A Note on Algebraic Decomposition Method for Masked Implementation

Shoichi Hirose

University of Fukui, Fukui, Japan
hrs_shch@u-fukui.ac.jp

Abstract. Side-channel attacks are a serious problem in the implementation of cryptosystems. Masking is an effective countermeasure to this problem and it has been actively studied for implementations of block ciphers. An obstacle to efficient masked implementation is the complexity of an evaluation of multiplication, which is quadratic in the order of masking. A natural approach to this problem is to explore ways to reduce the number of multiplications required to compute an S-box. Algebraic decomposition is another interesting approach proposed by Carlet et al. in 2015, which gives a way to represent an S-box as composition of polynomials with low algebraic degrees. In this paper, for the algebraic decomposition, we propose to use a special type of low-algebraic-degree polynomials, which we call generalized multiplication (GM) polynomials. The masking scheme for multiplication can be applied to GM polynomials, which is more efficient than the masking scheme for general low-algebraic-degree polynomials. Our performance evaluation based on some experimental results shows the effectiveness of masked implementation using the proposed decomposition compared to masked implementation using the decomposition of Carlet et al.

Keywords: Algebraic decomposition · Boolean function · masking · S-box

1 Introduction

Background. Side channel attacks introduced by Kocher [15] are a serious problem in implementing cryptosystems. Chari et al. [6] proposed a sound approach based on secret sharing [3,20] against a class of side-channel attacks analyzing power consumption [16]. It is usually called masking [17] in this context. The d-th order masking splits each internal variable into $(d+1)$ shares so that at most d shares do not leak any information of the internal variable. The complexity of a successful side channel attack against a masked implementation was shown to be exponential in the masking order d [6].

Masked implementation has often been discussed for block ciphers. A block cipher can be manipulated as a function over the finite field $\mathbb{F}_2$ or its extensions. For a scalar multiplication or an addition, the number of operations to compute shares of the result from the shares of an input is $O(d)$. It is also $O(d)$

for a square, while it is $O(d^2)$ for a multiplication. Due to the difference, a multiplication is especially called a nonlinear multiplication. Thus, for efficient masked implementation of block ciphers, it is appropriate to explore ways to reduce the number of nonlinear multiplications to compute an S-box. Algebraic decomposition [5] is another approach, which gives a way to represent an S-box as composition of polynomials with low algebraic degrees.

Our Contribution. We propose a method for algebraic decomposition inspired by the method of Carlet et al. [5] and the method of Goudarzi et al. [11] for reducing the number of nonlinear multiplications. The proposed method can be applied to any function from $\{0,1\}^n$ to $\{0,1\}^n$ for even n. It regards a given function $h(x)$ as a pair of bivariate polynomials $(h_0(x_0,x_1),h_1(x_0,x_1))$, where $h_b : \mathbb{F}_{2^{n/2}} \times \mathbb{F}_{2^{n/2}} \rightarrow \mathbb{F}_{2^{n/2}}$ for $b \in \{0,1\}$. Then, it decomposes h using pairs of linear combinations of $x_0^{2^{i_0}} x_1^{2^{i_1}}$, where $0 \leq i_b \leq n/2-1$ for $b \in \{0,1\}$. We call such a pair of linear combinations a generalized multiplication (GM) polynomial. The difference between our proposed method and the method of Carlet et al. [5] is that the former uses GM polynomials instead of polynomials of low algebraic degrees such as 2 or 3. To a GM polynomial, the masking scheme for a multiplication can be applied, which is more efficient than the masking scheme for a polynomial of low algebraic degree [5,18,21]. Due to this property, for masked implementation, in terms of the number of evaluations of nonlinear functions (GM polynomials, polynomials of low algebraic degree, or multiplications), the proposed decomposition method is more efficient than the method of Goudarzi et al. [11] for $n = 4,6,8$ and than the method of Carlet et al. [5] for $n = 4,6$ and for $n = 8$ if the masking order is higher than 1. We further provide detailed analysis on performance of masked implementation using the proposed decomposition and the decomposition by Carlet et al. [5] and show the effectiveness of our proposal.

Related Work. Ishai, Sahai and Wagner presented a higher-order masking method for multiplication over $\mathbb{F}_2$ in their seminal paper [13]. Rivain and Prouff [19] generalized the method of Ishai et al. [13] to any finite field multiplication and applied it to the AES S-box. Carlet et al. [4] extended the method of Rivain and Prouff [19] and proposed a generic method for masking any S-box based on cyclotomic classes and the Knuth-Eve polynomial evaluation algorithm [9,14]. Coron, Roy and Vivek [8] improved the method of Carlet et al. [4] and presented a heuristic but generic method for masking any S-box. Goudarzi et al. [11] generalized the method of Coron, Roy and Vivek [8] and proposed a method treating any S-box from $\{0,1\}^{w_\mathrm{i}\nu}$ to $\{0,1\}^{w_\mathrm{o}\nu}$ as a tuple of polynomials over $\mathbb{F}_{2^\nu}$.

Inspired by the work of Coron, Roy and Vivek [8], Carlet et al. [5] introduced a new approach to decompose any S-box using polynomials of low algebraic degrees. They also presented a masking method for such polynomials. The masking method adapted to polynomials with algebraic degree 2 is essentially equivalent to the masking method of Coron et al. [7] for $x \cdot l(x)$, where $l(x)$ is a linear function. Its efficiency was improved by Zhang, Qiu and Zhou [21] and Qiu et al. [18].

Organization. Section 2 introduces some notations and definitions necessary for the discussions. Section 3 presents the proposed algebraic decomposition method using GM polynomials and its experimental results including an example of the decomposition of a 4-bit S-box. Section 5 shows results on performance evaluation of masked implementation using the proposed decomposition and the decomposition of Carlet et al. Section 6 gives a brief concluding remark.

A preliminary version [12] has been extended to this manuscript as follows: (i) The description of the proposed method in Sect. 3 has been made simpler; (ii) Section 3.3 has been added; (iii) Section 5 has been largely extended and provided detailed analysis of the proposed method.

2 Preliminaries

2.1 Notations

For a positive integer ν, let $\mathbb{F}_{2^\nu}$ be the finite field with 2^ν elements. For a set $\mathcal{S}$ and a variable s, let $s \twoheadleftarrow \mathcal{S}$ represent that an element chosen uniformly at random from $\mathcal{S}$ is assigned to s.

2.2 Functions over Finite Fields

A function $h : \mathbb{F}_{2^\nu}^{w_\mathrm{i}} \to \mathbb{F}_{2^\nu}^{w_\mathrm{o}}$ is a tuple of functions $(h_0, h_1, \ldots, h_{w_\mathrm{o}-1})$, where $h_j : \mathbb{F}_{2^\nu}^{w_\mathrm{i}} \to \mathbb{F}_{2^\nu}$ for $0 \le j \le w_\mathrm{o} - 1$. h_j can be represented as

$$h_j(x_0, x_1, \ldots, x_{w_\mathrm{i}-1}) = \sum_{k_0=0}^{2^\nu-1} \cdots \sum_{k_{w_\mathrm{i}-1}=0}^{2^\nu-1} \alpha_{j,k_0,\ldots,k_{w_\mathrm{i}-1}} x_0^{k_0} \cdots x_{w_\mathrm{i}-1}^{k_{w_\mathrm{i}-1}} \;,$$

where $\alpha_{j,k_0,\ldots,k_{w_\mathrm{i}-1}} \in \mathbb{F}_{2^\nu}$. We only refer to the cases that $(w_\mathrm{i}, w_\mathrm{o}) \in \{1,2\} \times \{1,2\}$ in the remaining parts.

Definition 1 (Algebraic degree). *The algebraic degree of a function $h : \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu}$ such that*

$$h(x) = \sum_{k=0}^{2^\nu-1} \alpha_k x^k$$

is the maximum of the Hamming weight of the binary representation of k such that $\alpha_k \neq 0$ for $0 \le k \le 2^\nu - 1$.

Definition 2 (Linearized polynomial). *A function $\ell : \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu}$ is called a linearized polynomial if it can be represented as*

$$\ell(x) = \sum_{k=0}^{\nu-1} \alpha_k x^{2^k} \;,$$

where $\alpha_k \in \mathbb{F}_{2^\nu}$.

For any linearized polynomial $\ell(x)$, its algebraic degree is 1, and it holds that

$$\ell\Big(\sum_{i=0}^{d} x_i\Big) = \sum_{i=0}^{d} \ell(x_i) \ . \tag{1}$$

We introduce generalized multiplication polynomials, which are used in our proposed decomposition method:

Definition 3 (Generalized multiplication polynomial). *We call a function $m : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ a generalized multiplication (GM) polynomial if it can be represented as $m(x) = (m_0(x), m_1(x))$ such that, for $b \in \{0,1\}$, $m_b : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu}$ and*

$$m_b(x) = \sum_{k=0}^{\nu-1}\sum_{l=0}^{\nu-1} \alpha_{b,k,l} x_0^{2^k} x_1^{2^l} \ ,$$

where $x = (x_0, x_1) \in \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ and $\alpha_{b,k,l} \in \mathbb{F}_{2^\nu}$.

For any GM polynomial $m(x_0, x_1)$,

$$m\Big(\sum_{i=0}^{d} x_{0,i}, \sum_{j=0}^{d} x_{1,j}\Big) = \sum_{i=0}^{d}\sum_{j=0}^{d} m(x_{0,i}, x_{1,j}) \tag{2}$$

holds since

$$\Big(\sum_{i=0}^{d} x_{0,i}\Big)^{2^k}\Big(\sum_{j=0}^{d} x_{1,j}\Big)^{2^l} = \Big(\sum_{i=0}^{d} x_{0,i}^{2^k}\Big)\Big(\sum_{j=0}^{d} x_{1,j}^{2^l}\Big) = \sum_{i=0}^{d}\sum_{j=0}^{d} x_{0,i}^{2^k} x_{1,j}^{2^l} \ . \tag{3}$$

For Eq. (3), multiplication is the case that $k = l = 0$.

3 Algebraic Decomposition

In the remaining parts of the paper, ν is a positive integer and $n = 2\nu$.

3.1 Algebraic decomposition using GM polynomials

A function $h : \{0,1\}^n \to \{0,1\}^n$ is regarded as $h(x) = (h_0(x_0, x_1), h_1(x_0, x_1))$, where $x = (x_0, x_1)$ and $h_b : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu}$ for $b \in \{0,1\}$. The decomposition of h proceeds as follows:

1. For $1 \le i \le r$, $f_i : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ is a GM polynomial chosen uniformly at random.
2. For $1 \le i \le r$, $g_i : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ is defined as follows:

$$\begin{aligned} g_1(x) &= f_1(x) \ , \\ g_2(x) &= f_2(g_1(x) + \ell_0(x_0) + \ell_1(x_1)) \ , \\ g_i(x) &= f_i(g_{i-1}(x)) \qquad \text{for } 3 \le i \le r, \end{aligned}$$

where $\ell_b : \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ is a tuple of linearized polynomials for $b \in \{0,1\}$. Namely, $\ell_b(x_b) = (\ell_{b,0}(x_b), \ell_{b,1}(x_b))$ and $\ell_{b,0}$ and $\ell_{b,1}$ are linearized polynomials over $\mathbb{F}_{2^\nu}$. $\ell_{b,0}$ and $\ell_{b,1}$ are chosen uniformly at random.

3. For $1 \leq j \leq t$, $q_j : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ is defined as follows:
$$q_j(x) = \sum_{i=1}^{r} \Big(\ell_{j,i,0}(g_{i,0}(x)) + \ell_{j,i,1}(g_{i,1}(x)) \Big) + \ell_{j,0,0}(x_0) + \ell_{j,0,1}(x_1) \ , \qquad (4)$$
where $\ell_{j,0,0}, \ell_{j,0,1}, \ldots, \ell_{j,i,0}, \ell_{j,i,1}$ are tuples of linearized polynomials over $\mathbb{F}_{2^\nu}$ chosen uniformly at random.
4. Search GM polynomials $\mu_1, \ldots, \mu_t$ over $\mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$, tuples $\lambda_{0,0}, \lambda_{0,1}, \ldots, \lambda_{r,0}$, $\lambda_{r,1}$ of linearized polynomials over $\mathbb{F}_{2^\nu}$ and a tuple δ of constants in $\mathbb{F}_{2^\nu}$ satisfying
$$\begin{aligned} h(x) = \sum_{j=1}^{t} \mu_j(q_j(x)) + \sum_{i=1}^{r} \Big(\lambda_{i,0}(g_{i,0}(x)) + \lambda_{i,1}(g_{i,1}(x)) \Big) \\ + \lambda_{0,0}(x_0) + \lambda_{0,1}(x_1) + \delta \ . \qquad (5) \end{aligned}$$
If the search fails, then return to the first step.

The amount of computation for an evaluation of h based on the decomposition is summarized in Table 1.

Table 1. The amount of computation based on the proposed decomposition

# GM polynomials	$r + t$
# tuples of linearized polynomials	$2((r+1)(t+1)+1)$
# additions	$2((r+1)(t+1)+1)$

The algebraic decomposition method of Carlet et al. [5] is given in Appendix A. Essentially, our proposed method uses GM polynomials instead of polynomials with low algebraic degrees for the decomposition method of Carlet et al. Another minor difference is that our proposed method uses linearized polynomials to compose g_2 in the second step. Our preliminary experiments suggest that our proposed method does not work well without these linearized polynomials.

As well as the decomposition method of Carlet et al. [5], the search in the 4th step above can be carried out by solving a system of linear equations over $\mathbb{F}_{2^\nu}$:
$$A \cdot \boldsymbol{v}_b = \boldsymbol{c}_b \qquad (6)$$
for $b \in \{0,1\}$. $\boldsymbol{c}_b$ is a 2^n-dimensional column vector over $\mathbb{F}_{2^\nu}$ such that
$$\boldsymbol{c}_b = (h_b(e_1), h_b(e_2), \ldots, h_b(e_{2^n}))^{\mathrm{T}} \ ,$$

where $e_i = (e_{i,0}, e_{i,1}) \in \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$ and $e_{i_1} \neq e_{i_2}$ if $i_1 \neq i_2$. $\boldsymbol{v}_b$ is the column vector of unknowns representing the coefficients of GM polynomials $\mu_{1,b}, \ldots, \mu_{t,b}$, linearized polynomials $\lambda_{0,0,b}, \lambda_{0,1,b}, \ldots, \lambda_{r,0,b}, \lambda_{r,1,b}$ and δ_b. The matrix A, which does not depend on the value of b, is defined as follows:

$$A = \left(A_{q_1}\ A_{q_2}\ \cdots\ A_{q_t}\ A_{g_{1,0}}\ \cdots\ A_{g_{r,0}}\ A_{g_{1,1}}\ \cdots\ A_{g_{r,1}}\ A_{e_{*,0}}\ A_{e_{*,1}}\ \mathbf{1}\right)\ .$$

For $1 \leq j \leq t$, A_{q_j} is a $2^n \times \nu^2$ matrix such that

$$A_{q_j} = \begin{pmatrix} Q_{j,1}^{(0,0)} \cdots Q_{j,1}^{(0,\nu-1)} & Q_{j,1}^{(1,0)} \cdots Q_{j,1}^{(1,\nu-1)} & \cdots & Q_{j,1}^{(\nu-1,0)} \cdots Q_{j,1}^{(\nu-1,\nu-1)} \\ Q_{j,2}^{(0,0)} \cdots Q_{j,2}^{(0,\nu-1)} & Q_{j,2}^{(1,0)} \cdots Q_{j,2}^{(1,\nu-1)} & \cdots & Q_{j,2}^{(\nu-1,0)} \cdots Q_{j,2}^{(\nu-1,\nu-1)} \\ \cdots & \cdots & \cdots & \cdots \\ Q_{j,2^n}^{(0,0)} \cdots Q_{j,2^n}^{(0,\nu-1)} & Q_{j,2^n}^{(1,0)} \cdots Q_{j,2^n}^{(1,\nu-1)} & \cdots & Q_{j,2^n}^{(\nu-1,0)} \cdots Q_{j,2^n}^{(\nu-1,\nu-1)} \end{pmatrix}$$

and $Q_{j,i}^{(k,l)} = q_{j,0}(e_i)^{2^k} q_{j,1}(e_i)^{2^l}$ for $1 \leq i \leq 2^n$, $0 \leq k \leq \nu - 1$, and $0 \leq l \leq \nu - 1$. For $1 \leq i \leq r$ and $b' \in \{0, 1\}$, $A_{g_{i,b'}}$ is a $2^n \times \nu$ matrix represented as follows:

$$\begin{pmatrix} g_{i,b'}(e_1)^{2^0} & g_{i,b'}(e_1)^{2^1} & \cdots & g_{i,b'}(e_1)^{2^{\nu-1}} \\ g_{i,b'}(e_2)^{2^0} & g_{i,b'}(e_2)^{2^1} & \cdots & g_{i,b'}(e_2)^{2^{\nu-1}} \\ & \cdots & & \\ g_{i,b'}(e_{2^n})^{2^0} & g_{i,b'}(e_{2^n})^{2^1} & \cdots & g_{i,b'}(e_{2^n})^{2^{\nu-1}} \end{pmatrix}\ .$$

For $b' \in \{0, 1\}$,

$$A_{e_{*,b'}} = \begin{pmatrix} e_{1,b'}^{2^0} & e_{1,b'}^{2^1} & \cdots & e_{1,b'}^{2^{\nu-1}} \\ e_{2,b'}^{2^0} & e_{2,b'}^{2^1} & \cdots & e_{2,b'}^{2^{\nu-1}} \\ & \cdots & & \\ e_{2^n,b'}^{2^0} & e_{2^n,b'}^{2^1} & \cdots & e_{2^n,b'}^{2^{\nu-1}} \end{pmatrix}$$

is a $2^n \times \nu$ matrix. $\mathbf{1}$ is the column vector whose 2^n coordinates equal 1.

The matrix A has 2^n rows and $t \cdot \nu^2 + 2(r + 1)\nu + 1$ columns. In order for the system of linear equations Eq.(6) to have a solution for any $\boldsymbol{c}_b$, the rank of A must be 2^n and it is required that

$$t \cdot n^2/4 + (r + 1)n + 1 \geq 2^n\ . \tag{7}$$

It is also required that the algebraic degree of the polynomial in the right side of Eq.(5) is $n/2$ with respect to each of x_0 and x_1. Thus,

$$2^r \geq n/2\ . \tag{8}$$

Once we obtain a matrix A with its rank 2^n from some $g_1, \ldots, g_r$ and $q_1, \ldots, q_t$, we can use it to decompose any function $h : \{0, 1\}^n \to \{0, 1\}^n$.

3.2 Experimental Result

Table 2 lists the values of parameters of successful decomposition minimizing the number of GM polynomials, that is, $r + t$. For $n = 4, 6$, all optimal values satisfying inequalities (7) and (8) and minimizing $r+t$ are achieved. For $n=8$, optimal values are also achieved. On the other hand, though $(r,t)=(2,15)$ are also optimal, they cannot be achieved by one hundred trials. For $n=10$, all optimal values $(r,t)=(3,40),(4,39)$ as well as $(r,t)=(3,41),(4,40)$ cannot be achieved by one hundred trials. Notice that, for $n=4$ and $(r,t)=(1,2)$, two tuples of linearized polynomials used to compose g_2 is not necessary for the proposed algebraic decomposition method.

Table 2. Achievable parameters minimizing $r+t$. #GMP represents the number of GM polynomials. #TLP represents the number of tuples of linearized polynomials. #Add represents the number of additions.

n	(r,t)	#GMP	#TLP	#Add
4	$(1,2)$	3	12	12
	$(2,1)$	3	14	14
6	$(2,5)$	7	38	38
8	$(3,14)$	17	122	122
10	$(5,39)$	44	482	482

Remark 1. Precisely, in the third step, we only searched the tuples of linearized polynomials $\ell_{j,i,0}=(\ell_{j,i,0,0},\ell_{j,i,0,1})$ and $\ell_{j,i,1}=(\ell_{j,i,1,0},\ell_{j,i,1,1})$ such that $\ell_{j,i,0,1}=0$ and $\ell_{j,i,1,0}=0$ for $i\geq 1$. We found no degradation in the parameters minimizing $r+t$ achieved by this restricted search.

3.3 Example: Decomposition of SKINNY 4-Bit S-box

As an example, a decomposition of the 4-bit S-box of the tweakable block cipher SKINNY [2], which is given in Table 3, is presented. The decomposition is performed with $(r,t)=(1,2)$.

Table 3. The 4-bit S-box of SKINNY

input	0 1 2 3 4 5 6 7 8 9 a b c d e f
output	c 6 9 0 1 a 2 b 3 8 5 d 4 e 7 f

For $f_1(x_0,x_1)=(f_{1,0}(x_0,x_1),f_{1,1}(x_0,x_1))$,

$$f_{1,0}(x_0,x_1)=0x_0x_1+2x_0x_1^2+0x_0^2x_1+3x_0^2x_1^2\ ,$$
$$f_{1,1}(x_0,x_1)=0x_0x_1+1x_0x_1^2+2x_0^2x_1+3x_0^2x_1^2$$

are chosen. For Eq.(4),

$$q_1(x)=\ell_{1,1,0}(g_{1,0}(x))+\ell_{1,1,1}(g_{1,1}(x))+\ell_{1,0,0}(x_0)+\ell_{1,0,1}(x_1)\ ,$$
$$q_2(x)=\ell_{2,1,0}(g_{1,0}(x))+\ell_{2,1,1}(g_{1,1}(x))+\ell_{2,0,0}(x_0)+\ell_{2,0,1}(x_1)\ ,$$

and

$$\begin{aligned}
\ell_{1,1,0}(y_0) &= (0y_0 + 1y_0^2, 0) \ , \\
\ell_{1,1,1}(y_1) &= (0, 1y_1 + 1y_1^2) \ , \\
\ell_{1,0,0}(x_0) &= (3x_0 + 0x_0^2, 0x_0 + 2x_0^2) \ , \\
\ell_{1,0,1}(x_1) &= (0x_1 + 1x_1^2, 0x_1 + 1x_1^2) \ , \\
\ell_{2,1,0}(y_0) &= (2y_0 + 2y_0^2, 0) \ , \\
\ell_{2,1,1}(y_1) &= (0, 2y_1 + 1y_1^2) \ , \\
\ell_{2,0,0}(x_0) &= (1x_0 + 3x_0^2, 1x_0 + 3x_0^2) \ , \\
\ell_{2,0,1}(x_1) &= (3x_1 + 3x_1^2, 2x_1 + 0x_1^2)
\end{aligned}$$

are chosen. Notice that the second linearized polynomials of $\ell_{1,1,0}$ and $\ell_{2,1,0}$ are 0 as well as the first linearized polynomials of $\ell_{1,1,1}$ and $\ell_{2,1,1}$ as described in Remark 1. For Eq.(5),

$$\begin{aligned}
h(x) = \mu_1(q_1(x)) + \mu_2(q_2(x)) \\
+ \lambda_{1,0}(g_{1,0}(x)) + \lambda_{1,1}(g_{1,1}(x)) + \lambda_{0,0}(x_0) + \lambda_{0,1}(x_1) + \delta \ ,
\end{aligned}$$

where

$$\begin{aligned}
\mu_{1,0}(z_0, z_1) &= 0z_0z_1 + 0z_0z_1^2 + 1z_0^2z_1 + 2z_0^2z_1^2 \ , \\
\mu_{1,1}(z_0, z_1) &= 0z_0z_1 + 0z_0z_1^2 + 1z_0^2z_1 + 2z_0^2z_1^2 \ , \\
\mu_{2,0}(z_0, z_1) &= 2z_0z_1 + 2z_0z_1^2 + 3z_0^2z_1 + 0z_0^2z_1^2 \ , \\
\mu_{2,1}(z_0, z_1) &= 2z_0z_1 + 1z_0z_1^2 + 3z_0^2z_1 + 0z_0^2z_1^2 \ ,
\end{aligned}$$

$$\begin{aligned}
\lambda_{1,0}(y_0) &= (3y_0 + 3y_0^2, 0y_0 + 0y_0^2) \ , \\
\lambda_{1,1}(y_1) &= (0y_1 + 0y_1^2, 1y_1 + 0y_1^2) \ , \\
\lambda_{0,0}(x_0) &= (0x_0 + 0x_0^2, 3x_0 + 2x_0^2) \ , \\
\lambda_{0,1}(x_1) &= (2x_1 + 3x_1^2, 1x_1 + 0x_1^2) \ ,
\end{aligned}$$

and $\delta = (3, 0)$ are obtained by solving the system of linear equations (6).

4 Comparison

Table 4 shows the smallest number of nonlinear functions achieved by our decomposition method and the decomposition methods of Carlet et al. [5] and of Goudarzi et al. [11]. For algebraic decomposition of Carlet et al., methods using polynomials of algebraic degrees 2 and/or 3 were presented, and the most efficient method was shown to be the method using only quadratic polynomials (polynomials of algebraic degree 2), which is mentioned in Table 4. The decomposition method of Goudarzi et al. [11] for reducing the number of multiplications is able to process any function over $\{0,1\}^n$ by regarding it as a function over $\mathbb{F}_{2^\xi}^{n/\xi}$ for any ξ such that $\xi \mid n$. Table 4 mentions only the case that $\xi = n/2$.

In terms of the number of multiplications or GM polynomials, our decomposition is slightly more efficient than the decomposition of Goudarzi et al. [11]. On the other hand, if implementation adopts table lookup for evaluation of multiplications or GM polynomials, then our decomposition needs a lookup table for each GM polynomial, while the decomposition of Goudarzi et al. needs just a single lookup table for multiplication. Thus, the total table size of our decomposition is $2(r+t)$ times as large as that of the decomposition of Goudarzi et al. For example, for $n = 8$, the total table size of GM polynomials for our decomposition is $4352(= 17 \times 256)$ Bytes.

In terms of the number of quadratic polynomials or GM polynomials, our decomposition does not seem so good as decomposition of Carlet et al. [5] apparently from Table 4. For masked implementations, on the other hand, the masking scheme for GM polynomials is more efficient than the masking scheme for quadratic polynomials. Thus, in the next section, we will compare the performance of masked implementation using our proposed method with that of masked implementation using the method of Carlet et al. [5].

Table 4. Comparison of best achievable parameters

	$n = 4$	$n = 6$	$n = 8$
# quadratic polynomials [5]	3	5	11
# multiplications [11]	4	9	18
# GM polynomials (Ours)	3	7	17

5 Application to Masking

In this section, the decomposition method of Carlet et al. [5] is referred to as the CPRR15 decomposition.

Algorithm 1 presents an algorithm of the d-th order masking for a GM polynomial. Due to the property of GM polynomials shown by Eq. (2), it is similar to the d-th order masking for multiplication. For reference, the algorithm of the d-th order masking for a quadratic polynomial [18,21] is shown in Algorithm 2. It is a modified version of the algorithm presented by Carlet et al. [5]. It reduces the required number of random sequences. Both of Algorithms 1 and 2 were shown to satisfy strong d-non-interference (d-SNI) [1,21].

Table 5 shows complexity of the d-th order masking for an evaluation of a quadratic polynomial or a GM polynomial. From Tables 4 and 5, in terms of the number of evaluations of nonlinear functions (quadratic polynomials or GM polynomials), the proposed decomposition yields more efficient masking for n-bit S-boxes than the CPRR15 decomposition using quadratic polynomials for $n = 4, 6$, and for $n = 8$ if $d \geq 2$.

Algorithm 1: The d-th order masking for a GM polynomial $m : \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu} \to \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$, where $\nu = n/2$

```
input  : Shares $(a_0, \ldots, a_d)$ of $a$ and $(b_0, \ldots, b_d)$ of $b$
output: Shares $(c_0, c_1, \ldots, c_d)$ of $c = m(a, b)$
for $i = 0$ to $d$ do
    for $j = i + 1$ to $d$ do
        $r_{i,j} \twoheadleftarrow \mathbb{F}_{2^\nu} \times \mathbb{F}_{2^\nu}$;
        $r_{j,i} \leftarrow (r_{i,j} + m(a_i, b_j)) + m(a_j, b_i)$;
for $i = 0$ to $d$ do
    $c_i \leftarrow m(a_i, b_i)$;
    for $j = 0$ to $d$ do
        if $j \neq i$ then
            $c_i \leftarrow c_i + r_{i,j}$;
return $(c_0, c_1, \ldots, c_d)$
```

Let us see more details on the total cost required to evaluate masked implementation of a function over $\{0, 1\}^n$. Table 6 shows the values of parameters (r', t') of successful decomposition minimizing the number of quadratic polynomials $(r' + t')$ of the CPRR15 decomposition. The amount of computation for an evaluation of a function based on the CPRR15 decomposition is summarized in Table 7. The parameters (r', t') defined in Appendix A correspond to the parameters (r, t) of our proposed decomposition. Goudarzi and Rivain [10] claimed that $\lambda_0(x), \lambda_1(g_1(x)), \ldots, \lambda_{r'}(g_{r'}(x))$ can be avoided in Eq. (9) for every $n \in \{4, \ldots, 10\}$. Namely, it can be replaced with

$$h(x) = \sum_{j=1}^{t'} \mu_j(q_j(x)) \enspace .$$

The amount of computation in Table 7 is evaluated based on this fact.

To evaluate the total cost for the evaluation of masked implementation of a function, we assume that the evaluation of a quadratic polynomial, a GM polynomial or a (tuple of) linearized polynomial(s) is done by a table-lookup. Then, the total numbers of table-lookups and additions of the d-th order masked implementation are as follows: For the CPRR15 decomposition,

Table-lookups $(r' + t')(d + 1)(2d + 1) + (r' + 1)t'(d + 1)$,
Additions $(r' + t')(4d(d + 1) + 1) + (r't' + t' - 1)(d + 1)$,

and for our decomposition,

Table-lookups

$$(r + t)(d + 1)^2 + \begin{cases} 2(r + 1)(t + 1)(d + 1) & \text{if } r = 1, \\ 2((r + 1)(t + 1) + 1)(d + 1) & \text{otherwise,} \end{cases}$$

Algorithm 2: The d-th order masking for a quadratic polynomial $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$

```
input  : Shares $(a_0, a_1, \ldots, a_d)$ of $a$
output: Shares $(b_0, b_1, \ldots, b_d)$ of $b = f(a)$
for $i = 0$ to $d$ do
    for $j = i + 1$ to $d$ do
        $r_{i,j} \twoheadleftarrow \mathbb{F}_{2^n}$;
        $t_{i,j} \leftarrow f(r_{i,j})$;
        $t_{j,i} \leftarrow f(a_i + r_{i,j}) + f((a_i + r_{i,j}) + a_j) + f(a_j + r_{i,j})$;
for $i = 0$ to $d$ do
    $b_i \leftarrow f(a_i)$;
    for $j = 0$ to $d$ do
        if $j \neq i$ then
            $b_i \leftarrow b_i + t_{i,j}$;
if $d$ is odd then
    $b_1 \leftarrow b_1 + f(0)$;
return $(b_0, b_1, \ldots, b_d)$
```

Table 5. Complexity of the d-th order masking. "# eval," "# rand" and "# add," represent the required number of evaluations of a nonlinear function (a quadratic polynomial or a GM polynomial), random sequences and additions, respectively.

	# eval	# rand	# add
Quadratic poly. eval. (Algorithm 2)	$(d+1)(2d+1)$	$d(d+1)/2$	$4d(d+1)+1$
GP poly. eval. (Algorithm 1)	$(d+1)^2$	$d(d+1)/2$	$2d(d+1)$

Additions $(r+t)2d(d+1) + 2((r+1)(t+1)+1)(d+1)$.

They are also shown in Fig. 1, Fig. 2, and Fig. 3 for a function over $\{0,1\}^n$ for $n = 4, 6, 8$, respectively. In terms of the number of operations, the proposed decomposition is more efficient than the CPRR15 decomposition if $d \geq 3, 11, 21$ for $n = 4, 6, 8$, respectively.

In terms of the total table size, the proposed decomposition is more efficient than the CPRR15 decomposition. The total table size of the CPRR15 decomposition is

$$(r' + t')n2^n + (r'+1)t'n2^n \text{ bits},$$

and that of our decomposition is

$$(r+t)n2^n + \begin{cases} 2(r+1)(t+1)n2^{n/2} \text{ bits}, & \text{if } r = 1, \\ 2((r+1)(t+1)+1)n2^{n/2} \text{ bits}, & \text{otherwise}, \end{cases}$$

where the table-size reduction implied by Remark 1 is not taken into consideration. The total table sizes are shown in Table 8 for $n = 4, 6, 8$.

Table 6. Achievable parameters minimizing $r' + t'$ of the CPRR15 decomposition for functions over $\{0,1\}^n$

n	(r', t')
4	$(1,2)$, $(2,1)$
6	$(2,3)$
8	$(2,9)$, $(3,8)$

Table 7. The amount of computation based on the CPRR15 decomposition

# quadratic polynomials	$r' + t'$
# linearized polynomials	$(r'+1)t'$
# additions	$r't' + t' - 1$

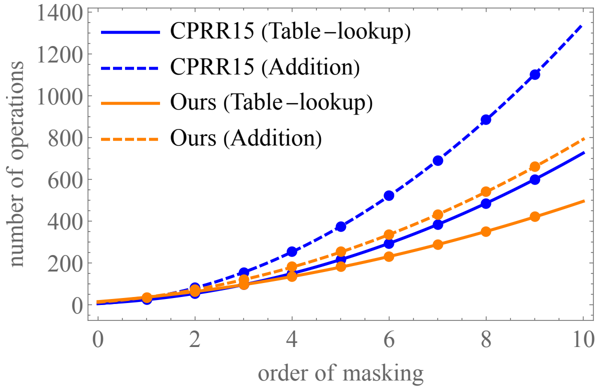

Fig. 1. The number of operations for $n = 4$. CPRR15 adopts $(r', t') = (2, 1)$, which yields smaller values than $(r', t') = (1, 2)$. Ours adopts $(r, t) = (1, 2)$.

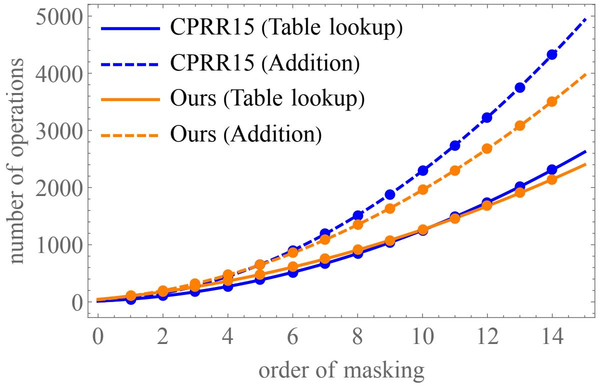

Fig. 2. The number of operations for $n = 6$

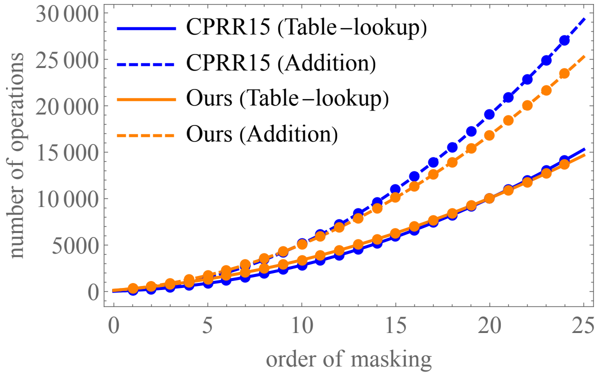

Fig. 3. The number of operations for $n = 8$. CPRR15 adopts $(r', t') = (2, 9)$, which yields smaller values than $(r', t') = (3, 8)$.

Table 8. Total table size (Bytes). CPRR15 adopts $(r', t') = (2, 1)$ for $n = 4$ and $(r', t') = (2, 9)$ for $n = 8$. They yield smaller values than $(r', t') = (1, 2)$ for $n = 4$ and $(r', t') = (3, 8)$ for $n = 8$, respectively. Ours adopts $(r, t) = (1, 2)$ for $n = 4$.

	$n = 4$	$n = 6$	$n = 8$
CPRR15	48	672	9728
Ours	48	564	6304

For the amount of random sequences, from Tables 2, 5 and 6, the CPRR15 decomposition is more efficient than the proposed decomposition for $n = 4, 6, 8$.

6 Conclusion

We have presented an algebraic decomposition method for masked implementation of any S-box. Essentially, our proposal is to use GM polynomials instead of polynomials with low algebraic degrees for decomposition. Future work is performance evaluation of masked implementation of S-boxes in software using the proposed decomposition method.

Acknowledgements

We would like to thank the reviewers for valuable comments. This work was supported in part by JSPS KAKENHI Grant Number JP18H05289.

A Algebraic decomposition by Carlet et al.

The decomposition of $h : \{0, 1\}^n \to \{0, 1\}^n$ proceeds as follows. h is regarded as a function over $\mathbb{F}_{2^n}$.

1. For $1 \leq i \leq r'$, $f_i : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is a low-algebraic-degree polynomial chosen uniformly at random.
2. For $1 \leq i \leq r'$, $g_i : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is defined as follows:

$$g_1(x) = f_1(x) \ ,$$
$$g_i(x) = f_i(g_{i-1}(x)) \quad \text{for } 2 \leq i \leq r' \ .$$

3. For $1 \leq j \leq t'$, $q_j : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is defined as follows:

$$q_j(x) = \sum_{i=1}^{r'} \ell_{j,i}(g_i(x)) + \ell_{j,0}(x) \ ,$$

where $\ell_{j,0}, \ell_{j,1}, \ldots, \ell_{j,r'}$ are linearized polynomials over $\mathbb{F}_{2^n}$ chosen uniformly at random.

4. Search low-algebraic-degree polynomials $\mu_1, \ldots, \mu_{t'}$ over $\mathbb{F}_{2^n}$, linearized polynomials $\lambda_0, \lambda_1, \ldots, \lambda_{r'}$ over $\mathbb{F}_{2^n}$, and a constant $\delta \in \mathbb{F}_{2^n}$ satisfying

$$h(x) = \sum_{j=1}^{t'} \mu_j(q_j(x)) + \sum_{i=1}^{r'} \lambda_i(g_i(x)) + \lambda_0(x) \ . \tag{9}$$

If the search fails, then return to the first step.

References

1. Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P., Grégoire, B., Strub, P., Zucchini, R.: Strong non-interference and type-directed higher-order masking. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 116–129. ACM (2016). https://doi.org/10.1145/2976749.2978427
2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5
3. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of AFIPS 1979 National Computer Conference. vol. 48, pp. 313–317 (1979)
4. Carlet, C., Goubin, L., Prouff, E., Quisquater, M., Rivain, M.: Higher-order masking schemes for S-boxes. In: Canteaut, A. (ed.) Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7549, pp. 366–384. Springer (2012). https://doi.org/10.1007/978-3-642-34047-5_21
5. Carlet, C., Prouff, E., Rivain, M., Roche, T.: Algebraic decomposition for probing security. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 742–763. Springer (2015). https://doi.org/10.1007/978-3-662-47989-6_36

6. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 398–412. Springer (1999). https://doi.org/10.1007/3-540-48405-1_26
7. Coron, J., Prouff, E., Rivain, M., Roche, T.: Higher-order side channel security and mask refreshing. In: Moriai, S. (ed.) Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8424, pp. 410–424. Springer (2013). https://doi.org/10.1007/978-3-662-43933-3_21
8. Coron, J., Roy, A., Vivek, S.: Fast evaluation of polynomials over binary finite fields and application to side-channel countermeasures. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8731, pp. 170–187. Springer (2014). https://doi.org/10.1007/978-3-662-44709-3_10
9. Eve, J.: The evaluation of polynomials. Numerische Mathematik **6**, 17–21 (1964)
10. Goudarzi, D., Rivain, M.: How fast can higher-order masking be in software? In: Coron, J., Nielsen, J.B. (eds.) Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10210, pp. 567–597 (2017). https://doi.org/10.1007/978-3-319-56620-7_20
11. Goudarzi, D., Rivain, M., Vergnaud, D., Vivek, S.: Generalized polynomial decomposition for S-boxes with application to side-channel countermeasures. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 154–171. Springer (2017). https://doi.org/10.1007/978-3-319-66787-4_8
12. Hirose, S.: Another algebraic decomposition method for masked implementation. In: Chen, B., Huang, X. (eds.) AC3 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 386, pp. 105–114. Springer (2021). https://doi.org/10.1007/978-3-030-80851-8_8
13. Ishai, Y., Sahai, A., Wagner, D.A.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 463–481. Springer (2003). https://doi.org/10.1007/978-3-540-45146-4_27
14. Knuth, D.E.: Evaluation of polynomials by computer. Commun. ACM **5**(12), 595–599 (1962). https://doi.org/10.1145/355580.369074
15. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer (1996). https://doi.org/10.1007/3-540-68697-5_9
16. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceed-

ings. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999). https://doi.org/10.1007/3-540-48405-1_25

17. Messerges, T.S.: Securing the AES finalists against power analysis attacks. In: Schneier, B. (ed.) Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1978, pp. 150–164. Springer (2000). https://doi.org/10.1007/3-540-44706-7_11
18. Qiu, S., Zhang, R., Zhou, Y., Cheng, W.: Corrections to ”Further improving efficiency of higher-order masking schemes by decreasing randomness complexity”. Cryptology ePrint Archive, Report 2017/1244 (2017)
19. Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: Mangard, S., Standaert, F. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6225, pp. 413–427. Springer (2010). https://doi.org/10.1007/978-3-642-15031-9_28
20. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
21. Zhang, R., Qiu, S., Zhou, Y.: Further improving efficiency of higher order masking schemes by decreasing randomness complexity. IEEE Trans. Inf. Forensics Secur. **12**(11), 2590–2598 (2017). https://doi.org/10.1109/TIFS.2017.2713323