conversely applying the "reverse subtraction" procedure to the set of all $[2n,n,d]$ codes (up to permutation) permits to construct a set of codes which contains (up to permutation) all $[2n+2,n+1,d+2]$ codes.

We now recall the recursive algorithm with a correction of $n-k$ in Step 2) from [1] into $\frac{n}{2}-k$:

Recursive algorithm

Input: S_n, the set of $[n, \frac{n}{2}, d]$ self-dual codes up to permutation
Output: The set of $[n+2, \frac{n}{2}+1, d+2]$ self-dual codes
For each code C_n of S_n do:

1) List all the words of weight d and construct the subcode C_d of dimension k generated by these words. Construct a generator matrix G_d of C_d composed only with words of weight d.
2) Let E be a code of dimension $\frac{n}{2}-k$ with generator matrix G_E such that $C_n = C_d + E$, constructs the extended codes C with generator matrices

$$\begin{bmatrix} 1 & 1 & \\ \vdots & \vdots & G_d \\ 1 & 1 & \\ a_1 & a_1 & \\ \vdots & \vdots & G_E \\ a_{\frac{n}{2}-k} & a_{\frac{n}{2}-k} & \end{bmatrix} \tag{1}$$

such that $a_i \in \{0,1\}$, $(1 \le i \le \frac{n}{2}-k)$.
3) Complete all the previous codes C by nonzero elements of $C^{\perp}/C$ in order to obtain a self-dual code D and check for codes with minimum distance $d+2$. For codes with weight $d+2$ check for the equivalence with already obtained self-dual $[n+2, \frac{n}{2}+1, d+2]$ codes.

The main result of [1] is the following:

Theorem 1: Applying the previous recursive algorithm to the set of all inequivalent (up to permutation) binary self-dual $[n, n/2, d]$ codes permits to find all inequivalent self-dual binary $[n+2, n/2+1, d+2]$ codes.

B. The Building-Up Construction

There are other constructions generating self-dual codes of length $n+2$ from self-dual codes of length n. In particular, we compare the above construction with two constructions; the building-up construction [21] by Kim, and Harada-Munemasa's construction [17] since both constructions generate all self-dual codes of length $n+2$ from the set of all self-dual codes of length n.

Theorem 2: ([21, building-up]) Let $G_0 = (\mathbf{r_i})$ be a generator matrix (may not be in standard form) of a self-dual code C_0 over $GF(2)$ of length n, where $\mathbf{r}_i$ is a row of G_0 for $1 \le i \le n/2$. Let $\mathbf{x}$ be a vector in $GF(2)^n$ with an odd weight. Define $y_i := \mathbf{x} \cdot \mathbf{r}_i$ for $1 \le i \le n/2$, where $\cdot$ denotes the usual inner product. Then the following matrix:

$$G = \left[\begin{array}{cc|c} 1 & 0 & \mathbf{x} \\ \hline y_1 & y_1 & \\ \vdots & \vdots & G_0 \\ y_{n/2} & y_{n/2} & \end{array}\right] \tag{2}$$

generates a self-dual code C over $GF(2)$ of length $n+2$.

The converse of the building-up construction holds as follows.

Theorem 3: ([21]) Any self-dual code C over $GF(2)$ of length n with minimum weight $d > 2$ is obtained from some self-dual code C_0 of length $n-2$ (up to equivalence) by the construction in Theorem 2.

The recursive construction is a special case of the building-up construction. The reason is as follows.

We show that the matrix in the form (1) together with a representative in $C^{\perp}/C$ whose weight is >2 can be written in the form (2) up to permutation equivalence. Suppose we are given the matrix in the form (1) above and let C be the code generated by this matrix. Then there are four cosets of C in $C^{\perp}$; that is, C, $\mathbf{z}_1 + C$, $\mathbf{z}_2 + C$, and $\mathbf{z}_1 + \mathbf{z}_2 + C$ for some nonzeroes $\mathbf{z}_1$, $\mathbf{z}_2 \in GF(2)^{n+2}$. We may assume that $\mathbf{z}_1 = (1,1,0,0,\ldots,0)$ since $\mathbf{z}_1$ is nonzero and orthogonal to C. Then the minimum weight of $C \cup (\mathbf{z}_1 + C)$ is 2, which is excluded. Hence, by permuting the first two columns of $\mathbf{z}_2$ if needed, we may put $\mathbf{z}_2 = (1, 0 \mid \mathbf{x})$ where $\mathbf{x} \in GF(2)^n$. As $C \cup (\mathbf{z}_2 + C)$ is designed to be self-dual, $\mathbf{z}_2$ is orthogonal to itself; hence $\mathbf{x}$ is odd. Then as $\mathbf{z}_2 \cdot (1, 1 \mid \mathbf{r}_i) = 0$, where $\mathbf{r}_i$ is a row of G_d in the form (1) for $1 \le i \le k$, we have $\mathbf{x} \cdot \mathbf{r}_i = 1$. Thus, by letting $y_i := \mathbf{x} \cdot \mathbf{r}_i = 1$ for $1 \le i \le k$, we obtain the matrix of the form (2). This implies that the recursive construction is a special case of the building-up construction.

C. The Harada-Munemasa Construction

In what follows, we recall Harada-Munemasa's construction [17]. We note that this is a binary version of Huffman's construction [18] for Hermitian self-dual codes over $GF(4)$.

Let G_1 be a generator matrix of a self-dual $[n, n/2, d]$ code C_1. Then the matrix

$$G_2 := \begin{bmatrix} a_1 & a_1 & \\ \vdots & \vdots & G_1 \\ a_{n/2-1} & a_{n/2-1} & \end{bmatrix} \tag{3}$$

where $a_i \in GF(2)$ for $(1 \le i \le n/2-1)$, generates a self-orthogonal $[n+2, n/2]$ code C_2. The matrix of the form (3) is a general form of (1) in the recursive construction. In order to reduce the possibilities of a_i's, they [17] consider the orbits of the vector $a^T := (a_1, \ldots, a_{n/2-1})^T$ under a certain subgroup of $GL(n/2-1, 2)$ to get equivalent self-dual codes of length $n+2$. After reducing the possibilities, as in the recursive construction, add to C_2 a coset $\mathbf{z}_2 + C_2$ from $C_2^{\perp}/C_2$ whose weight is >2 to get a self-dual $[n+2, n/2+1, d' > 2]$ code. Unlike the recursive construction, Harada-Munemasa's construction does not necessarily give self-dual $[n+2, n/2+1]$ codes with minimum weight $d' = d+2$.

D. Comparison of the Different Methods

The recursive construction is specially interesting when one wants to classify extremal codes since it permits to obtain a partial classification for a given minimum distance while other constructions do need to start from a whole classification.

More precisely, the recursive construction is more efficient than the building-up construction in generating many self-dual

TABLE I
NUMBER OF SELF-DUAL $[36, 18, 6]$ CODES WHOSE SUBCODE GENERATED BY CODEWORDS OF WEIGHT 6 HAS DIMENSION k

dim k	num	dim k	num	dim k	num
2	148	8	4615	14	8170
3	5	9	911	15	5311
4	666	10	7165	16	6290
5	45	11	2299	17	4492
6	2165	12	8411	18	3615
7	263	13	4100		

TABLE II
NUMBER OF EXTREMAL SELF-DUAL $[38, 19, 8]$ CODES WITH RESPECT TO THEIR ORDERS

$\|\mathrm{Aut}(C)\|$	num	$\|\mathrm{Aut}(C)\|$	num	$\|\mathrm{Aut}(C)\|$	num
1	2253	9	1	36	1
2	322	12	8	144	1
3	36	14	1	168	2
4	68	18	1	216	1
6	17	21	1	342	1
8	15	24	14	504	1

codes with higher minimum weight. This is because the recursive construction checks a relatively small number of possibilities of $a_i's$ in Step 2), whose complexity is $2^{n/2-k}$, where $k \geq 1$ depends on the given code. From our experimental results, the dimensions k of subcodes of the 58671 [36, 18, 6] codes generated by linearly independent vectors of weight 6 lie between 2 and 18. We give the possible values of k and the number num of their subcodes in Table I.

We see from our table that there are much more subcodes of large dimension than those of small dimension and this clearly shows the efficiency of our recursive algorithm.

On the other hand, the building-up construction [21] needs 2^{n-1} possibilities for the choice of odd vectors $\mathbf{x}$, generating all self-dual codes with various minimum distances. This complexity can be reduced to $2^{n/2}$ as remarked in [13], which is still higher than that of the recursive construction.

As described above, Harada-Munemasa's construction is effective if the given code has a large automorphism group in order to reduce the complexity of checking the equivalence. For example, if $n = 36$, then 41019 (respectively 11242) out of the 58671 self-dual [36, 18, 6] codes [17] have the automorphism group order 1 (respectively 2). Thus Harada-Munemasa's construction usually requires 2^{19} or 2^{18} possibilities to generate self-dual codes of length 38 with various minimum distances, given a $[36, 18, 6]$ self-dual code.

Overall, we conclude that when we classify binary self-dual $[38, 19, 8]$ codes, the recursive algorithm is much faster than the other two constructions.

IV. CLASSIFICATION OF THE $[38, 19, 8]$ SELF-DUAL CODES

A. *Construction of All* $[38, 19, 8]$ *Self-Dual Codes*

There are two possible weight enumerators W_1, W_2 and shadow weight enumerators S_1, S_2 for an extremal self-dual $[38, 19, 8]$ code [9]

$$W_1 = 1 + 171y^8 + 1862y^{10} + \cdots \quad (4)$$

$$S_1 = 114y^7 + 9044y^{11} + 118446y^{15} + \cdots; \quad (5)$$

$$W_2 = 1 + 203y^8 + 1702y^{10} + \cdots \quad (6)$$

$$S_2 = y^3 + 106y^7 + 9072y^{11} + 118390y^{15}. \quad (7)$$

In [9], two self-dual $[38, 19, 8]$ codes with W_1, denoted by R_3 and D_4, were given, where $|\mathrm{Aut}(R_3)| = 1$ and $|\mathrm{Aut}(D_4)| = 342$. In [16] one self-dual $[38, 19, 8]$ code C_{38} with W_2 was given with $|\mathrm{Aut}(C_{38})| = 1$. Then Harada [15] gave 40 self-dual $[38, 19, 8]$ codes with W_1 and W_2 and automorphism group orders 1, 2, 4, 8. Later, Kim [21] constructed 325 self-dual $[38, 19, 8]$ codes with W_1 and W_2 and automorphism group orders 1, 2, 3. Hence there are at least 368 inequivalent self-dual $[38, 19, 8]$ codes. We show that there are exactly 2744 inequivalent self-dual $[38, 19, 8]$ codes.

Starting from the 58671 $[36, 18, 6]$ codes of [17], we apply the recursive algorithm of Section III-A. The more expensive part of the algorithm is the inequivalence testing of the differently constructed codes. In order to optimize the computation we separated the 58671 $[36, 18, 6]$ codes into sets $S_{36,i}$ of 1000 codes. To each set, we apply the recursive algorithm to obtain a list $S_{38,i}$ of inequivalent $[38, 19, 8]$ codes derived from the set $S_{36,i}$. Each set $S_{38,i}$ contains a number of inequivalent codes. Then we compared all the $S_{38,i}$ sets to eventually obtained a list of all inequivalent $[38, 19, 8]$ self-dual codes. This method permits to avoid many costly inequivalence comparisons between codes, since separating the whole list of $[36, 18, 6]$ codes permits to avoid inequivalence testing as the $S_{38,i}$ list starts from an empty list.

The whole process took about three weeks on a CPU 2.53-GHz computer.

Now we obtain our main theorem below.

Theorem 4: There are exactly 2744 inequivalent extremal self-dual $[38, 19, 8]$ codes.

In Table II, we describe all extremal self-dual $[38, 19, 8]$ codes with respect to their orders, where $|\mathrm{Aut}(C)|$ and num stand for the order of automorphism group and the number of codes respectively.

As mentioned above, the previously known self-dual $[38, 19, 8]$ codes have automorphism group orders 1,2,3,4,8, and 342. Hence we list several new self-dual $[38, 19, 8]$ codes C_{38}^i with different automorphism group orders $|\mathrm{Aut}(C_{38}^i)| = i = 6$, 9, 12, 14, 18, 21, 24, 36, 144, 168, 216, 504 in Appendix. To save space, we only give one code for each order. We also list C_{38}^{342} which is equivalent to the double-circulant code D_4 in [9]. The list of all extremal self-dual $[38, 19, 8]$ codes can be obtained at http://www.unilim.fr/pages_perso/philippe.gaborit/SD/GF2/GF2I.htm.

B. *An Up-to-Date Table of the Number of Classified Optimal Self-Dual Codes*

In Table II, we give an up-to-date table of the classification of optimal Type I self-dual codes, where being optimal means that this is the best possible minimum distance among self-dual codes of a given length. These codes may not be extremal in the classical sense. For instance, an extremal self-dual code of length 34 will have minimum distance 8 if exists, but it is known that such a code cannot exist and the optimal minimum distance is 6. The highest length (up to now) for which Type I optimal codes are classified is length 38, which is done in this paper for the first time. Notice that it is length 48 for Type II codes.

Complete references for the self-dual codes can be found for instance in [19] and [23], except for length 38.

V. Classification of s-Extremal Codes

In this section, we classify s-extremal codes of length 38 and $d=8$ together with s-extremal codes of length 38 and $d=6$.

A. s-Extremal Codes

The notion of s-extremal codes was introduced by Bachoc and Gaborit in [2]. This type of codes is related to the notion of self-dual codes with long shadows introduced by Elkies in [11]. We recall the definition of s-extremal codes from [2].

Let C be a Type I self-dual binary code of length n. We denote by C_0 the doubly-even subcode of C. We denote by $\mathbf{x}$ an element of $C\backslash C_0$. The shadow S is defined by $S=C_0^{\perp}\backslash C$, we denote by $\mathbf{y}$ an element of $S\backslash C$. We have $C_0^{\perp}=C_0\cup C_1\cup C_2\cup C_3$ for $C_1=\mathbf{y}+C_0$, $C_2=\mathbf{x}+C_0$ and $C_3=\mathbf{x}+\mathbf{y}+C_0$. Then it is well known that $C=C_0\cup C_2$ and $S=C_1\cup C_3$. We have moreover the following three facts [9]:

1) for any $\mathbf{y}\in S$, $\text{weight}(\mathbf{y})\equiv \frac{n}{2} \pmod 4$
2) for any $\mathbf{y}\in S$ and $\mathbf{x}\in C_2 : \mathbf{x}\cdot\mathbf{y}=1$,
3) for any $\mathbf{y}\in S$ and $\mathbf{z}\in C_0 : \mathbf{x}\cdot\mathbf{z}=0$.

We denote the weight enumerators of C and S by W_C and W_S, respectively. From [9], there exist $c_0,\ldots,c_{[n/8]}\in R$ such that

$$\begin{cases} W_C(x,y)=\sum_{i=0}^{[n/8]} c_i(x^2+y^2)^{\frac{n}{2}-4i}\{x^2y^2(x^2-y^2)^2\}^i \\ W_S(x,y)=\sum_{i=0}^{[n/8]} c_i(-1)^i 2^{\frac{n}{2}-6i}(xy)^{\frac{n}{2}-4i}(x^4-y^4)^{2i} \end{cases}. \tag{8}$$

Let d be the minimum weight of C and s the minimum weight of its shadow.

Theorem 5: ([2]) Let C be a Type I self-dual binary code of length n with minimum weight d, and let S be its shadow with minimum weight s. Then, $2d+s\le 4+\frac{n}{2}$, unless $n\equiv 22 \bmod 24$ and $d=4[n/24]+6$, in which case $2d+s=8+\frac{n}{2}$.

A Type I code whose parameters (d,s) satisfy the equality in the previous bounds is called *s-extremal*. In that case, the polynomials W_C and W_S are uniquely determined.

A bound for n when the minimum weight d of an s-extremal code is divisible by 4 has been given in [12] and in [14], and a bound has also been given for $d=6$ [2, Theorem 4.1], and $d\equiv 2 \pmod 4$ with $d>6$ [14].

Theorem 6 : ([12], [14]) Let C be an s-extremal code with parameters (s,d) of length n. If $d\equiv 0 \pmod 4$, then $n<6d-2$.

Theorem 7: ([14]) Let C be an s-extremal code with parameters (s,d) of length n. If $d>6$ and $d\equiv 2 \pmod 4$, then $n<21d-82$.

Before proving our classification of s-extremal codes of length 38, we prove a result which permits in certain cases to relate the weight of the shadow of a code C with the weight of the shadow of a subtracted code by (11):

Theorem 8: If C is a $[4n+2,2n+1,d+2]$ self-dual code with $d\equiv 0 \pmod 4$, $d\neq 0$ and shadow weight $s\ge 3$, then there exist two coordinates of C on which the subtraction of (11) gives a self-dual $[4n,2n,d]$ code C' with shadow weight $s-1$.

Proof: Our proof is based on the existence of the following four vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, and $\mathbf{t}$ such that:

1) $\mathbf{y}=(\mathbf{y}'10)$, $\mathbf{y}\in S$ of weight s
2) $\mathbf{x}=(\mathbf{x}'11)$, $\mathbf{x}\in C_2$ of weight $d+2$
3) $\mathbf{z}=(\mathbf{z}'11)$, $\mathbf{z}\in C_0$
4) $\mathbf{t}=(\mathbf{t}'10)$, $\mathbf{t}\in C_0$

Let $\mathbf{y}\in S$ of weight s and $\mathbf{x}\in C_2$ of weight $d+2$. We have $\mathbf{x}\cdot\mathbf{y}=1$, that is, $\mathbf{x}$ and $\mathbf{y}$ meet in an odd number of positions. Then $x_i=1=y_i$ for some i. As the weight of $\mathbf{x}$ is even, there is a j such that $x_j=1$ and $y_j=0$. Up to permutation, we may assume that $\mathbf{x}=(\mathbf{x}'11)$ and $\mathbf{y}=(\mathbf{y}'10)$. Now it remains to show that there exist $\mathbf{z}$ and $\mathbf{t}$ given above. To do this, note that $C_0^{\perp}=C\cup S$. Hence the minimum distance of $C_0^{\perp}=\min\{d+2,s\}\ge 3$. Hence every two columns of a generator matrix of C_0 are linearly independent. (This means that C_0 has strength 2. See [20, p. 435], for the term.) Thus in each set of two columns of C_0 each binary 2-tuple occurs the same number $|C_0|/4$ of times. Therefore there exist $\mathbf{z}=(\mathbf{z}'11)\in C_0$ and $\mathbf{t}=(\mathbf{t}'10)\in C_0$.

Since the coordinates of $\mathbf{z}$ and $\mathbf{t}$ are, respectively, (11) and (10) on the last two positions, there exists a doubly-even code C_0'' of dimension $2n-2$ such that the doubly-even subcode C_0 of C can be written as

$$\begin{pmatrix} \mathbf{z}' & 1 & 1 \\ \mathbf{t}' & 1 & 0 \\ \hline & 0 & 0 \\ C_0'' & \vdots & \vdots \\ & 0 & 0 \end{pmatrix}.$$

Now if one subtracts (11) on the two last columns of C one obtains a code C', such that its doubly-even subcode C_0' has dimension $2n-1$, ($2n-2$ vectors of C_0'' and the vector $\mathbf{x}'$-which cannot be null since $d+2\neq 2$), the subcode C_0' can be written as

$$\begin{pmatrix} \mathbf{x}' \\ C_0'' \end{pmatrix}.$$

Overall, a generator matrix of C' can be written as

$$\begin{pmatrix} \mathbf{z}' \\ \mathbf{x}' \\ C_0'' \end{pmatrix}$$

with $\mathbf{x}'$ of weight d. And $C_2'=C_0'+\mathbf{z}'$. Let $\mathbf{c}'$ be in C_0'' and denote by $\mathbf{c}$ the extension of $\mathbf{c}'$ with (00), then $\mathbf{c}\in C_0$. Now $\mathbf{y}'\cdot\mathbf{c}'=0$ since $\mathbf{y}\cdot\mathbf{c}=0$ and $\mathbf{y}'\cdot\mathbf{x}'=0$ since $\mathbf{y}\cdot\mathbf{x}=1$, which proves that for $\mathbf{c}\in C_0'$, $\mathbf{c}\cdot\mathbf{y}'=0$. Moreover since $\mathbf{y}\cdot\mathbf{z}=0$, we deduce that $\mathbf{y}'\cdot\mathbf{z}'=1$. The latter results show that C' is a $[4n,2n]$ self-dual code with minimum distance d (since $\mathbf{x}'$ has weight d), such that $C'=C_0'\cup(\mathbf{z}'+C_0')$ and with shadow $S'=(\mathbf{y}'+C_0')\cup(\mathbf{y}'+\mathbf{z}'+C_0')$. Finally, we remark that by construction, for any vector of S' it is possible to add either (11),(01),(10), or (00) such that the extended vector is in S. Since all the weights of S' are congruent to $s-1 \pmod 4$

and since $\mathbf{y}'$ of weight $s-1$ is in S', we deduce that the minimum weight of S' is $s-1$ which proves the theorem. □

B. *Classification of s-extremal* $[38, 19, 8]$ *Codes*

Let C be an extremal self-dual $[38, 19, 8]$ code. If C satisfies W_1 in (4), then S_1 in (5) is also satisfied. So we have $d = 8$ and $s = 7$; hence $2d + s = 23 = n/2 + 4$. This implies that C is an s-extremal code with parameters (7,8). Clearly if C satisfies W_2, then C cannot be an s-extremal code since $2d+s = 19 < 23$. The s-extremal code can be obtained directly from the classification of all $[38, 19, 8]$ by a simple computation on the weight enumerator. We obtain:

Theorem 9: There are exactly 1730 s-extremal $[38, 19, 8]$ codes.

C. *Classification of s-Extremal* $[38, 19, 6]$ *Codes*

The case of $d = 6$ was mainly considered in [2], where s-extremal codes are known to exist for the lengths $22 \leq n \leq 44$. Two lengths 36 and 38 remained open in [2]. Later, s-extremal codes of length 36 and $d = 6$ were classified in [1]. The only open case is the classification of s-extremal codes of length 38 and $d = 6$. There are at least two such codes as shown in [2]. We show that there are exactly two s-extremal codes of length 38 and $d = 6$.

For a self-dual $[38, 19, 6]$ code to be s-extremal, the minimum weight of its shadow must be $s = 11$. A simple approach to find all s-extremal $[38, 19, 6]$ codes is to apply the recursive construction, starting from the set of all inequivalent $[36, 18, 4]$ self-dual codes. Unfortunately, since there are $436,633$ $[36, 18, 4]$ self-dual codes, such a computation would require more than 80 days, and although it is doable theoretically, in practice it remains largely too costly. Fortunately, by using the fact that such an s-extremal code has a shadow with high minimum weight it is possible to dramatically decrease this computation.

We have shown in Theorem 8 that it is possible to relate the weight of a shadow of a code to the that of the shadow of the subtracted code under certain conditions. We use this result to prove the following classification theorem:

Theorem 10: There are exactly two s-extremal $[38, 19, 6]$ codes.

Proof: Let C be an s-extremal $[38, 19, 6]$ code, then C has shadow weight $s = 11$. Applying Theorem 8, we deduce that there exist two coordinates on which the subtraction of (11) of C produces a $[36, 18, 4]$ self-dual code with shadow weight 10. Hence, if one applies the recursive algorithm starting from the set $S_{36,10}$ of all inequivalent $[36, 18, 4]$ self-dual codes with shadow weight 10, we construct the set of all $[38, 19, 6]$ self-dual codes (up to permutation) which by a subtraction of (11) on certain two columns give the set $S_{36,10}$. Hence, applying the recursive algorithm to $S_{36,10}$ gives a set of self-dual codes which contains all s-extremal $[38, 19, 6]$ codes. In practice, from the classification of [17], there are exactly 24 $[36, 18, 4]$ self-dual codes with shadow weight 10. The application of the recursive algorithm is then fast with these codes and we have that there are exactly two s-extremal $[38, 19, 6]$ codes. □

The two s-extremal codes $C_{38,1}$, $C_{38,2}$ have covering radius 11 and their generator matrices $G(C_{38,1})$, $G(C_{38,2})$ are as follows:

$$G(C_{38,1}) = \begin{bmatrix}
10000000000000000000000011111100001110 \\
01000000000000000000000011111100000001 \\
00100000000000000001010101010101111001 \\
00010000000000000001010101010101000101 \\
00001000000000000000000101010101110011 \\
00000100000000000000000101010110111111 \\
00000010000001000000000011110000000000 \\
00000001000001000000000011101110000000 \\
00000000100001010000000110100011000000 \\
00000000010001010000000110010000000000 \\
00000000001001010000010000100101010110 \\
00000000000101010000010011011010010110 \\
00000000000011000000000011000011000000 \\
00000000000000110000000011001100000000 \\
00000000000000001001000101010110011001 \\
00000000000000000101000110101001101001 \\
00000000000000000011000011111111001100 \\
00000000000000000000110011111111111100 \\
00000000000000000000001110011111000000
\end{bmatrix}$$

$$G(C_{38,2}) = \begin{bmatrix}
10000000000000000100010101010110010010 \\
01000000000000000100010101010110011101 \\
00100000000000000101000000000000100110 \\
00010000000000000101000000000000011001 \\
00001000000000000000000101010101000000 \\
00000100000000000000000101010110111100 \\
00000010000000010101010001011101110101 \\
00000001000000010101010001010010110101 \\
00000000100000000101010110000110110101 \\
00000000010000000101010110110101110101 \\
00000000001000000000000110101001110101 \\
00000000000100000000000101010110000110 \\
00000000000010010101010011100110110101 \\
00000000000001010101010000100101110101 \\
00000000000000110000000011001100000000 \\
00000000000000011000000011111111110000 \\
00000000000000000011000011111111111111 \\
00000000000000000000110011111111001111 \\
00000000000000000000001110011111000000
\end{bmatrix}.$$

Notice that these codes were already known from [2], but it was not known whether there exist other codes.

D. *Up-to-Date Tables for s-Extremal Codes*

In the following, we give up-to-date tables for s-extremal codes of minimum distance 6 and 8:

- $d = 6$.

For this minimum distance, we know that there are exactly two s-extremal codes of length 38 and $d = 6$ from Theorem 10. This was the only unknown case (see [1], [2]). Now we complete the classification of s-extremal codes of $d = 6$ in Table IV.

- $d = 8$.

In this case, s-extremal codes exist for $32 \leq n \leq 44$. More precisely, s-extremal codes of length 32 were known from the classification of extremal self-dual codes of length 32, and s-extremal codes of length 36 were done in [1]. We have completed the classification s-extremal codes of length 38 and $d = 8$ from Theorem 9. We list currently known codes for $d = 8$ in Table V.

VI. Covering Radii of Self-Dual Codes of Length 38

The *covering radius* $\rho(C)$ of a code C is the smallest integer R such that spheres of radius R around codewords cover $\mathbb{F}_2^n$.

TABLE III
NUMBER OF OPTIMAL TYPE I AND TYPE II CODES

n	d	num	n	d	num
2	2	1	22	6	1
4	2	1	24	8	1
6	2	1	26	6	1
8	4	1	28	6	3
10	2	1	30	6	13
12	4	1	32	8	8
14	4	1	34	6	938
16	4	3	36	8	41
18	4	2	38	8	2744
20	4	7			

TABLE IV
NUMBER OF s-EXTREMAL CODES WITH $d = 6$

n	num	n	num
22	1	34	17
24	1	36	5
26	1	38	2
28	2	40	1
30	9	42	1
32	19	44	1

TABLE V
NUMBER OF s-EXTREMAL CODES WITH $d = 8$

n	num	ref
32	3	[9]
36	25	[1]
38	1730	this paper
40	≥ 4	[9],[7]
42	≥ 17	[9],[6]
44	≥ 1	[9]

The following theorems give the lower and upper bound of $\rho(C)$ for a self-dual code over $GF(2)$.

Theorem 11: ([8], Theorem 1) Let C be a self-dual code of length n over $GF(2)$. Then $\sum_{i=0}^{\rho(C)} {}_nC_i \geq 2^{n/2}$. More precisely, $\sum_{2i \leq \rho(C)} {}_nC_{2i} \geq 2^{(n/2)-1}$ and $\sum_{2i+1 \leq \rho(C)} {}_nC_{2i+1} \geq 2^{(n/2)-1}$, where ${}_nC_i$ means n choose i.

Theorem 12: ([8], [20], Delsarte's bound) Let C be a self-dual code of length n over $GF(2)$ and s be the number of distinct nonzero weights in C. Then $\rho(C) \leq s$.

By Theorem 11, any self-dual $[38, 19]$ code has covering radius at least 6. On the other hand, the weight enumerators (4) and (6) of any self-dual $[38, 19, 8]$ code has 13 nonzero weights. Thus by Theorem 12, the covering radius of any self-dual $[38, 19, 8]$ code is at most 13. Combining both, we have $6 \leq \rho(C) \leq 13$ for any self-dual $[38, 19, 8]$ code C.

Using our classification of all self-dual $[38, 19, 8]$ codes, we have the following.

Theorem 13: All 2744 self-dual $[38, 19, 8]$ codes have covering radius 7.

Remark 14: If we choose a coset representative of weight 7 and using it as a vector $\mathbf{x}$ in Theorem 2, then the built code will be an extremal self-dual $[40, 20, 8]$ code. Hence for $n = 38$, any extremal self-dual $[38, 19, 8]$ code can produce an extremal self-dual $[40, 20, 8]$ code using the building-up construction. This is not always true for some lengths (e.g., $n = 24$, $n = 32$).

Proposition 15: Let C_1 be a self-dual code of length n and covering radius $\rho(C_1)$. Then any self-dual code C_2 of length $n+2$ obtained by the building-up construction (in particular, by the recursive algorithm) has covering radius $\rho(C_2) \leq \rho(C_1)+2$.

Proof: Let $\rho(C_1) = r$. We recall [20, Theorem 1.25.5], that the covering radius $\rho(C_1)$ of a linear code C_1 with parity check matrix H_1 is the smallest number s such that every nonzero syndrome is a combination of s or fewer columns of H_1, and some syndrome requires s columns. The generator matrix G_2 of C_2 by the building-up construction is of the form (2). This G_2 is also a parity check matrix of C_2 as C_2 is self-dual. Any syndrome $\mathbf{u} = [u_1, u_2, \ldots, u_{(n/2)+1}]^T$ with respect to G_2 can be written as $u_1[1, 0, \ldots, 0]^T + [0, u_2, \ldots, u_{(n/2)+1}]^T$. Now $[0, u_2, \ldots, u_{(n/2)+1}]^T$ is a linear combination of r or fewer columns of G_2 as G_1 has covering radius r, and $[1, 0, \ldots, 0]^T$ is the difference of the first columns of G_2 in the form (2). Hence $\mathbf{u}$ is a linear combination of at most $r + 2$ columns of G_2. Thus $\rho(C_2) \leq \rho(C_1) + 2$. □

Using Proposition 15, we have a better upper bound for the covering radius of a self-dual $[38, 19, 6]$ code than Delsarte's bound as follows.

Corollary 16: The covering radius $\rho(C)$ of any self-dual $[38, 19, 6]$ code is $6 \leq \rho(C) \leq 12$.

Proof: The lower bound is true for any even $[38, 19]$ code by Theorem 11. Delsarte's bound would imply $\rho(C) \leq 15$. For a better upper bound, we recall that any self-dual $[38, 19, 6]$ code can be constructed from a self-dual $[36, 18, 4]$ code by the recursive algorithm. Since the covering radius of any self-dual $[36, 18, d = 4, 6]$ code is at most 10 [17], it follows that the covering radius of any self-dual $[38, 19, 6]$ code is at most 12 by Proposition 15. □

APPENDIX

Let i = 6, 9, 12, 14, 18, 21, 24, 36, 144, 168, 216, 342, 504. Then $G(C_{38}^i)$ represents a generator matrix of a new self-dual $[38, 19, 8]$ code C_{38}^i with the automorphism group order $|\mathrm{Aut}(C_{38}^i)| = i$

$$
G(C_{38}^6) = \begin{bmatrix}
10000000000000000001010110011101101001 \\
01000000000000000001010110011101010110 \\
00100000000000000000010101101000011000 \\
00010000000000000000010101100111100111 \\
00001000000001000001100111011001010100 \\
00000100000001000001100111100110101000 \\
00000010000001000001110000100101100001 \\
00000001000001000001110011011010101101 \\
00000000100001000000100101011001101110 \\
00000000010001000000100110011010100001 \\
00000000001001000001000110010000010010 \\
00000000000101000001000100111100001100 \\
00000000000011000000000010101111100001 \\
00000000000000100001100000010000110101 \\
00000000000000010000000011010011110110 \\
00000000000000001001100010001100101000 \\
00000000000000000101010001111111011011 \\
00000000000000000011100100111111100010 \\
00000000000000000000001101011111101101
\end{bmatrix}
$$

$$G(C^{9}_{38})=\begin{bmatrix}
10000000000000000000011101010111111101\\
01000000000000000000011101010111000010\\
00100000000000000001010011111101100010\\
00010000000000000001010011110010010001\\
00001000000000000001100001101110010111\\
00000100000000000001100001010001100111\\
00000010000000000000111101011000110110\\
00000001000000000001111101001111111001\\
00000000100000000001101000100100111001\\
00000000010000000001101011100111111010\\
00000000001000000000001011101101000101\\
00000000000100000000000100011110110101\\
00000000000010000001001111010010110110\\
00000000000001000001000000100010110110\\
00000000000000100001111011011010100010\\
00000000000000010001001000011001101110\\
00000000000000001000100100011001010010\\
00000000000000000100010111101010101101\\
00000000000000000011111111000000001100
\end{bmatrix}$$

$$G(C^{12}_{38})=\begin{bmatrix}
10000000000000000011000110001010110100\\
01000000000000000011000110001010001011\\
00100000000000000011010110100010000101\\
00010000000000000011010110100001001010\\
00001000000000000100000111000001101101\\
00000100000000000100000111001101100010\\
00000010000000000010010000100101101011\\
00000001000000000010010011011010010111\\
00000000100000000111000011111010101011\\
00000000010000000111000100101100111010\\
00000000001000000010001010010100111110\\
00000000000100000001010001110011111010\\
00000000000010000000000110011001011000\\
00000000000001000011000110100101010100\\
00000000000000100101000101010101010111\\
00000000000000010110010010010011001111\\
00000000000000001111010000011100000110\\
00000000000000000000100000101100110110\\
00000000000000000000001000101010101101
\end{bmatrix}$$

$$G(C^{14}_{38})=\begin{bmatrix}
10000000000000000000101001101100111000\\
01000000000000000000101001101100000111\\
00100000000000000000111010000100010100\\
00010000000000000000111010000111100111\\
00001000000000000000101111111000111111\\
00000100000000000000101111101110011111\\
00000010000000000000110101010011000000\\
00000001000000000000110101100011111100\\
00000000100000000001111110100110000110\\
00000000010000000001111101010110110101\\
00000000001000000000001010111110011011\\
00000000000100000000000110000010011011\\
00000000000010000001001110000010010100\\
00000000000001000001000001001101100111\\
00000000000000100001010010011000100010\\
00000000000000010001100010010111101101\\
00000000000000001000001110011011011110\\
00000000000000000100111101100100100001\\
00000000000000000001111100110000001111
\end{bmatrix}$$

$$G(C^{18}_{38})=\begin{bmatrix}
10000000000000000100000111100100111000\\
01000000000000000100000111100100000111\\
00100000000000000101000100010111001111\\
00010000000000000101000100011000110011\\
00001000000000000000100101101000100001\\
00000100000000000000100101010111010001\\
00000010000000000101100000010010111010\\
00000001000000000101100011101101111001\\
00000000100000000100110101101110001001\\
00000000010000000100110110101101001010\\
00000000001000000101010110100111111001\\
00000000000100000101010111110010100010\\
00000000000010000100010010011000000101\\
00000000000001000100010011001110101101\\
00000000000000100100101100010010110100\\
00000000000000010101010001010101100100\\
00000000000000001100110011110011110000\\
00000000000000000011110001100110100100\\
00000000000000000000001110100110101011
\end{bmatrix}$$

$$G(C^{21}_{38})=\begin{bmatrix}
10000000000000000100001010111101011100\\
01000000000000000100001010111101100011\\
00100000000000000100100101000100101101\\
00010000000000000100100101001011011110\\
00001000000000000100101001000010111001\\
00000100000000000100101001111101001001\\
00000010000000000000000011001101011011\\
00000001000000000000000000110010011011\\
00000000100000000000110111100010110101\\
00000000010000000000110100100001111010\\
00000000001000000000110101111000100111\\
00000000000100000000111010001011010100\\
00000000000010000000010000010100111001\\
00000000000001000000011111100100111001\\
00000000000000100100001101100011101110\\
00000000000000010100111110100000100010\\
00000000000000001100110011110011110000\\
00000000000000000010011110010011101110\\
00000000000000000001100001010011101101
\end{bmatrix}$$

$$G(C^{24}_{38})=\begin{bmatrix}
10000000000000000000110110010111100110\\
01000000000000000000110110010111011001\\
00100000000000000000101110101111010001\\
00010000000000000000101110101100101101\\
00001000000001000000011101100111001100\\
00000100000001000000011101101000000000\\
00000010000001000000010000010011110011\\
00000001000001000000010000101111001111\\
00000000100000000000000111111001011110\\
00000000010000000000000100110110010001\\
00000000001001000000010001110101011110\\
00000000000101000000011101110110010010\\
00000000000011000000011001100001111100\\
00000000000000100000111000101011010111\\
00000000000000010000001011010111100111\\
00000000000000001000111101010001111001\\
00000000000000000100001110101101111010\\
00000000000000000010100101111001101101\\
00000000000000000001011001001010101101
\end{bmatrix}$$

$$G(C_{38}^{36})=\begin{bmatrix}
10000000000000000001110100011110101001\\
01000000000000000001110100011110010110\\
00100000000000000001010110110111011011\\
00010000000000000001010110111000011000\\
00001000000000000001000101010001101101\\
00000100000000000001000101101110100010\\
00000010000000000000011100101101111010\\
00000001000000000000011111010010000110\\
00000000100000000001001001010001111010\\
00000000010000000001001010010010110110\\
00000000001000000000101010011000001010\\
00000000000100000000100101101011110101\\
00000000000010000001101110100111111001\\
00000000000001000001100001010111000101\\
00000000000000100001111110010000101011\\
00000000000000010001001101010011010100\\
00000000000000001000100001010011101011\\
00000000000000000100010010100000010111\\
00000000000000000011111111000000111100
\end{bmatrix}$$

$$G(C_{38}^{144})=\begin{bmatrix}
10000000000000000000110101000110111110\\
01000000000000000000110101000110000001\\
00100000000000000000010101101110110011\\
00010000000000000000010101101101111100\\
00001000000000000000010111111011001100\\
00000100000000000000010111111011100011\\
00000010000000000000110101010011110011\\
00000001000000000000110101100000001100\\
00000000100000000001101100100101101011\\
00000000010000000001101111010101010100\\
00000000001000000001110001100101101110\\
00000000000100000001111101011001010010\\
00000000000010000000110101011001101110\\
00000000000001000000111010010101011101\\
00000000000000100000111011000011000110\\
00000000000000010000001011001100001010\\
00000000000000001001100111000011001010\\
00000000000000000101010100111100110101\\
00000000000000000011111100110011111100
\end{bmatrix}$$

$$G(C_{38}^{168})=\begin{bmatrix}
10000000000000000001000010111010001000\\
01000000000000000001000010111010110111\\
00100000000000000101000100010111001111\\
00010000000000000101000100011000001100\\
00001000000000000001001100001100010100\\
00000100000000000001001100110011100111\\
00000010000000000100000111010000100111\\
00000001000000000100000100101111011011\\
00000000100000000101001100111011001010\\
00000000010000000101001111111000001001\\
00000000001000000100101111110010110101\\
00000000000100000100100000000001110101\\
00000000000010000101101011001101001001\\
00000000000001000101100100111101110110\\
00000000000000100101101100110000001111\\
00000000000000010101000001100100100010\\
00000000000000001100101101100100101110\\
00000000000000000011100001010111011110\\
00000000000000000000011110010111101101
\end{bmatrix}$$

$$G(C_{38}^{216})=\begin{bmatrix}
10000000000000000000001101101111100110\\
01000000000000000000001101101111011001\\
00100000000000000000111010000100101101\\
00010000000000000000111010000111100010\\
00001000000000000010011111111110011011\\
00000100000000000010011111110010010100\\
00000010000000000010111000100011001111\\
00000001000000000010111011011100111100\\
00000000100000000000001000111001010001\\
00000000010000000000000111000101101101\\
00000000001000000000111110110000110110\\
00000000000100000000000010001100111010\\
00000000000010000000101110011111110011\\
00000000000001000010011101110000111010\\
00000000000000100010001110010110100010\\
00000000000000010000000010000110010111\\
00000000000000001000010011110110100010\\
00000000000000000110010000010101100100\\
00000000000000000001110011010011111001
\end{bmatrix}$$

$$G(C_{38}^{342})=\begin{bmatrix}
10000000000001000000001010101111111011\\
01000000000001000000001010101111000100\\
00100000000000000100001101101001111010\\
00010000000000000100001101100110111001\\
00001000000001000100101001000000101011\\
00000100000001000100101001111111100111\\
00000010000001000000101011100010011110\\
00000001000001000000101000011101101110\\
00000000100001000000101000000110110011\\
00000000010001000000101011000101111100\\
00000000001001000000000010110001111010\\
00000000000101000000001101000010000110\\
00000000000011000000001111110000110000\\
00000000000000100100100101001110000101\\
00000000000000010100001001101011101101\\
00000000000000001100101100010101100111\\
00000000000000000010101001011000010001\\
00000000000000000001001001111110110101\\
00000000000000000000011111100110010111
\end{bmatrix}$$

$$G(C_{38}^{504})=\begin{bmatrix}
10000000000000010010011000100100010110\\
01000000000000010010011000100100101001\\
00100000000000010010010110111100011000\\
00010000000000010010010110110011010100\\
00001000000000000010011001010011110000\\
00000100000000000010011001101100001100\\
00000010000000010010000100000110000011\\
00000001000000010010000111111010110000\\
00000000100000000000010111100110101101\\
00000000010000000000011000100101101110\\
00000000001000010000010100011110100100\\
00000000000100010000011111010101011101\\
00000000000010010000010010001010111100\\
00000000000001010010000000010101100010\\
00000000000000110010011010100111100111\\
00000000000000001000001100010110101110\\
00000000000000000110011001111110111001\\
00000000000000000001010010100011010010\\
00000000000000000000110111001000000101
\end{bmatrix}$$

ACKNOWLEDGMENT

All the computations were done with the MAGMA system [4]. While this work was under review, the authors learned that independently, similar results have been obtained by others (see [3], [5]).

REFERENCES

[1] C. A. Melchor and P. Gaborit, "On the classification of extremal $[36, 18, 8]$ binary self-dual codes," *IEEE Trans. Inform. Theory*, vol. 54, no. 10, pp. 4743–4750, Oct. 2008.

[2] C. Bachoc and P. Gaborit, "Designs and self-dual codes with long shadows," *J. Combin. Theory Ser. A*, vol. 105, pp. 15–34, 2004.

[3] K. Betsumiya, M. Harada, and A. Munemasa, A complete classification of doubly even self-dual codes of length 40, 2011, arXiv:1104.3727v2.

[4] W. Bosma and J. Cannon, *Handbook of Magma Functions*. Sydney, Australia: , 1995.

[5] S. Bouyuklieva and I. Bouyukliev, On the classification of binary self-dual codes 2011, arXiv:1106.5930v1.

[6] S. Buyuklieva, "New extremal self-dual codes of length 42 and 44," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1607–1612, 1997.

[7] S. Buyuklieva and V. Yorgov, "Singly-even self-dual codes of length 40," *Des. Codes Cryptog.*, vol. 9, pp. 131–141, 1996.

[8] G. D. Cohen, M. G. Karpovsky, H. F. Mattson, Jr, and J. R. Schatz, "Covering radius-survey and recents results," *IEEE Trans. Inf. Theory*, vol. IT-31, pp. 328–343, 1985.

[9] J. H. Conway and N. J. A. Sloane, "A new upper bound on the minimal distance of self-dual codes," *IEEE Trans. Inform. Theory*, vol. 36, pp. 1319–1333, 1990.

[10] S. T. Dougherty, T. A. Gulliver, and M. Harada, "Extremal binary self-dual codes," *IEEE Trans. Inform. Theory*, vol. 43, pp. 2036–2047, 1997.

[11] N. D. Elkies, "Lattices and codes with long shadows," *Math. Res. Lett.*, vol. 2, pp. 643–651, 1995.

[12] P. Gaborit, "A bound for certain s-extremal lattices and codes," *Arch. Math (Basel)*, vol. 89, no. 2, pp. 143–151, 2007.

[13] T. A. Gulliver, M. Harada, and J.-L. Kim, "Construction of some extremal self-dual codes," *Discrete Math*, vol. 263, pp. 81–91, 2003.

[14] S. Han and J.-L. Kim, "Upper bound for the length of s-extremal codes over $\mathbb{F}_2, \mathbb{F}_4$ and $\mathbb{F}_2 + u\mathbb{F}_2$," *IEEE Trans. Inform. Theory*, vol. 54, no. 1, pp. 418–422, 2008.

[15] M. Harada, "New extremal self-dual codes of lengths 36 and 38," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2541–2543, 1999.

[16] M. Harada and H. Kimura, "On extremal self-dual codes," *Math. J. Okayama Univ.*, vol. 37, pp. 1–14, 1995.

[17] M. Harada and A. Munemasa, Classification of Self-Dual Codes of Length 36 2010, arXiv:1012.5464v1 [math.CO].

[18] W. C. Huffman, "Characterization of quaternary extremal codes of lengths 18 and 20," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1613–1616, 1997.

[19] W. C. Huffman, "On the classification and enumeration of self-dual codes," *Finite Fields Appl.*, vol. 11, pp. 451–490, 2005.

[20] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[21] J.-L. Kim, "New extremal self-dual codes of lengths 36,38 and 58," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1575–1580, 2001.

[22] F. J. MacWilliams, N. J. A. Sloane, and J. G. Thompson, "Good self-dual codes exist," *Discrete Math.*, vol. 3, pp. 153–162, 1972.

[23] G. Nebe, E. Rains, and N. J. A. Sloane, *Slef-Dual Codes and Invariant Theory*. Berlin, Germany: Springer, 2006.

[24] V. Pless, "A classification of self-orthogonal codes over $GF(2)$," *Discrete Math.*, vol. 3, pp. 209–246, 1972.

[25] E. M. Rains and N. J. A. Sloane, "Self-dual codes," in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998, pp. 177–294.

Carlos Aguilar-Melchor , former student of the École Polytechnique de Paris, obtained his Ph.D. degree in 2006 at the LAAS-CNRS (Toulouse) under the supervision of Yves Deswarte. His research domains are security, privacy, codes and cryptography. In security and privacy, he has worked mainly on anonymous communications and on trust and replication issues on new generation networks. In codes and cryptography, he has worked on various privacy primitives such as private information retrieval, homomorphic encryption and ring signatures and on the enumeration of self-dual codes. He is currently Assistant Professor at the University of Limoges.

Philippe Gaborit obtained his Ph.D. degree from the University of Bordeaux I, France, in 1997. After a two years post-doc position at the University of Illinois at Chicago, he became assistant professor at the Univesity of Limoges, France, then full professor. His main research interests are coding theory, cryptography, privacy, security and watermarking.

Jon-Lark Kim (S'01–A'03) received the B.S. degree in mathematics from POSTECH, Pohang, Korea, in 1993, the M.S. degree in mathematics from Seoul National University, Seoul, Korea, in 1997, and the Ph.D. degree in mathematics from the University of Illinois at Chicago, in 2002. From 2002 to 2005, he was with the Department of Mathematics at the University of Nebraska-Lincoln as a Research Assistant Professor. Currently, he is an Associate Professor at the University of Louisville, KY. He was awarded a 2004 Kirkman Medal of the Institute of Combinatorics and its Applications. He is a member of the Editorial Board of both ''Designs, Codes, and Cryptography'' and ''International J. of Inform. and Coding Theory''. His areas of interest include Algebraic Coding Theory and its interaction with Algebra, Combinatorics, Cryptography, Number Theory, and Quantum Information.

Lin Sok received a B.Sc. degree in Mathematics from the Royal University of Phnom Penh, Phnom Penh, Cambodia, in 2003, a M.Sc. degree in Discrete Mathematics from the Université de la Méditerranée-Aix Marseille II, Marseille, France, in 2008.

He has been was a Ph.D. student at the Department of Comelec, Telecom ParisTech, Paris, France, since 2010. His research interests include coding theory, Boolean functions and discrete mathematics.

Patrick Solé received the Ingenieur and the Docteur Ingenieur degrees from Ecole Nationale Superieure des Telecommunications, Paris, France in 1984 and 1987, respectively, and the Habilitation a Diriger des Recherches degree from Universite de Nice, Sophia-Antipolis, France, in 1993.

He has held visiting positions at Syracuse University, Syracuse, NY, during 1987-1989, Macquarie University, Sydney, Australia, during 1994-1996, and at Universite des Sciences et Techniques de Lille, Lille, France, during 1999-2000. He has been a permanent member of Centre National de la Recherche Scientifique since 1989, and with the rank of Research Professor (Directeur de Recherche) since 1996. Since 2011, he has a joint affiliation with King AbdulAziz University, Jeddah, Saudi Arabia.

His research interests include coding theory (covering radius, codes over rings, geometric codes, quantum codes), interconnection networks (graph spectra, expanders), space time codes (lattices, theta series), and cryptography (Boolean functions). He is well-known for using methods of ring theory in algebraic coding. In particular the best paper award in Information Theory was awarded in 1994 to his paper on codes over the integers modulo 4, which has been quoted 720 times since. He is the author of more than 100 journal papers and one book (Codes Over Rings, World Scientific, 2008).