Achilles' Heel: the Unbalanced Mask Sets May Destroy a Masking Countermeasure*

Jingdian Ming[1,2], Wei Cheng[1], Huizhong Li[1,2], Guang Yang[1,2], Yongbin Zhou[1,2], and Qian Zhang[1,2]

[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
Email: {mingjingdian, zhouyongbin}@iie.ac.cn

Abstract. Low Entropy Masking Scheme (LEMS) has attracted wide attention for its low-cost feature of small fixed mask sets in Side-Channel-Analysis (SCA). To achieve the expected side channel security, it is necessary to find a balanced mask set to reduce the correlations between key dependent variables and their corresponding leakages. However, the security proof of LEMS, based on an inadequate assumption, might lead to consequent mask sets proposed without balance property, which could cause vulnerable LEMS implementations. This paper focusing on correcting and improving this scheme, first gives the formal definitions of univariate balance property on mask sets and extends it to multivariate settings. From these definitions, we propose three fundamental properties to analyze the balance of mask sets in Rotating Sbox Masking (RSM), the most popular LEMS implementations. To demonstrate the definitions and properties, three state-of-the-art RSM mask sets were selected as research objects. The corresponding attacks when any properties violated distinctly indicate the necessity of evaluating the balance property of the mask set in advance (during the design phase). However, it is found impossible to get a mask set for the RSM with all three properties satisfied, which means the vulnerabilities of RSM scheme in its unbalanced mask set are unavoidable. Thus, this promising masking scheme may be broken for its unqualified mask set.

Keywords: Side Channel Analysis, Masking Countermeasures, DPA Contest, RSM Scheme, Unbalanced Mask Set

1 Introduction

Side-Channel-Analysis have been proved to be a serious threat to practical security of hardware implementations. Since the pioneering work was proposed by Kocher et al. [14] in 1996, SCA has received much more attention due to its strong analytical power. Subsequently, many attacks have been emerged in succession, such as differential power analysis (DPA) [15], correlation power analysis (CPA) [16] and mutual information analysis (MIA) [20]. In SCA, an adversary can

* This report was submitted to USENIX 2018, unfortunately it was not accepted.

exploit the physical leakages underlying device during the algorithm execution (e.g., the execution time [11], the power consumption [4] or the electromagnetic radiations [19]) to recover some secret information. The recent Meltdown and Spectre Attacks are typical examples of utilizing SCA to steal secret data. As a consequence, countermeasures against SCA must be developed and applied to protect our secret information.

The countermeasures against SCA can be divided into five different categories according to their level of integration: chip level, system level, algorithm level, gate level and transistor level, respectively [18]. Among them, the algorithm-level countermeasures, such as shuffling and masking, have the unique advantage of device independence. Namely, there is no need to design different protection schemes for each type of devices (ASICs, DPSs, CPUs, GPUs and FPGAs). Specifically, masking [28–31] is the most investigated and popular technique to improve the security of cryptographic implementations against SCA. The masking countermeasure, by adding random values before and after sensitive operations, prevents the dependency between sensitive intermediate values and side channel leakages. While attacking masked implementations, the adversary needs to combine different leakages to compensate the effect induced by masks, which reduces the SNR (signal to noise ratio) of leakages and increases the difficulty of attacks. Despite all the above advantages, the main drawback of masking schemes is the significant overhead of their implementations for masking, demasking and generating massive random numbers, especially toward block ciphers such as AES. In practice, the resources (e.g. the code size, the execution time and the RAM needed) are limited by the application, so it is necessary to reduce the cost to make the masking scheme more practical.

To lower the cost of masked cryptographic implementations, Low Entropy Masking Scheme (LEMS) emerged with a fixed mask set, which is a strict subset of a full mask set composed of all possible values. Therefore, the design of mask sets has become a crucial issue in LEMS. Then, the RSM scheme [21] was proposed as a typical and complete LMES with low cost but high performance (RSM aims at keeping performances and complexity close to an unprotected AES design while being as robust against first-order SCA attacks as the state-of-the-art masking in hardware [21]). Its lightweight implementation is achieved by precisely designing the mask sets and embedding them into each round function structures. And RSM was adopted by DPA Contest v4 (both v4.1 and v4.2) as the official countermeasure against SCA and is extensively studied. Originally, the RSM scheme was designed to protect sensitive intermediates with a low entropy mask set against 1O-, 2O- and 3O- zero-offset correlation power attacks [24]. It was proven [22] that such security can be achieved with only 16 mask values. However, we find that the premise of a qualified mask set in [22] is dubious, which results in most mask sets based on it would not achieve expected security level. In researching from the perspective of inherent property of the mask sets, there are two unsolved questions: how to evaluate the balance of a mask set to filter out unqualified mask sets, and how to find a qualified mask set to make RSM scheme reach a high security level. This paper concentrates

on developing methods of evaluating whether a mask set is balanced, and how to find a qualified one. Unexpectedly, it proved that there are no qualified mask sets, and it means that the RSM scheme cannot meet desired security with any potential mask sets. Therefore, it is necessary to change the framework of RSM scheme to fix these unavoidable vulnerabilities.

This paper's contributions mainly lie in the following aspects. To analyze the balance of mask set thoroughly, it first give the formal definitions of balance property on mask sets. On the basis of these definitions, we propose three fundamental properties (necessary but not sufficient) to analyse the balance of mask sets in RSM scheme. Then we demonstrate their validity by attacking three state-of-the-art RSM-masked implementations (proposed by DPA Contest v4.2 [23], by Moradi *et al.* [17] and by Veshchikov *et al.* [5]) which cannot meet the three properties simultaneously. The attack results show that all three state-of-the-art RSM-masked implementations are insecure because of the lack of these three properties. Finally, we prove that there is no qualified mask set for RSM existing, which means it is impossible to make the RSM scheme achieve expected security level only by selecting a proper mask set. Thus the framework of RSM scheme must be modified to achieve a higher security level.

The rest of the paper is organized as follows. Section 2 reviews the details of LEMS and RSM scheme, then Section 3 gives the definitions of balance properties of mask sets and proposes three fundamental properties to analyze the validity of mask sets. Afterwards, in Section 4 it shows the attacks and experimental results toward RSM-masked implementations for each corresponding property unsatisfied to demonstrate their reasonability and validity. Section 5 proves that it is impossible to find a proper mask set for RSM. Finally, Section 6 contains observations and conclusions.

2 Low Entropy Masking Schemes

With the rapid development of communication, the device involving cryptographic applications is becoming increasingly lightweight, so it is necessary to make the implementations of cryptographic algorithms smaller and faster. In order to address the high overhead caused by masking scheme for lightweight implementations, low entropy masking scheme (LEMS) was proposed as an alternative to provide first-order side channel security for cryptographic implementations. In LEMS, all masks are chosen from a fixed mask set, that is a strict subset of full entropy masking set (e.g. for a n-bit full entropy mask set has 2^n elements from $\mathbb{F}_2^n$). Thus the idea behind LEMS is to reduce the cardinality of mask set while keeping immunity to certain order of side channel attacks, which is a practical tradeoff between performance and security. As a result, the selection of the Candidate mask set is the key for designing an optimal LEMS.

On one hand, the cardinality of mask set directly determines the number of a mask values which could be used to protect sensitive variables in cryptographic implementations. In this regard, Nassar *et al.* [2] formally analyzed the feasibility of using decreased mask set to protect implementations and showed that it's

possible to restrain both CPA and 2O-CPA with only 12 mask values (cardinality equals 12). Bhasin *et al.* [22] proved that a byte-oriented block cipher such as AES can be protected against 1O-, 2O- and 3O- zero-offset CPA with only 16 mask values (cardinality equals 16). Therefore, it's a reasonable choice to set the mask set's cardinality at 16.

On the other hand, the elements of mask set play key roles in securing practical implementations. By using entropy analysis, Nassar *et al.* [2] also intensively analyzed the dependency between the choices of mask values and corresponding entropies (conditional entropy and mutual information). They exhaustively searched for word size $n \leq 5$-bit and using SAT-solver for n up to 8-bit to obtain qualified mask sets. The following expression formally quantifies the amount of dependency between $\mathcal{L}$ and V [22], where $\mathcal{L}$ and V denote the side channel leakage and sensitive variable respectively.

$$\mathcal{L} \perp V \Longrightarrow \forall d \in \mathbb{N}, Var[\mathbb{E}[\mathcal{L}^d|V]] = 0 \tag{1}$$

$\mathbb{E}$ and Var denote the expectation and variance operator, and d denotes the d-th order moment. Since $\mathcal{L}$ connects with masked variables, let M denotes the mask set of LEMS. Then the quantified dependency can be re-written as

$$\begin{aligned} &\forall d \in \mathbb{N}, Var[\mathbb{E}[\mathcal{L}^d(V \odot M)|V]] = 0 \\ \Longrightarrow &\forall d \in \mathbb{N}, Var[\mathbb{E}[HW^d(V \oplus M)|V]] = 0 \end{aligned} \tag{2}$$

where symbol "$\odot$" denotes the masking operation. If it is boolean masking operation, "$\odot$" can be replaced by "$\oplus$". $\mathcal{L}(\cdot)$ is also used as a leakage model which typically replaced by Hamming Weight model (HW).

Although $Var[\mathbb{E}[\mathcal{L}^d(V \odot M)|V]] = 0$ is a necessary requirement to select a mask set, it's far from sufficient to select a qualified one for LEMS. Firstly, Hamming Weight model may be too simple to characterize the leakages of a real-world devices, especially when characterizing leakages of different sensitive bits with diverse leakages [6]. Thus $Var[Var[\mathcal{L}^d(V \odot M)|V]] = 0$ could be another necessary condition for mask set selection, which characterizing the variance of masked values. Secondly, Eq.2 is only used to characterize univariable dependencies between sensitive variables and corresponding leakages, but not for multivariable leakages. However, Eq.1 and Eq.2 could be intuitively seen as the balance properties of a mask set. Hereafter, this paper will recall the first complete LEMS scheme named RSM with some practical attacks to explain the necessary properties of the mask set to guarantee a certain security level.

2.1 Rotating Sbox Masking Scheme

On the basis of aforementioned requirements on mask sets, Rotating Sbox Masking scheme (RSM) [21] was proposed as a typical complete LEMS with $|M| = 16$ ($|M|$ means the number of elements in mask set M) and $Var[\mathbb{E}[HW^d(V \oplus M)|V]] = 0$ fully satisfied. RSM is very efficient (the performances in terms of speed and complexity is very near to unprotected implementation and far

better than usual masking structures) and also declared to be immune to 1O- and 2O- zero-offset CPA [21]. With its attractive properties in performance and resistance to SCA, RSM was adopted as a main countermeasure by DPA Contest v4 (both v4.1 and v4.2) to protect the public target AES implementation. In order to give a first sight of RSM-masked implementation, this study shows the first round of RSM-AES-256 which was used in DPA Contest v4.2 as Alg. 1, all rounds in AES share the same low entropy mask set.

Algorithm 1 The first round of RSM protected AES implementation in DPA Contest v4.2

Input: 16-bytes Plaintext Plain[16] and key[16],
Subkeys, 16-bytes first Roundkey[0][16] (only the first round considered)
16 mask values of 8-bit Mask[16],
/* Draw 16 4-bit values (uniformly random, unknown) offset[16] for the key blinding */
/* Draw of a shuffling function(uniformly random permutation), Shuffle0[16]: $[0, 15] \rightarrow [0, 15]$, bijective */
Output: the next round input $State[16]$

$State \leftarrow Plain[16]$
$Roundkey[0][0:15] \leftarrow key \oplus Mask[offset[0:15]]$
$State = State \oplus Roundkey[0][0:15]$
for $i \in Shuffle0([0, 15])$ **do**
    $X_i = MaskSbox_{offset[i]}(X_i)$
$State = SR(State)$ /* SR means ShiftRow */
$State = MC(State)$ /* MC means MixColumns */
for $i \in [0, 15]$ **do**
    $MaskCompensaton[i] = Mask_{offset[i]+1} \oplus SR(MC(Mask_{offset[i]+1}))$
$State = State \oplus MaskCompensaton[0:15]$
return $State$

Considering the different mask sets used in LEMS, the state-of-the-art mask sets are categorized into four classes: M_1 used in DPA Contest v4.1 which is a [8,4,4] linear code and its variant M_2 proposed by Moradi *et al.* [17], M_3 used in DPA Contest v4.2 which is not a linear code but more secure than M_1, and M_4 proposed by Veshchikov *et al.* [5] with sixteen variants. Note that mask set M_2 proposed by Moradi *et al.* is a variant of M_1 with changed order (we omit the mask set M_5 referenced in [7] since $|M_5| = 12$ with a lower security level when compared to mask set M_1, M_2 and M_3). These four mask sets are listed as follows.

$$
\begin{aligned}
M_1 = \{&0x00, 0x0f, 0x36, 0x39, 0x53, 0x5c, 0x65, 0x6a,\\
&0x95, 0x9a, 0xa3, 0xac, 0xc6, 0xc9, 0xf0, 0xff\}\\
M_2 = \{&0x00, 0x0f, 0x36, 0x39, 0x53, 0x95, 0x5c, 0xc9,\\
&0xff, 0xc6, 0xac, 0x9a, 0x6a, 0xa3, 0x65, 0xf0\}\\
M_3 = \{&0x03, 0x0c, 0x35, 0x3a, 0x50, 0x5f, 0x66, 0x69,\\
&0x96, 0x99, 0xa0, 0xaf, 0xc5, 0xca, 0xf3, 0xfc\}\\
M_4 = \{&0x13, 0x94, 0x25, 0xcb, 0x8e, 0x5f, 0xd9, 0x37,\\
&0x77, 0xc6, 0xa8, 0x38, 0x05, 0xea, 0x70, 0xe8\}
\end{aligned}
\tag{3}
$$

Note that only the first variant M_4 in [5] (there are totally 16 variant mask sets in [5]) is listed since all sixteen variants have similar properties.

From a perspective of evaluating the mask set M, it's obvious that all four mask sets satisfy the balance properties characterized by Eq.2. However, Moradi et al. [17] and Veshchikov et al. [5] both showed the practical attacks against RSM-masked AES-256 by exploiting vulnerabilities existing in mask set M_1. Other attacks [8, 25] also showed different ways to utilize these vulnerabilities to attack RSM-masked AES-256. For M_3, Liu et al. [3] presented bivariate first-order attacks by using flaws existing in M_3. Although there is no attack against RSM scheme with M_2 and M_4 for now, we will show in this paper that all four mask sets are vulnerable in practical attacks which exploiting flaws in mask sets. In summary, these attacks evidently show that Eq.2 is far from sufficient to quantify the balance property of a mask set. To intensively quantify the balance property of a mask set, next section first defines the balance properties of a mask set and then evaluates the balance of all four mask sets.

3 Balance of Mask Sets in LEMS

The balance of mask sets in LEMS has an appreciable effect on the security of masking implementations. To make it clear that which security level the LEMS can reach without estimating by attacks or experiments, we propose four definitions of univariable balance and multivariable balance for low entropy mask sets.

3.1 Definitions for Balanced Mask Set

The study grades the balance of mask sets from univariable balance up to multivariable balance. Univariable balance of mask sets is most important for LEMS to maintain its security level. In theory, the LEMS cannot protect against first-order attacks or high order zero-offset attacks if the mask set is not univariable balanced.

Definition 1 (Univariable Balance of n-bit Mask Sets). *Let M be a n-bit mask set, $v \in \left\{0, 1, \cdots, 2^{n-1}\right\}$ and v is uniformly distributed, the univariable*

balance of mask set M can be defined by:

$$ublc_n^d = Var[\mathbb{E}(\mathcal{L}^d(v, M)|v)] \tag{4}$$

if $ublc_n^d = 0$, then the n-bit mask set M is d-order univariable balanced.

where v denotes the n-bit intermediate value, d denotes the targeted order of resistance to zero-offset attacks, we can assume that $\mathcal{L}(v, M) = HW(v \odot M)$, the symbol "$\odot$" denotes the masking operation (like xor). Definition 1 means all $\mathbb{E}(\mathcal{L}^d(v, M)|v)$ are equal for a uniformly distributed variable v. Since the conditions when $n = 1$ and $n = 8$ are the commonest in cryptographic implementations, this paper defines univariable balance on single bit and single byte.

Definition 2 (Univariable Balance on Single Bit). *Let M be a single bit mask set, $v \in \{0, 1\}$ and v is uniformly distributed, if $ublc_1^d = Var[\mathbb{E}(\mathcal{L}^d(v, M)|v)] = 0$, then mask set M is d-th order univariable balanced on single bit.*

Definition 3 (Univariable Balance on Single Byte). *Let M be a 8-bit mask set, $v \in \{0, 1, \cdots, 255\}$ and v is uniformly distributed, if $ublc_8^d = Var[\mathbb{E}(\mathcal{L}^d(v, M)|v)] = 0$, then mask set M is d-th order univariable balanced on single byte.*

When the mask set is not univariable d-order balanced, we can inevitably launch a univariable d-order zero-offset attack on LEMS to get the secret key. However, the adversary can also combine different variables to launch multivariable attacks [17,25]. Therefore if LEMS implementations need to reach a higher security level, univariable balance for the mask set is a necessary condition but not sufficient [5], multivariable balance must also be considered.

Definition 4 (Multivariable Balance of n-bit Mask Sets). *Let M be a n-bit mask set, $v_1, v_2, \cdots, v_s, \in \left\{0, 1, \cdots, 2^{n-1}\right\}$ and $v_1, v_2 \cdots v_s$ is uniformly distributed, the univariable balance of mask set M can be defined by:*

$$mblc_n^{s,d} = Var[\mathbb{E}(\mathcal{L}^d(v_1, v_2, \cdots, v_s, M)|v_1 v_2 \cdots v_s)] \tag{5}$$

if $mblc_n^{s,d} = 0$ then the n-bit mask set M is d-order s-variable balanced.

We can assume that $\mathcal{L}(v_1, v_2, \cdots, v_s, M) = HW((v_1 \odot M) \circ (v_2 \odot M) \cdots \circ (v_s \odot M))$, where "$\circ$" denotes combined operation toward masked intermediate values, s denotes the number of intermediate variables the adversary can use to combine for attacks.

Remark 1. As above, define multivariable balance on single bit and single byte. For the sake of brevity, these definitions have been eliminated in this paper.

In theory, if the mask set is not s-variable balanced, the adversary can launch no more than s-varible SCA attacks to recover secret key (the d-th order and s-variable we mentioned in this paper are similar with [7]). However, the adversary may use less variables to attack (even only one variable) to successfully recover the key if the masking scheme is not carefully implemented [5, 25].

Table 1: the univariable balance of M_1, M_2, M_3, M_4 for LEMS with d=1

Mask Set	$ublc_1^1$	$ublc_8^1$	$mblc_1^{2,1}$ (0,1-th bits)
M_1	0	0	0
M_2	0	0	0
M_3	0	0	0
M_4	0	0	**0.0157**

Theoretically, we measure the balance of each mask sets on single bit, single byte and 2-variable bit (say for example, we combine 0-th and 1-th bit as a new 2-bit mask set), the results are shown as Table 1.

Clearly, These mask sets are always univariable balanced. But M_4, the latest mask set proposed by Veshchikov *et al.* [5], is not balanced when combined 0-th bit and 1-st bit of each mask value, which proves that the original method for mask sets selecting is problematic. In fact, combining other bits may also be unbalanced. For the sake of brevity, in this paper we just show one of them in table 1.

Applying these definitions to some specific implementations of LEMS such as RSM, the study proposes three fundamental properties to simultaneously evaluate the balance and entropy of RSM mask sets, which are both crucial to the security level of RSM implementation.

3.2 Three Fundamental Properties for RSM Mask Set

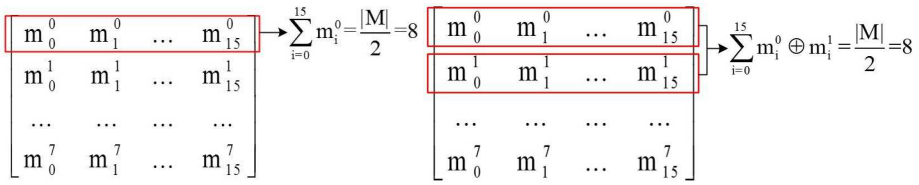
All definitions of balanced mask set in LEMS could be extended on algorithms for which the datapath is segmented in words of n-bits. For instance, RSM scheme is a representative LEMS, hence $n = 8$. To measure the security level easily, the study refines the four definitions into three fundamental properties. These three properties are necessary but not sufficient for secure SCA implementation of RSM. Specially, RSM [21] requires a mask set M with 16 mask values. So let $|M| = 16$, m_i^b denotes the b-th bit of i-th mask value in M,

Property 1 (Univariable Balance on Single Bit) *For* $\forall b \in \{0, 1, \cdots, 7\}$, *$M$ should satisfy that*

$$\sum_{i=0}^{|M|-1} m_i^b = \frac{|M|}{2} \tag{6}$$

Property 2 (Multivariable Balance on Single Bit) *For* $\forall s \in \{2, 3\}$ *and* $\forall b_1, \cdots, b_s \in \{0, 1, \cdots, 7\}$, *$M$ should satisfy that*

$$\sum_{i=0}^{|M|-1} (m_i^{b_1} \oplus \cdots m_i^{b_s}) = \frac{|M|}{2} \tag{7}$$



(a) **Property 1** with $b = 0$ (b) **Property 2** with e=2, $b_1 = 0$ and $b_2 = 1$

Fig. 1: Graphical illustration for **Property 1** and **Property 2**

if $s \in \{1, 7\}$ then **Property2** has no different with **Property 1**, and $s \in \{2, 3\}$ is the same with $s \in \{5, 6\}$. At last, $s = 4$ is a special case, any low entropy mask set can't meet the Eq.8 when s equals to 4, so we have no need to consider this condition.

Property 3 (Bivariable Balance on Adjacent Bytes) *Let M' be an adjacent byte combined mask set $M' = \{m_i \oplus m_{(i+1) mod\, 16} | i \in \{0, 1, \cdots, 15\}\}$, then M' should meet $|M'| = 16$, **Property 1** and **Property 2**.*

In fact, to make the mask set in RSM reach bivariable balance on single byte, it is necessary to combine all possible two mask values m_i and m_j. However, assume $j = (i + 1) mod\, 16$ is adequate in this paper to show the attacks and demonstrate the proof of unavoidable vulnerabilities of RSM. In order to make these properties easier to be understood, transform mask set M into a 8×16 binary matrix, where each column in the matrix represents a binary vector translated by its corresponding mask value, **Property 1** and **Property 2** can be expressed as Fig. 1 when $b = 0$ and $b_1 = 0$, $b_2 = 1$, respectively.

Based on **Property 3**, the number of elements in new set M' need not to be lower than mask set M, so we define entropy loss by:

$$\begin{aligned} entropy\ loss &= H(M) - H(M') \\ &= -\sum_{i=0}^{|M|-1} P(m_i) log P(m_i) + \sum_{i=0}^{|M'|-1} P(m'_i) log P(m'_i) \end{aligned} \quad (8)$$

To maintain the security while combine 2 adjacent masks, it is necessary to reduce entropy loss to zero, so that the combining intermediate values are under the same degree of protection. The entropy loss and balance of M_1, M_2, M_3, M_4 for RSM are showed in Table 2.

As showed in Table 2, M_1, M_3 and M_4 are bivariable unbalanced. M_2 is balanced on adjacent bytes but the entropy loss is 1 bit, which make the masked intermediate values not random enough and still correlated to their corresponding leakages. In conclusion, all of these mask sets don't meet **Property 3**.

Intuitively, if these three properties are not met, the mask set will be inevitably unbalanced and the RSM scheme is insecure accordingly. The next section shows attacks based on those unbalanced properties. There might be more

Table 2: the entropy loss and bivariable balance of M_1, M_2, M_3, M_4 for RSM with d=1 and s=2

Mask Set	$mblc_8^{2,1}$ (Adjacent Bytes)	$\|M'\|$	*entropy loss*
M_1	2.5908×10^3	4	2.25
M_2	0	8	1
M_3	2.5908×10^3	4	2.25
M_4	1.0280×10^3	14	0.25

effective attacks, but these attacks are feasible and can demonstrate the validity of these three necessary properties.

4 Attacking RSM Scheme by Utilizing the Unbalance of Mask Set

This section validates the properties by performing our attacks when one or more of these three properties are not satisfied. In order to take a fair comparison, we adopt the same RSM implementation framework as used in DPA Contest v4.2. The smart card implementation of RSM can be found from the DPA Contest v4 [23], and the study programmed the hex-file on a FunCard with an Atmel ATmega 163 micro-processor, which is the suggested platform from DPA Contest v4.

4.1 First-Order Attacks on Single Bit or Byte

The univariable first-order attack is a class of low cost methods which could be used by adversary to recover the secret key. When an unsuitable mask set which doesn't even meet **Property 1** is selected in RSM implementation, the mask set is not first-order univariable balanced, thus it is possible to directly launch a first-order attack.

The attacks can be launched with Hamming weight model or Bit model here. When $|M| < min(|M|)$ ($min(|M|)$ is the minimum number of elements in RSM mask set, [7, 8, 22] prove that if $|M|$=16, security can be achieved), if the $|M|$ is too small, the Hamming weight model can be used to attack directly. The first round implementation of DPA Contest v4.2 is showed as Alg. 1. It is clear that after $MaskSbox$, the output is $Sbox(plain \oplus key) \oplus m_{(offset(i)+1) mod 16}$, then assume the intermediate values is $Sbox(plain \oplus key) \oplus m_{random}$, which means all mask values are assumed to be m_{random}. Then we have $P(m_{random} = m_{(offset(i)+1) mod 16}) = \frac{1}{|M|} > \frac{1}{16}$, that is a large enough probability to guess right masks. Then we are able to recover the secret key.

Even if the number of mask set is suffuent, Bit model [9] can be used when there is an unbalanced bit in RSM mask set. For example, a mask set $M = \{0x01, 0x11, \cdots, 0xf1\}$, the last bit of each element in M is always 1, so the mask

Table 3: Binary representation of masks M_4 (proposed by Veshchikov *et al.* [5])

Masks	Bit Index								$\boldsymbol{m_i^1 \oplus m_i^0}$
	7	6	5	4	3	2	1	0	
0x13	0	0	0	1	0	0	1	1	**0**
0x94	1	0	0	1	0	1	0	0	**0**
0x25	0	0	1	0	0	1	0	1	**1**
0xcb	1	1	0	0	1	0	1	1	**0**
0x8e	1	0	0	0	1	1	1	0	**1**
0x5f	0	1	0	1	1	1	1	1	**0**
0xd9	1	1	0	1	1	0	0	1	**1**
0x37	0	0	1	1	0	1	1	1	**0**
0x77	0	1	1	1	0	1	1	1	**0**
0xc6	1	1	0	0	0	1	1	0	**1**
0xa8	1	0	1	0	1	0	0	0	**0**
0x38	0	0	1	1	1	0	0	0	**0**
0x05	0	0	0	0	0	1	0	1	**1**
0xea	1	1	1	0	1	0	1	0	**1**
0x70	0	1	1	1	0	0	0	0	**0**
0xe8	1	1	1	0	1	0	0	0	**0**
P(bit=0)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.625**

set does not protect the last bit of intermediate values, Bit model can be directly used to recover key. These attacks are well known and widely validated [9, 12], and we need not to repeat these attacks.

4.2 First-Order Attacks on Multi-Bit

Since **Property 1** is met but **Property 2** is not, the mask set is not multi-variable balanced on single bit, consequently we can launch a bivariable attack to recover the key. However, the unbalanced bits are restricted to the same univariable value, so it is possible to launch a univariable attacks to achieve the same results. As showed in Table 1 and Table 3, the mask set M_4 is not satisfied **Property 2**, so the attacked target of RSM implementation uses mask set M_4 (proposed by Veshchikov *et al.*). In fact, there are totally 16 mask sets proposed by Veshchikov *et al.*, none of them meet **Property 2**. To show the feasibility of attacks, the study randomly selects any of them. The value of each bit is equally likely to be 0 or 1, see Table 3.

On Table 3, we can see that although it is balanced for each bit, it is unbalanced while combining some different bits. For example, we combine 0-th bit and 1-th bit, 0 occurs 10 times while 1 occurs 6 times. Then calculate $P(m_i^0 \oplus m_i^1 = 0) = \frac{10}{16} = 0.625$. This unbalance can be utilized as two ways:

1) Find the leakage corresponding to the selected bits, then launch a bivariable attack [26, 27];
2) Combine the unbalanced bits as a new intermediate value, for example, we combine 0-th bit, 1-th bit as a 2-bit intermediate v, so $v = m_i^0\, m_i^1$. Calculate

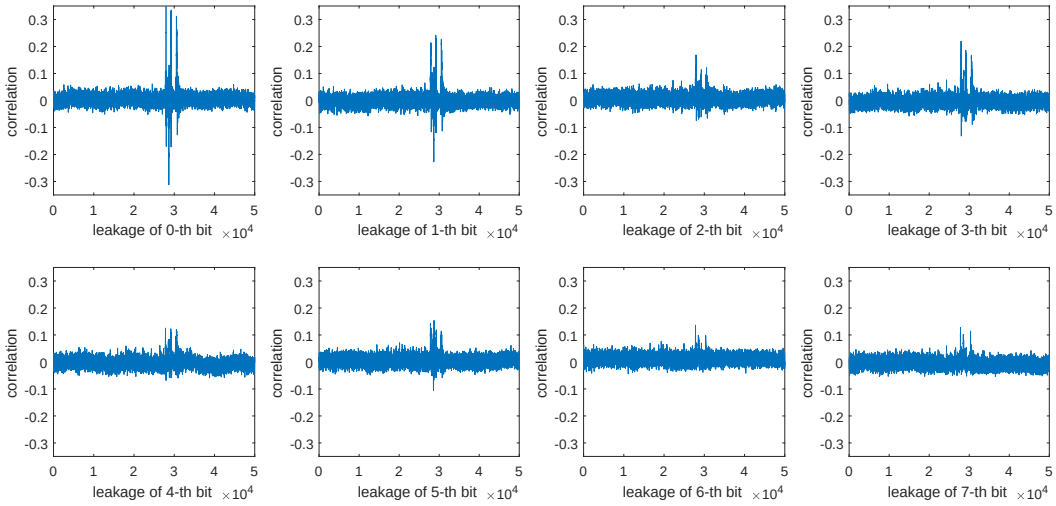


Fig. 2: The leakages for each bit of output after MaskSbox

the Hamming weight of v, $HW(v)$ and then launch an univariable first-order attack.

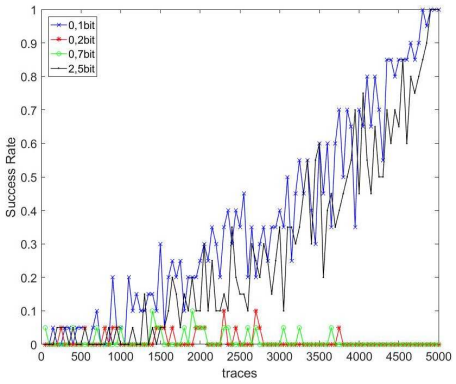
The traces are obtained from a SASEBO-W platform, the RSM framework is the same with DPA Contest v4.2 and mask set is replaced with the new one which proposed by Veshchikov *et al.* The sampling rate is set to 250MHz, and 50,000 points around the first round are taken to attack. First, we use the Pearson correlation coefficient to detect the leakage for each bit of output after MaskSbox with 5,000 traces gathered, and Fig. 2 shows the results. In fact, these leakages can directly affect our attacks.

Leakage for each bit varies in amplitude and position, and the points of interest may overlap for different bits, which may affect our attacks. Intuitively, if the leakages of unbalanced bits are not obvious, the multi-variable attacks on single bit will not be easy to success. Combine different unbalanced bits to attack and the success rate and guessing entropy [10] are shown in Fig. 3, Fig. 4.

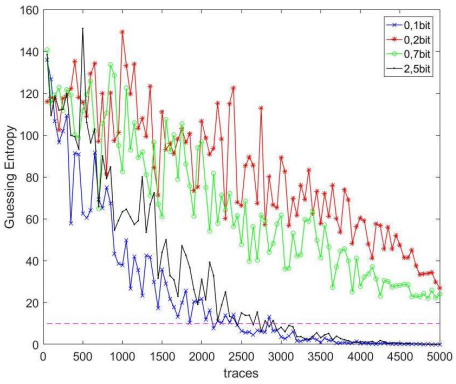
Fig. 3, Fig. 4 demonstrate that these two attacks are feasible and effective when **Property 2** is not met. Choosing different combinations can be dropped to various effects, some multi-bit attacks can recover secret key within 4,000 traces.

4.3 bivariable Attack Based on Unbalance of Adjacent Masks

Because mask sets proposed by DPA Contest v4 (v4.1 and v4.2) only meet **Property 1** and **Property 2** but not meet **Property 3**, this type of attacks is the most discussed [5, 17, 25]. When **Property 3** is not satisfied, we can combine different mask values to construct a bias then launch a bivariable attack.

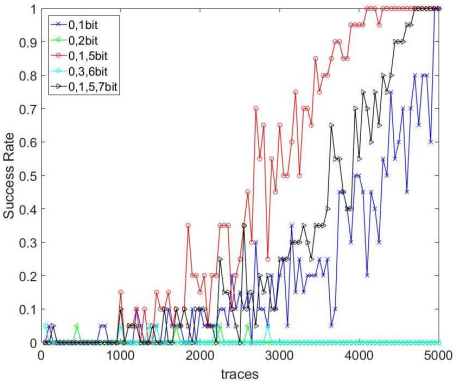


(a) The results of Success Rate

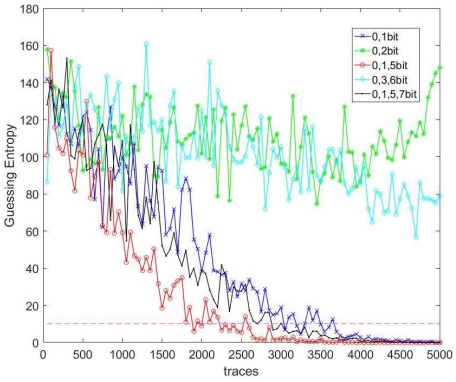


(b) The results of Guessing Entropy

Fig. 3: Combine two unbalanced bits then launch bivariable attack RSM with M_4 (proposed by Veshchikov *et al.*)



(a) The results of Success Rate



(b) The results of Guessing Entropy

Fig. 4: Combine multiple unbalanced bits then launch univariable first-order attack on RSM with M_4 (proposed by Veshchikov *et al.*)

Specifically, there are two intermediate values with adjacent masks protected in first round implementation of DPA Contest v4.2, which are $v_1 = plain \oplus key \oplus m_{offset(i)}$ and $v_2 = Sbox(plain \oplus key) \oplus m_{offset(i)+1 mod\, 16}$. Then we assume that $M' = \{m_i \oplus m_{(i+1) mod\, 16} | i \in \{0, \cdots 15\}\}$. If M' is not univariable balanced, RSM scheme is also vulnerable.

The attacks on the FPGA implementation with mask set M_1 (proposed by DPA Contest v4.1) have already been discussed [25], these attacks also utilize

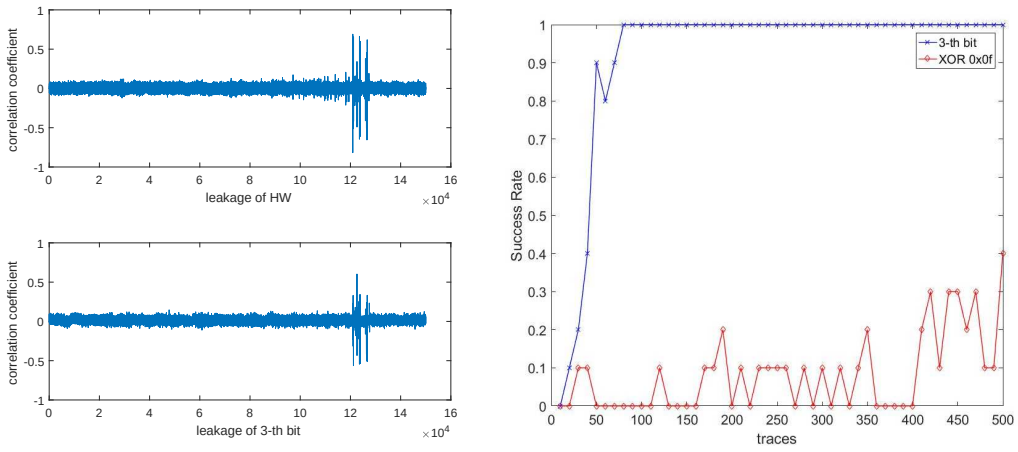
Table 4: Binary representation of M_3 (proposed by DPA Contest v4.2 *et al.* [23])

Masks	$m_i \oplus m_{(i+1)mod\,16}$	Bit Index							
		7	6	5	4	3	2	1	0
0x03	**0x0f**	0	0	0	0	**1**	1	1	1
0x0c	0x39	0	0	1	1	**1**	0	0	1
0x35	**0x0f**	0	0	0	0	**1**	1	1	1
0x3a	0x6a	0	1	1	0	**1**	0	1	0
0x50	**0x0f**	0	0	0	0	**1**	1	1	1
0x5f	0x39	0	0	1	1	**1**	0	0	1
0x66	**0x0f**	0	0	0	0	**1**	1	1	1
0x69	0xff	1	1	1	1	**1**	1	1	1
0x96	**0x0f**	0	0	0	0	**1**	1	1	1
0x99	0x39	0	0	1	1	**1**	0	0	1
0xa0	**0x0f**	0	0	0	0	**1**	1	1	1
0xaf	0x6a	0	1	1	0	**1**	0	1	0
0xc5	**0x0f**	0	0	0	0	**1**	1	1	1
0xca	0x39	0	0	1	1	**1**	0	0	1
0xf3	**0x0f**	0	0	0	0	**1**	1	1	1
0xfc	0xff	1	1	1	1	**1**	1	1	1
$\mathbf{P(m_i^b \oplus m_{(i+1)mod\,16}^b = 1)}$		0.5	0.125	0.5	0.375	**1**	0.625	0.75	0.875

the unbalance of combined mask values. In this paper, the target of RSM implementations we attack uses mask set M_3 (proposed by DPA Contest v4.2 [23]) and M_2 (proposed by Moradi *et al.* [17]). Mask set M_2 is {0x03, 0x0c, 0x35, 0x3a, 0x50, 0x5f, 0x66, 0x69, 0x96, 0x99, 0xa0, 0xaf, 0xc5, 0xca, 0xf3, 0xfc}, each bit of elements in M' is equally likely to be 0 or 1, as shown in Table 4. There are also two ways to attack:

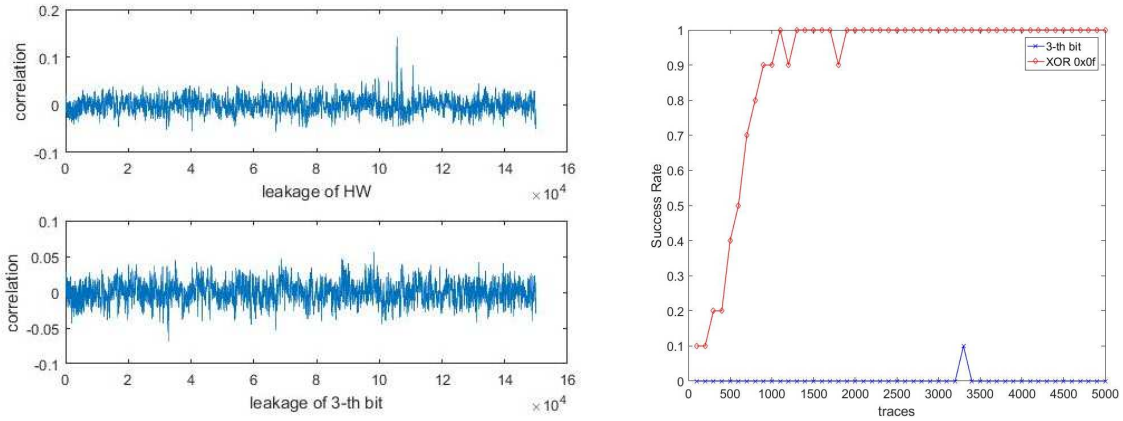
1) When we XOR adjacent masks, we have the probability of 50% to get 0x0f, which could be formulated as $\forall i \in \{0, 1, \cdots, 15\}, P(m_i \oplus m_{i+1} = 0x0f) = 0.5$. Then we can combine two intermediate mentioned above, and gain new intermediate $v_1 \oplus v_2 = plain \oplus key \oplus Sbox(plain \oplus key) \oplus m_{offset(i)} \oplus m_{(offset(i)+1)mod\,16}$. If we guess the unprotected intermediate $plain \oplus key \oplus Sbox(plain \oplus key)$ is masked by value 0x0f, we have the probability of 50% to guess right. The probability is high enough to launch a bivariable attack to recover the secret key.
2) When we XOR adjacent masks, almost all bits are unbalanced. The most unbalanced is 3-th bit, because $P(m_i^3 \oplus m_{(i+1)mod\,16}^3 = 1) = 1$. It means 3-th bit of the intermediate is totally unprotected, which is threatened by bivariable attacks with Bit model.

We attack two groups of traces respectively: one is published by DPA Contest v4.2, the other is collected from our laboratory. The published traces get higher SNR, which makes the leakage more obvious. The leakage of traces collected from our laboratory get more noise and is harder to conduct our attacks successfully. The sampling rate for collecting is set to 500MHz, and 150,000 points around the first round are taken to attack. Fig. 5 shows the result on 500 published



(a) Leakage of Hamming weight and 3-th bit (b) Comparing two attacks

Fig. 5: Attack public traces when M_3 (proposed by DPA Contest v4.2) is adopted



(a) Leakage of Hamming weight and 3-th bit (b) Comparing two attacks

Fig. 6: Attack traces collected from our laboratory when M_3 (proposed by DPA Contest v4.2) is adopted

traces and Fig. 6 shows the result on 5,000 collected traces from our laboratory.

It is obviously that SNR can strongly affect the correlation between single-bit leakage and sensitive intermediate values, when SNR is low, the leakage is

Table 5: Binary representation of M_2 (proposed by Moradi *et al.* [17])

Masks	$m_i \oplus m_{(i+1) mod\, 16}$	Bit Index							
		7	6	5	4	3	2	1	0
0x00	**0x0f**	0	0	0	0	1	1	1	1
0x0f	**0x39**	0	0	1	1	1	0	0	1
0x36	**0x0f**	0	0	0	0	1	1	1	1
0x39	**0x6a**	0	1	1	0	1	0	1	0
0x53	**0xc6**	1	1	0	0	0	1	1	0
0x95	**0xc9**	1	1	0	0	1	0	0	1
0x5c	**0x95**	1	0	0	1	0	1	1	0
0xc9	**0x36**	0	0	1	1	0	1	1	0
0xff	**0x39**	0	0	1	1	1	0	0	1
0xc6	**0x6a**	0	1	1	0	1	0	1	0
0xac	**0x36**	0	0	1	1	0	1	1	0
0x9a	**0xf0**	1	1	1	1	0	0	0	0
0xa6	**0xc9**	1	1	0	0	1	0	0	1
0xa3	**0xc6**	1	1	0	0	1	1	0	0
0x65	**0x95**	1	0	0	1	0	1	0	1
0xf0	**0xf0**	1	1	1	1	0	0	0	0
$P(m_i^b \oplus m_{(i+1) mod\, 16}^b=1)$		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

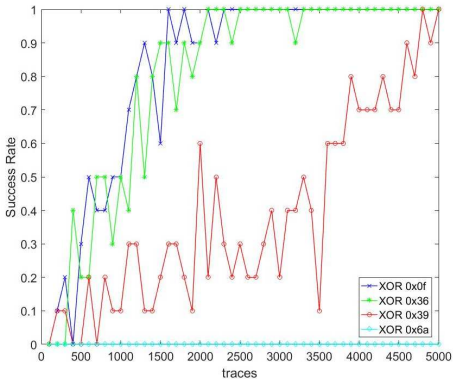
obvious and the success rate of attacking 3-th bit is high. However, we can always attack successfully within 1,000 traces.

Moradi *et al.* also found this unbalance feature, and modified the mask set. They change the order of the values in mask set in DPA Contest v4.1, and make M' more balanced. The values of each bit in M' are showed in Table 5.

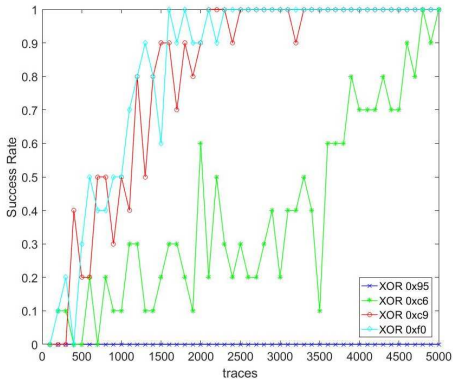
The modified mask set still doesn't meet **Property 3**. As for attacks, We can get $|M'| = \frac{16}{2} = 8$, it means the elements of M' are too few to resist against bivariable attacks. we have the probability of $\frac{1}{8} = 12.5\%$ to get that $m_i \oplus m_{(i+1) mod\, 16}$ could be compensated with each value in M'.

The sampling rate is also set to 500MHz, and 150,000 points around the first round are taken to attack. Then we calculate the new set M', there are only 8 unique elements in it, M'={0x0f, 0x36, 0x39, 0x6a, 0x95, 0xc6, 0xc9, 0xf0}. All of these elements occur with a probability of 12.5%, so all of them can be used to attack. The result is shown in Fig. 7.

In fact, the first four values equal the complement of last four values, so their attack results must be the same, which is consistent with the results shown in Fig. 7. In sum, all attacks based on unbalance of mask set are feasible. Similarly, these attacks are also feasible on FPGA implementation, besides, it is possible to mount univariable first order attack if the 2 combined intermediate variables are stored in the same registers successively.



(a) XOR first four values {0x0f, 0x36, 0x39, 0x6a}



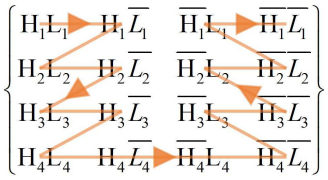
(b) XOR last four values {0x95, 0xc6, 0xc9, 0xf0}

Fig. 7: Attacks when M_2 (proposed by Moradi *et al.* [17]) is adopted

5 No Qualified Mask Set Exist for RSM

In [5,17,23], all authors try to find a qualified mask set to make RSM implementation reach a higher security level. Clearly, we can't use an exhaustive method to find it, since there are $A_{256}^{16} \approx 10^{38}$ mask sets, so we need to design a method to find a qualified mask set. In [5], they randomly generate a large number of mask sets then filter them based on inaccurate Eq.1, so all mask sets they found don't meet **Property 2** and the RSM implementation with these mask sets can be attacked by univariable first-order attacks as shown in Section 4.2.

In [23], the mask set is generated in a subtle way. Let $M = \{m_0, m_1, \cdots, m_{15}\}$, and we can assume that H denotes high 4-bit of m_i, and L denotes low 4-bit, then the mask set and the sequence of the mask values can be expressed as follows:



$\overline{H}$ and $\overline{L}$ denote the complement of H and L respectively. If high 4-bit set $\{H_1, H_2, H_3, H_4\}$ and low 4-bit set $\{L_1, L_2, L_3, L_4\}$ are [4,2,2] linear codes, the mask set M is a [8,4,4] linear codes accordingly. However, the sequence of mask values is the reason why $P(m_i \oplus m_{(i+1) mod\, 16} = 0x0f) = 0.5$, and $\forall i \in \{0, 1, \cdots, 15\}, m_i^3 \oplus m_{(i+1) mod\, 16}^3 = 1$, which makes the attacks mentioned in Section 4.3 feasible. The modified sequence proposed by Moradi *et al.* [17]

can make each bit of mask set balanced, but reordered mask set still can't meet **Property 3**. So can we reorder this mask set again to make it totally balanced with no entropy loss?

In fact, it is impossible by reordering the mask values to make these two mask sets in DPA Contest v4.1 and v4.2 meet all properties. The mask sets in DPA Contest v4 are [8, 4, 4] linear codes (v4.1) or [8,4,4] linear codes XOR with 0x03 (v4.2). So when XOR any two values from same mask set, there are only 16 outcomes, one of them is 0x00 which can't occur in set $M' = \{m_i \oplus m_{(i+1) mod\, 16} | i \in \{0, 1, \cdots, 15\}\}$. It means we always have $|M'| < 16$ no matter how we reorder the mask set M, and there will still be at least one mask value m_{rep} occurred more than once. If we calculate the probability $P(m_i \oplus m_{(i+1) mod\, 16} = m_{rep}) \geq \frac{2}{16} = 0.125$, this probability is not large enough to reduce the correlation between intermediate values and corresponding leakages to protect the secret key as shown in Section 4.3.

Furthermore, is there any other mask choosing method existing to find a mask set which can meet all three properties? The answer is also no. To prove it, we need to prove **Lemma** 1 first.

Lemma 1. *Let M be a mask set which meets **Property 1** and **Property 2**, G is a 8×16 binary matrix representation of M, each column in G represents a binary vector translated by its corresponding mask value. Then M still meets **Property 1** and **Property 2** when reorder the column or row values of G, or flip any row values of G.*

Proof. Change the columns or rows order of G, mask set M still meets **Property 1** and **Property 2**. Because **Property 1** only need the sum of each rows in G to equal to $\frac{16}{2} = 8$, and **Property 2** need the sum of XOR any two or three rows in G to equal to $\frac{16}{2} = 8$, reordering the columns or rows has no effect for M to satisfy the first two properties.

Let X, Y and Z be 16-bit values, and $X \oplus Y = Z$. Then we have $X \oplus \overline{Y} = \overline{X} \oplus Y = \overline{Z}$. If there are 8 bits in Z equal to 1, there are also 8 bits in $\overline{Z}$ equal to 1. So M still meets **Property 1** and **Property 2** when we flip any row values of G. ■

Based on **Property 1** and **Property 2**, we can reduce the number of candidate mask sets, but all remained candidate mask sets can't meet **Property 3**, so we have **Theorem** 1.

Theorem 1. *There is no 8-bit mask set in RSM which can satisfy all three properties.*

Proof. Assume a mask set M satisfies with all three properties, $M = \{m_0, \cdots, m_{15}\}$. And let $M' = \{m_i \oplus m_{(i+1) mod\, 16} | i \in \{0, 1, \cdots, 15\}\}$.

We can transform M into a binary matrix G, the size of G is 8×16, each row can be expressed as four 4-bit values as shown in Eq.9. Based on **Property 1** and **Lemma 1**, there are eight 0 bits and eight 1 bits among each row and we can change the rows or columns order optionally. Change the column order to

$$\left\{\begin{matrix} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \cdots & \cdots & \cdots & \cdots \\ \square & \square & \square & \square \end{matrix}\right\} \rightarrow \left\{\begin{matrix} f & f & 0 & 0 \\ \square & \square & \square & \square \\ \cdots & \cdots & \cdots & \cdots \\ \square & \square & \square & \square \end{matrix}\right\} \rightarrow \left\{\begin{matrix} f & f & 0 & 0 \\ f & 0 & f & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \square & \square & \square & \square \end{matrix}\right\} \tag{9}$$

$$\left\{\begin{matrix} f & f & 0 & 0 \\ f & 0 & f & 0 \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{matrix}\right\} \rightarrow \left\{\begin{matrix} f & f & 0 & 0 \\ f & 0 & f & 0 \\ 3 & 3 & 3 & 3 \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{matrix}\right\} \rightarrow \left\{\begin{matrix} f & f & 0 & 0 \\ f & 0 & f & 0 \\ 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{matrix}\right\} \rightarrow \left\{\begin{matrix} f & f & 0 & 0 \\ f & 0 & f & 0 \\ 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \\ 3 & c & c & 3 \\ 5 & a & a & 5 \\ 6 & 9 & 6 & 9 \\ 6 & 6 & 9 & 9 \end{matrix}\right\} \tag{10}$$

make the first row m_i^0 become $[f, f, 0, 0]$, which means the first two 4-bit values are f and the last two 4-bit values are 0. Then consider the first two rows $m_i^0\, m_i^1$, they represent the first two bits of mask set, and 00, 01, 10, 11 should each occur four times. So we can change the columns to make the second row become $[f, 0, f, 0]$. This transformation can be expressed as Eq.9, each "□" denotes a 4-bit value, and there are totally 8 rows in each matrix.

While the first two rows are finalized, we can filter out most unqualified mask values. Based on **principle 2**, the values when XOR any two or three rows should also be balanced. So each "□" in remaining six rows have to be chosen in $\{< 3, c >, < 5, a >, < 6, 9 >\}$, the two values in each pair is mutually complemented. Similarly, consider the first three rows $m_i^0\, m_i^1\, m_i^2$ in matrix G, 000, 001, 010, 011, 100, 101, 110, 111 should each occur twice, then we can change the column order to make the third row become $[3, 3, 3, 3]$. The Hamming distance between third row and other five rows has only three conditions, which are $HD([3, 3, 3, 3], [\square, \square, \square, \square]) = [4, 4, 0, 0]\, or\, [4, 2, 2, 0]\, or\, [2, 2, 2, 2]$ without considering the order of 4-bit values.

By the method of exclusion, it can be easily proved that the case $HD([3, 3, 3, 3], [\square, \square, \square, \square]) = [4, 2, 2, 0]$ is impossible without changing the first two rows, and $HD([3, 3, 3, 3], [\square, \square, \square, \square]) = [4, 4, 0, 0]$ occurs only when $[\square, \square, \square, \square]$=$[3, c, c, 3]$ (or $[c, 3, 3, c]$). There must be at least one row to meet $HD([3, 3, 3, 3], [\square, \square, \square, \square])$ $= [2, 2, 2, 2]$, rank this row on 4-th, then change the column order to make this row become [5, 5, 5, 5] without changing any finalized rows. Analyze the Hamming distance of the last four rows, each 4-bit value in the same row must belong to the same pair of $\{< 3, c >, < 5, a >, < 6, 9 >\}$. So the last four rows have to be $[3, c, c, 3]$ (or $[c, 3, 3, c]$),$[5, a, a, 5]$ (or $[a, 5, 5, a]$),$[6, 6, 9, 9]$ (or $[9, 9, 6, 6]$) and $[6, 9, 6, 9]$ (or $[9, 6, 9, 6]$). This transformation can be expressed as Eq.10.

After execution Eq.10, we use G' to denote the changed binary matrix. By flipping some rows, changing rows order or columns order, we can transform G' back to G, then transform G into mask set M which should meet all three properties.

Let $M_{max} = \{m_i \oplus m_j | i, j \in \{0, 1, \cdots, 15\}, i \neq j\}$, by eliminating duplicates we get $|M_{max}| < 16$, and we always have $|M'| \leq |M_{max}| < 16$, which proves there must be a mask m_{rep} fit in with $P(m_i \oplus m_{(i+1) mod\, 16} = m_{rep}) = \frac{2}{16} = 0.125$. So the mask set M can't satisfy **Property 3**, M is nonexistent. ■

In summary, we can't find a balanced mask set which meet all properties for RSM, so it is necessary to modify the framework of RSM scheme.

6 Conclusion

LEMS is essentially a masking countermeasure with a low entropy mask set, and the balance of mask set directly determines its effectiveness against SCA. In this paper, we find the flaws in original method which is used to measure the dependency between intermediate values and corresponding leakages [22]. These flaws have led to an incorrect inference for balanced mask set selection in LEMS. We fix the flaws by formally defining balance (both univariable and multivariable) of mask sets in LEMS. To prove the reasonability and validity of these definitions, we apply these definitions on RSM scheme which is a specific implementations of LEMS, then propose three fundamental properties (necessary but not sufficient) for the balanced mask set in RSM. From an adversary perspective, we analyze and evaluate three state-of-the-art RSM variants with different mask sets. The experimental results show that all RSM variants are insecure because of their unbalanced mask sets, and the one proposed by Veshchikov et al. [5] even can't thwart univariable first-order attacks. Finally, we prove that the mask set with all three properties satisfied does not exist, which means it is impossible to make RSM scheme resist bivariable attacks by changing the mask set. Furthermore, we believe that all state-of-the-art LEMS may have the same vulnerabilities if their mask set are not carefully selected.

References

1. Shivam Bhasin, Nicolas Bruneau, Jean-Luc Danger, Sylvain Guilley, Zakaria Najm. Analysis and Improvements of the DPA Contest v4 Implementation. SPACE 2014: 201-218
2. Maxime Nassar, Sylvain Guilley, Jean-Luc Danger. Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks. INDOCRYPT 2011: 22-39
3. Zeyi Liu, Neng Gao, Chenyang Tu, Yuan Ma, Zongbin Liu. Detecting Side Channel Vulnerabilities in Improved Rotating S-Box Masking Scheme - Presenting Four Non-profiled Attacks. SAC 2016: 41-57
4. Willi Geiselmann, Rainer Steinwandt. Power attacks on a side-channel resistant elliptic curve implementation. Inf. Process. Lett. 91(1): 29-32 (2004)
5. Nikita Veshchikov, Sylvain Guilley. Implementation flaws in the masking scheme of DPA Contest v4. IET Information Security 11(6): 356-362 (2017)
6. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, Olivier Rioul. Optimal side-channel attacks for multivariate leakages and multiple models. J. Cryptographic Engineering 7(4): 331-341 (2017)

7. Vincent Grosso, Francois-Xavier Standaert, Emmanuel Prouff. Low Entropy Masking Schemes, Revisited. CARDIS 2013: 33-43
8. Xin Ye, Thomas Eisenbarth. On the Vulnerability of Low Entropy Masking Schemes. CARDIS 2013: 44-60
9. Amir Moradi, Francois-Xavier Standaert. Moments-Correlating DPA. TIS@CCS 2016: 5-15
10. Fran?ois-Xavier Standaert, Tal Malkin, Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. EUROCRYPT 2009: 443-461
11. Arnab Das, Prakash Narayan. Capacities of time-varying multiple-access channels with side information. IEEE Trans. Information Theory 48(1): 4-25 (2002)
12. Thorben Moos, Amir Moradi. On the Easiness of Turning Higher-Order Leakages into First-Order. COSADE 2017: 153-170
13. Tobias Schneider, Amir Moradi, Tim Gneysu. Arithmetic Addition over Boolean Masking - Towards First- and Second-Order Resistance in Hardware. ACNS 2015: 559-578
14. Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. CRYPTO 1996: 104-113
15. Paul C. Kocher, Joshua Jaffe, Benjamin Jun. Differential Power Analysis. CRYPTO 1999: 388-397
16. Peter H. Schneider, Shankar Krishnamoorthy. Effects of correlations on accuracy of power analysis - an experimental study. ISLPED 1996: 113-116
17. Amir Moradi, Sylvain Guilley, Annelie Heuser. Detecting Hidden Leakages. ACNS 2014: 324-342
18. Andreas Gornik, Ivan Stoychev, Jrgen Oehm. A Novel Circuit Design Methodology to Reduce Side Channel Leakage. SPACE 2012: 1-15
19. Vincent Carlier, Herv Chabanne, Emmanuelle Dottax, Herv Pelletier. Electromagnetic Side Channels of an FPGA Implementation of AES. IACR Cryptology ePrint Archive 2004: 145 (2004)
20. Benedikt Gierlichs, Lejla Batina, Pim Tuyls. Mutual Information Analysis - A Universal Differential Side-Channel Attack. IACR Cryptology ePrint Archive 2007: 198
21. Maxime Nassar, Youssef Souissi, Sylvain Guilley, Jean-Luc Danger. RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs. DATE 2012: 1173-1178
22. Shivam Bhasin, Claude Carlet, Sylvain Guilley. Theory of masking with codewords in hardware: low-weight dth-order correlation-immune Boolean functions. IACR Cryptology ePrint Archive 2013: 303
23. DPA Contest v4.2. Documentation. Available at: `http://www.dpacontest.org/v4/42_doc.php`
24. Jason Waddle, David A. Wagner. Towards Efficient Second-Order Power Analysis. CHES 2004: 1-15
25. Sebastian Kutzner, Axel Poschmann. On the Security of RSM - Presenting 5 First- and Second-Order Attacks. COSADE 2014: 299-312
26. Katsuyuki Okeya, Kouichi Sakurai. A Second-Order DPA Attack Breaks a Window-Method Based Countermeasure against Side Channel Attacks. ISC 2002: 389-401
27. Eric Peeters, Francois-Xavier Standaert, Nicolas Donckers, Jean-Jacques Quisquater. Improved Higher-Order Side-Channel Attacks with FPGA Experiments. CHES 2005: 309-323
28. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. CRYPTO 1999: 398-412

29. Jean-Sbastien Coron, Louis Goubin. On Boolean and Arithmetic Masking against Differential Power Analysis. CHES 2000: 231-237
30. Stefan Mangard, Elisabeth Oswald, Thomas Popp. Power analysis attacks - revealing the secrets of smart cards. Springer 2007, ISBN 978-0-387-30857-9, pp. I-XXIII, 1-337
31. Emmanuel Prouff, Christophe Giraud, Sebastien Aumonier. Provably Secure S-Box Implementation Based on Fourier Transform. CHES 2006: 216-230
32. Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, Mike Hamburg: Meltdown. meltdownattack.com (2018)
33. Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, Yuval Yarom: Spectre Attacks: Exploiting Speculative Execution. meltdownattack.com (2018)