A Smaller and Faster Variant of RSM

Noritaka Yamashita, Kazuhiko Minematsu, Toshihiko Okamura and Yukiyasu Tsunoo
NEC Corporation
1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8666, Japan
Email: n-yamashita@ap.jp.nec.com

***Abstract*—Masking is one of the major countermeasures against side-channel attacks to cryptographic modules. Nassar *et al*. recently proposed a highly efficient masking method, called Rotating S-boxes Masking (RSM), which can be applied to a block cipher based on Substitution-Permutation Network. It arranges multiple masked S-boxes in parallel, which are rotated in each round. This rotation requires remasking process for each round to adjust current masks to those of the S-boxes. In this paper, we propose a method for reducing the complexity of RSM further by omitting the remasking process when the linear diffusion layer of the encryption algorithm has a certain algebraic property. Our method can be applied to AES with a reduced complexity from RSM, while keeping the equivalent security level.**

I. Introduction

Side-channel attacks are drawing attention as a threat to cryptographic modules. They exploit various types of side-channel information, including processing time [1], power consumption [2], and electromagnetic waves [3], to recover confidential information in the module. Among them Differential Power Analysis (DPA) is a strong attacking technique based on a statistical analysis of power consumption traces [2]. Since DPA was introduced by Kocher [2], numerous variants of DPA have been proposed [4]–[7]. Many countermeasures against DPA and the variants have therefore been proposed. The basic strategies of the countermeasures are classified into hiding [8] and masking [9]–[13]. Each strategy has its own pros and cons, however, we here focus on the masking for its applicability and effectiveness. Masking involves adding a random value (mask value) to the intermediate value. This makes the power consumption independent of the intermediate value of the process stored in a register, making analysis difficult. Messerges *et al*. proposed two masking countermeasures: boolean and arithmetic maskings [9]. In [9], when one wants to protect an encryption algorithm using a non-linear function called S-box, a mask value is randomly generated and then a masked S-box is computed before encryption starts. A countermeasure is shown in the reference [9]. However, it has a problem, namely the large size of RAM is required to calculate the masked S-box. In addition, calculation of the masked S-box takes time. Itoh *et al*. proposed a fixed value masking [10] to reduce the RAM and computation time for masked S-box. In [10], pairs of masks and masked S-boxes are stored in ROM in advance. With this method, one of the pairs is selected randomly during the encryption process. Since this method does not require recalculation of masked S-boxes, the RAM size does not increase and the calculation time can be shortened. This method, however, increases the ROM size because multiple masks and masked S-boxes have to be stored in ROM in advance. In addition, with this method, only one of masked S-boxes is selected during the encryption process. This implies an inefficiency in circuit usage.

Nassar *et al*. proposed Rotating S-boxes Masking (RSM) [13] to solve these problems of [9] and [10]. RSM assumes the underlying encryption algorithm to use S-boxes in parallel, such as a blockcipher based on the Substitution-Permutation Network (SPN), and arranges multiple masked S-boxes in parallel and rotates them in each round. This rotation allows different masks to be applied for each round, thereby improving the resistance against DPA. In addition, multiple masked S-boxes can be used in parallel, thereby improving the efficiency in terms of circuit usage and decreasing the vulnerability to a variant of DPA, called second-order zero-offset DPA [14]. On the other hand, however, the masks are varying in the linear operation for each round and they have to be remasked for each of the round processes. Because of this remasking, RSM requires additional memories to store data and circuits for remasking, and additional processing time.

In this paper, we propose a variant of RSM that simplifies some of the masking processes. Our method requires that the linear diffusion layer of the target cipher has a kind of simple algebraic property, such as a cyclicity. We insert two permutating processes reflecting property of the linear diffusion layer before and after the masked S-boxes so that the mask values after the linear operation are fixed. By imposing these limitations on the mask values and the permutating processes, we can omit the remasking operation for the intermediate rounds, which is necessary for RSM. This eliminates the memory for remasking and adder circuit, thereby reducing the circuit size. In addition, multiple masked S-boxes can be used in parallel, thereby decreasing the vulnerability to second-order zero-offset DPA as RSM. Though the restriction on the underlying block cipher is slightly stronger than RSM, our method is still applicable to AES. The effectiveness and security of the proposed countermeasure has been verified by implementation to FPGA. According to our experiments we think our proposal keeps the same security level as RSM. Moreover, the overhead of the proposed countermeasure is decreased by 10% compared with RSM.

II. Rotating S-boxes Masking

RSM was proposed as a DPA-countermeasure dedicated to AES. We extend the definition of RSM so that it is applicable to a block cipher of SPN structure with S-boxes and a linear diffusion layer.

We assume that a plaintext and a ciphertext are elements of $\mathrm{GF}(2^a)^t$ for some positive integers, a and t. The S-box is a

978-3-9815370-2-4/DATE14/©2014 EDAA

non-linear function over GF(2^a) and t S-boxes are aligned in one round. The diffusion layer is a linear function defined on t-dimensional vector space over GF(2^a). More formally, we use the following notations;

$X_r = (x_{r,0}, \cdots, x_{r,t-1}), x_{r,i} \in$ GF(2^a): r-th round input data
s : GF(2^a) $\rightarrow$ GF(2^a): S-box, a non-linear permutation used in the encryption algorithm
For $X = (x_0, \cdots, x_{t-1}) \in$ GF$(2^a)^t$, $S(X) := (s(x_0), \cdots, s(x_{t-1}))$.
P : GF$(2^a)^t \rightarrow$ GF$(2^a)^t$: the linear operation of the encryption algorithm
$K_r = (k_{r,0}, \cdots, k_{r,t-1}), k_{r,i} \in$ GF(2^a): r-th round key

Basic Encryption Algorithm
Using the above notations, the encryption algorithm is described as follows. We assume that the encryption consists of R rounds, and the plaintext and ciphertext are written as X_0 and X_{R+1}.

1. $X_1 = X_0 \oplus K_0$.
2. For $r = 1$ to $R-1$,
   $X_{r+1} = P(S(X_r)) \oplus K_r$.
3. $X_{R+1} = S(X_R) \oplus K_R$

A. Algorithm of Rotating S-boxes Masking

RSM performs nonlinear transformation by using rotating S-boxes that combines masked S-boxes with two barrel shifters. These components are described as follows:

Masked S-box
For $m, n \in$ GF(2^a), a masked S-box $s_{(m,n)}$ is defined as $s_{(m,n)}(x) = s(x \oplus m) \oplus n$.
For $M = (m_0, \cdots, m_{t-1}), N = (n_0, \cdots, n_{t-1}) \in GF(2^a)^t$,

$$S_{(M,N)}(X) := (s_{(m_0,n_0)}(x_0), \cdots, s_{(m_{t-1},n_{t-1})}(x_{t-1})). \quad (1)$$

Barrel Shifter
For an integer c, let $\tau_c(X)$ be the barrel shifter of X by c:

$$\tau_c(X) = (x_{c \bmod t}, x_{(c+1) \bmod t}, \cdots, x_{(c+t-1) \bmod t}),$$

Note that $S(\tau_c(X)) = \tau_c(S(X))$ holds true for any X and c. The algorithm of RSM is described as follows.

RSM Algorithm

1. $\tau(M)$, $P(\tau(N))$ and $\tau(N)$ are arranged.
2. For plaintext X_0, c_1 is chosen randomly from $\{0, 1, \cdots, t-1\}$.
3. Let X'_1 be the first round input data, where
$$X'_1 = X_0 \oplus K_0 \oplus \tau_{c_1}(M). \quad (2)$$
4. For $r = 1$ to $R-1$,
$$\begin{aligned} X''_{r+1} &= P(\tau_{c_r}(S_{(M,N)}(\tau^{-1}_{c_r}(X'_r)))) \oplus K_r. \\ X'_{r+1} &= X''_{r+1} \oplus (\tau_{c_r+1}(M) \oplus P(\tau_{c_r}(N))). \quad (3) \\ c_{r+1} &= (c_r + 1) \bmod t. \end{aligned}$$
5. Let X_{R+1} be the ciphertext, where
$$\begin{aligned} X''_{R+1} &= \tau_{c_R}(S_{(M,N)}(\tau^{-1}_{c_R}(X'_R))) \oplus K_R. \\ X_{R+1} &= X''_{R+1} \oplus \tau_{c_R}(N). \quad (4) \end{aligned}$$

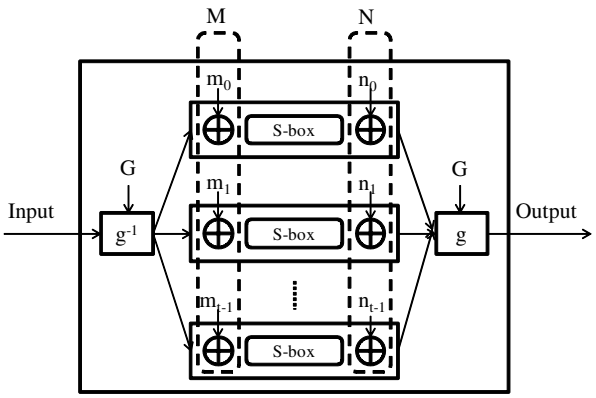

Fig. 1. Permutating S-boxes.

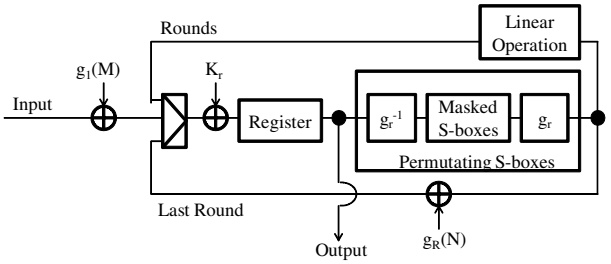

Fig. 2. Proposed Method.The function g_1 is chosen randomly from G.

In RSM the masked S-box is implemented as a table which is calculated in advance, using the mask constants and the original S-box. Hence the circuit size of masked S-boxes does not increase from the original. On the other hand, other masking methods need mask update S-boxes [15] or many masked S-boxes [10]. The additional parts of RSM are the barrel shifters, the mask addition processes and the random number generator. Therefore the increase in circuit size and the degradation in speed are relatively small compared with other masking methods. In addition, t masked S-boxes can be computed in parallel, which thus improves throughput per circuit size. The authors of [13] proposed, several sets of masks that resist first-order and second-order zero-offset CPA using a SAT-solver. The RSM algorithm needs the remasking process in Equation (3) for each round because the masks vary in the linear operation. Though these parts have relatively small sizes from masked S-box, RSM requires the processing time for remasking and the memory area used to store the masks that would be remasked. This observation implies that, though RSM is one of the simplest masking method to date, there may be a room for improvement.

III. Proposed Method

In this paper, we propose a variant of RSM that simplifies some of these masking processes. Our method requires that the linear diffusion layer of the target cipher has a kind of algebraic property. With this property we set the initial mask values and the permutation process before and after the masked S-boxes so that the mask values after the linear operation are fixed. By imposing these limitations on the mask values and the permutating process, we can omit the remasking operation for the intermediate rounds, which was necessary for RSM.

A. Proposed Algorithm

In our proposal, the barrel shifters of RSM are substituted with permutation units. In the following we describe the details of permutation units and the masked S-boxes. The combination of the permutation units and masked S-boxes is called permutating S-boxes, and is depicted by Figure 1.

Let G be a group of permutations of the elements of $X = (x_0, \cdots, x_{t-1})$ commutative to the linear operation P:

We require that for any $g \in G$, there exists $\tilde{g} \in G$ such that

$$P(g(X)) = \tilde{g}(P(X)). \tag{5}$$

We also require that G is not a singleton consisting of the identity permutation. In addition, it is necessary to make a structure of G simple as cyclic to reduce the complexity.

We determine M so that $M = P(N)$. Note that $S(g(X)) = g(S(X))$ holds for any permutation g over X. Using the same notations as RSM, our algorithm is defined as follows, and is depicted by Figure 2.

Proposed Algorithm

1. $g(M)$ and $g(N)$ are arranged.
2. For plaintext X_0, g_1 is chosen randomly from G.
3. Let X'_1 be the first round input data, where

$$X'_1 = X_0 \oplus K_0 \oplus g_1(M). \tag{6}$$

4. For $r = 1$ to $R - 1$, let

$$X'_{r+1} = P(g_r(S_{(M,N)}(g_r^{-1}(X'_r)))) \oplus K_r. \tag{7}$$
$$g_{r+1} = \tilde{g}_r. \tag{8}$$

5. Let X_{R+1} be the ciphertext computed as

$$X_{R+1} = g_R(S_{(M,N)}(g_R^{-1}(X'_R))) \oplus K_R \oplus g_R(N). \tag{9}$$

If $X'_r = X_r \oplus g_r(M)$, from Equations (7) and (8) we observe that

$$\begin{aligned} X'_{r+1} &= P(g_r(S_{(M,N)}(g_r^{-1}(X'_r)))) \oplus K_r \\ &= P(g_r(S(g_r^{-1}(X_r \oplus g_r(M)) \oplus M) \oplus N)) \oplus K_r \\ &= P(S(X_r)) \oplus P(g_r(N)) \oplus K_r \\ &= X_{r+1} \oplus \tilde{g}_r(P(N)) \\ &= X_{r+1} \oplus g_{r+1}(M). \end{aligned} \tag{10}$$

Equations (6), (9) and (10) show that the proposed algorithm generates the same output as the original encryption algorithm.

For example, as we show in the next section, AES has a linear diffusion layer which satisfies Equation (5).

B. Application to AES

We show that the proposed method can be applied to AES. AES uses an S-box working on GF(2^a) with $a = 8$, and all $t = 16$ S-boxes in a round are identical, which satisfies a necessary condition to apply the proposed method. The linear diffusion function of AES is defined as

$$P = MC \circ SR : \mathrm{GF}(2^8)^{16} \to \mathrm{GF}(2^8)^{16}, \text{where} \tag{11}$$

$$SR : X = (x_0, \cdots, x_{15}) \to SR(X) = (SR(x_0), \cdots, SR(x_{15}))$$

denotes the permutation called *ShiftRows* defined as

$$SR(x_i) = \begin{cases} x_i, & (i \bmod 4 = 0), \\ x_{(i+4) \bmod 16}, & (i \bmod 4 = 1), \\ x_{(i+8) \bmod 16}, & (i \bmod 4 = 2), \\ x_{(i+12) \bmod 16}, & (i \bmod 4 = 3). \end{cases} \tag{12}$$

MixColumns, denoted by *MC*, performs a multiplication of a constant matrix over GF(2^8). For $X = (x_0, \cdots, x_{15})$, MC applies the following matrix multiplications to $(x_{4j}, x_{4j+1}, x_{4j+2}, x_{4j+3})$ for $j = 0, 1, 2, 3$:

$$\begin{pmatrix} x_{4j} \\ x_{4j+1} \\ x_{4j+2} \\ x_{4j+3} \end{pmatrix} \to \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} x_{4j} \\ x_{4j+1} \\ x_{4j+2} \\ x_{4j+3} \end{pmatrix}, \tag{13}$$

where the elements of the matrix are defined in [16]. We denote the 4-byte left barrel shift of $X = (x_0, \cdots, x_{15})$ by τ:

$$\tau : (x_0, x_1, \cdots, x_{15}) \to (x_4, \cdots, x_{15}, x_0, x_1, x_2, x_3), \tag{14}$$

and define ϕ as a circulant permutation in a 4-byte block, i.e.,

$$\phi : (x_{4j}, x_{4j+1}, x_{4j+2}, x_{4j+3}) \to (x_{4j+1}, x_{4j+2}, x_{4j+3}, x_{4j}), \tag{15}$$

for $j = 0, 1, 2, 3$. These permutations are commutative and form a group

$$G = \{\tau^u \phi^v : u, v \in \{0, 1, 2, 3\}\}, \tag{16}$$

where $f^i(x)$ for function f denotes the i applications of f to x. G is a product of two cyclic permutations of order 4 and can be implemented with the same complexity as that of a cyclic permutation of order 16. From the definitions of SB, MC and τ, we have

$$SR(\tau(X)) = \tau(SR(X)), \text{and } MC(\tau(X)) = \tau(MC(X)). \tag{17}$$

For ϕ, from the cyclic property of the matrix shown in Equation (13), we have

$$MC(\phi(X)) = \phi(MC(X)). \tag{18}$$

For SR, it is easy to derive the following equation;

$$SR(\phi(X)) = (\tau^{-1}\phi)(SR(X)). \tag{19}$$

In general, for $g = \tau^u \phi^v$ we have

$$MC(SR(g(X))) = \tilde{g}(MC(SR(X))), \text{where } \tilde{g} = g\tau^{-v} \in G. \tag{20}$$

From Equation (20), we can apply our proposed method to AES with G. Because G is a direct product of two barrel shifts of 4 elements, the implementation of G is almost the same as circular permutation of 16 elements in RSM. Hence, the proposed method can omit the remasking process in RSM while keeping almost the same implementation cost of permutations before and after the substitution process.

Process flow of AES with the proposed method is shown in the following. We remark that only SR is applied in the

diffusion process of the last round of AES.

AES with Proposed Algorithm
Input is plaintext X_0, Output is ciphertext X_{11}.

1. $g(M)$ and $SR(g(N))$ are arranged.
2. Let $u, v \in \{0, 1, 2, 3\}$ be randomly chosen and let g_1 be $\tau^u \phi^v$.
3. Let X'_1 be the first round input data, where
$$X'_1 = X_0 \oplus K_0 \oplus g_1(M). \quad (21)$$
4. For $r = 1$ to 9,
$$X'_{r+1} = MC(SR(g_r(S_{(M,N)}(g_r^{-1}(X'_r))))) \oplus K_r. \quad (22)$$
$$g_{r+1} = g_r \tau^{-v}. \quad (23)$$
5. Let X_{11} be the ciphertext defined as
$$X_{11} = SR(g_{10}(S_{(M,N)}(g_{10}^{-1}(X'_{10})))) \oplus K_{10} \oplus SR(g_{10}(N)). \quad (24)$$

C. Security

First, we observe that the initial g_1 cannot be guessed by SPA, since for any $g_1 \in G$, all the masks are accessed in parallel. Thus, we here only discuss the security against DPA. In addition, we focus on the case when the last round is attacked, which is considered to be most practical.

In the proposed method, an intermediate value $x_{r,i} \oplus m_{r,i}$ is stored at the register for $g_r(M) = (m_{r,0}, \cdots, m_{r,t-1})$. Note that the last round output $x_{R+1,i}$ is not masked, and the contents of the register are changed from $x_{R,i} \oplus m_{R,i}$ to $x_{R+1,i}$ in the last round. Hence, assuming that $x_{R,i}$ and $x_{R+1,i}$ are independently random, the security of the proposed method against DPA focusing on the last round is evaluated with random variables $\{m_{R,0}, m_{R,1}, \cdots, m_{R,t-1}\}$. Note that these variables are derived by g_1, which is chosen randomly.

Let $p_{r,i}(m_j)$ be the probability that $m_{r,i}$ becomes m_j. Note that $p_{r,i}$ is uniform over M in RSM. On the application to AES of the proposed method, the order of G is $t = 16$ in (16), and there exists unique $g \in G$ such that g maps m_i to m_j for any i, j. Note that the map from g_1 to g_R is one-to-one from (23). Hence, if $g_1 \in G$ is chosen randomly, then $p_{R,i}(m_j)$ is $1/t = 1/16$ for any i, j, and the proposed method makes $p_{R,i}$ uniform. It means that both the proposed method and RSM will achieve the same security level for the attack at the last round if the same M is applied.

Next, we show how to optimize $\{m_0, \cdots, m_{t-1}\}$ for uniform $p_{r,i}$. Regarding $x_{R,i}$ and $x_{R+1,i}$ as independent uniform random variables over $GF(2^a)$, the following condition is required for M to thwart DPA;

Condition A: The Hamming weight of the sum of each bit of m_j is $t/2$, specifically, m_j satisfies
$$\bigoplus_{j=0}^{t-1} m_j = 0. \quad (25)$$

Let $\tilde{m}$ be a random variable over $\{m_0, \cdots, m_{t-1}\}$ with uniform distribution, $\tilde{x}$ be a random element of $GF(2^a)$ with uniform distribution, $E[x]$ be an expectation and $E[x|y]$ be a conditional expectation, for random variables, x and y. To ensure the securities against single-order and second-order zero-offset DPAs, M should be chosen according to the next condition B:

Condition B [13]: The following values, called optimal correlation coefficients [17], should be 0, where
$$\rho_{opt}^{(1)} = \frac{\sigma(E[HW(\tilde{x} \oplus \tilde{m})|HW(\tilde{x})])}{\sigma(HW(\tilde{x} \oplus \tilde{m}))}, \quad (26)$$
$$\rho_{opt}^{(2)} = \frac{\sigma(E[(HW(\tilde{x} \oplus \tilde{m}))^2|HW(\tilde{x})]))}{\sigma((HW(\tilde{x} \oplus \tilde{m}))^2)}. \quad (27)$$

Here, $\sigma(x)$ denotes the variance of x. Let $p(x, y)$ be simultaneous probability function, and $p(x)$ be probability function. Against MIA, M should be selected according to the next condition C:

Condition C [13]: The mutual information $I(HW(\tilde{x} \oplus \tilde{m}); \tilde{x})$ is the minimum, where
$$I(HW(\tilde{x} \oplus \tilde{m}); \tilde{x}) = \sum_{\tilde{x}} \sum_{\tilde{m}} p(\tilde{x} \oplus \tilde{m}, \tilde{x}) \log \frac{p(\tilde{x} \oplus \tilde{m}, \tilde{x})}{p(\tilde{x} \oplus \tilde{m}) p(\tilde{x})}. \quad (28)$$

Note that $I(HW(\tilde{x} \oplus \tilde{m}); \tilde{x})$ becomes 0 only when the mask takes all values of $GF(2^a)$.

While we have considered the attacks focusing on the last round, one may wonder the possibility of other attacks. Typically, an attack based on the Hamming weight of S-box would be of concern. In this case, the output mask, N, of the S-boxes may determine the security level. Here, it is preferable that N satisfies Conditions A, B, and C as well as the input mask M. RSM allows a such choice, as M and N can be independently chosen. Unfortunately, in our proposal N is determined by M hence we are not sure if M and N can satisfy all the conditions. So far we found an instance of (M, N) where M satisfies all conditions and N satisfies Condition A. On hardware implementation, however, such attack is generally much less effective than the ones at the last round, and the effect of N may be much less critical than that of M. The impact of N on security, and how we should choose (M, N) (for M satisfying all conditions), still need to be studied further.

IV. Experiments

We examine the effectiveness of our method proposed in Section III. More specifically we apply our method for AES-128, as shown in Section III-B, and evaluate the complexity and security.

A. Environment

We implemented three methods: unprotected AES, AES with RSM [13], and AES with the proposed method. Both RSM and the proposed method used the same mask M, which follows Conditions A, B, and C of Section III-C. The following analysis methods were used:

- Differential Power Analysis (DPA) [2]
- Correlation Power Analysis (CPA) [7]

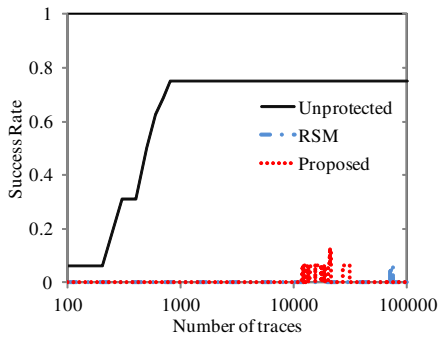

Fig. 3. Success Rate for DPA on FPGA.

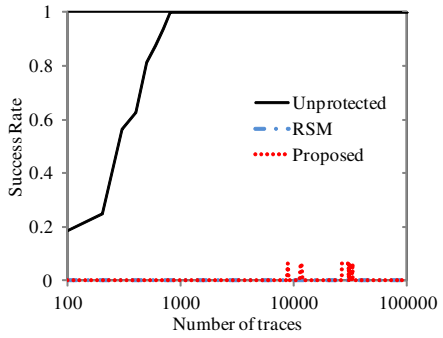

Fig. 4. Success Rate for CPA on FPGA.

- Variance-based Power Analysis (VPA) [18]
- Mutual Information Analysis (MIA) [19]

We employed Hamming distance model [7] for the power model used for these attacks. In this experiment, Hamming distance between an intermediate value of the last round and the corresponding ciphertext was used for analysis. We denote the the Hamming distance by *HD*. We used a Xilinx FPGA (VirtexII Pro XC2VP7) on SASEBO 1st board [20]. Our implementation of AES encryption was based on the reference code of SASEBO board. The power consumption traces were measured by using a Textronix oscilloscope (DPA7104).

B. Results

Figure 3 shows the success rate of DPA. The success rate indicates the rate of successfully recovered sub-keys from 16 sub-keys. It presents that, in RSM and proposed method, the attack failed even at 100,000 traces. On the other hand, in the unprotected circuits, analysis succeeded for 12 sub-keys at 3,000 traces. Since intermediate values are masked by M selected on Condition A, the correlation between *HD* and the power consumption is decreased. Therefore, both RSM and the proposed method have a sufficiently strong resistance to DPA.

Figure 4 shows the success rate of CPA. It presents that, in RSM and proposed method, analysis failed even at 100,000 traces. On the other hand, in the unprotected circuit, analysis succeeded for all sub-keys at 1,000 traces. Since intermediate values are masked by M with $\rho^{(1)}_{opt} = 0$, the correlation between *HD* and power consumption is decreased. Therefore, both RSM and the proposed method have a sufficiently strong resistance to CPA.

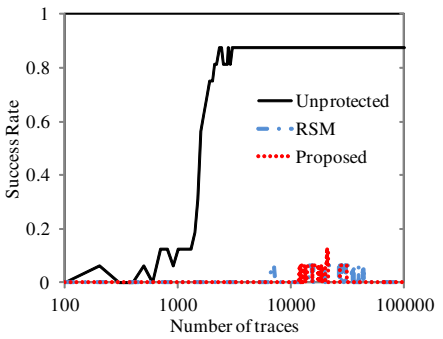

Fig. 5. Success Rate for VPA on FPGA.

TABLE I. MUTUAL INFORMATION BETWEEN POWER CONSUMPTION TRACES AND HD

Sub-Key	0	1	2	3
Unprotected	0.025082	0.027385	0.024325	0.033705
RSM	0.003122	0.003078	0.002977	0.003268
arrange	0.002514	0.002503	0.002515	0.002491

Sub-Key	4	5	6	7
Unprotected	0.025239	0.024781	0.037738	0.029845
RSM	0.002936	0.003064	0.003132	0.003125
Proposed	0.002503	0.002618	0.002475	0.00251

Sub-Key	8	9	10	11
Unprotected	0.020983	0.025855	0.022266	0.03266
RSM	0.003061	0.003214	0.003062	0.003101
Proposed	0.002548	0.002351	0.002469	0.00249

Sub-Key	12	13	14	15
Unprotected	0.030212	0.027025	0.029678	0.026494
RSM	0.003247	0.003037	0.003336	0.003218
Proposed	0.002348	0.002415	0.002447	0.002407

Figure 5 shows the success rate of VPA. It presents that, in RSM and proposed method, the result of VPA was almost same as that of DPA and CPA. Since intermediate values are masked by M with $\rho^{(2)}_{opt} = 0$, the correlation between *HD* and square of power consumption is decreased. Therefore, both RSM and the proposed method have a sufficiently strong resistance to VPA.

Table I shows the result of MIA at 100,000 traces. In Table I, the mutual information of the power consumption traces and *HD* obtained by the attack are presented for each sub-key. Let W be the power consumption trace ($W \in \mathbf{R}^d$, d is trace length), and H be a random variable denoting Humming distance ($H \in \{0, 1, \cdots, 8\}$), $p(x, y)$ be a simultaneous probability function for x and y, and $p(x)$ be a probability function. In this experiment, $p(h, w), p(h)$, and $p(w)$ are estimated using histograms. The mutual information is as follows:

$$I(H;W) = \sum_{h \in H} \sum_{w \in W} p(h,w) \log \frac{p(h,w)}{p(h)p(w)}. \qquad (29)$$

When $I(H;W)$ is large, the correlation between the power consumption traces and *HD* is high, which makes analysis successful. Table I also shows that the mutual information in the RSM or proposed method is one-tenth of that in the unprotected circuits. Since intermediate value are masked by M following Condition C, $I(H;W)$ is decreased in RSM and the proposed method. As a result MIA failed at RSM and the proposed method, while succeeded at the unprotected

TABLE II. COMPARISONS OF AES IMPLEMENTATIONS

Method	Slices	LUTs	FFs	Operating frequency[MHz]
Unprotected	1945	3605	395	123.2
Single	2034 (105%)	3839 (106%)	395 (100%)	119.8 (88%)
Fixed	19014 (978%)	37399 (1037%)	432 (109%)	71.4 (58%)
RSM	2882 (148%)	4892 (136%)	432 (109%)	51.8 (42%)
Proposed	2708 (139%)	4537 (126%)	432 (109%)	56.6 (46%)

circuits. This means that RSM and the proposed method have the sufficiently-high security against MIA.

Table II summarizes the size and speed of AES implementations. It shows the number of slices, lookup tables (LUTs), flip-flops (FFs), and the maximum operating frequency. The percentages below the figures indicate the ratio from the unprotected AES. We also consider AES with single masking and fixed masking [10] for comparison. A structure of single masking is that of RSM without rotating units. Therefore, all intermediate value stored the register are masked by M.

Although the increase in size of single masking from the unprotected one is 5%, the key is guessed by the brute force attack to mask M and its security level is low. Fixed masking required ten times larger size of the unprotected one, thus we cannot implement the circuit on the FPGA.

The increase in size of the proposed method from the unprotected one is 30%. The proposed method achieves smaller size and faster speed than RSM. In circuit size, the overhead of the proposed method is decreased by 10% compared with that of RSM. The improvement in size is rather marginal. This comes from the fact that the random number generator and the permutation unit are rather large and thus reduce the gain obtained by removing the remasking process of RSM. When the block size of the target cipher is large, the size of mask is large, too. Therefore, a great decrease in the circuits size can be expected by applying the proposed method. Still, considering that RSM is one of the most efficient yet secure masking method to date, an indication of further improvement would be useful for both in theory and practice.

V. CONCLUSION

This paper proposed a masking countermeasure that simplifies RSM using an algebraic property of the linear diffusion layer of a block cipher. The proposed method was applied to AES to demonstrate the effectiveness in terms of security and the reduction in circuit size using simulation and implementation in FPGA. We will study the security of the proposed method to other attacks and if the proposed method can be applied to other ciphers.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their helpful comments that improve the quality of this paper, and also thank Kouichi Nakamura, Tomonori Iida and Takayuki Kimura for assistance in experiments.

REFERENCES

[1] P. Kocher "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Crypto'96, pp.101-113 , 1996.

[2] P. Kocher, J. Jaffe and B. Jun, "Introduction to Differential Power Analysis and Related Attacks," , 1998.

[3] J-J. Quisquater, D. Samyde, "A new tool for non-intrusive analysis of smart cards based on electro-magnetic emissions. The SEMA and DEMA methods," Eurocrypt2000 rump session, May , 2000.

[4] L. Goubin, J. Patarin, "DES and Differential Power Analysis: The Duplication Method," CHES 1999, LNCS, vol. 1717, Springer, pp.158-172 , 1999.

[5] M. L. Akkar, R. Bevan, P. Dischamp and D. Moyart, "Power Analysis, What Is Now Possible...," ASIACRYPT 2000, LNCS, vol. 1976, Springer, pp.489-502 , 2000.

[6] R. Bevan, E.Knudsen, "Ways to Enhance DPA," ICISC 2002, LNCS, vol. 2587, Springer, pp. 327-342 , 2003.

[7] E. Brier, C. Clavier and F. Olivier, "Correlation Power Analysis with a Leakage Model," CHES 2004, LNCS, vol. 3156, Springer, pp.16-29 , 2004.

[8] C. Clavier, J. Coron, and N. Dabbous, "Differential Power Analysis in the Presence of Hardware Countermeasures," CHES2000, LNCS, vol. 1965, Springer, pp.252-263 , 2000.

[9] T. Messerges, "Securing the AES Finalists Against Power Analysis Attasks," FSE2000, LNCS, vol. 1978, Springer, pp.150-164 , 2001.

[10] K. Itoh, M. Takenaka, and N. Torii, "DPA Countermeasure Based on the Masking Method," ICICS2001, LNCS, vol. 2288, Springer, pp. 440-456 , 2002.

[11] E. Trichina, "Combinational Logic Design for AES SubByte Transformation On masked Data," Cryptology ePrint Archive, 2003/236 , 2003.

[12] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches," ICICS 2006, LNCS, vol. 4307, Springer, pp. 529-545 , 2006.

[13] M. Nassar, Y. Souissi, S. Guilley and J.-L.Danger, "RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs, " DATE 2012 , pp.1173-1178 , 2012.

[14] J. Waddle and D. Wagner, "Towards Efficient Second-Order Power Analysis," CHES 2004, LNCS, vol.3156, Springer, pp. 1-15, 2004.

[15] F. -X Standaert, E. Peeters, J.-J. Quisquater, "On the Masking Countermeasure and Higher-Order Power Analysis Attacks," ITCC 2005, vol. 1, pp.562-567.

[16] NIST FIPS-197, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[17] E. Prouff, M. Rivain, and R. Bevan, "Statistical Analysis of Second Order Differential Power Analysis," IEEE Trans. Computers, vol. 58, no. 6, pp. 799-811, 2009.

[18] S. Mangard, N. Pramstaller, and E. Oswald, "Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices," ICISC, LNCS, vol. 5461, Springer, pp. 253-267 ,2008.

[19] N. Veyrat-Charvillon and F.-X. Standaert, "Mutual Information Analysis: How, When and Why?," CHES 2009, LNCS, vol. 5747, Springer, pp. 429-443 ,2009.

[20] National Institute of Advanced Industrial Science and Technology, Research Center for Information Security, "Side-channel Attack Standard Evaluation Board Specification. Ver. 1.0," Sep. ,2007.