The Number of Boolean Functions with Multiplicative Complexity 2

Magnus Gausdal Find[1], Daniel Smith-Tone[1,2], and
Meltem Sönmez Turan [1,3]

[1] National Institute of Standards and Technology,
[2] University of Louisville
[3] Dakota Consulting Inc.
{magnus.find,daniel.smith,meltem.turan}@nist.gov

Abstract. Multiplicative complexity is a complexity measure defined as the minimum number of AND gates required to implement a given primitive by a circuit over the basis (AND, XOR, NOT). Implementations of ciphers with a small number of AND gates are preferred in protocols for fully homomorphic encryption, multi-party computation and zero-knowledge proofs. In 2002, Fischer and Peralta [12] showed that the number of n-variable Boolean functions with multiplicative complexity one equals $2\binom{2^n}{3}$. In this paper, we study Boolean functions with multiplicative complexity 2. By characterizing the structure of these functions in terms of affine equivalence relations, we provide a closed form formula for the number of Boolean functions with multiplicative complexity 2.
Keywords: Affine equivalence; Boolean functions; Cryptography; Multiplicative complexity; Self-mappings

1 Introduction

Multiplicative complexity is a complexity measure defined as the minimum number of AND gates required to implement a given primitive by a circuit over the basis (AND, XOR, NOT). In recent years, the relationships between multiplicative complexity and cryptography has been pointed out in several studies:

Multiplicative Complexity and Cryptography Many protocols for fully homomorphic encryption (e.g., [1]), multi-party computation (e.g., [2]) and zero-knowledge proofs of knowledge (e.g., [3]) operate on the circuit representation of a function in a gate-by-gate manner. In these and many other protocols, it is the case that processing AND gates is more expensive than processing XOR gates. We refer to [4] and the references therein for a comprehensive list of examples. Moreover, efficiency of some of the countermeasures against side channel attacks is related to the number of AND gates in the implementation. For example, complexity of the higher-order masking schemes for s-boxes mainly depends on the *masking complexity* which is defined as the minimal number of nonlinear multiplications required to evaluate a polynomial representation of an (n, m)-bit s-box over $\mathbb{F}_{2^n}$ [**?**]. In Eurocrypt'15, Albrecht et al. [4] used this motivation to design the family of block ciphers LowMC. On the other hand, having a certain multiplicative complexity is essential for security, e.g., Boyar et al. [6] showed that a cryptographic hash function must have a certain multiplicative complexity to be collision resistant.

Multiplicative Complexity and Circuit Design Determining the multiplicative complexity of a given function is computationally intractable, even for functions with a small number of variables. For general n, it is known that under standard cryptographic assumptions it is not possible to compute the multiplicative complexity in polynomial time in the length of the truth table [7]. The multiplicative complexity of a random n-variable Boolean function is at least $2^{n/2} - O(n)$ with high probability [8]. In 2010, Boyar et al. [9] proposed a two-stage heuristic method to minimize the gate complexity of Boolean circuits. In the first stage, the heuristic minimizes the number of AND gates required to implement the function, and then in the second stage, the linear components are optimized. Using this method, they constructed efficient circuits for the AES S-box over the basis (AND, XOR, NOT). In 2014, Turan and Peralta [10] studied the multiplicative complexity of five variable Boolean functions and showed that any five variable Boolean function can be implemented with at most four AND gates. Also in 2014, Zajac and Jókay [11] showed that any bijective 4×4 Sbox can be implemented with at most five AND gates.

Previous work In [13], Mirwald and Schnorr studied the multiplicative complexity of quadratic forms and showed that multiplicative complexity of a Boolean function f is M iff f is isomorphic to the canonical form $\bigoplus_{i=1}^{n} x_{2i-1}x_{2i}$. They also showed that the fraction of $2n$-ary forms that has multiplicative complexity n is $\prod_{k=0}^{n-1}(1 -$

$2^{-(2n-2k-1)}$). Boyar et al. [8] showed that the number of functions with multiplicative complexity M is at most $2^{M^2+2M+2Mn+n+1}$. For large values of n and M, this bound is essentially tight [8], but it is unclear to what extent this is true for small constant values of M. In 2002, Fischer and Peralta [12] showed that there are precisely $2\binom{2^n}{3}$ Boolean functions on n variables with multiplicative complexity one. Their result was based on properties on polynomial representations of such Boolean functions and the authors mention that this technique is unlikely to generalize to the case of even multiplicative complexity two.

This Paper In this work, we develop an alternative approach to count the number of Boolean functions with a given multiplicative complexity. Our approach relies on affine equivalence relations. First, we find the exhaustive list of affine equivalence classes with a given multiplicative complexity, and then, we obtain the number of functions with a certain multiplicative complexity by summing the number of Boolean functions within each equivalence class. From a theoretical perspective this gives an algorithm, that given as input M, outputs a formula for the number of functions in n variables with multiplicative complexity M. Using this approach, we reprove the result of Fischer and Peralta [12], and extend the result to show that the number of Boolean functions with multiplicative complexity exactly 2 equals:

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

We remark that this is asymptotically a factor of 5921 smaller than the bound from [8].

The organization of the paper is as follows. Section 2 gives definitions and preliminary information about Boolean functions and multiplicative complexity. Section 3 discusses affine transformations and equivalence classes. Section 4 studies the Boolean functions with multiplicative complexity one. Section 5 provides the equivalence classes of Boolean functions with multiplicative complexity two. Section 6 concludes the paper, with some future directions.

2 Preliminaries

Let $\mathbb{F}_2$ be the binary field. 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. A Boolean function $f \in \mathcal{B}_n$ can be represented uniquely by the list of output values for each input $T_f = (f(0,\ldots,0), f(0,\ldots,0,1),\ldots,f(1,\ldots,1))$. This list is called the *truth table* (representation) of f. Since the truth table has length 2^n and there are two possibilities for each, $|\mathcal{B}_n| = 2^{2^n}$. Another way of representing a Boolean function $f \in \mathcal{B}_n$ is by the unique multilinear polynomial called the *algebraic normal form* (ANF)

$$f(x_1,\ldots,x_n) = \sum_{u \in \mathbb{F}_2^n} a_u x^u, \tag{1}$$

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 d_f, is the highest degree of monomials occurring in its ANF. Functions with degree 2 are called quadratic and functions with degree 1 are called affine.

The *multiplicative complexity* of a Boolean function f is the minimum number of AND gates (multiplications in $\mathbb{F}_2$) that are sufficient to evaluate the function over the basis (AND, XOR, NOT) where all gates have fanin 2. It is known that a function with degree d has multiplicative complexity at least $d-1$ [14]. This bound is called the *degree bound.*

3 Affine Transformations and Equivalence Classes

Definition 1 *[15] A map $S \colon \mathcal{B}_n \to \mathcal{B}_n$ is called an* affine transformation *if $g \overset{S}{\mapsto} f$ is defined by*

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

where A is a non-singular $n \times n$ matrix over $\mathbb{F}_2$; $\mathbf{a}, \mathbf{b}, \mathbf{x}$ *are column vectors in $\mathbb{F}_2^n$ and $c \in \mathbb{F}_2$.*

An affine transformation can be characterized by the values of $A, \mathbf{a}, \mathbf{b}, c$, and we represent the affine transformation from $f(\mathbf{x})$ to $f(A\mathbf{x} + \mathbf{a}) + \mathbf{b}^\top \mathbf{x} + c$ using the tuple $S = (A, \mathbf{a}, \mathbf{b}, c)$.

Directly from the definition of an affine transformation, it follows that the relation

$$R = \{(f, g) |\ \exists \text{ an affine transformation from } f \text{ to } g\},$$

is an equivalence relation on $\mathcal{B}_n$. This relation imposes equivalence classes on $\mathcal{B}_n$, and two functions in the same class are said to be *affine equivalent*. An algorithm to determine whether two functions are equivalent is given in [16].

By counting the number of choices of the A, $\mathbf{a}$, $\mathbf{b}$, and c from Definition 1, we get that for all $n \in \mathbb{N}$, the total number of distinct affine transformations applicable to any given function $f \in \mathcal{B}_n$ is

$$\tau_n = 2^{2n+1} \prod_{i=0}^{n-1} (2^n - 2^i).$$

It was shown in 1972 by Berlekamp and Welch that $\mathcal{B}_5$ has 48 equivalence classes [15]. Maiorana [17] proved that $\mathcal{B}_6$ has 150 357 equivalence classes. This was independently verified by Fuller [16] and Braeken et al. [18]. It was shown by Hou [19] that $\mathcal{B}_7$ has 63 379 147 320 777 408 548($\approx 2^{65.78}$) classes. See Table 1 for the equivalence classes with n=2,3,4 variables.

It should be noted that multiplicative complexity is *affine invariant*, i.e., the multiplicative complexity of a Boolean function does not change after applying an affine transformation to the function. Hence functions in the same equivalence class all have the same mulitiplicative complexity.

3.1 Self-Mappings

In this section we establish a few facts on a particular kind of affine transformations, called *self-mappings*.

Definition 2 *[16] A self mapping of $f \in \mathcal{B}_n$ is an affine transformation such that $f(\mathbf{x}) = f(A\mathbf{x}+\mathbf{a})+\mathbf{b}^\top\mathbf{x}+c$, where A is a non-singular $n \times n$ matrix over $\mathbb{F}_2$; $\mathbf{a}, \mathbf{b}, \mathbf{x}$ are column vectors in $\mathbb{F}_2^n$ and $c \in \mathbb{F}_2$.*

n	Equivalence Class $[f]$	$dim(f)$	$\vert[f]\vert$	# Self Mappings	# Affine Transformations
2	$[x_1]$	1	8	24	192
	$[x_1x_2]$	2	8	24	
3	$[x_1]$	1	16	1344	21504
	$[x_1x_2]$	2	112	192	
	$[x_1x_2x_3]$	3	128	168	
4	$[x_1]$	1	32	322 560	10321920
	$[x_1x_2]$	2	1120	9216	
	$[x_1x_2x_3]$	3	3840	2688	
	$[x_1x_2 + x_3x_4]$	4	896	11 520	
	$[x_1x_2x_3 + x_1x_4]$	4	26 880	384	
	$[x_1x_2x_3x_4]$	4	512	20 160	
	$[x_1x_2x_3x_4 + x_1x_2]$	4	17920	579	
	$[x_1x_2x_3x_4 + x_1x_2 + x_3x_4]$	4	14336	720	

Table 1. Equivalence classes for $n = 2, 3, 4$.

The collection of affine transformations forms a group $\mathcal{A}_n$ under the operation $\otimes$ defined by the composition of the affine transformations $(A_1, \mathbf{a}_1, \mathbf{b}_1, c_1)$ and $(A_2, \mathbf{a}_2, \mathbf{b}_2, c_2) \in A_n$. The group operation $\otimes$ can be expressed as

$$(A_1, \mathbf{a}_1, \mathbf{b}_1, c_1) \otimes (A_2, \mathbf{a}_2, \mathbf{b}_2, c_2) = (A_2A_1, A_2\mathbf{a}_1 + \mathbf{a}_2, A_1^\top \mathbf{b_2} + \mathbf{b}_1, \mathbf{b}_2^\top \mathbf{a}_1 + c_1 + c_2).$$

Let $\Theta(f)$ be the set of self-mappings of $f \in \mathcal{B}_n$. We first remark that $\Theta(f)$ is closed under the group operation $\otimes$. This follows from the fact that $\otimes$ corresponds to composition of affine transformations. For every pair of self-mappings (S_1, S_2), $(S_1 \otimes S_2)(f(\mathbf{x})) = S_2(S_1(f(\mathbf{x}))) = S_2((f\mathbf{x})) = f(\mathbf{x})$. Since $\Theta(f)$ is finite, $\Theta(f)$ forms a subgroup of $\mathcal{A}_n$.

3.2 Size of Equivalence Classes

By the Orbit-Stabilizer Theorem, the size of the equivalence class $[f]$, $f \in \mathcal{B}_n$, is given by

$$|[f]| = \frac{\tau_n}{\theta(f, n)}, \tag{2}$$

where τ_n is the number of affine transformations in A_n and $\theta(f, n)$ is the number of self mappings of $f \in \mathcal{B}_n$.

Definition 3 *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 $f_n(x_1, \ldots, x_\ell, x_{\ell+1}, \ldots, x_n) = f(x_1, \ldots, x_\ell)$.*

For the purposes of this paper, we are interested in determining the size of the equivalence class $[f_n]$, $f \in \mathcal{B}_n$, given the size of $[f]$, $f \in \mathcal{B}_\ell$, $\ell < n$, where f_n is the embedding of f in $\mathcal{B}_n$.

Lemma 4 *Let $f, g \in \mathcal{B}_\ell$. Let f_n and g_n be the embedding of f and g in $\mathcal{B}_n$, $n \geq \ell$, respectively. Then f and g are affine equivalent if and only if f_n and g_n are affine equivalent.*

Proof. Suppose that f and g are affine equivalent. Then there exists an affine transformation $S = (A, \mathbf{a}, \mathbf{b}, c)$ such that $S(f) = g$. Consider the affine transformation

$$\tilde{S} = \left(\begin{bmatrix} A & \mathit{0} \\ \mathit{0} & I \end{bmatrix}, \begin{bmatrix} \mathbf{a} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, c \right),$$

where A is $\ell \times \ell$, I is $(n-\ell) \times (n-\ell)$, and $\mathbf{a}$ and $\mathbf{b}$ are ℓ-dimensional. It is obvious that for all $\mathbf{x} \in \mathbb{F}_2^n$ with $\mathbf{x} = \mathbf{x_1} || \mathbf{x_2}$ where $\mathbf{x_1} \in \mathbb{F}_2^\ell$ and $\mathbf{x_2} \in \mathbb{F}_2^{n-\ell}$ that

$$\begin{aligned} \tilde{S}(f_n)(\mathbf{x}) &= f_n\left(\begin{bmatrix} A & \mathit{0} \\ \mathit{0} & I \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a} \\ \mathbf{0} \end{bmatrix} \right) + \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + c \\ &= f_n((A\mathbf{x_1} + \mathbf{a})\,||\mathbf{x_2}) + \mathbf{b}^\top \mathbf{x_1} + c \\ &= f(A\mathbf{x_1} + \mathbf{a}) + \mathbf{b}^\top \mathbf{x_1} + c \\ &= g(\mathbf{x_1}) \\ &= g_n(\mathbf{x}). \end{aligned}$$

Thus f_n and g_n are affine equivalent.

For the converse, suppose that f_n and g_n are affine equivalent. Then there exists an affine transformation $S \in \mathcal{A}_n$ such that $S(f_n) = g_n$. We may write

$$S = \left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}, \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}, \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix}, c \right),$$

where A is $\ell \times \ell$, B is $\ell \times (n-\ell)$, C is $(n-\ell) \times \ell$, D is $(n-\ell) \times (n-\ell)$, $\mathbf{a_1}$ and $\mathbf{b_1}$ are ℓ-dimensional, and $\mathbf{a_2}$ and $\mathbf{b_2}$ are $(n-\ell)$-dimensional. Let $\mathbf{x_1} = \begin{bmatrix} x_1 \cdots x_\ell \end{bmatrix}^\top$ and $\mathbf{x_2} = \begin{bmatrix} x_{\ell+1} \cdots x_n \end{bmatrix}^\top$. The variables $x_{\ell+1}, \ldots, x_n$ do not occur in the ANF of g_n, so if the variables occur in

$$f_n\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}\right),$$

they must have been added to the expression by the affine transformation and are therefore linear in $\mathbf{x_2}$. Thus

$$f_n\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}\right) = f_n\left(\begin{bmatrix} A & \mathit{0} \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}\right) + \begin{bmatrix} \mathbf{0} \; \mathbf{b_2'} \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix}.$$

Adding $\begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + c$ to both sides of this equation we obtain on the left hand side g_n and on the right hand side

$$f_n\left(\begin{bmatrix} A & \mathit{0} \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}\right) + \begin{bmatrix} \mathbf{b_1} \\ (\mathbf{b_2} + \mathbf{b_2'}) \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + c.$$

Since the output of f_n doesn't involve the variables $x_{\ell+1}, \ldots, x_n$, we learn that $\mathbf{b_2'} = \mathbf{b_2}$. Clearly A is of full rank. Then for all $\mathbf{x} = \mathbf{x_1}||\mathbf{x_2}$,

$$\begin{aligned} g(\mathbf{x_1}) = g_n(\mathbf{x_1}||\mathbf{x_2}) &= f_n\left(\begin{bmatrix} A & \mathit{0} \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}\right) + \begin{bmatrix} \mathbf{b_1} \\ \mathbf{0} \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + c \\ &= f_n([A\mathbf{x_1} + \mathbf{a_1}]||[C\mathbf{x_1} + D\mathbf{x_2} + \mathbf{a_2}]) + \mathbf{b_1}^\top\mathbf{x_1} + c \\ &= f(A\mathbf{x_1} + \mathbf{a_1}) + \mathbf{b_1}^\top\mathbf{x_1} + c. \end{aligned}$$

Therefore f and g are affine equivalent, as was required. □

Definition 5 *Let L_f denote the number of distinct input variables appearing in the ANF of f. The* dimension *of f, denoted $dim(f)$ is defined by*

$$dim(f) = \min_{g \in [f]} L_g.$$

A couple of properties of the dimension of a Boolean function are apparent. First, dimension is affine invariant, that is, if $g \in [f]$ then $dim(g) = dim(f)$. Second, for $dim(f) \geq 2$, if $dim(f) = L_f$, every variable of f occurs in a monomial of degree at least two, that is all variables occur nonlinearly. Third, the dimension of f is at least the degree of f. In the proof of Lemma 7 we will need the need the following proposition, the proof of which is given in the appendix.

Proposition 6 *Let $f \in \mathcal{B}_\ell$ with dim$(f) = \ell \geq 2$ and let $\mathbf{x} = (x_1, \ldots, x_n)^\top$, $n \geq \ell$. For any full rank linear map $L\colon \mathbb{F}_2^n \to \mathbb{F}_2^\ell$ and for any constant vector $\mathbf{a} \in \mathbb{F}_2^\ell$, every variable x_i occurring in $L\mathbf{x} + \mathbf{a}$ occurs in a nonlinear term in $f(L\mathbf{x} + \mathbf{a})$.*

Lemma 7 *Let $\theta(f, \ell)$ be the number of self-mappings of $f \in \mathcal{B}_\ell$. Let f_n be the embedding of f in $\mathcal{B}_n, n \geq \ell$. If the dimension of f is $\ell \geq 2$ then the number of self-mappings of f_n is*

$$\theta(f, n) = 2^{n-\ell}\theta(f, \ell) \prod_{i=\ell}^{n-1} (2^n - 2^i).$$

Proof. Consider S_n, a self-mapping of f_n. We may write

$$S_n = \left(\begin{bmatrix} A \ B \\ C \ D \end{bmatrix}, \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}, \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix}, c \right),$$

where A is $\ell \times \ell$, B is $\ell \times (n-\ell)$, C is $(n-\ell) \times \ell$, D is $(n-\ell) \times (n-\ell)$, $\mathbf{a_1}$ and $\mathbf{b_1}$ are ℓ-dimensional, and $\mathbf{a_2}$ and $\mathbf{b_2}$ are $(n-\ell)$-dimensional. For every $\mathbf{x} = \mathbf{x_1}||\mathbf{x_2}$ we have

$$\begin{aligned} f_n(\mathbf{x}) &= f_n\left(\begin{bmatrix} A \ B \\ C \ D \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix} \right) + \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + c \\ &= f\left(\begin{bmatrix} A \ B \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \mathbf{a_1} \right) + \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix}^\top \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + c \end{aligned} \tag{3}$$

By Proposition 6 given in Appendix and the fact that x_i for $i > \ell$ doesn't occur in the ANF of f_n, we conclude that B is the zero matrix. This fact further implies, from (3), that $\mathbf{b_2} = \mathbf{0}$. Therefore

$$f(\mathbf{x_1}) = f_n(\mathbf{x}) = S_n(f_n)(\mathbf{x}) = f(A\mathbf{x_1} + \mathbf{a_1}) + \mathbf{b_1}^\top \mathbf{x_1} + c.$$

Since S_n is an affine transformation and $B = 0$, it is necessary for A to be nonsingular. Consequently, $S = (A, \mathbf{a_1}, \mathbf{b_1}, c)$ is a self-mapping of f.

Since by the above any self-mapping of f_n is of the form

$$\left(\begin{bmatrix} A \ 0 \\ C \ D \end{bmatrix}, \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}, \begin{bmatrix} \mathbf{b_1} \\ \mathbf{0} \end{bmatrix}, c \right),$$

where $S = (A, \mathbf{a_1}, \mathbf{b_1}, c)$ is a self-mapping of f, we conclude that for every self-mapping S of f, for every $(n-\ell) \times \ell$ matrix C, for every nonsingular $(n-\ell) \times (n-\ell)$ matrix D and for every $(n-\ell)$-dimensional vector $\mathbf{a_2}$, there is an unique self-mapping of f_n.

There are $\theta(f, \ell)$ self-mappings S of f, $2^{\ell(n-\ell)}$ possible choices of C and $2^{n-\ell}$ choices of $\mathbf{a_2}$. Since, in addition, D is required to be invertible, there are $\prod_{i=0}^{n-\ell-1}(2^{n-\ell} - 2^i)$ choices of D. Thus

$$\theta(f, n) = 2^{n-\ell}\theta(f, \ell)2^{\ell(n-\ell)} \prod_{i=0}^{n-\ell-1} (2^{n-\ell} - 2^i) = 2^{n-\ell}\theta(f, \ell) \prod_{i=\ell}^{n-1} (2^n - 2^i),$$

as required. □

We note here explicitly that the condition that $dim(f) = \ell$ in the above lemma is a necessary condition. It can be shown that if $dim(f) < \ell$ that the result is a strict inequality instead of equality. This lemma allows us to prove the main tool of this paper.

Theorem 8 *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*

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

Proof. Using (2) and Lemma 7, the size of the equivalence class $[f_n]$ can be written as

$$
\begin{aligned}
|[f_n]| = \frac{\tau_n}{\theta(f,n)} &= \frac{2^{2n+1}\prod_{i=0}^{n-1}(2^n - 2^i)}{\theta(f,n)}, \\
&= \frac{2^{2n+1}\prod_{i=0}^{n-1}(2^n - 2^i)}{2^{n-\ell}\theta(f,\ell)\prod_{i=\ell}^{n-1}(2^n - 2^i)}, \\
&= \frac{2^{n+\ell+1}\prod_{i=0}^{\ell-1}(2^n - 2^i)}{\theta(f,\ell)}, \\
&= 2^{n-\ell}\frac{2^{2\ell+1}\prod_{i=0}^{\ell-1}(2^l - 2^i)}{\theta(f,\ell)}\prod_{i=0}^{\ell-1}\frac{(2^n - 2^i)}{(2^l - 2^i)}, \\
&= 2^{n-\ell}\frac{\tau_\ell}{\theta(f,\ell)}\prod_{i=0}^{\ell-1}\frac{(2^n - 2^i)}{(2^l - 2^i)}, \\
&= 2^{n-\ell}|[f_\ell]|\prod_{i=0}^{\ell-1}\frac{2^n - 2^i}{2^\ell - 2^i}.
\end{aligned}
$$

□

4 Boolean Functions with Multiplicative Complexity One

Fischer and Peralta [12] provided the number of Boolean functions with multiplicative complexity one. In this section, we reprove their result using an alternative method which can easily be extended for multiplicative complexity two.

Proposition 9 *Let f be an n-variable Boolean function with multiplicative complexity 1. f is affine equivalent to $x_1 \cdot x_2$.*

Proof. Let $f \in \mathcal{B}_n$ have multiplicative complexity exactly 1. Then f is of the form

$$f(\mathbf{x}) = (\mathbf{a}^\top \mathbf{x}) \cdot (\mathbf{b}^\top \mathbf{x}) + \mathbf{c}^\top \mathbf{x} + d,$$

where $\mathbf{a}$ and $\mathbf{b}$ are distinct and nonzero. Note that when $\mathbf{a}$ and $\mathbf{b}$ do not satisfy these conditions, the multiplicative complexity of f is 0.

Define the function $g \in \mathcal{B}_n$ by

$$g(x_1, \dots, x_n) = x_1 \cdot x_2.$$

We want to show that f and g are affine equivalent. It suffices to show that there exists an invertible A such that

$$f(\mathbf{x}) = g(A\mathbf{x}) + \mathbf{c}^\top \mathbf{x} + d.$$

One can let the first row of A be $\mathbf{a}^\top$, the second row be $\mathbf{b}^\top$. The last $n-2$ vectors can be chosen arbitrarily under the condition that they together with $\mathbf{a}, \mathbf{b}$ form a basis of $\mathbb{F}_2^n$. This is possible since $\mathbf{a}, \mathbf{b}$ are distinct and therefore linearly independent. □

Thus it suffices to count the number of functions in the equivalence class from Proposition 9. To this end, we use Theorem 8.

Proposition 10 *Let $n > 1$. The number of n-variable Boolean functions with multiplicative complexity 1 is exactly $2\binom{2^n}{3}$.*

Proof. The size of the equivalence class $[x_1x_2]$, $x_1x_2 \in \mathcal{B}_2$ corresponds to the number of non-affine functions in $\mathcal{B}_2$, which is equal to 8. Since the dimension of this function is 2, we can use Theorem 8. The size of the equivalence class $[x_1x_2]$, $x_1x_2 \in \mathcal{B}_n$ is

$$2^{n-2} 8 \left(\frac{2^n - 1}{2^2 - 1} \right) \left(\frac{2^n - 2}{2^2 - 2} \right) = 2 \binom{2^n}{3}$$

□

5 Boolean Functions with Multiplicative Complexity Two

In this section, we use the proof technique from the previous section to count the number of functions with multiplicative complexity 2. We start by showing that there exist exactly three equivalence classes with multiplicative complexity 2.

Proposition 11 *Let f be an n-variable Boolean function with multiplicative complexity 2. Then f is affine equivalent to exactly one of the following three functions:*

1. $x_1x_2x_3$
2. $x_1x_2x_3 + x_1x_4$
3. $x_1x_2 + x_3x_4$

Proof. First we show that each function with multiplicative complexity 2 falls in one of three classes. Consider a circuit, C, containing exactly two AND gates computing a function with multiplicative complexity two. Suppose first that there is no directed path from either of the AND gate to the other. In this case there exists affine forms $r_1, , \ldots, r_5$ such that the circuit computes the function

$$C(x) = r_1(x) \cdot r_2(x) + r_3(x) \cdot r_4(x) + r_5(x).$$

This function is affine equivalent to $x_1x_2+x_3x_4$ via an affine transformation similar to the one demonstrated in the proof of Proposition 9.

Now suppose that there exist a directed path from one of the AND gates to the other. Call the functions computed by the two AND gates f_{A_1}, f_{A_2}, respectively. Suppose the topologically minimal AND gate computes the function $f_{A_1}(x) = r_1(x)r_2(x)$ for suitably chosen affine forms r_1, r_2.

We now claim that there exist affine forms r_3, r_4, r_5 such that the circuit computes the function

$$(r_1 \cdot r_2 + r_3) \cdot r_4 + r_5.$$

First, we can assume that f_{A_1} occurs only in one of the two inputs of f_{A_2}. To see this, notice that

$$(f_{A_1} + r_3) \cdot (f_{A_1} + r_4) = (f_{A_1} + r_3) \cdot (r_4 + r_3 + 1).$$

Therefore the topologically last AND gate, f_{A_2}, computes the function

$$f_{A_2} = (f_{A_1} + r_3) \cdot r_4.$$

So we have that the output of the circuit is either $f_{A_2} + r_5$, or $f_{A_2} + f_{A_1} + r_5$. for some affine function r_5.

We claim that the output can be assumed to be of the first form. Suppose $A_2 = (A_1+r_3)\cdot r_4$, and let $A_2' = (A_1+r_3)\cdot(r_4+1)$, $r_5' = r_5+r_3$

$$A_2' + r_5' = (A_1 + r_3) \cdot (1 + r_4) + r_3 + r_5 = A_2 + A_1 + r_5.$$

This proves the claim. Now suppose that $r_3 \in \mathrm{span}_{\mathbb{F}_2}\{r_1, r_2, r_4\}$, and let

$$r_3 = c_1 r_1 + c_2 r_2 + c_4 r_4.$$

By adding constants to r_1, r_2 we can assume that $c_1 = c_2 = 0$. That is we have that the function computed is:

$$(r_1 r_2 + c_4 r_4) r_4 + r_5,$$

again we can assume that $c_4 = 0$ since otherwise

$$(r_1 r_2 + r_4) r_4 + r_5 = r_1 r_2 r_4 + r_4 + r_5.$$

We observe that the affine functions r_1, r_2, r_4 must be linearly independent (otherwise the function has multiplicative complexity at most 1). We conclude that this function is affine equivelant to $x_1 x_2 x_3$.

Now suppose that r_3 is linearly independent of r_1, r_2, r_4. In this case the function computed is

$$r_1 r_2 r_4 + r_3 r_4 + r_5,$$

for linearly independent functions r_1, r_2, r_3, r_4. This function is clearly linearly equivalent to the function $x_1 x_2 x_3 + x_1 x_4$.

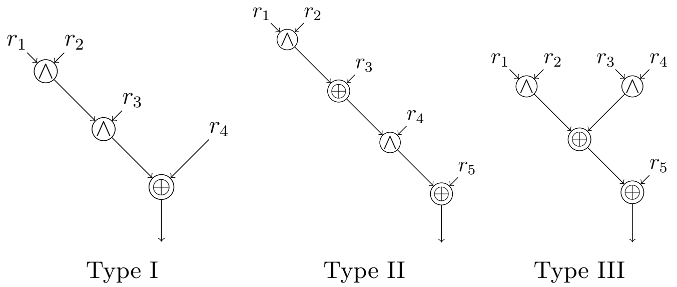
Finally we notice that these three functions indeed belong to three distinct equivalence classes. The function (3) has degree two, and is therefore not affine equivalent to a function of degree 3. Furthermore, by Lemma 4 a simple calculation shows that $[x_1 x_2 x_3]$ has dimension 3 whereas $[x_1 x_2 x_3 + x_1 x_4]$ has dimension 4. Thus these equivalence classes are distinct. □

The result readily implies that there are three different circuit types computing functions with multiplicative complexity 2. These are shown in Figure 1.

Theorem 12 *The number of n-variable Boolean functions with multiplicative complexity 2 is exactly*

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

Proof. By Proposition 11, it suffices to count the number of functions in each of the following three equivalence classes:



Type I Type II Type III

Fig. 1. The three different circuit types for the three equivalence classes.r_i denotes an affine form on the input variables. Every function with multiplicative complexity 2 can be computed by exactly one of the three displayed circuits.

1. $[x_1x_2x_3]$
2. $[x_1x_2x_3 + x_1x_4]$
3. $[x_1x_2 + x_3x_4]$

Type I: The size of the equivalence class $[x_1x_2x_3]$, $x_1x_2x_3 \in \mathcal{B}_3$ is the number of functions with degree 3, which is equal to 128. The dimension of this function is 3. Using Theorem 8, the size of the equivalence class $[x_1x_2x_3]$, $x_1x_2x_3 \in \mathcal{B}_n$ is

$$\frac{2^{n+1}(2^n-1)(2^n-2)(2^n-4)}{21}$$

Type II: Let $f_1, f_2 \in \mathcal{B}_4$ be $x_1x_2x_3$ and $x_1x_2x_3+x_1x_4$ respectively. It is easy to see that, for $n = 4$, $|[f_1]| + |[f_2]|$ equals the number of Boolean functions in $\mathcal{B}_4$ with degree 3, which is equal to 30720. Using the result presented above, $|[f_1]| = 3840$, hence $|f_2| = 30720-3840 = 26880$. The dimension of f_2 is 4. Using Theorem 8, the size of the equivalence class $[x_1x_2x_3 + x_1x_4]$, $x_1x_2x_3 + x_1x_4 \in \mathcal{B}_n$ is

$$\frac{2^n(2^n-1)(2^n-2)(2^n-4)(2^n-8)}{12}.$$

Type III: Let $f_1, f_2 \in \mathcal{B}_4$ be x_1x_2 and $x_1x_2+x_3x_4$ respectively. For $n = 4$, there are two equivalence classes ($[x_1x_2]$ and $[x_1x_2 + x_3x_4]$) that include the quadratic functions. Hence, $|[f_1]| + |[f_2]|$ is equal to the total number of quadratic function, which is 2016. Using that $|[x_1x_2]| = 2\binom{2^4}{3} = 560$, using Proposition 10. Then, the size of $[f_2]$ is

896 for $n = 4$. The dimension of f_2 is 4. Using Theorem 8, the size of the equivalence class $[x_1x_2 + x_3x_4]$, $x_1x_2x_3 + x_1x_4 \in \mathcal{B}_n$ is

$$\frac{2^n(2^n - 1)(2^n - 2)(2^n - 4)(2^n - 8)}{360}.$$

We conclude that the total number of functions with multiplicative complexity 2 is

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

which concludes the proof. □

6 Conclusion

One can count the number of Boolean functions with multiplicative complexity M by exhaustively listing the equivalence classes with multiplicative complexity M and finding the size of each class. In this paper, we showed that there are only three equivalence classes $[x_1x_2x_3]$, $[x_1x_2x_3 + x_1x_4]$ and $[x_1x_2 + x_3x_4]$ with multiplicative complexity two, and the number of n-bit Boolean functions with multiplicative complexity two is $2^n(2^n-1)(2^n-2)(2^n-4)(\frac{2}{21}+\frac{2^n-8}{12}+\frac{2^n-8}{360})$. However, for multiplicative complexity three, it is hard to list the equivalence classes exhaustively. For functions on $n = 4$, one can show that $[x_1x_2x_3x_4]$,$[x_1x_2x_3x_4 + x_1x_2]$ and $[x_1x_2x_3x_4 + x_1x_2 + x_3x_4]$ are the only equivalence classes with multiplicative complexity 3. For $n = 5$, the exhaustive list of classes with multiplicative complexity 3 is not known. Turan and Peralta [10] showed that the number of such classes is between 16 and 24. For $n = 6$, there are 2497 equivalence classes having degree at most 4. This provides an upper bound on the number of equivalence classes that can be computed by circuits with three AND gates, since some of these might have multiplicative complexity 4 or more.

Acknowledgments

The authors thank René Peralta and Çağdaş Çalık for helpful discussions and suggestions.

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. 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.
3. Joan Boyar, Ivan Damgård, and René Peralta. Short non-interactive cryptographic proofs. *J. Cryptology*, 13(4):449–472, 2000.
4. 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.
5. Nicolas Courtois, Daniel Hulme, and Theodosis Mourouzis. Solving circuit optimisation problems in cryptography and cryptanalysis, 2011.
6. Joan Boyar, Magnus Find, and René Peralta. Four measures of nonlinearity. In Paul G. Spirakis and Maria J. Serna, editors, *CIAC*, volume 7878 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 2013.
7. Magnus Gausdal Find. On the complexity of computing two nonlinearity measures. In Edward A. Hirsch, Sergei O. Kuznetsov, Jean-Éric Pin, and Nikolay K. Vereshchagin, editors, *Computer Science - Theory and Applications - 9th International Computer Science Symposium in Russia, CSR 2014, Moscow, Russia, June 7-11, 2014. Proceedings*, volume 8476 of *Lecture Notes in Computer Science*, pages 167–175. Springer, 2014.
8. 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.
9. Joan Boyar and René Peralta. A new combinational logic minimization technique with applications to cryptology. In Paola Festa, editor, *SEA*, volume 6049 of *Lecture Notes in Computer Science*, pages 178–189. Springer, 2010.
10. Meltem Sönmez Turan and René Peralta. The multiplicative complexity of boolean functions on four and five variables. In Thomas Eisenbarth and Erdinç Öztürk, editors, *Lightweight Cryptography for Security and Privacy - Third International Workshop, LightSec 2014, Istanbul, Turkey, September 1-2, 2014, Revised Selected Papers*, volume 8898 of *Lecture Notes in Computer Science*, pages 21–33. Springer, 2014.
11. Pavol Zajac and Matus Jokay. Multiplicative complexity of bijective 4 x 4 s-boxes. *Cryptography and Communications*, 6(3):255–277, 2014.
12. M. J. Fischer and R. Peralta. Counting predicates of conjunctive complexity one. *Yale Technical Report 1222*, February 2002 2002.

13. Roland Mirwald and Claus-Peter Schnorr. The multiplicative complexity of quadratic Boolean forms. *Theor. Comput. Sci.*, 102(2):307–328, 1992.
14. Claus-Peter Schnorr. The multiplicative complexity of Boolean functions. In *AAECC*, pages 45–58, 1988.
15. 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.
16. Joanne Elizabeth Fuller. *Analysis of affine equivalent boolean functions for cryptography.* PhD thesis, Queensland University of Technology, 2003.
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.

Appendix

Proof of Propositon 6

Proof. Suppose without loss of generality (by renaming the variables if necessary) that the variable x_1 occurs in $L\mathbf{x} + \mathbf{a}$. By elementary row operations, there exists a nonsingular linear transformation P such that the matrix representation of PL is in reduced row echelon form. We may further, without loss of generality, assume that the first ℓ columns of the matrix representation of PL form an $\ell \times \ell$ identity matrix. Define $\mathbf{z} = P(L\mathbf{x} + \mathbf{a}) \in \mathbb{F}_2^\ell$.

Let $F = f \circ P^{-1}$, so that $F(\mathbf{z}) = f(L\mathbf{x} + \mathbf{a})$. Since $dim(f) = \ell$ and $F \in [f]$, we have that $dim(F) = \ell$; therefore, the variable z_1 occurs in nonlinear terms in F. Let $d \geq 2$ be the maximum among degrees of monomials in $z_1, \ldots, z_\ell$ in which z_1 occurs.

Clearly, for all $1 \leq i \leq \ell$,

$$z_i = x_i + L_i(x_{\ell+1}, \ldots, x_n) + \tilde{a}_i.$$

Thus for an arbitrary permutation σ of $1, \ldots, \ell$, we have

$$z_{\sigma(1)} \cdots z_{\sigma(d)} = x_{\sigma(1)} \cdots x_{\sigma(d)} + p(x_{\ell+1}, \ldots, x_n) + \text{ lower degree terms},$$

where $deg(p) \leq d$.

Since the maximum degree monomials in which z_1 occurs are of degree d, the collection of degree d monomials in $x_1, \ldots, x_\ell$ is linearly independent, and the collection of the degree d monomials in $z_1, \ldots, z_\ell$ is linearly dependent, for any degree d monomial in the ANF of F containing z_1 there is a degree d monomial in the ANF of $f(L\mathbf{x} + \mathbf{a})$ containing x_1. Since $d \geq 2$, the variable x_1 occurs in a nonlinear term in $f(L\mathbf{x} + \mathbf{a})$. □