**Noname manuscript No.**
(will be inserted by the editor)

# Boolean Functions with Multiplicative Complexity 3 and 4

**Çağdaş Çalık · Meltem Sönmez Turan · René Peralta**

Received: date / Accepted: date

**Abstract** Multiplicative complexity (MC) is defined as the minimum number of AND gates required to implement a function with a circuit over the basis (AND, XOR, NOT). Boolean functions with MC 1 and 2 have been characterized in Fisher and Peralta (2002), and Find et al. (2017), respectively. In this work, we identify the affine equivalence classes for functions with MC 3 and 4. In order to achieve this, we utilize the notion of the dimension $dim(f)$ of a Boolean function in relation to its linearity dimension, and provide a new lower bound suggesting that the multiplicative complexity of $f$ is at least $\lceil dim(f)/2 \rceil$. For MC 3, this implies that there are no equivalence classes other than those 24 identified in Çalık et al. (2018). Using the techniques from Çalık et al. and the new relation between the dimension and MC, we identify all 1277 equivalence classes having MC 4. We also provide a closed formula for the number of $n$-variable functions with MC 3 and 4. These results allow us to construct AND-optimal circuits for Boolean functions that have MC 4 or less, independent of the number of variables they are defined on.

**Keywords** Affine equivalence · Boolean functions · Multiplicative complexity.

**Mathematics Subject Classification (2010)** 94A60 · 06E30

Ç. Çalık
NIST Computer Security Division, 100 Bureau Dr, Gaithersburg, MD 20899
Tel.: +1-301-975-4024
Fax: +1-301-975-8670
E-mail: cagdas.calik@nist.gov

M. Sönmez Turan
NIST Computer Security Division, 100 Bureau Dr, Gaithersburg, MD 20899

René Peralta
NIST Computer Security Division, 100 Bureau Dr, Gaithersburg, MD 20899

## 1 Introduction

In cryptographic protocols such as fully-homomorphic encryption (e.g., [1]), zero-knowledge proofs (e.g., [2]), and secure multi-party computation (e.g. [3]), Boolean circuits using fewer nonlinear gates are preferred for efficiency. This promoted the design of symmetric primitives (e.g., Rasta [4], LowMC [5]), which are inherently designed to use only a small number of AND gates.

*Multiplicative Complexity* (MC) is defined as the minimum number of AND gates required to implement a given function by a circuit over the basis (AND, XOR, NOT). The MC of a random $n$-variable Boolean function $f$, denoted $C_\wedge(f)$, is at least $2^{n/2} - \mathcal{O}(n)$ with high probability [6]. The MC of a random Boolean function is hard to calculate even for a small number of variables. For up to 6 variables, the MC of each Boolean function has been established in [7, 8]. For arbitrary $n$, it is known that under standard cryptographic assumptions, computing the MC in polynomial time in the length of the truth table [9] is not possible. There are, however, results for special classes of Boolean functions. In [10], Mirwald and Schnorr studied the MC of quadratic functions and showed that $C_\wedge(f) = k$, iff $f$ is isomorphic to the canonical form $\bigoplus_{i=1}^{k} x_{2i-1}x_{2i}$. In [11], Brandão et al. studied the MC of symmetric Boolean functions and constructed circuits for all such functions with up to 25 variables.

A particular value of interest is the number of $n$-variable Boolean functions with MC $k$, denoted $\lambda(n,k)$. In [6], it is shown that $\lambda(n,k) \leq 2^{k^2+2k+2kn+n+1}$. In 2002, Fischer and Peralta [12] showed that $\lambda(n,1)$ is equal to $2\binom{2^n}{3}$. In 2017, Find et al. [13] characterized the Boolean functions with MC 2 by using the fact that MC is invariant with respect to affine transformations and showed that

$$\lambda(n,2) = 2^n(2^n-1)(2^n-2)(2^n-4)\left(\frac{2}{21} + \frac{2^n-8}{12} + \frac{2^n-8}{360}\right). \qquad (1)$$

In this work, we focus on Boolean functions with MC 3 and 4. We utilize the notion of the dimension $dim(f)$ of a Boolean function in relation to its linearity dimension [14], and provide a new lower bound suggesting that $C_\wedge(f) \geq \lceil dim(f)/2 \rceil$. For MC 3, this implies that there are no other equivalence classes other than those 24 identified in [8]. For MC 4, using the techniques from [8] and the new relation between dimension and MC, we identify 1277 equivalence classes. We also provide a closed formula for the number of $n$-variable functions with MC 3 and 4, i.e., $\lambda(n,3)$ and $\lambda(n,4)$.

The techniques allow us to construct AND-optimal circuits for Boolean functions that have MC 4 or less, independent of the number of variables they are defined on. Knowledge of all equivalence classes with MC 4 or less can also be used to determine that a function has MC greater than 4, if it does not belong to any of those classes.

The organization of the paper is as follows. Section 2 gives definitions and preliminary information about Boolean functions and Boolean circuits. Section 3 explains the relation between dimension and MC, and presents the new lower bound. Section 4 provides the affine equivalence classes of Boolean functions with MC 3 and 4. Section 5 concludes the paper with discussion of future research directions.

## 2 Preliminaries

### 2.1 Boolean Functions

Let $\mathbb{F}_2$ be the binary field with 2 elements and $\mathbb{F}_2^n$ be the $n$-dimensional vector space over $\mathbb{F}_2$. There is a one-to-one mapping between the elements of $\mathbb{F}_2^n$ and the integers modulo $2^n$ so that $a = (a_{n-1}, \ldots, a_0) \in \mathbb{F}_2^n$ maps to the integer $\sum_{i=0}^{n-1} a_i 2^i$. For simplicity, we will occasionally use an integer when an element of $\mathbb{F}_2^n$ is expected. The unit vectors $e_i \in \mathbb{F}_2^n$ are defined to be vectors whose $i^{th}$ entry is 1 and the remaining entries are zeros.

An $n$-variable Boolean function $f$ is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let $\mathcal{B}_n$ be the set of $n$-variable Boolean functions. The *truth table* $T_f$ of a function $f \in \mathcal{B}_n$ is the ordered list of output values:

$$T_f = (f(0), f(1), \ldots, f(2^n - 1)). \tag{2}$$

The *algebraic normal form* (ANF) of $f$ is the multivariate polynomial

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}_2^n} a_u x^u, \tag{3}$$

where $a_u \in \mathbb{F}_2$ and $x^u = x_1^{u_1} x_2^{u_2} \cdots x_n^{u_n}$ is a *monomial* containing the variables $x_i$ where $u_i = 1$. The degree of the monomial $x^u$ is the number of variables appearing in $x^u$. The *degree* of a Boolean function, denoted $deg(f)$, is the highest degree among the monomials appearing in its ANF.

The *Walsh-Hadamard transform* of a Boolean function $f$ is the integer-valued function defined as

$$W_f(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\alpha \cdot x}, \alpha \in \mathbb{F}_2^n. \tag{4}$$

The vector $[W_f(0), \ldots, W_f(2^n - 1)]$ is called the *Walsh spectrum* of $f$. The *autocorrelation* function of a Boolean function $f$ is defined as

$$C_f(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+f(x+\alpha)}, \alpha \in \mathbb{F}_2^n. \tag{5}$$

The vector $[C_f(0), \ldots, C_f(2^n - 1)]$ is called the *autocorrelation spectrum* of $f$.

The vector $\alpha \in \mathbb{F}_2^n$ is a *linear structure* of $f$, if $f(x)+f(x+\alpha)$ is a constant function [14]. In this case, the autocorrelation value $C_f(\alpha)$ becomes either $-2^n$ or $2^n$. The set of linear structures of a Boolean function forms a vector space, whose dimension $d_l(f)$ is called the *linearity dimension* of $f$. The linearity dimension can be computed from the autocorrelation function as follows:

$$d_l(f) = \log_2 \#\{|C_f(\alpha)| = 2^n, \alpha \in \mathbb{F}_2^n\}. \tag{6}$$

Two functions $f, g \in \mathcal{B}_n$ are *affine equivalent* if $f$ can be written as

$$f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{a}) + \mathbf{b}^\top \mathbf{x} + c, \text{ for all } \mathbf{x}, \tag{7}$$

where $A$ is a non-singular $n \times n$ matrix over $\mathbb{F}_2$; $\mathbf{a}, \mathbf{b}$ are column vectors in $\mathbb{F}_2^n$ and $c \in \mathbb{F}_2$. The parameters $T = (A, a, b, c)$ above constitute an *affine transformation* that maps $g$ to $f$. We use $[f]$ to denote the affine equivalence class of the function $f$.

Some of the relevant cryptographic properties of Boolean functions such as degree, multiplicative complexity, linearity dimension, distribution of the absolute values in the Walsh spectrum and in the autocorrelation spectrum are invariant under affine transformations. A method for determining whether two functions are affine equivalent is given in [15].

In 1972, Berlekamp and Welch showed that $\mathcal{B}_5$ has 48 equivalence classes [16]. For $n = 6$, Maiorana [17] proved that there are 150 357 equivalence classes, which was later independently verified by Fuller [15] and by Braeken et al. [18]. Hou [19] showed that $\mathcal{B}_7$ has approximately $2^{65.78}$ classes.

The effect of an affine transformation on a Boolean functions autocorrelation spectrum is known and explained in the following proposition.

**Proposition 1** *[20] If $g \in \mathcal{B}_n$ can be transformed to $f \in \mathcal{B}_n$ using the transformation $T = (A, a, b, c)$, then their autocorrelation spectrums are related in the following way:*

$$C_f(\alpha) = (-1)^{\alpha (A^{-1})^\top b} C_g(A\alpha). \tag{8}$$

**Corollary 1** *Let $f \in \mathcal{B}_n$, and let $A$ be an invertible $n\,x\,n$ matrix. If $\{\alpha_i\}_{i=1}^k$ are linear structures of $f$, then $\{A\alpha_i\}_{i=1}^k$ are linear structures of $f(Ax)$.*

**Proposition 2** *Let $f \in \mathcal{B}_n$ and $e_i$ is the all-zero unit vector except the $i$th bit. If $e_i \in \mathrm{F}_2^n$ is a linear structure of $f$, then $f$ can be written as*

$$f(x) = g(x) + cx_i, \tag{9}$$

*where $g \in \mathcal{B}_n$ does not depend on $x_i$ and $c \in \mathrm{F}_2$ satisfies*

$$c = \begin{cases} 0, & \text{if } C_f(e_i) = 2^n, \\ 1, & \text{if } C_f(e_i) = -2^n. \end{cases}$$

*Proof* *Any Boolean function $f \in \mathcal{B}_n$ can be expressed as*

$$f(x) = x_i g_1(x) + g_2(x), \tag{10}$$

*where $g_1, g_2 \in \mathcal{B}_n$ do not depend on the variable $x_i$. Then, one can obtain $f(x + e_i) = (x_i + 1)g_1(x) + g_2(x)$, which leads $f(x) + f(x + e_i) = g_1(x)$. The vector $e_i$ being a linear structure of $f$ implies that $g_1(x)$ is constant. From (10), $g_1(x) = 0$ implies $f(x) = g_2(x)$ and $x_i$ does not appear in the ANF of $f$, and $g_1(x) = 1$ implies $f(x) = x_i + g_2(x)$ and $x_i$ appears as a linear term in the ANF.*

## 2.2 Boolean Circuits

A *Boolean circuit* $C$ with $n$ inputs and $m$ outputs is a directed acyclic graph, where the inputs and the gates are the nodes, and the edges correspond to the Boolean-valued *wires*. The *fanin* and *fanout* of a node is the number of wires going in and out of the node, respectively. The nodes with fanin zero are called the *input nodes* and are labeled with an input variable from $\{\mathtt{x}_1, \ldots, \mathtt{x}_n\}$. The circuits considered in this study only contain gates from the complete basis (AND, XOR, NOT) and have exactly one node with fanout zero (i.e., $m = 1$), which is called the *output node*. For our purposes, we assume AND gates have fan-in two, but XOR gates have arbitrary fan-in $> 0$.

Boolean functions can be partitioned into those $f$ for which $f(0) = 0$ and those $f$ for which $f(0) = 1$. One set can be mapped bijectively into the other by the transformation $g(\mathbf{x}) = f(\mathbf{x})+1$. A function $f(\mathbf{x})$ for which $f(0) = 0$ can be computed by a circuit which is both optimal with respect to multiplicative complexity and has no negations. Thus, without loss of generality, we will only consider circuits that do not have the constant 1 as input.

Each Boolean circuit $C$ with $n$ input nodes computes a Boolean function $f \in B_n$. When a Boolean vector $\mathbf{x} \in \{0, 1\}^n$ is fed to the input nodes, the logic gates compute the function where the output node gets the value $f(\mathbf{x})$.

We use the following notation from [8]:

| | |
|---|---|
| $\mathtt{A}$: | the set of AND gates |
| $\mathtt{B}$: | the set of XOR gates |
| $\mathtt{a}_i$: | ith AND gate of the circuit, $1 \le i \le k$ |
| $\mathtt{b}_i$: | ith XOR gate of the circuit, $1 \le i \le 2k + 1$ |
| $S_i$: | the set of AND gates that are inputs to the $\mathtt{b}_i$ |
| $L_i$ | the set of input nodes to $\mathtt{b}_i$ |

The *canonical form of a circuit* [8] has the following properties:

1. The circuit output is always an XOR gate.
2. The output of AND gate is always an input to an XOR gate.
3. The two inputs of an AND gate are outputs of XOR gates.
4. The inputs of XOR gates are either inputs to the circuit or outputs of AND gates.
5. There are no negation gates.
6. The AND gates are numbered topologically, with no gate being an ancestor of a lower-numbered gate.
7. XOR gates have fanout 1 or zero (for the output gate).
8. The AND gate $\mathtt{a}_i$ has inputs $\mathtt{b}_{2i-1}$ and $\mathtt{b}_{2i}$.

It is easy to verify that any Boolean circuit with $k$ AND gates can be converted into the canonical form with $k$ AND gates and $2k + 1$ XOR gates.

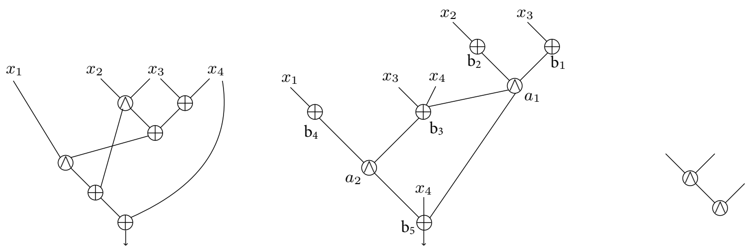
Given a set $V$ of nodes, let $\mathcal{X}_V$ denote the Boolean function computed as $\bigoplus_{v \in V} v$. [1] The output of the $i$-th XOR gate is $F_{\mathtt{b}_i} = \mathcal{X}_{L_i} \oplus \mathcal{X}_{S_i}$, and the output of the $i$-th AND gate is

$$F_{\mathtt{a}_i} = (\mathcal{X}_{L_{2i-1}} \oplus \mathcal{X}_{S_{2i-1}}) \wedge (\mathcal{X}_{L_{2i}} \oplus \mathcal{X}_{S_{2i}}). \tag{11}$$

[1] We abuse notation here, identifying a node with the function it computes.

Given a circuit, the ordered list $(L_1, \ldots, L_{2k+1}, S_1, \ldots, S_{2k+1})$ is called the *trace of the circuit*. The ordered list $[(S_1, S_2), (S_3, S_4) \ldots, (S_{2k-1}, S_{2k})]$ shows the relations between the AND gates, and is called the *topology* of the circuit. The ordered list $(L_1, \ldots, L_{2k+1})$ shows the linear inputs to the XOR gates, and is called the *input to the topology*. For readability, we will be depicting topologies through diagrams rather than as lists of sets.

*Example 1* Let $f \in B_4$ be $f = x_1x_2x_3 + x_1x_3 + x_1x_4 + x_2x_3 + x_4$. A circuit computing $f$, with its canonical form and topology is shown in Figure 1. The trace for that circuit is $(\{x_3\}, \{x_2\}, \{x_3, x_4\}, \{x_1\}, \{x_4\}, \emptyset, \emptyset, \{\mathsf{a}_1\}, \emptyset, \{\mathsf{a}_1, \mathsf{a}_2\})$. The topology of the circuit is $[(\emptyset, \emptyset), (\{a_1\}, \emptyset)]$. The input to the topology is $(\{x_3\}, \{x_2\}, \{x_3, x_4\}, \{x_1\}, \{x_4\})$.



(a) Circuit (b) Canonical form (c) Topology

Fig. 1: Circuit and topology computing $f$.

Using a topology $[(S_1, S_2), (S_3, S_4) \ldots, (S_{2k-1}, S_{2k})]$ with $k$ AND gates, $2^{2k+2}$ new topologies with $k + 1$ AND gates can be constructed by appending $(S_{2k+1}, S_{2k+2}) \subseteq \{\mathsf{a}_1, ..., \mathsf{a}_k\}$ to the original topology. Details of topology construction, and identification of isomorphic topologies are described in [8].

## 3 A New Lower Bound on Multiplicative Complexity

A general lower bound on the multiplicative complexity of Boolean functions is the *degree bound*, which states that the multiplicative complexity of a Boolean function $f$ is at least $deg(f) - 1$ [21]. In this section, we provide a new lower bound on the multiplicative complexity based on the dimension of the Boolean function.

**Definition 1** [13] Let $N_f$ be the number of distinct input variables appearing in the ANF of $f \in B_n$. The dimension of $f$, denoted $dim(f)$ is defined as the smallest number of variables that appear in the ANFs of functions that are affine equivalent to $f$;

$$dim(f) = \min_{g \in [f]} N_g. \tag{12}$$

We can now relate the dimension of a function to its linearity dimension, at the same time providing an efficient way to compute it.

**Lemma 1** *Let $f \in \mathcal{B}_n$. If $dim(f) = n$, $d_l(f) = 0$.*

*Proof* *Let $dim(f) = n$. If $d_l(f) > 0$, then $f$ has a non-zero linear structure $\alpha$. From linear algebra, we know there exists an invertible $n\, x\, n$ matrix $A$ that satisfies $A\alpha = e_n$. By Corollary 1, $e_n$ is a linear structure of $f(Ax)$, and by Proposition 2, $f(Ax)$ is independent of $x_n$ or $x_n$ only appears linearly in the ANF of $f(Ax)$. Then one of $f(Ax)$ or $f(Ax) + x_n$ does not depend on $x_n$. This contradicts $dim(f) = n$. Thus $d_l(f) = 0$.*

**Theorem 1** *Let $f \in \mathcal{B}_n$. Then $dim(f) + d_l(f) = n$.*

*Proof* *If $dim(f) = n$ the result follows from Lemma 1. If $\delta = dim(f) < n$, then $f$ is affine equivalent to a function $g \in B_n$ with exactly $\delta$ variables appearing in its ANF. Without loss of generality, assume the variables $x_1, \dots, x_\delta$ appear in the ANF of $g$. Let $g' \in \mathcal{B}_\delta$ satisfying $g'(x_1, \dots, x_\delta) = g(x_1, \dots, x_n)$ for all $x \in F_2^n$. Since $dim(g') = \delta$, Lemma 1 implies $d_l(g') = 0$ (i.e., there are no linear structures of $g'$). Since the output of $g \in B_n$ is independent of the values of the variables $x_{\delta+1}, x_{\delta+2}, \dots,$ and $x_n$, and there are no other linear structures of $g$ based on the first $\delta$ variables, $\{e_i\}_{i=\delta+1}^n$ is a basis for the linear structures of $g$. Then the linearity dimension is $n - \delta$ and $dim(f) + d_l(f) = n$ holds.*

**Theorem 2** *The MC of a Boolean function $f \in \mathcal{B}_n$ is at least $\lceil dim(f)/2 \rceil$.*

*Proof* *Let $f$ be an arbitrary Boolean function and $C_\wedge(f) = k$. There exists a circuit implementing $f$ with $k$ AND gates. The topology of the circuit with $k$ AND gates has $2k$ linear inputs. Any set of $2k$ linear functions on $n > 2k$ variables can be mapped to functions having at most $2k$ variables by an affine transformation. Therefore, $dim(f) \leq 2C_\wedge(f)$, which implies that the multiplicative complexity of $f$ is greater than or equal to $\lceil dim(f)/2 \rceil$.*

Note that the dimension bound is tighter than the degree bound for multiplicative complexity when $deg(f) \leq \lceil dim(f)/2 \rceil$.

*Example 2* Let $f$ be the symmetric Boolean function $\Sigma_4^8$, i.e., $f = x_1x_2x_3x_4 + \ldots + x_5x_6x_7x_8$. According to the degree bound, the $C_\wedge(f) \geq 3$. By Theorem 2, $C_\wedge(f) \geq 4$.

## 4 Boolean Functions with Multiplicative Complexity *k*

The characterization of Boolean functions with respect to MC can be realized by working on the equivalence classes rather than examining functions individually, since MC is invariant under affine transformation. In this section, we propose an iterative method to construct the list of all affine equivalence classes of Boolean functions with a given MC $k$.

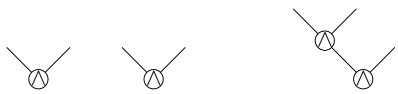
The method first constructs topologies with $i = 1, \ldots, k$ AND gates in an iterative manner. At $i$th step, topologies with $i$ AND gates are constructed as described in [8]. Then, the topologies are evaluated by supplying linear function inputs $X = (L_1, \ldots, L_{2i})$, with dimension at most $2i$. This process generates a set of Boolean functions with MC at most $i$. The functions whose MC is less than $i$ are omitted from the list, by checking whether they belong to the equivalence classes with MC less than $i$. The remaining set of functions are processed to make sure that exactly one function from each equivalence class remains in the set. These functions become the representatives of their classes, and are stored with an associated MC value of $i$. The method is repeated until $i = k$. The choice of the representatives is arbitrary and does not have any affect on the results.

### 4.1 Equivalence Classes with MC 1 and 2

As previously shown in [12,13], Boolean functions with MC 1 are affine equivalent to $x_1x_2$ and can be generated using the topology given in Fig 2.

Fig. 2: Topology with 1 AND gate

There are two topologies with 2 AND gates as illustrated in Figure 3. Find et al. [13] showed that a Boolean function with MC 2 is affine equivalent to exactly one of these following three functions $x_1x_2x_3$, $x_1x_2x_3 + x_1x_4$ and $x_1x_2 + x_3x_4$.



(a) Topology 1 (b) Topology 2

Fig. 3: Topologies with 2 AND gates.

### 4.2 Equivalence Classes with MC 3

According to Theorem 2, Boolean functions with MC 3 can have up to 6 independent inputs. The MC distribution of all 150 357 affine equivalence classes on 6-variables is given in [8]. Figure 4 shows the graphical representations of the topologies with 3 AND gates.

Evaluating topologies with linear inputs having dimension up to 6 gives the exhaustive list of equivalence classes having MC 3 as shown in Table 1.

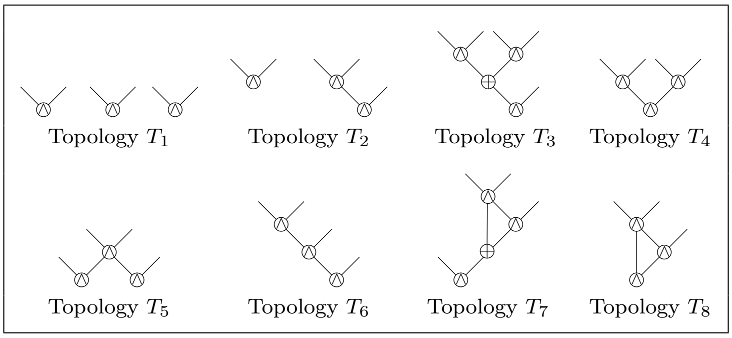



Fig. 4: Topologies with 3 AND gates

There are three equivalence classes with dimension 4, and all of these classes can be generated by either of the topologies $T_4, T_6, T_7$ and $T_8$. For dimension 5 and 6, there are 14 and 7 classes, respectively.

### 4.3 Equivalence Classes with MC 4

According to Theorem 2, Boolean functions with MC 4 can have up to 8 independent inputs. Different from the MC 3 case, it is not feasible to exhaustively list the equivalence classes for Boolean functions with 7 and 8 inputs. This makes it less efficient sometimes to decide whether two functions are in the same equivalence class or not for those cases.

After evaluating 84 topologies with 4 AND gates, 26 classes with dimension 5, 888 classes with dimension 6, 321 classes with dimension 7, and 42 classes with dimension 8 were obtained. The complete list of affine equivalence classes with MC 4 is published on [22].

### 4.4 Number of Boolean functions with MC $\leq 4$

Let $\lambda(n, k)$ be the number of $n$-variable Boolean functions with MC $k$. Boyar et al. [6] showed that $\lambda(n, k) \leq 2^{k^2+2k+2kn+n+1}$. The exact formulas for $k = 1, 2$ are given in [12] and [13], respectively.

The size of an equivalence class for a given $f \in B_n$ is calculated using the techniques provided in Corollary 4.8 in [23]. Table 1 provides the size of the equivalence classes with MC 3, defined in $B_{dim(f)}$. For example, the size 512 of the equivalence class $\mathrm{x}_1\mathrm{x}_2\mathrm{x}_3\mathrm{x}_4$ is defined in $B_4$.

**Definition 2** Let $f \in \mathcal{B}_\ell$. The *embedding* of $f$ in $\mathcal{B}_n$, $n \geq \ell$ is defined as the $n$-variable Boolean function that satisfies $f_n(x_1, \ldots, x_\ell, x_{\ell+1}, \ldots, x_n) = f(x_1, \ldots, x_\ell)$.

The following theorem proved in [13] determines the size of equivalence classes when a Boolean function is embedded in higher number of variables.

| **Dimension = 4** | | |
|---|---|---|
| Representative | Size of the class | Generated by |
| $x_1x_2x_3x_4$ | 512 | $T_4$,$T_6$,$T_7$,$T_8$ |
| $x_1x_2$+$x_1x_2x_3x_4$ | 17 920 | $T_4$,$T_6$,$T_7$,$T_8$ |
| $x_2x_3$+$x_1x_4$+$x_1x_2x_3x_4$ | 14 336 | $T_4$,$T_6$,$T_7$,$T_8$ |
| **Dimension = 5** | | |
| Representative | Size of the class | Generated by |
| $x_1x_2x_3x_4$+$x_1x_2x_5$ | 2 222 080 | $T_4$,$T_6$,$T_7$,$T_8$ |
| $x_1x_3x_4$+$x_1x_2x_5$ | 1 777 664 | $T_3$ |
| $x_2x_3$+$x_1x_2x_3x_4$+$x_2x_3x_5$+$x_1x_4x_5$ | 28 442 624 | $T_4$,$T_8$ |
| $x_1x_2x_3x_4$+$x_1x_5$ | 3 809 280 | $T_6$,$T_7$ |
| $x_3x_4$+$x_1x_2x_3x_4$+$x_1x_5$+$x_1x_2x_5$ | 106 659 840 | $T_6$,$T_7$ |
| $x_1x_2x_3$+$x_4x_5$ | 5 079 040 | $T_2$,$T_5$ |
| $x_1x_2x_3x_4$+$x_1x_5$+$x_1x_2x_5$ | 26 664 960 | $T_6$,$T_7$ |
| $x_1x_3$+$x_1x_2x_3x_4$+$x_1x_2x_5$ | 19 998 720 | $T_6$,$T_7$ |
| $x_3x_4$+$x_1x_3x_4$+$x_1x_2x_5$ | 17 776 640 | $T_3$,$T_4$,$T_8$ |
| $x_1x_2x_3$+$x_2x_4$+$x_1x_5$ | 3 333 120 | $T_2$,$T_3$,$T_5$,$T_6$,$T_7$ |
| $x_2x_3$+$x_1x_2x_3x_4$+$x_1x_5$ | 26 664 960 | $T_6$,$T_7$ |
| $x_1x_2x_3x_4$+$x_2x_3x_5$+$x_1x_4x_5$ | 284 426 240 | $T_4$,$T_8$ |
| $x_1x_2x_3x_4$+$x_1x_2x_5$+$x_3x_5$ | 213 319 680 | $T_4$,$T_8$ |
| $x_3x_4$+$x_1x_2x_3x_4$+$x_1x_2x_5$ | 35 553 280 | $T_4$,$T_6$,$T_7$,$T_8$ |
| **Dimension = 6** | | |
| Representative | Size of the class | Generated by |
| $x_1x_2x_3x_4$+$x_3x_4x_5$+$x_1x_2x_6$+$x_5x_6$ | 143 350 824 960 | $T_4$,$T_8$ |
| $x_3x_4$+$x_1x_2x_3x_4$+$x_1x_2x_5$+$x_1x_6$ | 26 878 279 680 | $T_6$,$T_7$ |
| $x_3x_4$+$x_1x_3x_4$+$x_1x_2x_5$+$x_1x_6$ | 2 239 856 640 | $T_3$ |
| $x_1x_3x_4$+$x_1x_2x_5$+$x_1x_6$ | 223 985 664 | $T_3$ |
| $x_3x_4$+$x_2x_5$+$x_1x_6$ | 1 777 664 | $T_1$ |
| $x_1x_2x_3x_4$+$x_1x_2x_5$+$x_1x_6$ | 6 719 569 920 | $T_6$,$T_7$ |
| $x_1x_2x_3$+$x_4x_5$+$x_1x_6$ | 4 479 713 280 | $T_2$,$T_5$ |

Table 1: The list of affine equivalence classes with MC 3. The size of each class (i.e., the number of functions in the class) is given for the dimension it belongs to.

**Theorem 3** *[13] Let $f \in \mathcal{B}_\ell$, with $dim(f) = \ell$. Let $f_n$ be the embedding of $f$ in $\mathcal{B}_n$, $n \geq \ell$. The size of the equivalence class $[f_n]$ is*

$$|[f_n]| = 2^{n-\ell}|[f_\ell]| \prod_{i=0}^{\ell-1} \frac{2^n - 2^i}{2^\ell - 2^i}. \tag{13}$$

Let $\beta(d,k)$ be the sum of sizes of equivalence classes with multiplicative complexity $k$ and dimension $d$. For example, $\beta(3,4) = 32\,768$ is the total of the size of the equivalence classes $[x_1x_2x_3x_4]$, $[x_1x_2{+}x_1x_2x_3x_4]$ and $[x_2x_3{+}x_1x_4{+}x_1x_2x_3x_4]$. Then, using the Theorem 3, the number of Boolean functions with MC 3 in $B_n$ is equal to the sum of the sizes of each equivalence class embedded in $B_n$. This number can be calculated as

$$\lambda(n,3) = \sum_{d=4}^{6} \left( 2^{n-d} \prod_{i=0}^{d-1} \frac{2^n - 2^i}{2^d - 2^i} \beta(d,3) \right) \tag{14}$$

where

$$\begin{aligned}\beta(4,3) &= 32\,768,\\ \beta(5,3) &= 775\,728\,128,\\ \beta(6,3) &= 183\,894\,007\,808.\end{aligned}$$

Similarly, the number of Boolean functions with MC 4 can be calculated as

$$\lambda(n,4) = \sum_{d=5}^{8}\left(2^{n-d}\prod_{i=0}^{d-1}\frac{2^n-2^i}{2^d-2^i}\beta(d,4)\right) \tag{15}$$

where

$$\begin{aligned}\beta(5,4) &= 3\,515\,396\,096,\\ \beta(6,4) &= 7\,944\,313\,921\,970\,176,\\ \beta(7,4) &= 8\,217\,135\,092\,528\,316\,416,\\ \beta(8,4) &= 5\,502\,415\,308\,673\,798\,144.\end{aligned}$$

## 5 Constructing circuits for Boolean functions with MC 4 or less

The techniques defined in [8] and the exhaustive list of affine equivalence classes having MC up to 4 allow us to construct AND-optimal circuits for Boolean functions that have MC up to 4, independent of the number of variables they are defined on.

Given a Boolean function $f \in B_n$, we first compute $dim(f)$. If $dim(f) > 8$, we conclude that $f$ does not belong to any of the equivalence classes with MC less than or equal to 4, hence $C_\wedge(f) > 4$. Otherwise, we determine the equivalence class that it belongs to among the 1305 equivalence classes having MC up to 4. Next, we find the affine transformation between the representative of the class and $f$. Applying the same transformation to the linear inputs of the topology implementing the representative, a circuit that implements $f$ with a minimal number of AND gates can be obtained.

As an optimization, instead of working on $B_n$, the number of variables in $f$ can be reduced to $dim(f)$ by an affine transformation. Then, the algorithms for identifying the equivalence class of $f$ and finding the transformation between $f$ and the representative of its class can be performed in $B_{dim(f)}$ more efficiently.

## 6 Conclusion and Future Work

The relation between dimension and multiplicative complexity of Boolean functions enabled us to exhaustively list all affine equivalence classes with MC 3 and 4. The MC distribution of Boolean functions with dimension up to 6 were provided in [8]. In this work, we showed that there are exactly 24 equivalence classes for MC 3. For MC 4, in addition to the classes found in

[8], we determined the equivalences classes having dimension 7 and 8, which makes a total of 1277 equivalence classes. Table 2 provides the number of affine equivalence classes with respect to MC and dimension. The contributions of this paper were written in bold. Note that it is easy to see that for the shaded cells the number of affine equivalence classes is zero. We also provide a closed formula for the number of $n$-variable functions with MC 3 and 4.

| MC | Dimension | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Total |
| 1 | 1 | | | | | | | | | | | 1 |
| 2 | | 1 | 2 | | | | | | | | | 3 |
| **3** | | | 3 | 14 | 7 | | | | | | | **24** |
| **4** | | | | 26 | 888 | **321** | **42** | | | | | **1277** |
| 5 | | | | | 148483 | ? | ? | ? | **575** | | | ? |
| 6 | | | | | 931 | ? | ? | ? | ? | ? | ? | ? |

Table 2: The number of affine equivalence classes with respect to MC and dimension.

For each equivalence class representative, an AND-optimal circuit has been constructed. This allows us to construct optimal circuits for any Boolean function with MC up to 4 independent of the number of variables the functions are defined on. The method can also be used to determine that a function has MC greater than 4, if it does not belong to any of the equivalence classes with MC 4 or less.

The table also includes the known cases for $n = 5, 6$. The identification of classes with MC 5 is still in progress. The techniques require more computation resources as the dimension and MC increase. Different techniques or optimizations may be necessary to find the missing values in the table.

## References


1. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 309–325. ACM, 2012.
2. Joan Boyar, Ivan Damgård, and René Peralta. Short Non-Interactive Cryptographic Proofs. *J. Cryptology*, 13(4):449–472, 2000.
3. Vladimir Kolesnikov and Thomas Schneider. Improved Garbled Circuit: Free XOR Gates and Applications. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 486–498. Springer, 2008.
4. Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. In *CRYPTO (1)*, volume 10991 of *Lecture Notes in Computer Science*, pages 662–692. Springer, 2018.

5. Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.
6. Joan Boyar, René Peralta, and Denis Pochuev. On the Multiplicative Complexity of Boolean Functions over the Basis ($\wedge$, $\oplus$, 1). *Theor. Comput. Sci.*, 235(1):43–57, 2000.
7. Meltem Sönmez Turan and René Peralta. *The Multiplicative Complexity of Boolean Functions on Four and Five Variables*, pages 21–33. Springer International Publishing, Cham, 2015.
8. Çağdaş Çalık, Meltem Sönmez Turan, and René Peralta. The multiplicative complexity of 6-variable boolean functions. *Cryptogr. Commun.*, 11(1):93–107, 2019.
9. Magnus Gausdal Find. On the Complexity of Computing Two Nonlinearity Measures. In *Computer Science - Theory and Applications - 9th International Computer Science Symposium in Russia, CSR 2014, Moscow, Russia, June 7-11, 2014. Proceedings*, pages 167–175, 2014.
10. Roland Mirwald and Claus-Peter Schnorr. The Multiplicative Complexity of Quadratic Boolean Forms. *Theor. Comput. Sci.*, 102(2):307–328, 1992.
11. Luís T. A. N. Brandão, Çağdaş Çalık, Meltem Sönmez Turan, and René Peralta. Upper bounds on the multiplicative complexity of symmetric boolean functions. *Cryptogr. Commun.*, 11(6):1339–1362, 2019.
12. M. J. Fischer and R. Peralta. Counting Predicates of Conjunctive Complexity One. *Yale Technical Report 1222*, February 2002.
13. Magnus Gausdal Find, Daniel Smith-Tone, and Meltem Sönmez Turan. The Number of Boolean Functions with Multiplicative Complexity 2. *IJICoT*, 4(4):222–236, 2017.
14. Kaisa Nyberg. On the Construction of Highly Nonlinear Permutations. In Rainer A. Rueppel, editor, *Advances in Cryptology - EUROCRYPT '92, Workshop on the Theory and Application of of Cryptographic Techniques, Balatonfüred, Hungary, May 24-28, 1992, Proceedings*, volume 658 of *Lecture Notes in Computer Science*, pages 92–98. Springer, 1992.
15. Joanne Elizabeth Fuller. *Analysis of Affine Equivalent Boolean Functions for Cryptography.* PhD thesis, Queensland University of Technology, 2003.
16. Elwyn R. Berlekamp and Lloyd R. Welch. Weight Distributions of the Cosets of the (32, 6) Reed-Muller Code. *IEEE Transactions on Information Theory*, 18(1):203–207, 1972.
17. James A. Maiorana. A classification of the cosets of the Reed-Muller code R(1,6). *Mathematics of Computation*, 57(195):403–414, 1991.
18. An Braeken, Yuri L. Borissov, Svetla Nikova, and Bart Preneel. Classification of Boolean Functions of 6 Variables or Less with Respect to Some Cryptographic Properties. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 324–334. Springer, 2005.
19. Xiang-Dong Hou. AGL($m$,2) acting on R($r$, $m$)/R($s$,$m$). *Journal of Algebra*, 171(3):927–938, 1995.
20. Bart Preneel. *Analysis and Design of Cryptographic Hash Functions.* PhD thesis, Katholieke Universiteit Leuven, 1993.
21. Claus-Peter Schnorr. The Multiplicative Complexity of Boolean Functions. In *AAECC*, pages 45–58, 1988.
22. NIST Computer Security Division. *Circuit Complexity Project Repository,* `https://github.com/usnistgov/Circuits/`.
23. Erdener Uyan. *Analysis of Boolean Functions with respect to Walsh Spectrum.* PhD thesis, Middle East Technical University, 2013.