On the Multiplicative Complexity of 6-variable Boolean Functions

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

National Institute of Standards and Technology, Gaithersburg, MD, USA

July 5, 2017 BFA 2017 Os, Norway

What is Multiplicative Complexity?

Multiplicative complexity is a complexity measure that is defined as the minimum number of AND gates required to implement a function f by a circuit over the basis (AND, XOR, NOT).

Why do we count the AND gates?

- **Lightweight Cryptography:** Efficient implementations needed for resource-constrained devices (e.g. RFID tags). The technique of minimizing the number of AND gates, and then optimizing the linear components leads to the implementations with low gate complexity.
- **Secure multi-party computation:** Reducing the number of AND gates improves the efficiency of secure multi-party protocols (e.g. conducting online auctions in a way that the winning bid can be determined without opening the losing bids).
- **Side channel attacks:** Minimizing the number of AND gates is necessary when implementing a masking scheme to prevent side-channel attacks.
- **Cryptanalysis of cryptographic primitives:** Primitives with low multiplicative complexity may be susceptible to algebraic cryptanalysis.

Some Properties of Multiplicative Complexity

- Multiplicative complexity of a function with degree d is at least $d-1$.
- Multiplicative complexity is invariant w.r.t affine transformation.
  - f and g are affine equivalent, if there exists an affine transformation of the form $f(x) = g(Ax + a) + b \cdot x + c$, where A is a non-singular $n \times n$ matrix over $\mathbb{F}_2$; x, a are column vectors over $\mathbb{F}_2$; b is a row vector over $\mathbb{F}_2$.
  - If f and g are affine equivalent, they are said to be in the same equivalence class and they have the same multiplicative complexity.
- Multiplicative complexity of a randomly selected n-bit Boolean function is at least $2^{n/2} - \mathcal{O}(n)$. No specific n-bit Boolean function has been proven to have multiplicative complexity larger than $n-1$ for any n.

4- and 5-bit Boolean Functions (Turan and Peralta, 2014)

Turan and Peralta (2014) showed that multiplicative complexity is

- ≤ 3 for $f \in B_4$ (8 equivalence classes),
- ≤ 4 for $f \in B_5$ (48 equivalence classes).

Method

1. Find a simple representative from each equivalence class.
2. Find a circuit with small number of AND gates.
3. Check if it is optimal using the degree bound.

Equivalence classes for $n = 4$

Class	Representative
1	x_1
2	x_1x_2
3	$x_1x_2 + x_3x_4$
4	$x_1x_2x_3$
5	$x_1x_2x_3 + x_1x_4$
6	$x_1x_2x_3x_4$
7	$x_1x_2x_3x_4 + x_1x_2$
8	$x_1x_2x_3x_4 + x_1x_2 + x_3x_4$

6-bit Boolean Functions

The approach of Turan & Peralta does not work for $n = 6$, since

- The number of equivalence classes is 150 537, and
- Simple heuristics do not find optimal circuits, as representatives are more complex.
- For some classes, it is not possible to verify optimality using the degree bound.

Our approach

Exhaustively construct all Boolean circuits with 1,2, 3, . . . AND gates, and mark the Boolean functions that can be generated by the circuits until all 6-bit Boolean functions are generated.

6-bit Boolean Functions

The approach of Turan & Peralta does not work for $n = 6$, since

- The number of equivalence classes is 150 537, and
- Simple heuristics do not find optimal circuits, as representatives are more complex.
- For some classes, it is not possible to verify optimality using the degree bound.

Our approach

Exhaustively construct all Boolean circuits with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until ~~all 6-bit Boolean functions are generated~~ a function from each equivalence class is generated.

6-bit Boolean Functions

The approach of Turan & Peralta does not work for $n = 6$, since

- The number of equivalence classes is 150 537, and
- Simple heuristics do not find optimal circuits, as representatives are more complex.
- For some classes, it is not possible to verify optimality using the degree bound.

Our approach

Exhaustively construct all Boolean ~~circuits~~ topologies with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until a function from each equivalence class is generated.

Boolean circuit and Topology of a circuit (Codish et al, 2015)

Definition (Boolean circuit)

For a given $n \in \mathbb{N}$, let $X_n = \{x_1, x_2, \ldots, x_n\}$ denote the n inputs to a circuit. A Boolean circuit C with n inputs and k AND gates is a pair $\mathcal{C} = (\mathcal{A}, \mathcal{O})$, where:

- $\mathcal{A} = \{a_1, \ldots, a_k\}$ is a list of k AND gates, where the i-th AND gate inputs L_i and R_i with $L_i, R_i \in \langle 1, x_1, \ldots, x_n, L_1.R_1, \ldots, L_{i-1}.R_{i-1} \rangle$.
- $\mathcal{O} \in \langle 1, x_1, \ldots, x_n, L_1.R_1, \ldots, L_k.R_k \rangle$ is the output gate.

Boolean circuit and Topology of a circuit (Codish et al, 2015)

Definition (Boolean circuit)

For a given $n \in \mathbb{N}$, let $X_n = \{x_1, x_2, \ldots, x_n\}$ denote the n inputs to a circuit. A Boolean circuit C with n inputs and k AND gates is a pair $\mathcal{C} = (\mathcal{A}, \mathcal{O})$, where:

- $\mathcal{A} = \{a_1, \ldots, a_k\}$ is a list of k AND gates, where the i-th AND gate inputs L_i and R_i with $L_i, R_i \in \langle 1, x_1, \ldots, x_n, L_1.R_1, \ldots, L_{i-1}.R_{i-1}\rangle$.
- $\mathcal{O} \in \langle 1, x_1, \ldots, x_n, L_1.R_1, \ldots, L_k.R_k\rangle$ is the output gate.

Definition (Topology)

A topology of a circuit $C = (\mathcal{A}, \mathcal{O})$ is the set of AND gates $\mathcal{A}$, except that $L \cup R \subset \mathcal{A}$ for all $\langle L, R\rangle \in \mathcal{A}$. Given an AND-XOR circuit $C = \langle \mathcal{A}, \mathcal{O}\rangle$, the topology of $\mathcal{C}$ is $\langle \langle L \cap \mathcal{A}, R \cap \mathcal{A}\rangle \,|\, \langle L, R\rangle \in \mathcal{A}\rangle$.

Example: Boolean Circuit and Topology

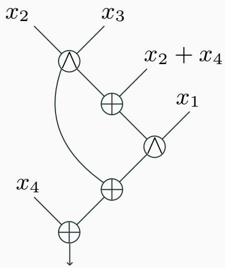
Let $f = x_1x_2x_3 + x_1x_2 + x_1x_4 + x_2x_3 + x_4$.

The circuit $C = \langle \mathcal{A}, \mathcal{O} \rangle$ is represented as $\mathcal{A} = \langle a_1, a_2 \rangle$

$a_1 = \langle \{x_2\}, \{x_3\} \rangle$

$a_2 = \langle \{a_1, x_2, x_4\}, \{x_1\} \rangle$

$\mathcal{O} = \langle \{x_4\}, \{a_1, a_2\} \rangle$



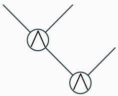
The topology of C is represented as

$\mathcal{A} = \langle a_1, a_2 \rangle$

$a_1 = \langle \emptyset, \emptyset \rangle$

$a_2 = \langle \{a_1\}, \emptyset \rangle$

$\mathcal{O} = \langle \emptyset, \{a_1, a_2\} \rangle$



Constructing Circuit Topologies

Let T_k be the set of all topologies with k AND gates. We use an iterative method to construct T_{k+1} as follows:

1. Let S be an empty set.
2. For each topology $t \in T_k$,
    2.1 For all choices of (L_{k+1}, R_{k+1}) (L_{k+1} and R_{k+1} can take on all 2^k possible combinations of previous k AND gates),
        2.1.1 Let t' be a new topology constructed by adding a new AND gate a_{k+1} with inputs (L_{k+1}, R_{k+1}) to t.
        2.1.2 $S = S \cup t'$
3. We eliminate redundant topologies (due to symmetry). $T_{k+1} = S$.

Constructing Circuit Topologies

Let T_k be the set of all topologies with k AND gates. We use an iterative method to construct T_{k+1} as follows:

1. Let S be an empty set.
2. For each topology $t \in T_k$,
   2.1 For all choices of (L_{k+1}, R_{k+1}) (L_{k+1} and R_{k+1} can take on all 2^k possible combinations of previous k AND gates),
      2.1.1 Let t' be a new topology constructed by adding a new AND gate a_{k+1} with inputs (L_{k+1}, R_{k+1}) to t.
      2.1.2 $S = S \cup t'$
3. We eliminate redundant topologies (due to symmetry). $T_{k+1} = S$.

Number of topologies for k up to 6

k	1	2	3	4	5	6
$\|T_k\|$	1	2	8	84	3 170	475 248

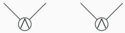
Constructing Circuit Topologies

Topologies with 1 AND gate

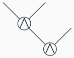
Constructing Circuit Topologies

Topologies with 1 AND gate

Topologies with 2 AND gates



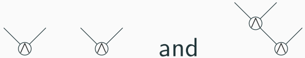
and



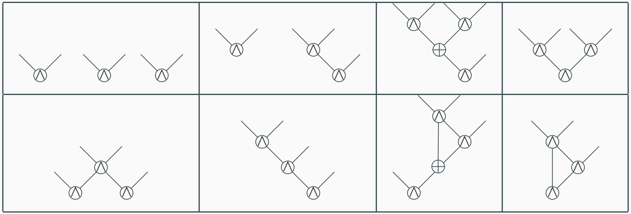
Constructing Circuit Topologies

Topologies with 1 AND gate

Topologies with 2 AND gates

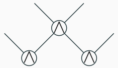


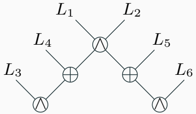
Topologies with 3 AND gates



Evaluating Topologies to Generate Boolean Functions

- A topology with k AND gates can be supplied $2k$ linear function inputs $X = (L_1, \ldots, L_{2k})$. Trying all inputs becomes quickly infeasible since there are 2^{2kn} choices (2^{60} inputs for $n = 6,\ k = 5$).
- Any affine transformation of the inputs $A(X) = (A(L_1), \ldots, A(L_{2k}))$ will produce a function from the same equivalence class. Hence, the inputs that are affine transformations of each other need not be considered.
- The number of inputs corresponds to the Gaussian binomial coefficient $\binom{2k}{n}_2$ ($\approx 2^{26}$ inputs for $n = 6,\ k = 5$).





Computation Summary

- Generated all topologies ≤ 6 AND gates.
- For each topology having $k = 1, 2, 3, 4, 5$ AND gates, all equivalence classes each topology can produce is found.
- 149 426 equivalence classes out of 150 357 generated with at most 5 AND gates.
- Remaining 931 equivalence classes were generated from a selection of 6 AND gate topologies.
- Computations were done on a cluster (Intel Xeon E5-2630 processor, 64GB RAM) and took 38 422 core hours.

Multiplicative Complexity Distribution for $n = 6$

Multiplicative complexity distribution of the equivalence classes and functions for $n = 6$

MC	#classes	#functions	$\log_2(\#functions)$
0	1	128	7.00
1	1	83 328	16.34
2	3	73 757 184	26.13
3	24	281 721 079 808	38.03
4	914	7 944 756 861 878 272	52.81
5	148 483	18 344 082 080 963 133 440	63.99
6	931	94 716 954 089 619 456	56.39

Conclusion

- Multiplicative complexity distribution of 6-bit Boolean functions is found.
- Showed that the multiplicative complexity is ≤ 6 for $f \in B_6$.
- Showed that there exists $f \in B_6$ with multiplicative complexity 6, e.g.,
  - A function with 6 monomials: $x_1x_5 + x_3x_6 + x_3x_4x_5 + x_2x_4 + x_1x_2x_6 + x_1x_2x_3x_4x_5x_6$
  - A function with algebraic degree 4: $x_4x_5 + x_3x_4x_5 + x_2x_5 + x_2x_4 + x_2x_4x_6 + x_1x_5x_6 + x_1x_4 + x_1x_3 + x_1x_2x_4x_5 + x_1x_2x_3x_6$

References

1. Sönmez Turan M., Peralta R., "The multiplicative complexity of Boolean functions on four and five variables", International Workshop on Lightweight Cryptography for Security and Privacy, 2014
2. M. Codish, L. Cruz-Filipe, M. Frank, P. Scheneider-Kamp, "When Six Gates are Not Enough", https://arxiv.org/pdf/1508.05737.pdf, 2015
3. Fuller, J.E. "Analysis of affine equivalent boolean functions for cryptography" Ph.D. thesis, Queensland University of Technology, 2003

Thanks!