On the Multiplicative Complexity of Cubic Boolean Functions

Meltem Sönmez Turan and René Peralta

National Institute of Standards and Technology, Gaithersburg MD

Abstract

Multiplicative complexity is a relevant complexity measure for many advanced cryptographic protocols such as multi-party computation, fully homomorphic encryption, and zero-knowledge proofs, where processing `AND` gates is more expensive than processing `XOR` gates. For Boolean functions, multiplicative complexity is defined as the minimum number of `AND` gates that are sufficient to implement a function with a circuit over the basis (`AND`, `XOR`, `NOT`). In this paper, we study the multiplicative complexity of cubic Boolean functions. We propose a method to implement a cubic Boolean function with a small number of `AND` gates and provide upper bounds on the multiplicative complexity that are better than the known generic bounds.

1 Introduction

In many advanced cryptographic protocols such as multi-party computation (e.g., [1]), fully homomorphic encryption (e.g., [2]), and zero-knowledge proofs (e.g., [3]), processing nonlinear operations is more expensive than processing linear operations. Hence, having efficient implementations of these protocols in terms of nonlinear gates is of interest. This desired feature promoted the design of new symmetric-key primitives (e.g., Rasta [4], LowMC [5]) that use a small number of `AND` gates.

The *Multiplicative Complexity* (MC) of a Boolean function f, denoted $C_\wedge(f)$, is defined as the minimum number of `AND` gates that is sufficient to implement f with a circuit over the basis (`AND`, `XOR`, `NOT`). The MC of a Boolean function having degree d is at least $d-1$ [6]. Boyar et al. [7] showed that the MC of an n-variable random Boolean function is at least $2^{n/2} - \mathcal{O}(n)$ with high probability. There is no known asymptotically efficient method to calculate the MC of a random Boolean function. In practice, it is hard to calculate the MC even for Boolean functions with only seven variables. For up to 6 variables, the MC of each Boolean function has been established in [8, 9]. For arbitrary n, it is known that under standard cryptographic assumptions, computing the MC in polynomial time in the length of the truth table is not possible [10]. Even if the function is given in the form of a circuit, the problem is $coNP$-hard, as being able to determine MC would allow one to decide if the circuit encodes a tautology [10].

There are known bounds for special classes of Boolean functions. The MC of affine Boolean functions is zero. In [11], Mirwald and Schnorr showed that the MC of a quadratic function f is k, iff f is affine equivalent to the canonical form $\bigoplus_{i=1}^{k} x_{2i-1}x_{2i}$. This implies the MC of quadratic functions is at most $\lfloor \frac{n}{2} \rfloor$. In [12], Brandão et al. studied the MC of symmetric Boolean functions and constructed circuits for all such functions with up to 25 variables. The exact MC of the elementary symmetric functions Σ_k^n is also known for k less than or equal to 3 and for k larger than or equal to $n-3$ [13]. In 2017, Find et al. [14] characterized the Boolean functions with MC 2 by using the fact that MC is invariant with respect to affine transformations. In 2020, Çalık et al. extended the result to Boolean functions with MC up to 4 [15].

In this paper, we study the MC of cubic Boolean functions. We enumerate the equivalence classes of cubic functions with MC up to 4 and provide a generic implementation method. This method provides upper bounds on the MC of cubic Boolean functions that are significantly better than the upper bounds for random Boolean functions.

2 Preliminaries

Let $\mathbb{F}_2$ be the finite field with two elements. An n-variable Boolean function f is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let B_n be the set of n-variable Boolean functions and B_n^c be the set of n-variable cubic

Boolean functions.

The *algebraic normal form* (ANF) of f is the multivariate polynomial $f(x_1, \dots, x_n) = \sum_{u \in \mathbb{F}_2^n} a_u x^u$, 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.

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

$$f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{a}) + \mathbf{b}^\top \mathbf{x} + c \tag{1}$$

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 $c \in \mathbb{F}_2$. We use $[f]$ to denote the affine equivalence class of the function f. Degree and multiplicative complexity are invariant under affine transformations.

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, i.e., $dim(f) = \min_{g \in [f]} N_g$.

3 Cubic Boolean Functions with MC ≤ 4

In this section we provide some results on the MC of cubic Boolean functions. These results mainly follow from earlier studies [8, 9, 14, 15], and can be considered as special cases for cubic Boolean functions.

By the degree bound, the MC of a cubic Boolean function is at least two. Proposition 3.1 follows from [14] that exhaustively lists the affine equivalence classes with MC 2 as $[x_1x_2x_3]$, $[x_1x_2x_3+x_1x_4]$ and $[x_1x_2 + x_3x_4]$.

Proposition 3.1 *Let f be an n-variable cubic Boolean function with MC 2. Then f is affine equivalent to exactly one of the following two functions: $x_1x_2x_3$ and $x_1x_2x_3 + x_1x_4$.*

Next, we characterize the cubic Boolean functions with MC 3. As shown in [8], there are no cubic Boolean functions with MC 3 for $n = 4$. The dimension of a Boolean function with MC k is at most $2C_\wedge(f)$ [15], hence the dimension of Boolean functions with MC 3 is either 5 or 6.

Proposition 3.2 *Let f be an n-variable cubic Boolean function with dimension 5 and MC 3. Then f is affine equivalent to exactly one of the following four functions $x_1x_3x_4 + x_1x_2x_5$, $x_1x_2x_3 + x_4x_5$, $x_3x_4 + x_1x_3x_4 + x_1x_2x_5$ and $x_1x_2x_3 + x_2x_4 + x_1x_5$.*

Proposition 3.3 *Let f be an n-variable cubic Boolean function with dimension 6 and MC 3. Then f is affine equivalent to exactly one of the following three functions $x_3x_4 + x_1x_3x_4 + x_1x_2x_5 + x_1x_6$, $x_1x_3x_4 + x_1x_2x_5 + x_1x_6$ and $x_1x_2x_3 + x_4x_5 + x_1x_6$.*

Table 1 shows the affine equivalence classes for cubic functions with MC 4. The functions listed in Proposition 3.2, Proposition 3.3 and Table 1 are obtained by extracting cubic equivalance classes from [16].

Table 1: Affine equivalence class representations for cubic Boolean functions with MC 4

Dimension	*Equivalence class*
5	$x_2x_3 + x_1x_3x_4 + x_1x_2x_5$
	$x_2x_4 + x_3x_4 + x_1x_3x_4 + x_1x_2x_5 + x_3x_5$
6	$x_1x_3x_4 + x_1x_2x_5 + x_2x_6$
	$x_1x_3x_4 + x_1x_2x_5 + x_3x_5 + x_2x_6$
	$x_3x_4x_5 + x_1x_2x_6$
	$x_2x_3 + x_1x_3x_4 + x_1x_2x_5 + x_1x_6$
	$x_2x_3x_4 + x_1x_3x_5 + x_1x_2x_6$
	$x_2x_3x_4 + x_1x_3x_5 + x_4x_5 + x_1x_2x_6 + x_3x_6$
	$x_2x_3 + x_1x_4 + x_3x_4x_5 + x_1x_2x_6$
	$x_1x_4 + x_2x_3x_4 + x_1x_3x_5 + x_1x_2x_6$
	$x_1x_4 + x_2x_3x_4 + x_2x_5 + x_1x_3x_5 + x_1x_2x_6x_1x_2x_3 + x_3x_4 + x_2x_5 + x_1x_6$
	$x_2x_4 + x_3x_4 + x_1x_3x_4 + x_1x_2x_5 + x_3x_5 + x_1x_6$
	$x_2x_4 + x_3x_4 + x_2x_3x_4 + x_3x_5 + x_1x_3x_5 + x_1x_2x_6$
	$x_1x_3 + x_3x_4x_5 + x_1x_2x_6$
7	$x_1x_2 + x_1x_2x_3 + x_1x_2x_4 + x_3x_4 + x_1x_2x_6 + x_5x_6 + x_3x_7 + x_4x_7 + x_5x_7 + x_6x_7$
	$x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_1x_5x_6 + x_3x_5x_6 + x_4x_5x_6 + x_1x_7 + x_3x_7 + x_4x_7$
	$x_1x_2x_3 + x_1x_2x_4 + x_3x_4 + x_1x_2x_5 + x_1x_2x_6 + x_5x_6 + x_1x_7 + x_2x_7 + x_4x_7 + x_6x_7$
	$x_1x_2x_3 + x_1x_2x_4 + x_3x_4x_5 + x_5x_6 + x_1x_7 + x_2x_7 + x_3x_7 + x_4x_7 + x_3x_4x_7 + x_6x_7$
	$x_3x_4 + x_1x_2x_5 + x_3x_5 + x_3x_4x_5 + x_1x_2x_6 + x_3x_4x_6 + x_6x_7$
	$x_3x_4 + x_1x_2x_5 + x_5x_6 + x_1x_2x_7 + x_3x_4x_7 + x_5x_7 + x_6x_7$
	$x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 + x_1x_5 + x_2x_5 + x_4x_5 + x_1x_2x_6 + x_3x_4x_6 + x_6x_7$
	$x_1x_2x_3 + x_1x_5 + x_2x_5 + x_3x_5 + x_1x_2x_6 + x_3x_4x_6 + x_1x_5x_6 + x_2x_5x_6 + x_3x_5x_6 + x_6x_7$
	$x_1x_2 + x_3x_4 + x_1x_2x_5 + x_5x_6 + x_7 + x_3x_4x_7 + x_5x_7 + x_6x_7$
	$x_1x_2 + x_3x_4 + x_5x_6 + x_1x_2x_7 + x_3x_4x_7 + x_5x_7 + x_6x_7$
	$x_1x_2x_3 + x_3x_4x_5 + x_5x_6 + x_1x_2x_7 + x_3x_7 + x_3x_4x_7 + x_6x_7$
	$x_1x_2x_7 + x_3x_7 + x_4x_7 + x_3x_4x_7 + x_5x_7 + x_6x_7 + x_5x_6x_7$
	$x_3x_4 + x_5x_6 + x_1x_7 + x_2x_7 + x_1x_2x_7 + x_3x_4x_7 + x_6x_7 + x_5x_6x_7$
8	$x_1x_2 + x_1x_2x_5 + x_3x_4x_5 + x_5x_6 + x_1x_2x_7 + x_3x_4x_7 + x_7x_8$
	$x_1x_2x_7 + x_3x_4x_7 + x_5x_6x_7 + x_7x_8$
	$x_3x_4x_5 + x_1x_2x_6 + x_5x_6 + x_3x_4x_7 + x_1x_2x_8 + x_7x_8$
	$x_1x_2x_3 + x_3x_4 + x_1x_2x_5 + x_5x_6 + x_1x_2x_7 + x_7x_8$
	$x_1x_2x_5 + x_3x_4x_5 + x_5x_6 + x_1x_2x_7 + x_3x_4x_7 + x_7x_8$
	$x_1x_2 + x_5x_6 + x_1x_2x_7 + x_3x_4x_7 + x_5x_6x_7 + x_7x_8$

4 Constructing Circuits for Cubic Boolean Functions

Next, we provide an iterative method to implement cubic Boolean functions that uses a small number of `AND` gates. The method decomposes an n-variable cubic Boolean function f such that $f = x_nf_1 + f_2$, where f_1 is a quadratic function defined on $(x_1, \dots, x_{n-1})$ and f_2 is a function of degree at most three defined on $(x_1, \dots, x_{n-1})$. The method implements the functions f_1 and f_2 independently and computes f using one additional `AND` gate. The quadratic function f_1 is implemented using at most $\lfloor \frac{n-1}{2} \rfloor$ `AND` gates, as shown in [11]. The function f_2 is then recursively implemented. The recursion stops when f_2 is sub-cubic or when the number of variables in f_2 is small (e.g., $n = 6$). At that point, the function can be implemented optimally. Note that the decomposition can be done using any of the input variables. The natural greedy approach is to factor out a variable that appears in the largest number of cubic terms.

The method provides an upper bound on the MC of n-variable cubic Boolean functions, denoted $\mathrm{MaxMC}(B_n^c)$, using the following relation

$$\mathrm{MaxMC}(B_n^c) \leq \mathrm{MaxMC}(B_{n-1}^c) + \lfloor \frac{n-1}{2} \rfloor + 1. \quad (2)$$

For $n = 6$, it is known that the MC of cubic Boolean functions is at most 5 and this bound is tight, i.e., there exists cubic Boolean functions with MC 5 [9]. For $n = 7$, there are 179 affine equivalence classes for cubic Boolean functions [17, 18]. After applying the method presented here, we observed that the MC of cubic Boolean function for $n = 7$ is at most 8. Using this bound and the relation given in (2), we obtain

$$\text{MaxMC}(B_n^c) \leq \frac{1}{2}(\lfloor\frac{n-1}{2}\rfloor^2 + \lfloor\frac{n-1}{2}\rfloor + (\lfloor\frac{n}{2}\rfloor - 1)\lfloor\frac{n}{2}\rfloor + 2(n-8)) \in \frac{3n^2}{8} + O(n). \tag{3}$$

If we factor out two variables, as in $f(x_1, \ldots, x_n) = x_n f_1 + x_{n-1} f_2 + f_3$, where, without loss of generality, f_3 is at most cubic on variables $x_1, \ldots, x_{n-2}$ and f_1, f_2 are at most quadratic on variables $x_1, \ldots, x_{n-1}$, we obtain the recurrence

$$\begin{aligned}
\text{MaxMC}(B_n^c) &\leq \text{MaxMC}(B_{n-2}^c) + MC(f_1, f_2) + 2 && (4)\\
&\leq \text{MaxMC}(B_{n-2}^c) + \lfloor\frac{3(n-1)}{4}\rfloor + 2 \in \frac{3n^2}{16} + O(n). && (5)
\end{aligned}$$

The last inequality holds by a theorem of Mirwald and Schnorr [11]. We note that in practice this bound is likely not tight as, having calculated a circuit for $\{f_1, f_2\}$, the number of additional AND gates needed to calculate f_3 is likely smaller than $C_\wedge(f_3)$.

Table 2 provides upper bounds on the MC of n-variable cubic Boolean functions. The bounds for $n \geq 8$ are obtained using the bound from (5). The table also provides the best known bounds for the generic Boolean functions, given in [12].

Table 2: Upper bounds on the MC of n-variable Boolean functions

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Cubic functions	-	2	2	4	5	8	12	16	20	25	30	36	41	48	54
All functions	1	2	3	4	6	13	26	41	57	88	120	183	247	374	502

One can divide B_n into the set of functions B_n^+ for which $f(0) = 0$ and the set of functions B_n^- for which $f(0) = 1$. Function in B_n^+ can be optimally computed (with respect to multiplicative complexity) over the basis $(\texttt{AND}, \texttt{XOR})$. That is, negation (adding the constant 1) is not needed. An optimal circuit for a function $f()$ in B_n^- can be constructed from an optimal circuit for $f()+1 \in B_n^+$ by adding 1 to the output gate. Thus the number of functions in B_n that can be computed with at most k AND gates is exactly twice the number of functions in B_n^+ that can be computed with at most k AND gates. With this observation, a slight modification of the proof of Lemma 15 in [7], shows that the number of functions in B_n that can be computed with at most k AND gates is bounded above by $2^{k^2+2kn+n+2}$.

The cardinality of B_n^c is $(2^{\binom{n}{3}} - 1)2^{\binom{n}{2}+n+1}$. Thus, letting $\tau = \text{MaxMC}(B_n^c)$, we have

$$\begin{aligned}
(2^{\binom{n}{3}} - 1)2^{\binom{n}{2}+n+1} &\leq 2^{\tau^2+2\tau n+n+2}\\
\binom{n}{3} + \binom{n}{2} + n &\leq \tau^2 + 2\tau n + n + 2\\
n^3 - n &\leq 6\tau^2 + 12\tau n + 12
\end{aligned}$$

$$\frac{\sqrt{6}}{6}(n^3 + 6n^2 - n - 12)^{\frac{1}{2}} - n \leq \tau, \tag{6}$$

which shows that $\text{MaxMC}(B_n^c)$ is $\Omega(n^{3/2})$. Thus

$$\Omega(n^{3/2}) \leq \text{MaxMC}(B_n^c) \leq O(n^2). \tag{7}$$

Closing the gap in (7) is an interesting open problem.

5 Conclusion and Discussion

In this paper, we studied the multiplicative complexity of cubic Boolean functions. We first enumerated the equivalence classes of cubic Boolean functions with up to MC 4. Next, we provided a method to implement cubic Boolean functions that decomposes the input function into an expression of functions defined on a smaller number of variables. Using this method, we provide upper bounds on MC of cubic Boolean functions that are significantly better than the upper bounds for random Boolean functions. The methods in this paper can also be extended to implement Boolean functions with small Hamming distance to cubic Boolean functions, e.g., functions with small second-order nonlinearity. The extended algorithm and the proofs will be provided in the full version of the paper.

Acknowledgement

The authors thank Ç. Çalık for helpful discussions and implementation support, and Luís Brandão for useful suggestions.

References

[1] 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*, volume 5126 of *Lecture Notes in Computer Science*, pages 486–498. Springer, 2008.

[2] 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.

[3] Joan Boyar, Ivan Damgård, and René Peralta. Short Non-Interactive Cryptographic Proofs. *J. Cryptology*, 13(4):449–472, 2000.

[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] C. P. Schnorr. The Multiplicative Complexity of Boolean Functions. In Teo Mora, editor, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC 1988)*, volume 357 of *LNCS*, pages 45–58, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[7] 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.

[8] Meltem Turan Sönmez and René Peralta. *The Multiplicative Complexity of Boolean Functions on Four and Five Variables*, pages 21–33. Springer International Publishing, Cham, 2015.

[9] Çağdaş Çalık, Meltem Sönmez Turan, and René Peralta. The Multiplicative Complexity of 6-variable Boolean Functions. *Cryptogr. Commun.*, 11(1):93–107, 2019.

[10] 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.

[11] Roland Mirwald and Claus-Peter Schnorr. The Multiplicative Complexity of Quadratic Boolean Forms. *Theor. Comput. Sci.*, 102(2):307–328, 1992.

[12] 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.

[13] Joan Boyar and René Peralta. Tight bounds for the multiplicative complexity of symmetric functions. *Theor. Comput. Sci.*, 396(1-3):223–246, 2008.

[14] 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.

[15] Çağdaş Çalık, Meltem Sönmez Turan, and René Peralta. Boolean Functions with Multiplicative Complexity 3 and 4. *Cryptogr. Commun.*, 12(5):935–946, 2020.

[16] NIST Computer Security Division. *Circuit Complexity Project Repository*, `https://github.com/usnistgov/Circuits/`.

[17] X.D. Hou. AGL(m, 2) Acting on R(r, m)/R(s, m). *Journal of Algebra*, 171(3):921–938, 1995.

[18] An Braeken, Yuri Borissov, Svetla Nikova, and Bart Preneel. Classification of Boolean Functions of 6 Variables or Less with Respect to Cryptographic Properties. Cryptology ePrint Archive, Report 2004/248, 2004. `https://eprint.iacr.org/2004/248`.