On the Multiplicative Complexity of Boolean Functions and Bitsliced Higher-Order Masking

Dahmun Goudarzi and Matthieu Rivain

CHES 2016, Santa-Barbara

CryptoExperts
WE INNOVATE TO SECURE YOUR BUSINESS

ENS
ÉCOLE NORMALE SUPÉRIEURE

Higher-Order Masking

$$x = x_1 + x_2 + \cdots + x_d$$

Higher-Order Masking

$$x = x_1 + x_2 + \cdots + x_d$$

- Linear operations: $O(d)$

Higher-Order Masking

$$x = x_1 + x_2 + \cdots + x_d$$

- Linear operations: $O(d)$
- Non-linear operations: $O(d^2)$

Higher-Order Masking

$$x = x_1 + x_2 + \cdots + x_d$$

- Linear operations: $O(d)$
- Non-linear operations: $O(d^2)$

$\rightarrow$ Challenge for blockciphers: S-boxes

Ishai-Sahai-Wagner Multiplication

$$\left(\bigoplus_i a_i\right) \cdot \left(\bigoplus_i b_i\right) = \bigoplus_{i,j} a_i \cdot b_j + \text{fresh random}$$

- Variant: CPRR evaluation for quadratic functions (Coron etal, FSE 2013)

The Polynomial Method

- Sbox seen as a (univariate) polynomial over $GF(2^n)$

- Specific S-boxes, *e.g.* AES

$$S(x) = \mathsf{Aff}(x^{254})$$

- Generic methods:
  - CRV decomposition (CHES 2014):

$$S(x) = \sum_{i=0}^{t-1} g_i(x) \cdot h_i(x) + h_t(x)$$

  - Algebraic decomposition (CRYPTO 2015):

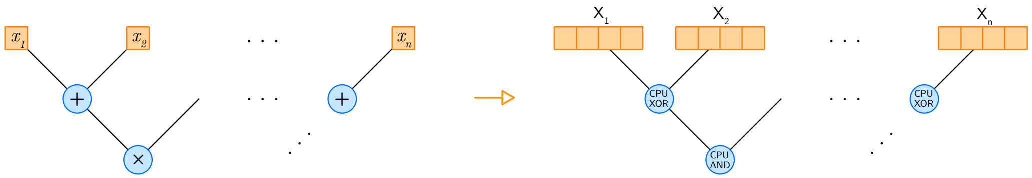
$$S(x) = \sum_{i=0}^{t-1} h_i(g_i(x)) + h_t(x)$$

The Bitslice Method

- Sbox seen as boolean circuit

The Bitslice Method

- Sbox seen as boolean circuit

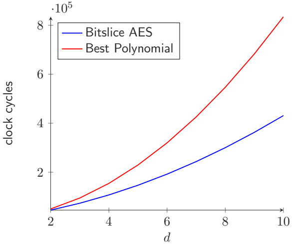


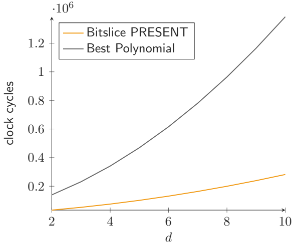
Bitslice for S-boxes

- Find a compact Boolean circuit at the S-box
- 16 S-box computed with one bitsliced computation
- Higher-Order Masking:
  - XOR $\rightarrow d$ XORs
  - AND $\rightarrow$ ISW-AND
- Minimizing the $O(d^2) \rightarrow$ minimizing the number of ISW-AND

Polynomial vs Bitslice approach

- How Fast Can Higher-Order Masking Be in Software?, eprint 2016





- Motivation: bitslice for generic s-box evaluations

Multiplicative Complexity of Boolean Functions

Boolean functions

- Span: $\langle f_1, f_2 \ldots, f_m \rangle = \left\{ \sum_{i=0}^{m} a_i \, f_i \mid a_i \in \mathbb{F}_2 \right\}$
- $\mathcal{M}_n = \left\{ x \mapsto x^u = x_1^{u_1} \cdot x_2^{u_2} \cdots x_n^{u_n} \mid u \in \{0,1\}^n \right\}$ is the set of monomials
- Algebraic Normal Form (ANF):

$$f(x) = \sum_{u \in \{0,1\}^n} a_u \, x^u, \text{ i.e. } f \in \langle \mathcal{M}_n \rangle$$

- S-box: $S(x) = (f_1(x), f_2(x), \ldots, f_n(x))$

Multiplicative Complexity

- $C(f)$: minimum number of multiplications to compute f

Multiplicative Complexity

- $C(f)$: minimum number of multiplications to compute f
- $C(f_1, f_2, \ldots, f_n) \leq C(\mathcal{M}_n) = 2^n - (n+1)$

Multiplicative Complexity

- $C(f)$: minimum number of multiplications to compute f
- $C(f_1, f_2, \ldots, f_n) \leq C(\mathcal{M}_n) = 2^n - (n+1)$
- $\exists f \in \langle \mathcal{M}_n \rangle,\ C(f) > 2^{\frac{n}{2}} - n$

Multiplicative Complexity

- $C(f)$: minimum number of multiplications to compute f
- $C(f_1, f_2, \ldots, f_n) \leq C(\mathcal{M}_n) = 2^n - (n+1)$
- $\exists f \in \langle \mathcal{M}_n \rangle$, $C(f) > 2^{\frac{n}{2}} - n$
- Method to find optimal solution for $n \leq 5$: SAT-Solver

Multiplicative Complexity

- $C(f)$: minimum number of multiplications to compute f
- $C(f_1, f_2, \ldots, f_n) \leq C(\mathcal{M}_n) = 2^n - (n+1)$
- $\exists f \in \langle \mathcal{M}_n \rangle$, $C(f) > 2^{\frac{n}{2}} - n$
- Method to find optimal solution for $n \leq 5$: SAT-Solver
- Constructive method [BPP00]:

$$C(f) \approx 2^{\frac{n}{2}+1} - \frac{n}{2} - 2$$

CryptoExperts

ENS
École Normale Supérieure

Our results

- Generalization of BPP for S-boxes:

$$C(S) \approx \sqrt{n}2^{\frac{n}{2}+1} - \frac{3}{2}n - \frac{1}{2}\log n$$

- New method: generalization of CRV

$$C(S) \approx \sqrt{n}2^{\frac{n}{2}+1} - 2n - 1$$

n	4	5	6	7	8	9	10
BPP extended	8	16	29	47	87	120	190
Our generic method ($C_{n,n}$)	8	17	31	50	77	122	190
Our improved method ($C^*_{n,n}$)	7	13	23	38	61	96	145

Table: Multiplicative complexities of n bits s-boxes.

New Generic Decomposition Method

Decomposition of a Single Boolean Function

$$f(x) = \sum_{i=0}^{t} g_i(x) \cdot h_i(x)$$

Decomposition of a Single Boolean Function

$$f(x) = \sum_{i=0}^{t} g_i(x) \cdot h_i(x)$$

- g_i: random linear combinations from $\mathcal{B} = \{\phi_j\}_j$

  $a_{i,j} \xleftarrow{\$} \{0,1\}$ $\qquad$ $g_i \leftarrow \sum_j a_{i,j}\phi_j$

CryptoExperts

ENS
ÉCOLE NORMALE SUPÉRIEURE

Decomposition of a Single Boolean Function

$$f(x) = \sum_{i=0}^{t} g_i(x) \cdot h_i(x)$$

- g_i: random linear combinations from $\mathcal{B} = \{\phi_j\}_j$

  $a_{i,j} \xleftarrow{\$} \{0,1\} \qquad g_i \leftarrow \sum_j a_{i,j}\phi_j$

- find $c_{i,j}$ s.t $h_i = \sum_j c_{i,j}\phi_j$ solving a linear system:

$$f(x) = \sum_i (\sum_j a_{i,j}\phi_j(x))(\sum_j c_{i,j}\phi_j(x)), \ \forall x$$

Decomposition of a Single Boolean Function

$$f(x) = \sum_i (\sum_j a_{i,j}\phi_j(x))(\sum_j c_{i,j}\phi_j(x)), \ \forall x$$

- $\{e_i\}_{i=1}^{2^n} = \mathbb{F}_2^n$
- $A_1 c_1 + A_2 c_2 + \cdots + A_t c_t = (f(e_1), f(e_2), \ldots, f(e_{2^n}))$

$$A_i = \begin{pmatrix} \phi_1(e_1)\cdot g_i(e_1) & \phi_2(e_1)\cdot g_i(e_1) & \ldots & \phi_{|\mathcal{B}|}(e_1)\cdot g_i(e_1) \\ \phi_1(e_2)\cdot g_i(e_2) & \phi_2(e_2)\cdot g_i(e_2) & \ldots & \phi_{|\mathcal{B}|}(e_2)\cdot g_i(e_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(e_{2^n})\cdot g_i(e_{2^n}) & \phi_2(e_{2^n})\cdot g_i(e_{2^n}) & \ldots & \phi_{|\mathcal{B}|}(e_{2^n})\cdot g_i(e_{2^n}) \end{pmatrix}$$

Conditions

- $(t+1)|\mathcal{B}|$ unknowns, 2^n equations:

$$(t+1)|\mathcal{B}| \geq 2^n$$

- Condition on the sum: $t \geq \lceil \frac{2^n}{|\mathcal{B}|} \rceil - 1$
- Condition on the basis: $\mathcal{B} \times \mathcal{B}$ has to span all Boolean functions

How to Construct the Basis $\mathcal{B}$

- Start from $\mathcal{B}_0$ such that $\mathcal{B}_0 \times \mathcal{B}_0 = \langle \mathcal{M}_n \rangle$
- from $\mathcal{B}_0$ to $\mathcal{B}$:
  - $\phi, \psi \xleftarrow{\$} \langle \mathcal{B} \rangle$
  - $\mathcal{B} \leftarrow \phi \cdot \psi$

Costs

- r multiplications for $\mathcal{B}$

$$r = |\mathcal{B}| - n - 1, \qquad |\mathcal{B}| \geq |\mathcal{B}_0|$$

- t multiplications for decomposition products

$$t \geq \lceil \frac{2^n}{|\mathcal{B}|} \rceil - 1$$

- Cost: $r + t$

n	4	5	6	7	8	9	10
(r, t)	(2,3)	(5,3)	(9,5)	(16,6)	(25,9)	(41,11)	(59,17)
$C_{n,n}$	5	8	14	22	34	52	78

Decomposition of the S-box

- Sbox: $x \to (f_1(x), f_2(x), \ldots, f_n(x))$
- Apply n Boolean decompositions on the f_i's
- Costs: $r + t \cdot n$ multiplications

n	4	5	6	7	8	9	10
(r, t)	(4,1)	(7,2)	(13,3)	(22,4)	(37,5)	(59,7)	(90,10)
$C_{n,n}$	8	17	31	50	77	122	190

- Works for any S-boxes

S-box Dependent Improvements

Basis Update Improvements

- Start with $\mathcal{B}_1 \supseteq \mathcal{B}_0$
- Decompose $f_1 = \sum_i g_{1,i} \cdot h_{1,i}$ with $\mathcal{B}_1$

Basis Update Improvements

- Start with $\mathcal{B}_1 \supseteq \mathcal{B}_0$
- Decompose $f_1 = \sum_i g_{1,i} \cdot h_{1,i}$ with $\mathcal{B}_1$
- Set $\mathcal{B}_2 = \mathcal{B}_1 \cup \{g_{1,i} \cdot h_{1,i}\}$
- Decompose $f_2 = \sum_i g_{2,i} \cdot h_{2,i}$ with $\mathcal{B}_2$

Basis Update Improvements

- Start with $\mathcal{B}_1 \supseteq \mathcal{B}_0$
- Decompose $f_1 = \sum_i g_{1,i} \cdot h_{1,i}$ with $\mathcal{B}_1$

- Set $\mathcal{B}_2 = \mathcal{B}_1 \cup \{g_{1,i} \cdot h_{1,i}\}$
- Decompose $f_2 = \sum_i g_{2,i} \cdot h_{2,i}$ with $\mathcal{B}_2$

- Set $\mathcal{B}_3 = \mathcal{B}_2 \cup \{g_{2,i} \cdot h_{2,i}\}$
- Decompose $f_3 = \sum_i g_{3,i} \cdot h_{3,i}$ with $\mathcal{B}_3$

Basis Update Improvements

- Start with $\mathcal{B}_1 \supseteq \mathcal{B}_0$
- Decompose $f_1 = \sum_i g_{1,i} \cdot h_{1,i}$ with $\mathcal{B}_1$

- Set $\mathcal{B}_2 = \mathcal{B}_1 \cup \{g_{1,i} \cdot h_{1,i}\}$
- Decompose $f_2 = \sum_i g_{2,i} \cdot h_{2,i}$ with $\mathcal{B}_2$

- Set $\mathcal{B}_3 = \mathcal{B}_2 \cup \{g_{2,i} \cdot h_{2,i}\}$
- Decompose $f_3 = \sum_i g_{3,i} \cdot h_{3,i}$ with $\mathcal{B}_3$

  ...
- $\mathcal{B}_n = \mathcal{B}_{n-1} \cup \{g_{n-1,i} \cdot h_{n-1,i}\}$
- Decompose $f_n = \sum_i g_{n,i} \cdot h_{n,i}$ with $\mathcal{B}_{n-1}$

Basis Update Improvements

- Start with $\mathcal{B}_1 \supseteq \mathcal{B}_0$
- Decompose $f_1 = \sum_i g_{1,i} \cdot h_{1,i}$ with $\mathcal{B}_1$ $\quad t_1 = \lceil \frac{2^n}{|\mathcal{B}_1|} \rceil - 1$

- Set $\mathcal{B}_2 = \mathcal{B}_1 \cup \{g_{1,i} \cdot h_{1,i}\}$
- Decompose $f_2 = \sum_i g_{2,i} \cdot h_{2,i}$ with $\mathcal{B}_2$ $\quad t_2 = \lceil \frac{2^n}{|\mathcal{B}_2|} \rceil - 1$

- Set $\mathcal{B}_3 = \mathcal{B}_2 \cup \{g_{2,i} \cdot h_{2,i}\}$
- Decompose $f_3 = \sum_i g_{3,i} \cdot h_{3,i}$ with $\mathcal{B}_3$ $\quad t_3 = \lceil \frac{2^n}{|\mathcal{B}_3|} \rceil - 1$

...

- $\mathcal{B}_n = \mathcal{B}_{n-1} \cup \{g_{n-1,i} \cdot h_{n-1,i}\}$
- Decompose $f_n = \sum_i g_{n,i} \cdot h_{n,i}$ with $\mathcal{B}_{n-1}$ $\quad t_n = \lceil \frac{2^n}{|\mathcal{B}_n|} \rceil - 1$

- Costs: $r + t_1 + t_2 + \ldots + t_n$

Rank Drop

- $A_1c_1 + A_2c_2 + \cdots + A_tc_t = (f(e_0), f(e_1), \ldots, f(e_{2^n}))$
- System $A \cdot c = b$ with rank$(A) = 2^n - \delta$ works for $\frac{1}{2^\delta}$ boolean functions
- Try $O(2^\delta)$ systems
- Reduced parameter: $(t+1)|\mathcal{B}| \geq 2^n - \delta$

$$\rightarrow t \geq \lceil \frac{2^n - \delta}{|\mathcal{B}|} \rceil - 1$$

Results

Sbox	Serpent	SC2000 S_5	SC2000 S_6	CLEFIA
n	4	5	6	8
Our generic method	7	17	31	77
Our improved method	6	11	21	62
Gain	1	6	10	15

Implementation

Parallelization

- 16 S-box $\rightarrow$ 16-bit bitsliced registers
- But 32-bit architecture
- 2 16-bit ISW-AND $\Rightarrow$ 1 32-bits ISW-AND
- At the circuit level: grouping AND gates per pair

A circuit for AES with parallelizable AND gates

$t_2 = y_{12} \wedge y_{15}$	$t_{23} = t_{19} \oplus y_{21}$	$t_{34} = t_{23} \oplus t_{33}$	$z_2 = t_{33} \wedge x_7$
$t_3 = y_3 \wedge y_6$	$t_{15} = y_8 \wedge y_{10}$	$t_{35} = t_{27} \oplus t_{33}$	$z_3 = t_{43} \wedge y_{16}$
$t_5 = y_4 \wedge x_7$	$t_{26} = t_{21} \wedge t_{23}$	$t_{42} = t_{29} \oplus t_{33}$	$z_4 = t_{40} \wedge y_1$
$t_7 = y_{13} \wedge y_{16}$	$t_{16} = t_{15} \oplus t_{12}$	$z_{14} = t_{29} \wedge y_2$	$z_6 = t_{42} \wedge y_{11}$
$t_8 = y_5 \wedge y_1$	$t_{18} = t_6 \oplus t_{16}$	$t_{36} = t_{24} \wedge t_{35}$	$z_7 = t_{45} \wedge y_{17}$
$t_{10} = y_2 \wedge y_7$	$t_{20} = t_{11} \oplus t_{16}$	$t_{37} = t_{36} \oplus t_{34}$	$z_8 = t_{41} \wedge y_{10}$
$t_{12} = y_9 \wedge y_{11}$	$t_{24} = t_{20} \oplus y_{18}$	$t_{38} = t_{27} \oplus t_{36}$	$z_9 = t_{44} \wedge y_{12}$
$t_{13} = y_{14} \wedge y_{17}$	$t_{30} = t_{23} \oplus t_{24}$	$t_{39} = t_{29} \wedge t_{38}$	$z_{10} = t_{37} \wedge y_3$
$t_4 = t_3 \oplus t_2$	$t_{22} = t_{18} \oplus y_{19}$	$z_5 = t_{29} \wedge y_7$	$z_{11} = t_{33} \wedge y_4$
$t_6 = t_5 \oplus t_2$	$t_{25} = t_{21} \oplus t_{22}$	$t_{44} = t_{33} \oplus t_{37}$	$z_{12} = t_{43} \wedge y_{13}$
$t_9 = t_8 \oplus t_7$	$t_{27} = t_{24} \oplus t_{26}$	$t_{40} = t_{25} \oplus t_{39}$	$z_{13} = t_{40} \wedge y_5$
$t_{11} = t_{10} \oplus t_7$	$t_{31} = t_{22} \oplus t_{26}$	$t_{41} = t_{40} \oplus t_{37}$	$z_{15} = t_{42} \wedge y_9$
$t_{14} = t_{13} \oplus t_{12}$	$t_{28} = t_{25} \wedge t_{27}$	$t_{43} = t_{29} \oplus t_{40}$	$z_{16} = t_{45} \wedge y_{14}$
$t_{17} = t_4 \oplus t_{14}$	$t_{32} = t_{31} \wedge t_{30}$	$t_{45} = t_{42} \oplus t_{t41}$	$z_{17} = t_{41} \wedge y_8$
$t_{19} = t_9 \oplus t_{14}$	$t_{29} = t_{28} \oplus t_{22}$	$z_0 = t_{44} \wedge y_{15}$	
$t_{21} = t_{17} \oplus y_{20}$	$t_{33} = t_{33} \oplus t_{24}$	$z_1 = t_{37} \wedge y_6$	

Parallelization

- Parallelization level: $k = \frac{\text{architecture size}}{\text{nb of Sboxes}}$
- Generic method: $\mathsf{MC} = \lceil \frac{r}{k} \rceil + \lceil \frac{n \cdot t}{k} \rceil$
- Improved method: results for specific s-boxes

CryptoExperts

ENS
École Normale Supérieure

Performance Comparison in ARM

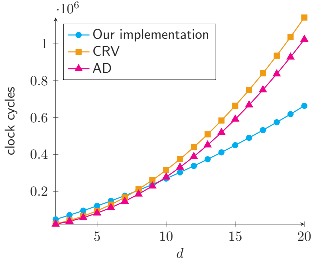


Figure: 16 Sboxes ($n = 8$), $k = 2 \rightarrow 31 \times 2$ multiplications .

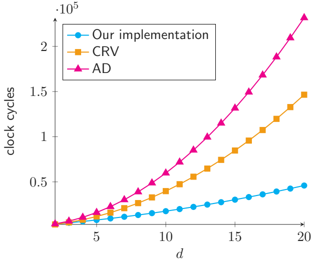


Figure: 16 Sboxes ($n = 4$), $k = 2 \rightarrow 3 \times 2$ multiplications.

Questions?