Table Re-Computation Based Low Entropy Inner Product Masking Scheme

Jingdian Ming[1], Yongbin Zhou[2,3,4✉], Wei Cheng[5,6], Huizhong Li[3,4]
[1]*Cyber Security Innovation Center, Jiaxing Research Institute, Zhejiang University,* Jiaxing, China
[2]*School of Cyber Security, Nanjing University of Science and Technology,* Nanjing, China
[3]*Institute of Information Engineering, Chinese Academy of Science,* Beijing, China
[4]*School of Cyber Security, University of Chinese Academy of Science,* Beijing, China
[5]*LTCI, Télécom Paris, Institut Polytechnique de Paris,* Palaiseau, France
[6]*Secure-IC S.A.S.,* Paris, France
Email: mingjingdian@outlook.com, zhouyongbin@njust.edu.cn, wei.cheng@telecom-paris.fr, lihuizhong9422@gmail.com

***Abstract*—Masking is a popular countermeasure due to its provable security. Table re-computation based Boolean masking (BM) is efficient at small masking share number, and addition chain based inner product masking (IPM) provides higher security order than BM. As a result, the natural question is: can we design a masking scheme that costs close to that of re-computation based BM while providing security comparable to that of addition chain based IPM? In this paper, we propose a table re-computation based IPM scheme that provides 3^{rd}-order security while being slightly more expensive than table re-computation based BM. Furthermore, we improve the side-channel security of IPM by randomly selecting the parameter L from an elaborated low entropy set, which we call low entropy inner product masking (LE-IPM). In an Intel Core i7-4790 CPU and ARM Cortex M4 based MCU for AES, we implemented four masking schemes, namely the addition chain based IPM and table re-computation based BM, IPM, and LE-IPM. Our proposals perform slightly slower (by about 0.8 times) than table re-computation based BM but significantly faster (at least 30 times) than addition chain based IPM. Furthermore, we assess the security of our proposals using a standard method named test vector leakage assessment methodology (TVLA). Our proposals provide the expected security against side-channel attacks according to the evaluation.**

***Index Terms*—Side-Channel Attacks, Masking Scheme, Table Re-Computation, Inner Product Masking**

I. Introduction

Side-Channel Attacks (SCAs) exploit various physical leakages of a cryptographic device to recover its sensitive data [5, 20]. During the past two decades, SCAs have been proved to be serious threats to the practical security of cryptographic devices [5]. As a consequence, the need for SCA protection has gained widespread acceptance in the industry, and there are international standards requiring cryptographic modules to defend against SCAs. For instance, NIST FIPS 140-3 [1] and ISO/IEC 17825:2016 [2], which both are security standards for security modules, claim that cryptographic modules with high-level security must concern with the mitigation of SCAs.

Masking [3, 4, 7] is one of the most investigated countermeasures against SCAs. Essentially, the masking countermeasure randomizes the dependency between sensitive intermediate and side-channel leakages by splitting the sensitive intermediate into n shares [4]. As a consequence, the effort required to recover the secret information grows exponentially with the security order d [3, 20].

Boolean masking (BM) is the most popular masking due to its simplicity [20]. In BM, the sensitive intermediate is split into shares using boolean operations, so that the security order d is no more than $n-1$. However, the main disadvantage of BM is the significant overhead associated with their implementation. Because of the high complexity of non-linear operations, most research efforts have concentrated on improving non-linear operations (e.g., SBoxes in AES) for masking schemes [3]. There are primarily two cost-effective solutions to this problem. The first is to devise a scheme for implementing SBoxes by computing over finite fields [4, 11], and the second method involves recalculating the table as a masked SBox [6, 13, 16]. The two approaches are referred to as addition chain based masking and table re-computation based masking in this paper. However, the implementation cost of masking remains at least quadratic to the security order d [6], limiting the practical use of BM in need of higher order security.

To strengthen the BM, inner product masking (IPM) [7, 9, 10] is proposed as a promising approach to provide higher order security while keeping share number n small. For instance, in addition chain based masking, it has been demonstrated that the IPM scheme implemented with $n = 2$ provides approximately 3^{rd}-order security [9], which is the same as BM with $n = 4$. In terms of the cost, the overhead of n shares IPM is higher than that of an n shares BM but lower than that of an $n+1$ share BM [8]. Due to its popularity and good performance, we focus on the IPM scheme in this paper. However, the state-of-the-art IPM implementations are based on addition chains, which are inefficient compared to table re-computation based BM if the share number is small [12]. Thus, a natural question arises: can we bridge the gap by designing a table re-computation-based IPM to guarantee the security of cryptographic implementations?

Our Contribution. In this paper, we investigate the design

This work is supported in part by National Key R&D Program of China(No. 2022YFB3103800), National Natural Science Foundation of China (No.U1936209, No.62002353) and Yunnan Provincial Major Science and Technology Special Plan Projects (No.202103AA080015).

978-3-9819263-7-8/DATE23/© 2023 EDAA

of two shares IPM based on table re-computation, which costs close to the table re-computation based BM and is significantly more efficient than addition chain based IPM. Furthermore, we propose the low entropy IPM (LE-IPM), which provides greater security than IPM with slightly higher costs. Our contributions are as follows.

First and foremost, we propose a table re-computation based IPM with two shares that achieves 3^{rd}-order security using the optimal parameter [9]. More importantly, the proposed IPM implementation is slightly expensive than table re-computation based BM and much more efficient than addition chain based IPM. In addition, we improve the IPM by allowing the parameter L to be randomly chosen from an elaborated low entropy set, which we refer to as LE-IPM in this paper. Actually, the overhead of LE-IPM is slightly higher than that of IPM, but it achieves higher security as a result.

As a concrete illustration, in two share cases, we benchmark the efficiency of four masked implementations: addition chain based IPM [8], table re-computation based BM [12] and the proposed table re-computation based IPM and LE-IPM. The four masked implementations are specifically implemented on an Intel Core i7-4790 CPU. The results show that our proposals run roughly 0.8 times as table re-computation based BM, but at least 30 times faster than addition chain based IPM in total running time. Besides, we use the Keil uVision5 simulator to evaluate the instruction cycles of the four masked implementations. The results are comparable to those obtained with the Intel Core i7-4790 CPU.

In addition, we test the security of our proposals in real-world experiments using an ARM Cortex-M4 architecture. For security evaluation, the test vector leakage assessment methodology (TVLA) [19], which is adopted in ISO/IEC 17825 [2] as a standard method for leakage assessment, is utilized. We mount TVLA from the 1^{st}-order to the 4^{th}-order moments of collected traces. The results show that no available leakage is detected in our proposed implementations.

II. Preliminaries

A. Boolean Masking

When masking is involved to protect the physical implementation of a cryptographic algorithm from d^{th}-order attacks, every sensitive intermediate x is randomly split into n shares $x_1, \ldots, x_n$ in such a way that the following relation is satisfied:

$$x = x_1 \oplus x_2 \cdots \oplus x_n \,, \tag{1}$$

where x denotes the sensitive variable and x_i denotes the i-th split share. And it is called BM since this relation is satisfied for a group of boolean operations. Usually, the d shares $\{x_2, \cdots, x_n\}$ which are called masks are randomly picked up, and the $\{x_1\}$ which is called the masked value is processed in such a way that it satisfies Eq. (1).

Particularly, look-up tables based BM is a popular countermeasure for side-channel attacks at small masking share number. When compared to addition chain based masking, this class of countermeasures for masking SBoxes has the advantage of supporting pre-processing [12]. Thus, it significantly reduces the amount of computation required once the unmasked inputs are available [13]. Indeed, the "online" computation can be as quick as a table look-up. Note that the masks will be canceled out in linear operation if the masks for all block values are the same [14]. Refreshing the masks before and after the non-linear operation is a popular solution. Another refresh operation is required to ensure that all blocks are protected by different masks [12]. It is not difficult to prove the security of the table re-computation based BM scheme, because each intermediate is independent of the sensitive key.

The size of the randomized look-up table, on the other hand, grows linearly with the number of masking shares. As a result, the RAM memory required to store pre-processed tables becomes infeasible for higher security orders, which limits the practical use of BM scheme.

B. Inner Product Masking

To strengthen BM scheme against side-channel analysis, IPM [7, 8] is proposed as an alternative with high algebraic complexity. In IPM, the sensitive value is split into n shares using mixed operations, as shown in Eq. (2).

$$x = (l_1 \otimes x_1) \oplus (l_2 \otimes x_2) \cdots \oplus (l_n \otimes x_n) \,, \tag{2}$$

where the vector $\boldsymbol{l} = (1, l_2, \cdots, l_n)$ is fixed and public (l_1 is set to 1), and x_i denotes the i-th split share. The product $l_i \otimes x_i$ is over $\mathbb{F}_{2^m}$.

Cheng *et al.* [9] demonstrated in 2020 how to select optimal $\boldsymbol{l}$ for IPM in terms of maximizing its side-channel resistance. By properly configuring the parameters, IPM is able to achieve 3^{rd}-order security with only two shares, whereas BM achieves the same security order with four shares. Moreover, if they are all based on addition chains, the cost of an n share IPM is higher than that of an n share BM but lower than that of an $n+1$ share BM. Specifically, the cost of two shares IPM is even lower than that of BM with $n = 3$ [8], but its security order is identical to that of BM with $n = 4$ [9].

However, the state-of-the-art IPM implementations are all based on addition chains [7, 8, 21], resulting in a higher cost while the share number is small. Thus, we propose a table re-computation based IPM with two shares in this paper. With the same share number, our proposal is significantly less expensive than addition chain-based IPM.

III. Table Re-computation Based IPM

A. Table Re-computation for SBoxes

The re-computation for masked table of IPM is similar to the one of BM. In general, one share of input (or output) is certain in each Substitution, namely the r_{in} and r_{out}. The algorithm is shown in Alg. 1.

In Alg. 1, we can see that two tables L and Ln are needed. Specifically, $L[i]$ outputs the $l_2 \otimes i$ over the finite field, while $Ln[i]$ outputs the inverse multiplication $l_2^{-1} \otimes i$. Since the parameter l_2 in IPM is public, the tables L and Ln are also public. It is not difficult to verify that the masked SBox can be

TABLE I: Comparison of the costs on the block cipher encryption processing

Masking Scheme	Offline		Online			
			Non-linear processing			Linear processing
	#XOR	#LUT	#XOR	#LUT	#MUL	
AD based IPM [8]	–	–	$8 \times \alpha \times n_{rd}$	$2 \times n_{rd}$	$8 \times \alpha \times n_{rd}$	$2 \times \beta \times n_{rd}$
TRC based BM [12]	$2{\times}2^m$	2^m	$4 \times n_{rd}$	n_{rd}	–	$2 \times \beta \times n_{rd}$
TRC based IPM	$2{\times}2^m$	$3{\times}2^m$	$4 \times n_{rd}$	$3 \times n_{rd}$	–	$2 \times \beta \times n_{rd}$

[1] n_{rd} denotes the number of rounds in an unprotected encryption. We have that n_{rd} equals 10 for AES-128.
[2] α denotes the number of IPM multiplicative gadgets in the addition chain for the SBox [15]. For example, there are at least 4 IPM multiplicative gadgets included in the addition chain for AES SBox.
[3] β denotes the number of operations for linear processing per round.

Algorithm 1 Table re-computation in IPM.

Input: Two randomness r_{in} and r_{out}, the unprotected SBox $SBox[2^m]$, Two public tables L and Ln
Output: Masked SBox $MSBox[2^m]$
for i = 0 to 2^m-1 **do**
  $tmp \leftarrow SBox[r_{in} \oplus L[i]]$
  $MSBox[i] = Ln[tmp \oplus r_{out}]$
end for
return $MSBox$

used in IPM, and this substitution processing can be expressed as follows.

① $x_2 \leftarrow x_2 \oplus Ln[r_{in} \oplus x_1]$// two shares r_{in}, x_2.
② $x_2 \leftarrow MSBox[x_2]$// two shares r_{out}, x_2.
③ $x_2 \leftarrow x_2 \oplus Ln[x_1 \oplus r_{out}]$// two shares x_1, x_2.

We subsequently prove the security of the table re-computation based IPM scheme. Similar to the proof for BM, the intermediates in Alg. 1 are independent of the sensitive key, since the key is not included in this algorithm. In the IPM masked encryption, the sensitive intermediate x is split into x_1 and x_2 in the form of inner product, yielding $x = x_1 \oplus l_2 \otimes x_2$. As for the operations in substitution, it can be proven that the sensitive intermediate x (*resp.* $SBox[x]$) is split into r_{in} and x_2 (*resp.* r_{out} and x_2) while the inner product form remains unchanged. Then r_{out} is refreshed by x_1 to guarantee the security for linear operations. Note that the linear processing of our proposal is identical to that of BM and addition chain based IPM, which has been proven to be secure [8].

In order to make a clearer comparison of the costs of our proposal and the other two masking, we list the required number of basic operations of these masking schemes in per encryption, as shown in Table I. Specifically, LUT denotes a look-up table operation, and MUL denotes the multiplication over the finite field. Note that the MUL operation costs much higher than XOR or LUT. Because of the high complexity of the multiplication over the finite field, the MUL operation is much more expensive than XOR or LUT. The pre-computation for the masked table is processed offline for the table re-computation based BM and IPM, as shown in Table I, so they are much more efficient on non-linear processing than addition chain based IPM. We can see that the difference between table re-computation based BM and IPM is primarily due to the number of efficient LUT operations in processing.

B. Table Re-Computation based IPM for AES as a Study Case

In this section, we present a table re-computation based IPM for AES as a case study. The masked SBox is re-computed offline utilizing Alg. 1, and we have $m = 8$ in this case. The online processing of IPM masked AES is shown in Alg. 2.

Algorithm 2 Table re-computation based IPM Masked AES-128 implementation.

Input: 16 bytes plaintext $p[16]$ and keys $key[16]$, 18 bytes randomness $r[16]$, r_{in} and r_{out}, masked SBox $MSBox[256]$, two public table $L[256]$ and $Ln[256]$
Output: 16 bytes ciphertext $c[16]$
1: $k[176] \leftarrow$ KeyExpansion(key) // Round key k
2: $[ST1, ST2] \leftarrow [r \oplus p, Ln[r]]$ // Store the two shares
3: $ST1 \leftarrow ST1 \oplus k[0:15]$
4: **for** i = 1 to 9 **do**
5: **for** j = 0 to 15 **do**
6: $ST2[j] \leftarrow Ln[ST1[j] \oplus r_{in}] \oplus ST2[j]$
7: $ST2[j] \leftarrow MSbox[ST2[j]]$
8: $ST2[j] \leftarrow Ln[ST1[j] \oplus r_{out}] \oplus ST2[j]$
9: **end for**
10: $ST1, ST2 \leftarrow$ ShiftRows($ST1$),ShiftRows($ST2$)
11: $ST1, ST2 \leftarrow$ MixColumns($ST1$),MixColumns($ST2$)
12: $ST1 \leftarrow ST1 \oplus k[i*16 : i*16+15]$
13: **end for**
14: **for** j = 0 to 15 **do**
15: $ST2[j] \leftarrow Ln[ST1[j] \oplus r_{in}] \oplus ST2[j]$
16: $ST2[j] \leftarrow MSbox[ST2[j]]$
17: $ST2[j] \leftarrow Ln[ST1[j] \oplus r_{out}] \oplus ST2[j]$
18: **end for**
19: $ST1, ST2 \leftarrow$ ShiftRows($ST1$), ShiftRows($ST2$)
20: $ST1 \leftarrow ST1 \oplus k[160:175]$
21: $c \leftarrow ST1 \oplus L[ST2]$
22: **return** c

We can see that totally 18 bytes of randomness are needed in this algorithm. Among them, 16 bytes of randomness $r[16]$ are used for encoding the plaintexts, and 2 bytes of randomness r_{in} and r_{out} are used for substitution. Extra 3×2^8 bytes are required to store the tables $MSBox$, L and Ln. With the help of Alg. 2, we can demonstrate the security of table

re-computation based masking easier. In Alg. 2, the sensitive intermediate per operation is split into two shares stored in $ST1$ and $ST2$. The security for non-linear SBox has been proven in Sec. III-A. Overall, the IPM masked AES has a similar running time to the BM masked one and is nearly twice as long as the unprotected one.

C. Low Entropy Inner Product Masking

In IPM scheme, the vector $\boldsymbol{l}$ is public [7]. Consequently, the table L and Ln are also public in table re-computation based IPM. To increase the security level of IPM, a direct approach is to keep the parameter $\boldsymbol{l}$ private.

Actually, IPM is a type of code-based masking scheme, and it has been proved that there are multiple optimal vectors $\boldsymbol{l}$ for IPM scheme [9]. Based on this finding, the IPM scheme can be improved in the following ways. Firstly, all optimal vectors are included in the public set $\mathcal{V}$. Before running the IPM scheme, the vector $\boldsymbol{l}$ is randomly selected from the public set $\mathcal{V}$. Since the size of public set $\mathcal{V}$ is much less than 2^m, we refer to it low entropy inner product masking (LE-IPM) in this paper. Intuitively, the side-channel resistance of LE-IPM should be higher than the original one, because the private $\boldsymbol{l}$ is still the optimal parameter in a coding-theoretic approach [9].

In a two shares case, the extra cost for LE-IPM is negligible when compared to that of the IPM scheme, since the main overhead is additionally selecting the parameter from the set $\mathcal{V}$. Specifically, the extra tables L and Ln are determined by the public parameter l_2. And there are a total of 12 optimal values for l_2 [17] to make IPM achieve 3^{rd}-order security, which is listed below in the form of a set $\mathcal{V}$.

$$\mathcal{V} = \{\,23,\ 46,\ 51,\ 54,\ 81,\ 92,\ 95, 102,\ 108,\ 162,\ 165,\ 184\,\}.$$

Since the size of $\mathcal{V}$ equals 12 for the finite field used in AES [17], the memory cost for storing L and Ln in table re-computation based LE-IPM is 12 times that of table re-computation based IPM. As for AES, the memory for storing public tables is 6 KB ($12 \times 256 \times 2$ Bytes). However, we find that if $i \in \mathcal{V}$, then $i^{-1} \in \mathcal{V}$. It also holds for other finite fields in [9], and the formal proof will be our future work. With this observation, the Ln can be generated by L with an offset setting, and the public table for AES occupies only 3 KB.

IV. Efficiency Evaluation

A. Evaluation of Running Time On CPU

On an Intel Core i7-4790 CPU running at 3.60GHz, the addition chain-based IPM and table re-computation-based BM and IPM are implemented. As discussed in Sec. II, the two shares IPM is able to achieve 3^{rd}-order security if the parameter l_2 is an optimal one. Meanwhile, the two shares BM can only achieve first-order security. Note that in LE-IPM, the parameter l_2 is randomly selected from the low entropy set $\mathcal{V}$ and is kept private to the adversary, resulting in a security order greater than three. We believe that the security order of LE-IPM is lower than four due to the low entropy set $\mathcal{V}$, and we intend to prove this in the future.

In this experiment, we evaluate the running time when the codes are compiled with a generic "x86-64" option and the "-O3" option. The results are shown in Table II. Note that the running time of per encryption is measured as an average of a total of 2,000,000 runs. In Table II, we can observe that the table re-computation based BM, IPM and LE-IPM are considerably faster (almost 40 times on x86 compiled option and 30 times for "-O3" compiled option) than addition chain based IPM in terms of online running time and total. As for the comparison of table re-computation based masking, we can see that the cost of LE-IPM is a little bit higher than IPM, and the proposed IPM and LE-IPM are roughly 20% higher than BM in total running time.

B. Evaluation of Embedded Implementations

We additionally take ARM Cortex-M4 as a software target. Specifically, we evaluate the performance of four masking schemes on an STM32F4 MCU based on ARM Cortex-M4. Thanks to the simulator in Keil uVision5, the performance results are straightforwardly in cycles, as shown in Table III.

It is obvious that the results on the MCU are different from those of CPU, especially for the offline processing. Compared to the re-computation based BM, the additional overheads of our proposed IPM and LE-IPM are nearly 50% and 90% on the typical 32-bit processor, respectively. Fortunately, the cost of offline processing accounts for a small proportion of that of the total encryption, and the costs of the three masking schemes in online processing are very close. So that the costs of IPM and LE-IPM are roughly 20% higher than that of table re-computation based BM as well. More importantly, the table re-computation based IPM and LE-IPM are much more efficient (about 35 times) than the addition chain based IPM.

V. Security Evaluation

A. Experimental Setup

Our measurement setup is shown in Fig. 1. It consists of the ChipWhisperer-Lite board, the CW308 UFO board and the CW308T-STM32F4 target board. The target board contains a 32-bit ARM Cortex-M4 MCU with an STM32F405 device.

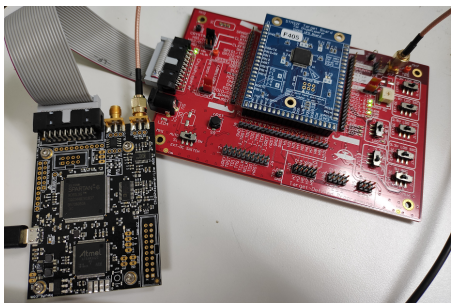


Fig. 1: Our environment for collecting power traces.

It is a relatively ideal environment with low noise for power analysis since the highest Signal-to-Noise Ratio (SNR) is close to 100. For comparison, the highest SNR of DPA Contest v4.1, which is a public data set for side-channel analysis, is roughly 30 [18]. Consequently, this environment has been widely used in the field of side-channel analysis [11].

TABLE II: Comparison of the security and running time in masked AES implementations on an Intel Core i7-4790 CPU running at 3.60GHz.

Security & **Running Time**	**Masking scheme with two shares**			
	AD based IPM [8]	TRC based BM [12]	TRC based IPM (our proposal)	TRC based LE-IPM (our proposal)
Security order	3^{rd}-order	1^{st}-order	3^{rd}-order	$\geq 3^{rd}$-order
Compiling with a generic x86 option ("-x86-64")				
Offline [ms]	–	0.598	0.655 (9.53% ↑)	0.914 (52.84% ↑)
Online [ms]	183.336	3.568	3.705 (3.84% ↑)	3.858 (8.13% ↑)
Total [ms]	183.336	4.166	4.360 (4.66% ↑)	4.772 (14.55% ↑)
Compiling with an optimized option ("-O3")				
Offline [ms]	–	0.181	0.221 (22.10% ↑)	0.233 (28.73% ↑)
Online [ms]	24.937	0.513	0.652 (27.10% ↑)	0.661 (28.85% ↑)
Total [ms]	24.937	0.694	0.873 (25.79% ↑)	0.894 (28.82% ↑)

TABLE III: Comparison of the security and clock cycles in masked AES implementations on a Cortex-M4 MCU.

Security & **#Cycles**	**Masking scheme with two shares**			
	AD based IPM [8]	TRC based BM [12]	TRC based IPM (our proposal)	TRC based LE-IPM (our proposal)
Security order	3^{rd}-order	1^{st}-order	3^{rd}-order	$\geq 3^{rd}$-order
Offline	–	3988	6036 (51.35% ↑)	7656 (91.98% ↑)
Online	1340177	26184	29421 (12.40% ↑)	30278 (15.64% ↑)
Total	1340177	30172	35457 (17.52% ↑)	37934 (25.73% ↑)

B. Evaluation of Side-Channel Resistance

To evaluate the side-channel resistance, we apply test vector leakage assessment methodology (TVLA) [19] to the addition chain based and table re-computation based IPM implementations, which are achieved on the experimental setup. TVLA was proposed at a NIST non-invasive attack testing workshop in 2011 and has since become one of the most popular methods for assessing leakage. Specifically, TVLA is based on student's t-test, and aims to detect the difference in the expectation of two leakage groups [19], e.g., the two leakage groups can be divided by one bit of the sensitive intermediate. If the absolute value of the output of t-test, expressed as $|t|$, is greater than 4.5, the corresponding leakages are considered to be related to the sensitive intermediate, consequently the tested implementations are deemed vulnerable.

Firstly, we implement a two shares addition chain based IPM scheme for AES SBox referring to the public higher-order masked implementation by Balasch *et al.* [8]. The sampling rate is set to be 30 MS/s. A total of 500,000 traces for the masking implementation are recorded for t-test, and 150 points around the Substitution are taken into the test. The collected traces are then subjected to the t-test, and the groups are divided by the least significant bit. Note that in the higher order t-tests, the leakages are pre-processed by higher order moments after a centralization. The results are shown in Fig. 2.

We can see in Fig. 2 that there are no leakages related to the sensitive intermediate according to the 1^{st}-, 2^{nd}- and 3^{rd}-order t-tests, but there are according to the 4^{th}-order t-test.

Since the XOR in step three runs after an look-up table operation for the affine function, the leakages corresponding to x_1 and x_2 partially overlap. Thus, the overlapped leakages are related to sensitive values according to 4^{th}-order t-test.

We implement the table re-computation based IPM and LE-IPM schemes for AES SBox, which are proposed in this paper. Similarly, 500,000 traces are recorded for t-tests, with 150 points used in the detection. We then perform the t-test on the collected traces, and the results are shown in Fig. 3 and Fig. 4. The standard t-test method detects no available leakages related to the sensitive intermediate.

VI. Conclusion

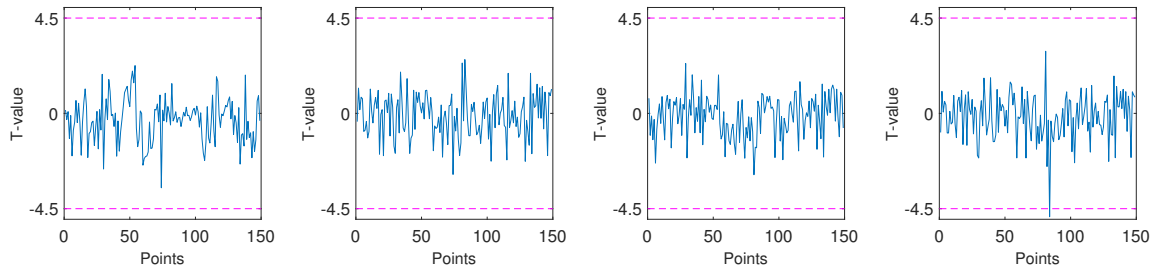
In this paper, we propose a table re-computation based IPM scheme, which provides a higher security order without compromising the costs thanks to the public parameter L in IPM. Moreover, we improve the side-channel security of IPM by keeping the parameter L to be randomly selected from an elaborated low entropy set. We have implemented the masking schemes for AES in real world devices. The results show they run slightly slower than BM scheme but much faster than addition chain based IPM. However, because our proposals are with two shares, they can only approach 3^{rd}-order security. Our future work will include the design and evaluation of a table re-computation based IPM with more shares.

References

[1] Security Requirements for Cryptographic Modules, 2009. [Online]. Available: https://csrc.nist.gov/projects/fips-140-3-development

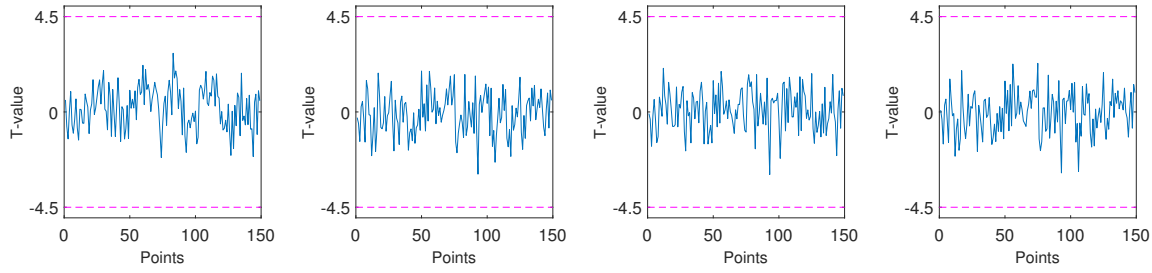
[2] ISO/IEC 17825:2016 Information technology – Security techniques – Testing methods for the mitigation of non-invasive attack classes against cryptographic modules, 2016. Available: https://www.iso.org/standard/60612.html

[3] Matthieu Rivain, Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. CHES 2010: 413-427.



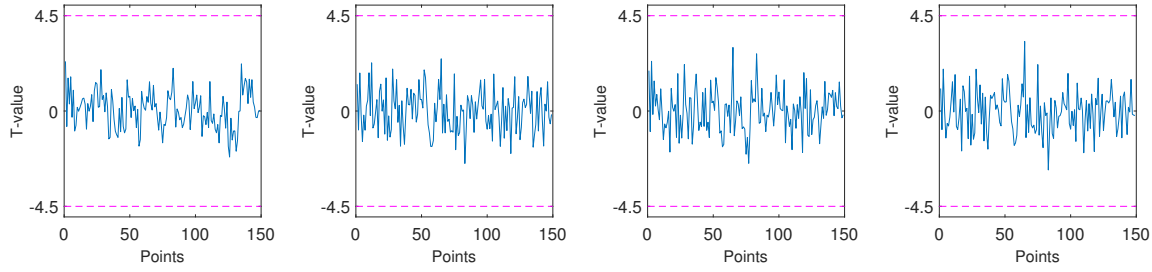
(a) 1^{st}-order t-test (b) 2^{nd}-order t-test (c) 3^{rd}-order t-test (d) 4^{th}-order t-test

Fig. 2: T-test on addition chain based IPM implemented on a 32-bit ARM Cortex-M4 MCU.



(a) 1^{st}-order t-test (b) 2^{nd}-order t-test (c) 3^{rd}-order t-test (d) 4^{th}-order t-test

Fig. 3: T-test on table re-computation based IPM implemented on a 32-bit ARM Cortex-M4 MCU.



(a) 1^{st}-order t-test (b) 2^{nd}-order t-test (c) 3^{rd}-order t-test (d) 4^{th}-order t-test

Fig. 4: T-test on table re-computation based LE-IPM implemented on a 32-bit ARM Cortex-M4 MCU.

[4] Yuval Ishai, Amit Sahai, David A. Wagner. Private Circuits: Securing Hardware against Probing Attacks. CRYPTO 2003: 463-481.

[5] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. CHES 2000: 238-251.

[6] Jean-Sebastien Coron, Franck Rondepierre, Rina Zeitoun. High Order Masking of Look-up Tables with Common Shares. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018(1): 40-72 (2018).

[7] Josep Balasch, Sebastian Faust, Benedikt Gierlichs. Inner Product Masking Revisited. EUROCRYPT (1) 2015: 486-510.

[8] Josep Balasch, Sebastian Faust, Benedikt Gierlichs, Clara Paglialonga, Francois-Xavier Standaert. Consolidating Inner Product Masking. ASIACRYPT (1) 2017: 724-754.

[9] Wei Cheng, Sylvain Guilley, Claude Carlet, Sihem Mesnager, Jean-Luc Danger. Optimizing Inner Product Masking Scheme by a Coding Theory Approach. IEEE Trans. Inf. Forensics Secur. 16: 220-235 (2021).

[10] Romain Poussier, Qian Guo, Francois-Xavier Standaert, Claude Carlet, Sylvain Guilley. Connecting and Improving Direct Sum Masking and Inner Product Masking. CARDIS 2017: 123-141.

[11] Jingdian Ming, Huizhong Li, Yongbin Zhou, Wei Cheng, Zehua Qiao. Revealing the Weakness of Addition Chain Based Masked SBox Implementations. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2021(4): 326-350 (2021).

[12] Emmanuel Prouff, Remi Strullu, Ryad Benadjila, Eleonora Cagli, Cecile Dumas. Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database. IACR Cryptol. ePrint Arch. 2018: 53 (2018).

[13] Annapurna Valiveti, Srinivas Vivek. Higher-Order Lookup Table Masking in Essentially Constant Memory. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2021(4): 546-586 (2021).

[14] Jingdian Ming, Yongbin Zhou, Huizhong Li, Qian Zhang. A secure and highly efficient first-order masking scheme for AES linear operations. Cybersecurity. 4(1): 14 (2021).

[15] Jean-Sebastien Coron, Arnab Roy, Srinivas Vivek. Fast Evaluation of Polynomials over Binary Finite Fields and Application to Side-Channel Countermeasures. CHES 2014: 170-187.

[16] Jean-Sebastien Coron. Higher Order Masking of Look-Up Tables. EUROCRYPT 2014: 441-458.

[17] Wei Cheng. What can information guess? Towards information leakage quantification in side-channel analysis. Information Theory [cs.IT]. Institut Polytechnique de Paris, 2021.

[18] Jingdian Ming, Yongbin Zhou, Wei Cheng, Huizhong Li, Guang Yang, Qian Zhang. Mind the Balance: Revealing the Vulnerabilities in Low Entropy Masking Schemes. IEEE Transactions on Information Forensics and Security. 15: 3694-3708, 2020.

[19] G. Goodwill, B. Jun, J. Jun, P. Rohatgi. A Testing Methodology for Side-Channel Resistance Validatio. NIST non-invasive attack testing workshop, 2011.

[20] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. CRYPTO 1999: 398-412.

[21] Qianmei Wu, Wei Cheng, Sylvain Guilley, Fan Zhang, Wei Fu. On Efficient and Secure Code-based Masking: A Pragmatic Evaluation. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022(3): 192-222 (2022)