IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2021, No. 4, pp. 326–350. DOI:10.46586/tches.v2021.i4.326-350

Revealing the Weakness of Addition Chain Based Masked SBox Implementations

Jingdian Ming[1,2], Huizhong Li[1,2], Yongbin Zhou[1,2,3†], Wei Cheng[4] and Zehua Qiao[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{mingjingdian,lihuizhong,zhouyongbin,qiaozehua}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] School of Cyber Security, Nanjing University of Science and Technology, Nanjing, China
[4] Télécom Paris, Polytechnique de Paris, Palaiseau, France
wei.cheng@telecom-paris.fr

Abstract. Addition chain is a well-known approach for implementing higher-order masked SBoxes. However, this approach induces more computations of intermediate monomials over $\mathbb{F}_{2^n}$, which in turn leak more information related to the sensitive variables and may decrease its side-channel resistance consequently. In this paper, we introduce a new notion named polygon degree to measure the resistance of monomial computations. With the help of this notion, we select several typical addition chain implementations with the strongest or the weakest resistance. In practical experiments based on an ARM Cortex-M4 architecture, we collect power and electromagnetic traces in consideration of different noise levels. The results show that the resistance of the weakest masked SBox implementation is close to that of an unprotected implementation, while the strongest one can also be broken with fewer than 1,500 traces due to extra leakages. Moreover, we study the resistance of addition chain implementations against profiled attacks. We find that some monomials with smaller output size leak more information than the SBox output. The work by Duc *et al.* at JOC 2019 showed that for a balanced function, the smaller the output size is, the less information is leaked. Thus, our attacks demonstrate that this property of balanced functions does not apply to unbalanced functions.

Keywords: Side-channel attacks, masking countermeasure, addition chain implementation · resistance evaluation

1 Introduction

Side-channel attacks (SCAs) exploit various physical leakages, e.g., the running time [Koc96], the power consumption [CRR02] or the electro-magnetic emanations [CCDP04], of a cryptosystem to recover its sensitive data. Since the pioneering work proposed in [Koc96], many implementations of block ciphers have been practically broken by SCAs [CRR02]. Consequently, protecting cryptographic implementations against SCAs has been a challenging and longstanding issue for the embedded systems industry.

Among all countermeasures against SCAs, masking is the most widely used since it is not only provably secure, but also device-independent. Specifically, the basic idea of masking is to apply secret sharing schemes [Sha79]. Namely, each sensitive variable x is split into $d+1$ shares such that $x = x_0 \perp x_1 \cdots \perp x_d$, where d is called the masking order. In that case, an implementation should be resistant against d-th order attacks, in which

†Yongbin Zhou is the corresponding author.

Received: 2021-04-15 Accepted: 2021-06-15 Published: 2021-08-11

the adversary combines leakage information from at most d intermediate variables. In this paper, we shall consider that $\perp$ is the exclusive-or (XOR) operation.

When protecting a cryptographic algorithm, the linear operations are simple to mask. If F is a linear function, we have $F(x_0 \oplus x_1) = F(x_0) \oplus F(x_1)$ and it suffices to compute the shares $F(x_i)$ separately. In comparison, masked non-linear operations are more difficult to implement. There are mainly two ways with an acceptable cost to solve this problem: 1) implement by look-up tables (LUT), or 2) compute the unrolled functions over a finite field. The first solution costs at least 4 times more in running time than that of the second one [Cor14, CRZ18] in higher-order masked implementations. As for implementations by computing over a finite field, the known methods are based on similar ideas. Specifically, the non-linear operation can be expressed as a sequence of squares and multiplications over $\mathbb{F}_{2^n}$. These non-linear multiplications can be then implemented using previously known schemes, such as ISW [ISW03]. The Rivain-Prouff masking scheme is the first provably secure higher-order masking for AES [RP10] using addition chain, as shown in Fig. 1. In this way, the AES SBox can be masked at any order d. Later, it was extended to a generic method for higher-order masking in [CGP+12] by Carlet *et al.* based on the fact that given n-bit SBox can be represented by a polynomial $\sum_{i=0}^{2^n-1} u_i x^i$ over $\mathbb{F}_{2^n}$ using Lagrange's interpolation theorem. Hence, any n-bit SBox can be expressed as a sequence of linear squares and non-linear multiplications over $\mathbb{F}_{2^n}$. Then from a theoretical perspective, Roy and Coron *et al.* [RV13, CRV15] further reduced the complexity of several well-known SBoxes. The best-known method for fast polynomial evaluation was proposed by Carlet *et al.* [CPRR15]. From an implementation perspective, Coron *et al.* proposed to use common shares to further improve the addition chain in parallel implementations [CGPZ16]. Since SBox implementations based on these methods can be expressed as several squares and multiplications, we refer to them as *addition chain based masked SBoxes* in this paper. Actually, lots of masking schemes, such as Boolean masking [RP10], Mixed Additive and Multiplicative Masking [MQ18], Inner Product Masking [CGC+21], are using addition chain to implement SBoxes.

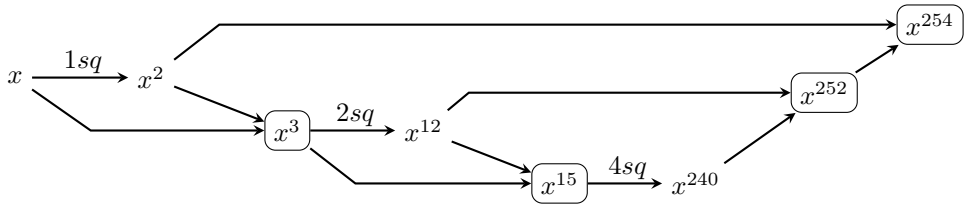


Figure 1: The computation of x^{254} used in [RP10]. Monomial with border means that the computation for this monomial is a multiplication, and the number over the arrow represents the number of squares.

However, addition chain implementations induce lots of extra computations. According to [MR04], information leakage is inherent in the physical execution of any cryptographic algorithm. Thus the extra computations are likely to lead to more information leakages and reduce the side-channel resistance of cryptographic implementations. There are three main factors in consideration to measure the side-channel resistance $\mathcal{R}$ of cryptographic implementations: noise level $\mathcal{P}_\mathcal{N}$, protection level $\mathcal{P}_P$ and the function $\mathcal{P}_F$ itself. Their relationship can be expressed as Eq.(1).

$$\mathcal{R} = \mathcal{P}_\mathcal{N} \circ \mathcal{P}_P \circ \mathcal{P}_F. \tag{1}$$

Specifically, $\mathcal{P}_\mathcal{N}$ is influenced by the target platform, measurement environment, and other physical factors. For simplification, it is often modeled as the variance of Gaussian noise under the Hamming weight leakage model [PR13]. $\mathcal{P}_P$ denotes the resistance gain from

the protected scheme. While the masking scheme is adapted to the implementation, the protection level can be regarded as masking order d. As for $\mathcal{P}_F$, [Pro05] has proved that the resistance of functions against SCAs differs greatly. Thus, even if a d^{th}-order masked implementation is resistant to d^{th}-order attacks, an adversary can launch a $(d+1)^{th}$-order attack on some weaker intermediate functions with a low number of traces (close to that of a d^{th}-order attack). Since addition chain implementations lead to a lot of extra computations, it is crucial to figure out the impact of extra computations on the side-channel resistance of the complete implementation.

Related work about measuring $\mathcal{P}_F$ of a function, like the confusion coefficient (CC) [FLD12] and the transparency order (TO) [Pro05], reveals the inherent mathematical properties of SBoxes related to side-channel attacks. However, a CC value is mostly used to decouple the contributions of physical implementations and cryptographic algorithms on side-channel leakages, while we mainly focus on the intrinsic resiliency of a function itself. As for TO and its two variants [CSM+17, LZM+20], they can only be utilized to evaluate the side-channel resistance of balanced functions, which will be demonstrated in Section 4. Overall, nearly half of computations for the monomials over $\mathbb{F}_{2^n}$ are not balanced functions, e.g., $F(x) = x^{15}$ over $\mathbb{F}_{2^8}$ in Fig. 1, which cannot be quantified by the known notions.

Our contributions. In this paper, we investigate the side-channel security of addition chain based masked SBox implementations. We find that the induced unbalanced functions extremely impact the resistance against SCAs. Our contributions are fourfold as follows.

Firstly, the side-channel security of addition chain implementations has been studied in [PR13, DDF19]. They assumed that the leakages in each operation (square or multiplication) are under the same noisy model, meanwhile sensitive information in these leakages follows the same bound related to a noise parameter ($\mathcal{P}_\mathcal{N}$). It can be seen as a simplified scenario, but the leakages from each operation are actually different and related to the function itself. Thus, we study how induced extra operations impact the security of addition chain implementations. We find if the addition chain is not carefully chosen, the side-channel resistance of a masked SBox may be similar to an unprotected implementation.

Since almost half of the monomial functions over $\mathbb{F}_{2^n}$ are unbalanced, we introduce the notion of polygon degree to quantify the resistance of a general function in the Hamming weight leakage model. We demonstrate that this notion is independent of the masking order for Boolean masking scheme under higher-order attacks. Then we demonstrate its practical implications by simulated experiments over $\mathbb{F}_{2^4}$, $\mathbb{F}_{2^6}$ and $\mathbb{F}_{2^8}$.

Moreover, we describe two adversaries with limited and unlimited computational power, and demonstrate how to measure the resistance of an addition chain implementation using polygon degree under the adversaries' attacks. We launch correlation power analysis (CPA) and correlation electromagnetic analysis (CEMA) on practical addition chain based SBox implementations, since they have been proved to be the most efficient non-profiled attacks under the Hamming weight leakage model [DPRS11]. CPA and CEMA are performed corresponding to different noise scenarios. All target functions are implemented on an STM32 chip based on an ARM Cortex-M4 architecture. The results show that all of them, including the theoretically strongest implementations, can be broken with 1,500 traces.

We also study the resistance of addition chain implementations under profiled attacks, specifically template attack and deep learning based profiled attack. For simplicity, we attack the worst monomial computation and the SBox computation (output step) in AES. It is a typical comparison to demonstrate the weakness for an addition chain implementation with unbalanced functions. In our simulation, with increasing noise, it is more efficient to attack unbalanced monomial computations rather than the SBox output for profiled attacks. The work in [DFS19] showed that the fewer leakage values a function has, the less information leaks for balanced functions. Our attacks demonstrate the property for

balanced functions is not applicable to unbalanced functions.

2 Preliminaries

2.1 Notations

Let x denote an n-bit value and the Hamming weight of x is denoted by $HW(x)$. Let F denote the function from $\{0,1\}^n$ to $\{0,1\}^m$ (usually $n \in \{4,6,8\}$). If m equals 1, then the function is called Boolean function. An (n,m)-function F can be seen as a multi-output Boolean function, so the function F can be expressed as $(f_1, \ldots, f_m)$, where f_j $(1 \leq j \leq m)$ denotes the Boolean function from n-bit inputs to the j-th bit of outputs of F. We use $\mathcal{F}$ to denote the whole processing required to compute an (n,m)-function, where F_i denotes the i-th intermediate function in $\mathcal{F}$ and w denotes the total number of intermediate functions. So we have $\mathcal{F} =< F_1, F_2, \ldots, F_w >$ for a sensitive input variable x. Thus, the processing in Fig. 1 can be expressed as $\mathcal{F}_{rp} =< x, x^2, x^3, x^6, x^{12}, \ldots, x^{254} >$.

Let $\dot{K}$ denote the secret key of the cryptographic algorithm, and K denotes the guessed key while attacking. $T = \{T_i | i = 1, 2, \ldots, N\}$ denotes the plaintexts of all N traces, and T_i denotes the plaintext of i-th trace. We suppose that the computations in the cryptographic implementation are done on n-bit words, which means that these intermediates can be seen as elements over $\mathbb{F}_2^n$, so the $\dot{K}$, K, T_i belong to $\mathbb{F}_2^n$. And we denote $\mathcal{L}(T_i, \dot{K})$ as a leakage during an execution of a cryptographic algorithm for plaintext T_i with correct key $\dot{K}$. In this paper, we distinguish the additions of integers in $\mathbb{R}$, denoted by $+$, and the additions mod 2, denoted by $\oplus$ (XOR operation).

2.2 Addition Chain Based Masked SBoxes

The principle of masking is to split a sensitive value into $d+1$ shares $x_0, x_1 \ldots x_d$, with the relation:

$$x = x_0 \oplus x_1 \ldots \oplus x_d, \tag{2}$$

where x denotes the sensitive value, x_i the shares and d the masking order. Usually, the d shares $\{x_1, \ldots, x_d\}$, called masks, are randomly picked and the x_0, called the masked value, is processed such that it satisfies Eq.(2). It has been shown that the complexity of mounting a successful side-channel attack against a masked implementation increases exponentially with the order d [PR13, DDF19] when the noise level is relatively high.

It is trivial to compute linear functions in a masked fashion, since it can be achieved by computing $y_i = F(x_i)$ for $0 \leq i \leq d$. However, it is not trivial to mask a non-linear function. It has been shown that any n-bit SBox can be represented by a polynomial $SBox(x) = \sum u_i x^i$ over $\mathbb{F}_{2^n}$ [CGP+12], and the u_i can be obtained from the look-up table by applying Lagrange's Interpolation Theorem. Thus, the common approach in turn decomposes each power function in terms of squares and non-linear multiplications, where the non-linear multiplications can be implemented, e.g., using the ISW scheme [ISW03]. The addition chain [Knu97] is defined as:

Definition 1 (Addition Chain)**.** An addition chain S for α ($\alpha \in \mathbb{N}$) is a sequence of integers

$$a_0 = 1, a_1, a_2, ..., a_r = \alpha, \tag{3}$$

for every $i = 1, 2, \ldots, r$, there exist some $0 \leq j, k \leq i$ such that $a_i = a_j + a_k$.

In fact, the exponential computation for each monomial in an SBox can be expressed as an addition chain. Note that when performing an exponentiation of the multiplicative group over a finite field, the exponent can be reduced modulo the group order, which equals the field size minus one. And the Frobenius endomorphism (i.e. squaring for a

binary field) is essentially for free. Thus, two exponents β_1 and β_2 that satisfy $\beta_1{=}2^i\beta_2$ mod (2^n-1) require the same number of costly multiplications. To implement SBox efficiently, the main work in past years focuses on evaluating any SBox with a low number of multiplications [CGP+12, CRV15, CPRR15]. Specifically, [CGP+12] introduces cyclotomic classes. All the power functions with exponents within a given cyclotomic class have the same complexity for multiplications over $\mathbb{F}_{2^n}$. The cyclotomic class, which is denoted by C_β, is defined as:

Definition 2 (Cyclotomic Class)**.** Let $\beta \in \{0, 1, \ldots, 2^n - 2\}$ over $\mathbb{F}_2^n$. The cyclotomic class of β, denoted by C_β, is defined as

$$C_\beta = \left\{\beta \cdot 2^i \,(\mathrm{mod} 2^n - 1) \,|i = 0, 1, \ldots, n-1\right\} \tag{4}$$

Namely, a power x^{β_1} can be computed from a power x^{β_2} without any non-linear multiplication if and only if β_1 and β_2 lie in the same cyclotomic class. The fast implementation using cyclotomic classes is called CC addition chain [CRV15]. Roy and Vivek [RV13] further reduced the complexity of several well-known SBox addition chain implementations, and the best-known method for fast polynomial evaluation was proposed by Carlet *et al.* in [CPRR15]. In addition, the best polynomial evaluation can be achieved by computing different addition chain [CGPZ16], which costs the same number of squares and multiplications.

However, the number of intermediate computations gets increased when implementing the SBoxes through addition chain, which may lead to more leakages. Thus, the side-channel resistance of SBox implementations may get seriously decreased. In other words, the adversary may use much fewer traces to attack the computation of certain monomials rather than SBox outputs.

2.3 Measuring for Side-channel Resistance of a Function

An SBox is a non-linear function that is widely used as a fundamental component in most block ciphers. Therefore, TO was proposed to focus on the intrinsic resiliency of SBoxes. Moreover, TO is used to evaluate the side-channel resistance of balanced functions. Note that an (n, m)-function F is said to be balanced if every element $y \in \mathbb{F}_2^m$ admits the same number 2^{n-m} of pre-images by F.

Consequently, to measure the side-channel resistance of the function itself, our work refers to the basic idea of TO. Under the assumption of the Hamming weight leakage model, TO was introduced by Prouff in 2005 [Pro05]. It quantifies the basic Differential Power Analysis (DPA) resilience from the mathematical properties of the SBox. Specifically, the basic starting point of TO research is the first-order single-bit DPA attack. A single-bit DPA attack is done by computing a differential leakage whose values are related to the selection function and to the power consumption function. The differential leakage on j-th bit, denoted by $\Delta_{K,\dot{K}}(T, j)$, can be simply expressed as:

$$\Delta_{K,\dot{K}}(T, j) = \mathbb{E}[\mathcal{L}|f_j(T_i \oplus K) = 1] - \mathbb{E}[\mathcal{L}|f_j(T_i \oplus K) = 0], \tag{5}$$

To further formulate a DPA attack on a balanced function F under the Hamming weight model, Eq.(5) can be expressed as:

$$\Delta_{K,\dot{K}}(T, j) = \frac{\sum_{i=1}^{N} f_j(T_i \oplus K) \cdot HW[F(T_i, \dot{K})]}{\sum_{i=1}^{N} f_j(T_i \oplus K)} - \frac{\sum_{i=1}^{N} [1 - f_j(T_i \oplus K)] \cdot HW[F(T_i, \dot{K})]}{\sum_{i=1}^{N} [1 - f_j(T_i \oplus K)]}. \tag{6}$$

Based on a DPA attack, TO measures the difference between the score for the correct key and the average score for the other key hypotheses with respect to a specific SBox. The

TO of the function F can be expressed as:

$$\mathrm{TO}(F) = \max_{\gamma \in \mathbb{F}_2^m} \left(|m - 2HW(\gamma)| - \frac{1}{2^{2n} - 2^n} \sum_{a \in \mathbb{F}_2^{n*}} \left| \sum_{i=1}^{m} (-1)^{\gamma_i} \mathcal{AT}_{f_i}(a) \right| \right).$$

where $\gamma \in \mathbb{F}_2^m$ denotes the register initial state which is assumed to be constant, and $\mathcal{AT}_{F_i}(a)$ denotes the *autocorrelation transform* of the function F_i with respect to a [Pro05]. Smaller TO indicates that it is more difficult for attackers to distinguish the correct key from other key hypotheses. Except for the original notion of TO, there are two variants of the transparency order so far: modified transparency order (MTO) [CSM+17] and reVisited transparency order (VTO) [LZM+20]. The two variants are based on the same idea of TO.

3 Side-channel Resistance of Addition Chains

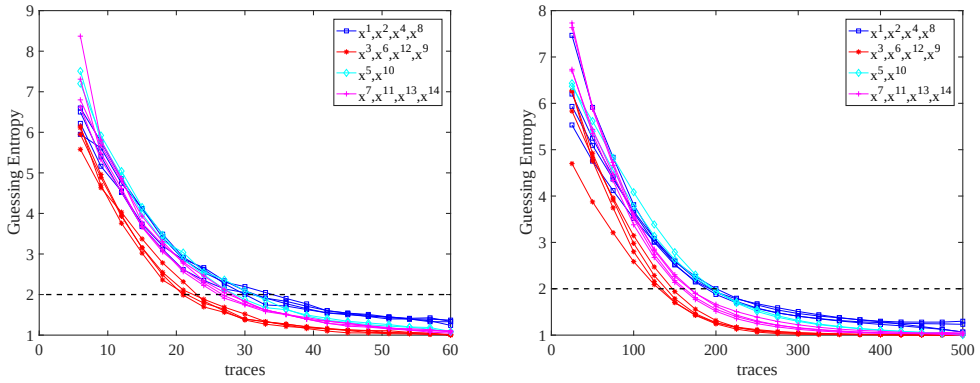
Addition chain based masked SBoxes have long been treated as secure implementations against side-channel attacks. Specifically, a sensitive variable is split into $d+1$ shares and the adversary is required to perform a $(d+1)^{th}$-order attack. However, these computations may largely differ in side-channel resistance, and attacks utilizing some computations among them might be much more efficient than utilizing leakages from others. To analyze side-channel resistance of intermediate computations, it is necessary to treat the addition chain implementation as several divided operations.

When analyzing side-channel security of a cryptographic implementation like AES, each operation can be utilized to collect the leakages, (e.g., AddRoundKey, SubBytes, ShiftRows and MixColumns). Similarly, we can utilize the leakages generated by any squares and non-linear multiplications operation. Some of them might leak much more than others, which leads to lower side-channel security. To verify our perspective, we show our attacks on all monomial computations over $\mathbb{F}_{2^4}$ in simulation (the irreducible polynomial is $x^4 + x + 1$). Specifically, the simulated attacks are done on the output of the monomial computations.

To carry out higher-order attacks, we simulate the leakages $\mathcal{L}_0(x_0), \ldots, \mathcal{L}_d(x_d)$ by $\mathcal{L}_i(x_i) = HW(x_i) + \mathcal{N}_i$, where x_i denotes the i-th share of the sensitive variable x and $\mathcal{N}_i$ denotes a Gaussian random variable centered in zero with standard deviation σ. All x_i and $\mathcal{N}_i$ are mutually independent. Each share is simulated with only 1 point of interest (PoI), and the combined leakages are obtained by normalized product [PRB09], i.e., $\mathcal{C}_d(x) = \prod_{i=0}^{d} [\mathcal{L}(x_i) - \mathbb{E}(\mathcal{L}(x_i))]$. It has been shown that the product combining is more efficient than the other combining functions [PRB09].

In these simulated scenarios, Guessing Entropy (GE) is utilized to evaluate the effectiveness of higher-order attacks. We perform higher-order attacks with different noise levels and get similar results. For the sake of brevity, the results with a low noise level are shown in Fig. 2, while the results with a high noise level are shown in Appendix A. From these results, we have several interesting findings. Side-channel resistance of different computations on the monomials over $\mathbb{F}_{2^4}$ is different, and these differences get more pronounced with increasing order d and the standard deviation σ. Moreover, the computations for powers x^{β_1} and x^{β_2} result in similar side-channel resistance if β_1 and β_2 lie in the same cyclotomic class.

Since the extra computations can actually reduce the side-channel resistance of SBox implementations, it is necessary to find a method to quantify the crucial property of these unbalanced functions.



(a) $d = 1$ (b) $d = 2$

Figure 2: The results of GE for $n = 4$ and $\sigma = 0.1$.

4 Quantifying Side-channel Resistance of a Function

The first work about side-channel resistance evaluation for a function, named TO, was derived from single-bit DPA, and then extended to multi-bit DPA. From then on, there are some other notions to quantify the side-channel resistance of a function, such as MTO and VTO. However, all these notions can only be used to evaluate side-channel resistance of balanced functions. Overall, nearly half of computations for the monomials in $\mathbb{F}_{2^n}$ are not balanced functions. Therefore, they cannot be quantified by the known notions in theory. In this section, we introduce the notion of polygon degree (PD) to quantify the resistance of a function against first-order attacks, then analyze the applicability of PD for quantifying resistance of a function against higher-order attacks.

4.1 Polygon Degree

We first explain why TO and its two variants cannot be used to quantify side-channel resistance for unbalanced functions. As shown in Eq.(6), the single-bit DPA attack works when the leakages for the target bit are different. However, if the function F is unbalanced, the Boolean function of j-th bit f_j $(1 \le j \le m)$ may also be unbalanced. Namely, the target bit may always be 0 or 1, which leads to the fact that the leakages can not be divided into two groups based on this bit. Thus, the denominator in Eq.(6) might be 0, and the following calculations for TO or its two variants become meaningless.

Actually, it is not a hard problem. In unbalanced function F, some output bits may always be 0 or 1, which are useless for distinguishing the secret key $\dot{K}$. In this case, the differential value $\Delta_{K,\dot{K}}$ equals to 0, then Eq.(6) becomes:

$$\Delta_{K,\dot{K}}(T,j) = \begin{cases} 0, & \text{if } f_j \equiv 0 \text{ or } f_j \equiv 1 \\ \dfrac{\sum_{i=1}^{N} f_j(T_i \oplus K) \cdot \mathcal{L}(T_i, \dot{K})}{\sum_{i=1}^{N} f_j(T_i \oplus K)} - \dfrac{\sum_{i=1}^{N} [1 - f_j(T_i \oplus K)] \cdot \mathcal{L}(T_i, \dot{K})}{\sum_{i=1}^{N} [1 - f_j(T_i \oplus K)]}, & \text{otherwise} \end{cases}.$$

We assume leakage in the Hamming weight model with independent additive noise [DPRS11]. Specifically, the leakages are assumed to satisfy $\mathcal{L}(T_i, \dot{K}) = HW[F(T_i \oplus \dot{K})] + \mathcal{N}$, where $\mathcal{N}$ denotes a Gaussian random variable which is centered in zero. Since $\mathbb{E}(\mathcal{L}|x) = HW(x)$,

we omit the noise then we have:

$$\Delta_{K,\dot{K}}(F,T,j)=\begin{cases}0, & \text{if } f_j\equiv 0 \text{ or } f_j\equiv 1\\ \dfrac{\sum_{i=1}^{N} f_j(T_i\oplus K)\cdot HW[F(T_i\oplus K)]}{\sum_{i=1}^{N} f_j(T_i\oplus K)}-\dfrac{\sum_{i=1}^{N}[1-f_j(T_i\oplus K)]\cdot HW[F(T_i\oplus \dot{K})]}{\sum_{i=1}^{N}[1-f_j(T_i\oplus K)]}, & \text{otherwise}\end{cases}.$$

Note that $T_i \in \mathbb{F}_2^n$, and each T_i is randomly picked by the adversary. If N is large, we can approximately set N to 2^n while the inputs of F loop through $\mathbb{F}_2^n$. Let α denote the XOR of the secret key and other key hypotheses $\dot{K}\oplus K$. Then the normalized differential leakage of the secret key and other key hypotheses on j-th bit is denoted by δ_α and defined as follows.

$$\delta_\alpha(F,j)=\begin{cases}0, & \text{if } f_j\equiv 0 \text{ or } f_j\equiv 1\\ \dfrac{\sum_{i=0}^{2^n-1} f_j(i\oplus\alpha)\cdot HW[F(i)]}{m\sum_{i=0}^{2^n-1} f_j(i\oplus\alpha)}-\dfrac{\sum_{i=0}^{2^n-1}[1-f_j(i\oplus\alpha)]\cdot HW[F(i)]}{m\sum_{i=0}^{2^n-1}[1-f_j(i\oplus\alpha)]}, & \text{otherwise}\end{cases},$$

since $0 \le HW[F(i)] \le m$, we have $0 \le \delta_\alpha(F,j) < 1$.

As for multi-bit DPA attack, the adversary utilizes single-bit DPA attack on all m bits and then combines the results. So the differential leakage of multi-bit DPA attack can be expressed as:

$$\delta_\alpha(F)=\frac{1}{m}\sum_{j=1}^{m}\delta_\alpha(F,j).$$

When we distinguish the secret key $\dot{K}$ from other key hypotheses K, we have the same basic ideas with the side-channel efficiency metric (standard score) discussed in [WO11], which is called average distinguishing score. Namely, the average distinguishing score is calculated by computing the difference between the score of the distinguisher for the good key and the average score for the wrong hypotheses, the difference being normalized with the variance of the scores. Therefore, we introduce the notion to quantify the resistance of a function as follows.

Definition 3 (Polygon Degree)**.** Let F denote a (n,m)-function, the polygon degree of F, denoted by $PD(F)$, is defined by:

$$PD(F)=\frac{1}{2^n}\sum_{\alpha\in\mathbb{F}_{2^n}}\left(|\delta_0(F)|-|\delta_\alpha(F)|\right). \tag{7}$$

The smaller the polygon degree of a function, the stronger it resists against side-channel attacks. In order to determine what a reasonable high polygon degree is, we introduce the range of the polygon degree of a function.

Theorem 1. *Let F denote a (n,m)-function, the polygon degree of F, denoted by $PD(F)$, satisfies the following relation:*

$$0 \le PD(F) < 1. \tag{8}$$

Proof. If $F(x)$ equals a constant for all $x \in \mathbb{F}_{2^n}$, we have $PD(F)$=0. For other functions F, it is obvious to get $PD(F) > 0$. Since $\delta_\alpha(F) < 1$, we can easily derive $PD(F) < 1$. □

In higher-order attacks, the normalized leakages corresponding to different shares are combined by product, namely $\mathcal{C}_d(x)=\prod_{i=0}^{d}[\mathcal{L}(x_i)-\mathbb{E}(\mathcal{L}(x_i)]$. Thanks to Lemma 1 [RPD09], the expectation of combined leakages is a linear function of $HW(x)$ when the leakages of each share follow the Hamming weight model. Thus, PD can also be applied to measure the resistance against higher-order attacks. Lemma 1 is given as follows.

Table 1: The *PD* of different cyclotomic classes for $n \in \{4, 6\}$ based on irreducible polynomials $x^4 + x + 1$ and $x^6 + x + 1$. For the sake of brevity, the complete table for $n = 8$ is shown in Appendix B.

$n = 4$			
classes	*PD*	classes	*PD*
x, x^2, x^4, x^8	0.1563	x^3, x^6, x^9, x^{12}	0.2984
x^5, x^{10}	0.1641	$x^7, x^{11}, x^{13}, x^{14}$	0.1836
$n = 6$			
classes	*PD*	classes	*PD*
$x, x^2, x^4, x^8, x^{16}, x^{32}$	0.1146	$x^{13}, x^{19}, x^{26}, x^{38}, x^{41}, x^{52}$	0.1428
$x^3, x^6, x^{12}, x^{24}, x^{33}, x^{48}$	0.1456	$x^{15}, x^{30}, x^{39}, x^{51}, x^{57}, x^{60}$	0.1482
$x^5, x^{10}, x^{17}, x^{20}, x^{34}, x^{40}$	0.1363	x^{21}, x^{42}	0.3180
$x^7, x^{14}, x^{28}, x^{35}, x^{49}, x^{56}$	0.2046	$x^{23}, x^{29}, x^{43}, x^{46}, x^{53}, x^{58}$	0.1393
x^9, x^{18}, x^{36}	0.1095	x^{27}, x^{45}, x^{54}	0.1037
$x^{11}, x^{22}, x^{25}, x^{37}, x^{44}, x^{50}$	0.1402	$x^{31}, x^{47}, x^{55}, x^{59}, x^{61}, x^{62}$	0.1395

Lemma 1. *[RPD09] Let x be in $\mathbb{F}_2^n$. If every $\mathcal{L}(x_i)$ follows the Hamming weight model, then the expectation of $\mathcal{C}_d(x)$ satisfies:*

$$\mathbb{E}[\mathcal{C}_d(x)] = (-\frac{1}{2})^d (HW(x) - \frac{n}{2}). \tag{9}$$

Consequently, *PD* can be easily extended to higher-order cases. Note that the combined noise does not need to follow a Gaussian distribution. Since *PD* is based on the difference of means of two leakage groups, it is only necessary that the expectation of combined leakage follows a linear transformation of the Hamming weight distribution, as stated in Lemma 1.

4.2 Soundness of Polygon Degree

We evaluate all monomials over $\mathbb{F}_{2^n}$ for $n \in \{4, 6, 8\}$ with the help of *PD*. We find that the powers x^{β_1} and x^{β_2} fall into a same *PD* value if β_1 and β_2 lie in a same cyclotomic class. For the sake of brevity, only the results based on irreducible polynomials $x^4 + x + 1$ and $x^6 + x + 1$ are shown in Table 1. Note that there are totally 34 cyclotomic classes for $n = 8$, so we show the *PD* values based on irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ in Table 4 in Appendix B.

To make these *PD* values easier to understand, we divide the *PD* values into 4 levels and differentiate them with different color. As shown in Table 1, the ordered *PD* values for $n = 4$ can be expressed as: $PD(C_3) < PD(C_7) < PD(C_5) < PD(C_1)$, which satisfies the attacking results as shown in Section 3.2. To further verify the soundness of *PD* for quantifying side-channel resistance of monomials over $\mathbb{F}_{2^n}$, we also launch higher-order attacks on monomials for $n = 6$ and $n = 8$. Similar to the 4-bit simulation-based experiments above, we simulate the leakages $\mathcal{L}_0(x_0), \ldots, \mathcal{L}_d(x_d)$ as $\mathcal{L}_i(x_i) = HW(x_i) + N_i$. We launch higher-order attacks on the simulated leakages with normalized combination.

As for $n = 6$, there are a total of 62 monomials over $\mathbb{F}_{2^6}$ to be simulated. To make the attack results more intuitive, we show the mean of the number of required traces for each cyclotomic class to reach a GE below 4. The results with a low noise level are shown in a histogram as Fig. 3. It can be clearly seen that the smaller the *PD* of a monomial function is, the higher is its resistance against side-channel attacks. The number of traces for the monomial function with the lowest *PD* is approximately 2–3 times than of the highest *PD*. Moreover, we also simulate our attacks with a high noise level, and the results are shown

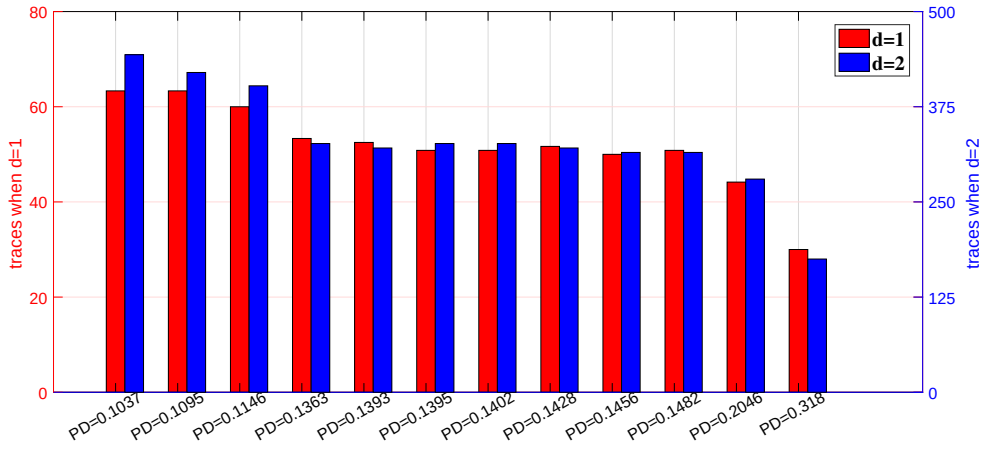


Figure 3: Number of traces for GE to be below 4 (in y-axis) versus the different PD (in x-axis) for $n = 6$ and $\sigma = 0.1$.

in Appendix A. We find that the needed traces also match the PD values well in high noisy situations. This matches with the 4-bit simulation.

As for 8-bit simulation, there are 254 monomials and a total of 34 cyclotomic classes over $\mathbb{F}_{2^8}$ to be simulated. It would be crowded if we continue to use a histogram to show the results. To demonstrate the inverse relationship between PD of the function and its side-channel resistance intuitively, we show the number of traces for the GE to be below 10 in a scatter plot. Then we use inverse functions to fit these points. The inverse function can be expressed as $Num = a/PD + b$, where Num denotes the number of traces for GE to be below 10, and a, b are function parameters.

Since there are numerous monomials over $\mathbb{F}_{2^8}$, we only simulate the attacks with a low noise level as shown in Fig. 4. We believe that the results with a high noise level would be similar to it based on the similarity of results in different noise levels over $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^6}$. It can be intuitively seen that these points are basically around the corresponding fit functions, which further verifies the property that the PD of a monomial function has an inverse relationship with its side-channel resistance. Moreover, most attacks on different cyclotomic classes require similar number of traces while their PD values are squeezed around 0.12. This finding might be helpful to avoid several weak monomial computations while designing addition chain implementations.

4.3 Information-Theoretic Evaluation on Monomial Function

Information-theoretic (IT) metrics measure the total information leakage irrespective of specific side-channel attacks, e.g., second-order CPA in Section 3. Mutual information (MI), as a well-known IT metric, has been widely used in side-channel analysis [SMY09, CS19]. Therefore, we use MI to evaluate the side-channel resistance of different monomials from an IT viewpoint.

As illustrated in [SVO+10], the multivariate joint distribution is the most effective way to utilize the information leakage in masked implementations. Let $\mathcal{L} = (\mathcal{L}_0, \ldots, \mathcal{L}_d)$ be the multivariate leakage where $\mathcal{L}_i$ are defined as in Section 4.2, then mutual information between the sensitive variable X and the leakage is denoted as $I(\mathcal{L}; X)$. Considering $d = 1$ in the first-order Boolean masking, the MI results of monomials over $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^6}$ are depicted in Fig. 5(a) and 5(b), respectively.

The first observation from Fig. 5 is that, as shown in Table. 1, all monomials can be

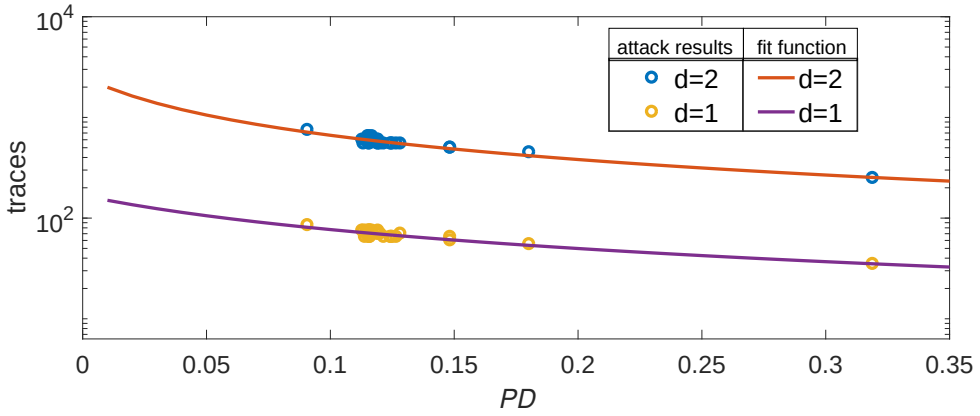


Figure 4: Number of traces for the GE to be below 10 (in y-axis) versus the different *PD* (in x-axis) for $n = 8$ and $\sigma = 0.1$.

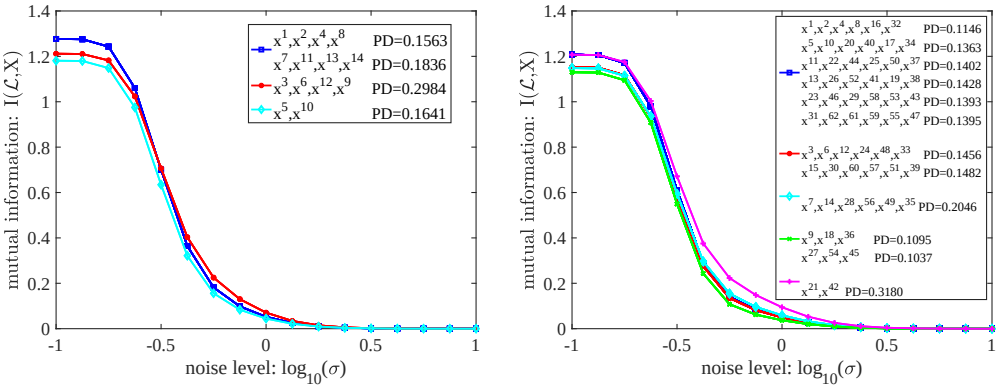
grouped into few classes in the sense of information leakage. In particular, there are only three classes in $\mathbb{F}_{2^4}$ and five classes in $\mathbb{F}_{2^6}$ for all monomials, and monomials with the same range of outputs are classified into the same classes when using MI metric. The monomials in each group can be viewed as equivalent where the amount of monomials' information leakage (during computation) is identical.

However, the results of MI metric do not exactly match the attack-based results by second-order CPA as shown in Fig. 2(a) and Fig. 3. For example, two monomial x^1 and x^7 leak identically under MI metric for $n = 4$, but x^1 is one of the strongest monomials against second-order CPA and x^7 belongs to the second weakest class. Moreover, the classification by MI metric might be contrary to the one by attack-based results. For example, x^1 is one of the weakest monomials by MI metric when $n = 6$ and $\sigma = 0.1$, but it belongs to the third strongest class under simulated attacks. Summing up, the classification by MI metric does not always match the attack-based results.

In summary, information-theoretic analysis allows us to understand how a sensitive variable leaks during manipulating. However, not all leakages can be exploited by side-channel distinguishers. We focus on CPA as a distinguisher in this paper since it has been proved to be optimal under the Hamming weight model [DPRS11]. In this respect, our *PD* can be utilized as a proper indicator for evaluating monomials, which matches with attack-based results. Therefore, we use *PD* values to evaluate side-channel resistance of addition chain implementations. Besides, we find that using other distinguishers could get results closer to IT analysis. For instance, preliminary results by mutual information analysis (MIA) in a noiseless scenario are in accordance with IT analysis, which is not surprising. However, MIA can be inefficient compared to CPA and faces several challenges in practice, e.g., estimation of statistical distributions of leakages. We leave further investigation as a topic for future work.

5 Experiments on Masked SBox Implementations

In this section, we list two instantiated adversaries with different computational resources, and demonstrate how to evaluate side-channel resistance of addition chain implementations using *PD*. We verify our analysis by practical experiments on selected masked AES SBox implementations. In consideration of different noise levels, we collect power and electromagnetic traces respectively.



(a) $n = 4$ (b) $n = 6$

Figure 5: Mutual information of monomial functions for $d = 1$.

5.1 Application of Polygon Degree in Addition Chain

Before moving to practical experiments, we first specify a few representative (more or less powerful) attacks against addition chain implementation. Two instantiated adversaries are described as:

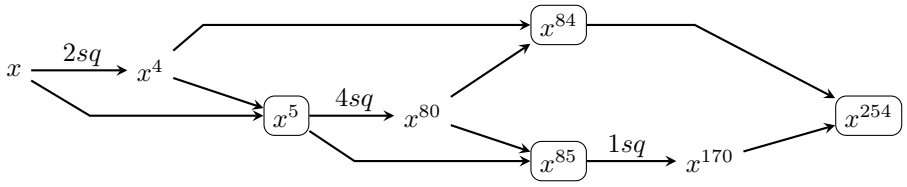
- $\mathcal{A}_1$ has limited computational resources, so he is only able to find leakages corresponding to one sensitive intermediate. Then he processes these leakages by normalized product combining, and launch higher-order SCAs.
- $\mathcal{A}_2$ has enough computational resources to find all existing leakages. So he is able to launch higher-order attacks on all sensitive intermediates, then sums all results together to achieve a higher success rate.

Considering the "bucket principle", side-channel resistance of a sequence of computations is determined by the worst one for $\mathcal{A}_1$. For example, if the computation of the monomial x^{15} has the lowest side-channel resistance in the addition chain [RP10], then the resistance of this implementation will not be better than the resistance of x^{15}. With the notion PD, we are able to quantify the side-channel resistance of each monomial function, and the security of whole processing for computing is determined by the worst one. Thus side-channel resistance of a whole processing for $\mathcal{A}_1$ can be expressed as $\max\{PD(F_1), PD(F_2) \ldots, PD(F_w)\}$, where F_i denotes the i-th intermediate function in whole processing $\mathcal{F}$.

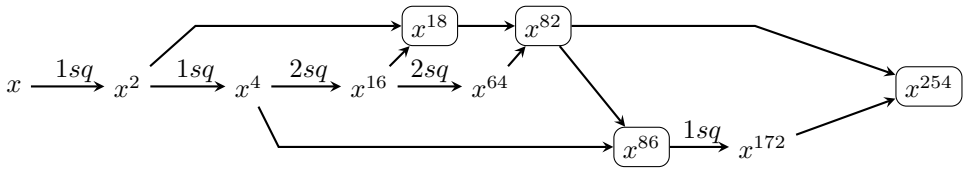
As for $\mathcal{A}_2$, the adversary can sum all results of higher-order SCAs on all sensitive intermediates. According to Eq.(7), the PD value is achieved by calculating all differences between the score of the distinguisher for right key and the other key hypotheses. While summing the results of attacks on different function F_1 and F_2, it can be equivalent to sum all the score of the distinguisher for each key hypotheses. Then the difference between right key and a wrong hypotheses α can be directly added, which is expressed as $(|\delta_0(F_1)| + |\delta_0(F_2)|) - (|\delta_a(F_1)| + |\delta_\alpha(F_2)|)$. After adding differences between right key and all other key hypotheses, it becomes $PD(F_1) + PD(F_2)$. Thus, for $\mathcal{A}_2$, side-channel resistance of a sequence of computations $\mathcal{F}$ can be expressed as $\sum_{i=1}^{w} PD(F_i)$.

5.2 Practical Results

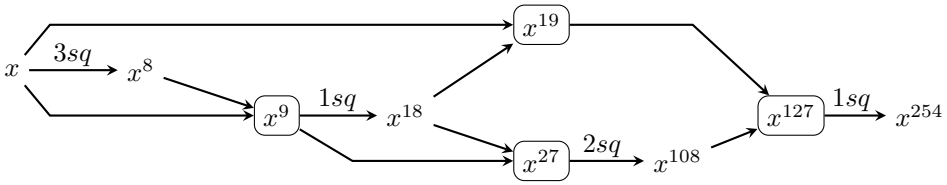
Since the polynomial function of SBox in AES is relatively simple, we show our practical results on AES addition chain based SBox implementations. [CGP+12] has proved that at least 7 squares and 4 multiplications are needed for an addition chain based AES SBox. So w equals to 12 for addition chains with the highest efficiency from x to x^{254}.



(a) One of the weakest addition chains $\mathcal{F}_c$ for x^{254} against both $\mathcal{A}_1$ and $\mathcal{A}_2$



(b) One of the strongest addition chains $\mathcal{F}_b$ for x^{254} against both $\mathcal{A}_1$ and $\mathcal{A}_2$



(c) One recommended computation $\mathcal{F}_c$ for x^{254} ($x^{18} \cdot x$ and $x^{18} \cdot x^9$ can be proceeded in parallel [CGPZ16])

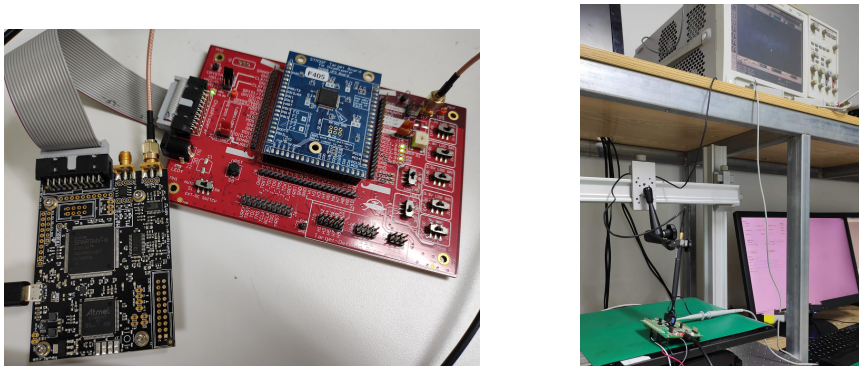
Figure 6: Three typical addition chains, and they are with the highest efficiency for AES S-box implementation (4 multiplications and 7 squares).

We find all feasible and the most efficient addition chains by a brute-force algorithm, which can be described as follows.

1. Init: Put all elements in cyclotomic class C_1 to the candidate set $Pool$.
2. Take two elements a_1, a_2 in $Pool$, then find $a_1 + a_2$ from all cyclotomic class C, denoted C_{found}. Put all elements in C_{found} to $Pool$.
3. Repeat step 2 for three times, then we will have $a_1, \ldots, a_8$. If $a_7 + a_8$ can be found in cyclotomic class C_{127}, make a_9 equals to 254.
4. Count the squares number in the addition chain $a_1, \ldots, a_9$. If only 7 squares are needed, save this chain.
5. Go through each selection for a_{2i-1} and a_{2i} in order, and save the qualified addition chains. Then we will get all feasible addition chains.

In sum, we get 1,330 addition chains with the highest efficiency for the AES SBox. Then we measure their resistance using the PD value against $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. For $\mathcal{A}_1$, we get 90 addition chains with the weakest resistance and 180 addition chains with the strongest resistance. As for $\mathcal{A}_2$, we get 5 weakest and 3 strongest addition chains. The addition chain in Fig. 6(a) is the weakest addition chain against both $\mathcal{A}_1$ and $\mathcal{A}_2$, and that in Fig. 6(b) is the strongest one against both two adversaries. Meanwhile, we recommend the computation as shown in Fig. 6(c). It is also the strongest one against $\mathcal{A}_1$ and can be accelerated by parallel computation via common shares [CGPZ16]. These addition chains are denoted by $\mathcal{F}_a$, $\mathcal{F}_b$ and $\mathcal{F}_c$ respectively, and the addition chain proposed in [RP10] is denoted by $\mathcal{F}_{rp}$[1]. While measuring side-channel resistance using $\max[PD(F)]$ against $\mathcal{A}_1$, the monomials in cyclotomic classes C_{15}, C_{85}, C_9 and C_9 are with the weakest resistance

[1]The cycle counts for the computation of $\mathcal{F}_{rp}$, $\mathcal{F}_a$ and $\mathcal{F}_c$ are all the same, which are 1,911 and 3,503 for $d = 1$ and $d = 2$. However, the cycle counts of the computation for the strongest addition chain $\mathcal{F}_b$ is slightly higher than the others (1,972 and 3,584 for $d = 1$ and $d = 2$), because it requires to store the intermediate shares of x^2, x^4 and x^{16} during several squares from x to x^{64}.



(a) Power traces (b) Electromagnetic traces

Figure 7: Our experimental environment for collecting power and electro-magnetic leakages.

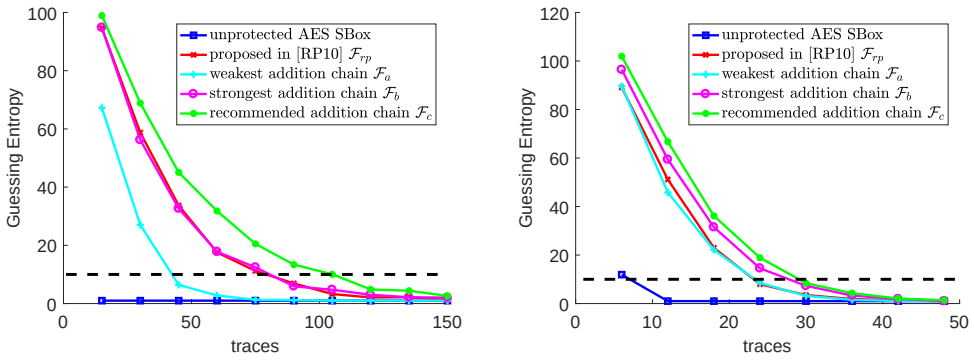
from $\mathcal{F}_{rp}$, $\mathcal{F}_a$, $\mathcal{F}_b$ and $\mathcal{F}_c$ respectively. To measure the side-channel resistance against $\mathcal{A}_2$, the metric $\sum[PD(F)]$ is used as shown in Table. 2.

Table 2: The measurement of side-channel resistance of four addition chains using PD.

Addition Chain	$\mathcal{F}_a$	$\mathcal{F}_{rp}$	$\mathcal{F}_b$	$\mathcal{F}_c$
$\max[PD(F)]$	0.3190	0.1408	0.1202	0.1202
$\sum[PD(F)]$	1.0074	0.9229	0.6589	0.7369

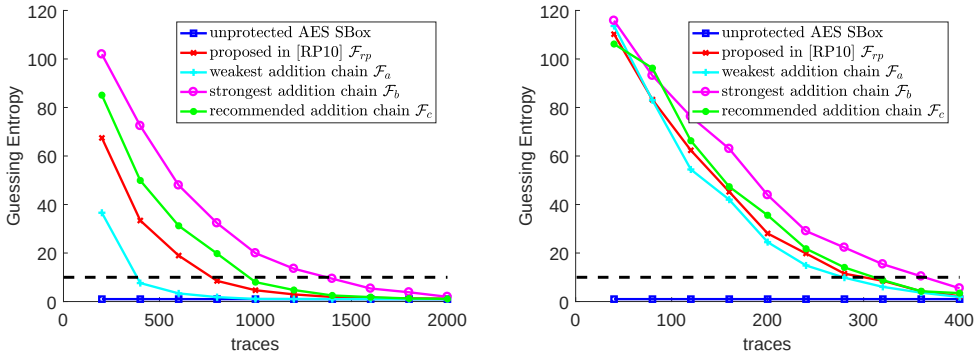
Power analysis. Our measurement setup for power analysis is shown in Fig. 7(a). It consists of the ChipWhisperer-Lite board, the CW308 UFO board and CW308T-STM32F4 target board. The target board contains a 32-bit ARM Cortex-M4 CPU with an STM32F405 device. It is a relatively ideal environment with low noise for power analysis, since the highest Signal Noise Ratio (SNR) is close to 100. We implement each masked addition chain based AES SBox for $d = 1$ and $d = 2$ in line with the public higher-order masked implementation by Coron [CRZ18]. A total of 3,000 traces and 10,000 traces of each addition chain are recorded for $d = 1$ and $d = 2$ respectively, and 24,400 points are used for attacks. Then we mount higher-order CPA on the leakages by simulating the two instantiated adversaries. To simulate $\mathcal{A}_1$, we attack each addition chain implementation by utilizing the leakages from the weakest monomial computation respectively, and compute the guessing entropy to evaluate the effectiveness. The higher-order leakages are obtained by combining leakages from each share with the normalized product. We simulate $\mathcal{A}_2$ by combining attack results on all sensitive intermediates. Moreover, we perform a first-order CPA on an unprotected look-up table AES implementation as a reference. The results for $d = 1$ and $d = 2$ are shown as Fig. 8 and Fig. 9 respectively.

From these results, we can see that these addition chain implementations can be broken with a small amount of traces. The side-channel resistances of the strongest addition chain $\mathcal{F}_b$ and recommended one $\mathcal{F}_c$ are always better than others, which is in line with their PD values. As for $\mathcal{A}_2$, the side-channel resistances of these additional chains seem to be very close. More importantly as shown in Fig. 8, the results of second-order CPA on four addition chain based masked implementations are close to those of CPA on an unprotected implementation. Besides, we see in Fig. 8 that $\mathcal{F}_c$ provides better security than $\mathcal{F}_b$, which is not in line with the results shown in Table 2. The reason is that the noise level of combined leakages also affects their side-channel resistance. Specifically, for the strongest and recommended addition chain implementations in our environment, Pearson correlation coefficients between the Hamming weight of the intermediate and combined leakages are 0.3247 and 0.3045 respectively. Thus, $\mathcal{F}_c$ shows better side-channel resistance than $\mathcal{F}_b$ in



(a) Guessing entropy for $\mathcal{A}_1$ (b) Guessing entropy for $\mathcal{A}_2$

Figure 8: The results of second-order CPA on four different first-order masked addition chain implementations.



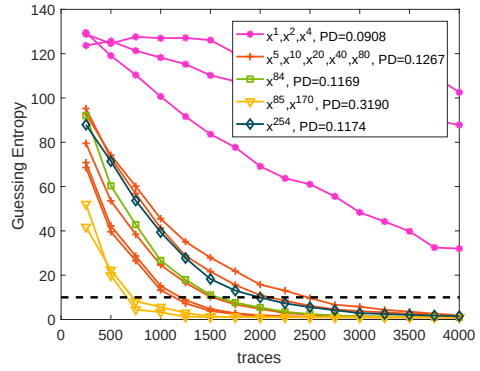
(a) Guessing entropy for $\mathcal{A}_1$ (b) Guessing entropy for $\mathcal{A}_2$

Figure 9: The results of third-order CPA on four different second-order masked addition chain implementations.

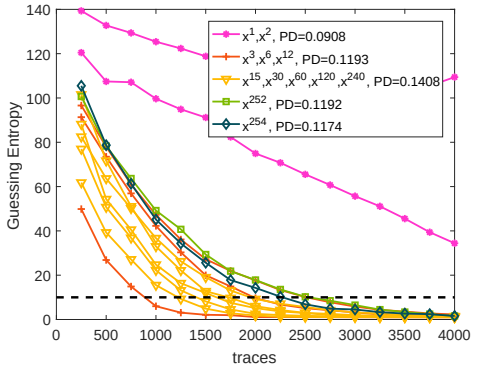
Fig. 8.

Electromagnetic analysis. Our experimental environment for electromagnetic analysis is shown in Fig. 7(b). The addition chain based masked implementations are running on an STM32F407 that is also a Cortex-M4 based micro-controller. Its electromagnetic consumption is measured through an electromagnetic near field probe RS H 400-1 on the surface of the micro-controller. Furthermore, each masked addition chain based AES SBox is implemented in line with the public one by Coron [CRZ18]. The traces are obtained through an Agilent DSO90404A Digital Storage Oscilloscope with a high impedance adapter, and the sampling rate is set to 1GHz. A total of 40,000 traces of each addition chain implementation is recorded and 25,000 points around the SBox implementation are used for the attacks. The collected electromagnetic traces are with a higher noise, since the highest SNR of PoIs is lower than 2. Considering the enormous costs for performing higher-order attacks, we do not collect the electromagnetic traces for $d \geq 2$.

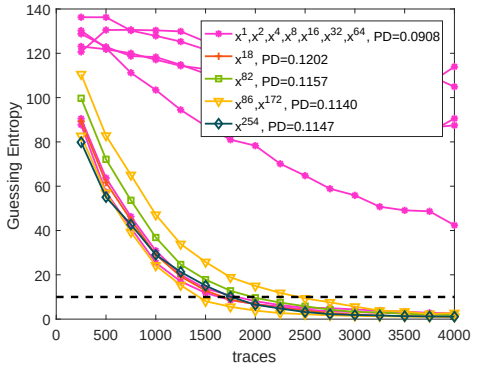
We perform second-order CEMA on the electromagnetic leakages by simulating the two instantiated adversaries. An important finding is that the SNR of our electromagnetic measurements is not only much less than that of our power measurements, but also widely varies for different monomial computations. Therefore, the monomial computation with maximum PD may not be the weakest against practical attacks. To simulate $\mathcal{A}_1$, we



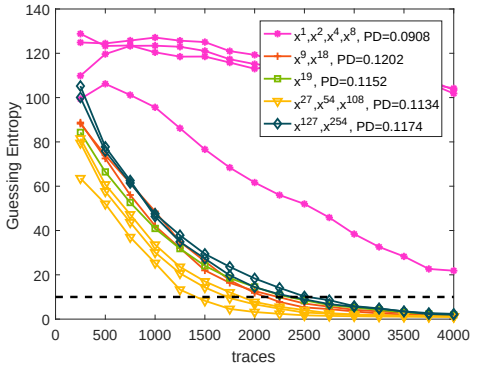
(a) The addition chain $\mathcal{F}_a$ with the weakest resistance



(b) The addition chain proposed in [RP10] $\mathcal{F}_{rp}$



(c) The addition chain $\mathcal{F}_b$ with the strongest resistance



(d) The addition chain $\mathcal{F}_c$ with strong resistance and parallel feasibility

Figure 10: The results of second-order CEMA toward each monomial computation in four different masked addition chain implementations.

attack addition chain implementations by utilizing the leakages from each monomial correspondingly. The second-order leakages are again obtained by combining leakages with the normalized product. The results are shown in Fig. 10. It can be seen that comparing the most efficient attacks on four addition chains, the practical resistance of them is consistent with our theoretical analysis. Namely, the addition chain $\mathcal{F}_a$ is the weakest while $\mathcal{F}_b$ and $\mathcal{F}_c$ are the strongest. However, the difference of attack results aiming at monomial computations is much larger than expected from our theoretical results. The reason is that all intermediate computations are assumed to leak information under the same noise level in our simulation, which is not the case in our practical experiments. For instance, in the weakest addition chain based implementation the SNR corresponding to a share of x^2 and x^{85} are 0.19 and 1.42 respectively. This difference might be related to the architecture of the target platform.

We simulate $\mathcal{A}_2$ by composing attack results on all sensitive intermediates, and the results are shown as Fig. 11. Moreover, we perform a first-order CEMA on an unprotected look-up table SBox implementation running on the same setup. We can see that the worst results of second-order CEMA have been very close to the results of first-order CEMA on the unprotected implementation. It means that if the addition chain is chosen without care, the protection provided by masking might be nullified. Note that attacks on the other three addition chain become less efficient for the adversary $\mathcal{A}_2$. The reason is that

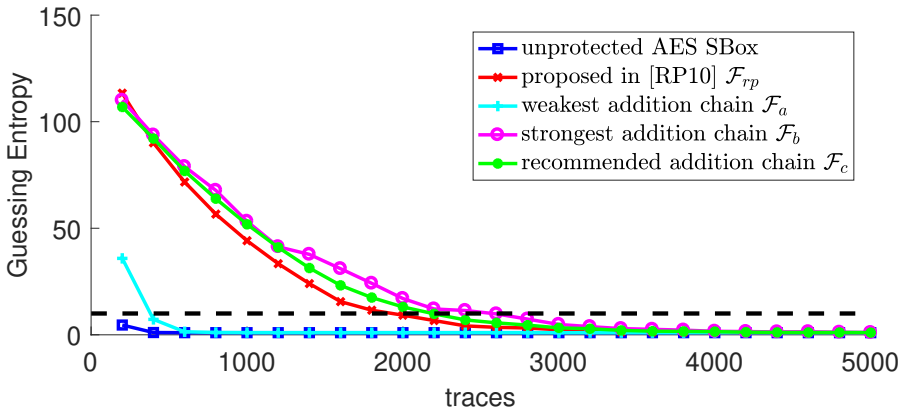


Figure 11: The combined results of second-order CEMA on four typical masked addition chain implementations.

Table 3: Comparison of the number of required traces for electromagnetic analysis to reach a GE lower than 10 for different addition chain implementations.

Adversary	Unprotected	Addition Chain			
		$\mathcal{F}_a$	$\mathcal{F}_{rp}$	$\mathcal{F}_b$	$\mathcal{F}_c$
$\mathcal{A}_1$	200	750	1,000	1,500	1,500
$\mathcal{A}_2$	200	300	2,000	2,600	2,200

the inefficient results on some monomials are again combined and negatively affect the total attack result. If $\mathcal{A}_2$ is able to filter out these inefficient results, the composing attack might perform better. However, this requires specific knowledge about the target platform. Our main results for electromagnetic analysis are summarized in Table 3.

6 Evaluation on Profiled Attacks

To further verify the practical soundness of the *PD*, we evaluate the addition chain implementations using profiled attacks.

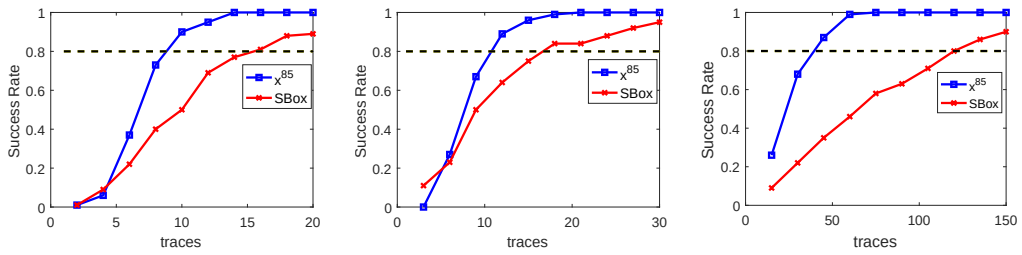
6.1 Template Attack

Template Attack (TA) is the first profiled attack [CRR02]. In 2013, efficient template attack (ETA) [CK13] was proposed as an improved variant of TA. In this paper, ETA is adapted to evaluate the addition chain implementations.

We first perform ETA on simulated second-order leakages. The leakages from computations of x^{85} and the SBox are simulated using the Hamming weight model, i.e., $\mathcal{L}(x_i) = HW(x_i) + \mathcal{N}_i$. Then we simulate 20 PoIs for each share. The leakages vector, including 20 PoIs for i-th share of j-th trace, can be denoted by $\boldsymbol{\mathcal{L}_i^j}$. We profile 256 efficient templates for each share using 15,000 traces.

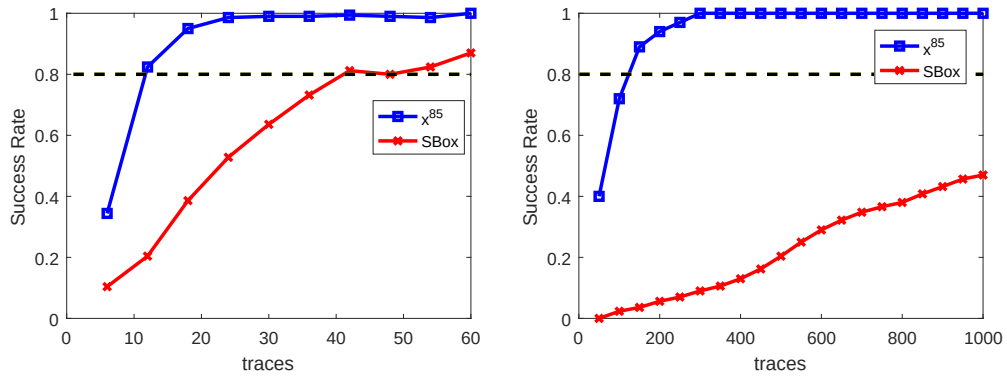
As for the attacking step, 5,000 traces are utilized for evaluation, and the success rate is used to evaluate the effectiveness of attacks. In the attack phase, we match the leakages to profiled templates, which are denoted as $\boldsymbol{M_i}$ and $i \in \{0, 1\}$. Then we get the probability $P(X_i^j = x_i^j | \boldsymbol{\mathcal{L}_i^j}, \boldsymbol{M_i})$ for each trace utilizing efficient templates. The probability for x^j can be expressed as

$$P(x^j | \boldsymbol{\mathcal{L}^j}, \boldsymbol{M}) = \sum_{\mathcal{S}} \prod_{i=0}^{d} P(x_i^j | \boldsymbol{\mathcal{L}_i^j}, \boldsymbol{M_i}),$$



(a) $\sigma = 0.1$ (b) $\sigma = 1$ (c) $\sigma = 2$

Figure 12: The success rate for ETA on simulated protected leakages with different noise levels.



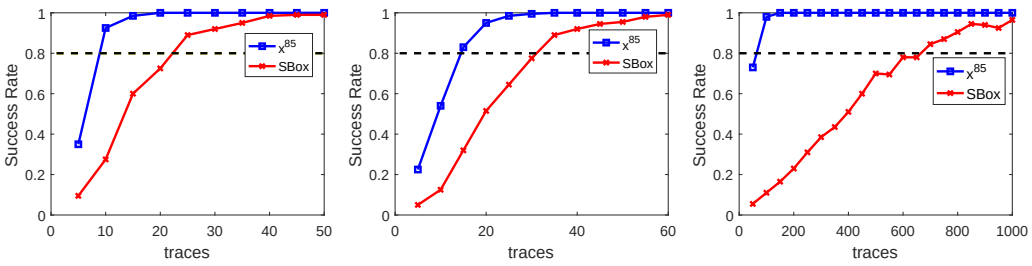
(a) Power leakages (b) Electromagnetic leakages

Figure 13: The success rate for ETA on practical leakages.

where $\mathcal{S}$ is the set $\{(x_0^j, \ldots, x_d^j) | x^j = x_0^j \oplus \ldots x_d^j\}$, and $\mathcal{L}^j$ denotes the leakages of all shares on j^{th} trace. With the help of the inverse mapping and plaintext T, $P(x^j)$ can be mapped to $P^j(k)$. We add up the $P^j(k)$ of each trace, and select the key hypothesis corresponding to $max[P(k)]$.

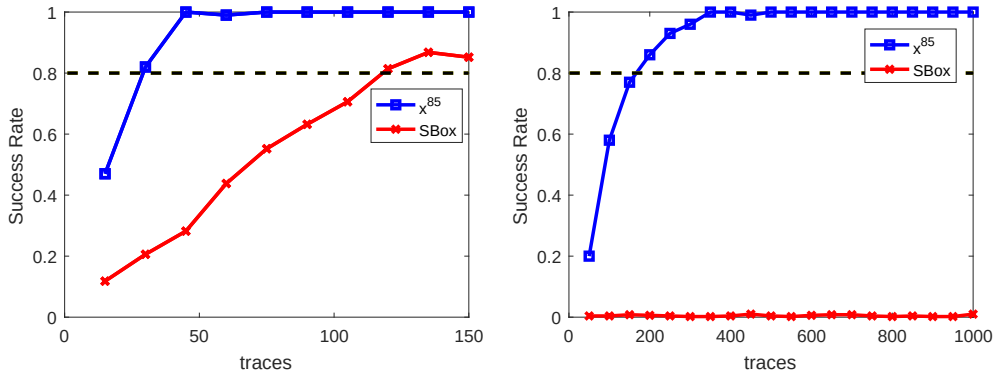
Simulated results with different noise levels are shown in Fig. 12. We can see that with increasing noise, it is counter-intuitive that targeting x^{85} becomes more and more efficient. In [DFS19], Duc *et al.* have shown that the fewer outputs a function has, the less information is leaked. It is invalid when the function is unbalanced. Specifically in x^{85}, 256 inputs are mapped to only four outputs. Thus, there are only four templates that need to be matched for x_1 while x_0 has been certain. So this match is more likely to be correct, which leads to higher success rate. The time required to attack changes as well. Aiming at x^{85}, the attack is faster than aiming at SBox, since fewer templates are needed to be matched due to the smaller output size.

We also launch ETA on practical first-order masked addition chain based SBox implementation as shown in Fig. 13. The experimental setup is the same as mentioned before. As for power analysis, 2,500 traces are used and 20 PoIs are selected to build the templates for each share in the profiling step. As for electromagnetic analysis, 37,000 traces are used and 20 PoIs are selected to build the templates for each share in the profiling step. It is obvious that the attack targeting x^{85} becomes more efficient than targeting the SBox output in both power and electromagnetic analysis.



(a) $\sigma = 0.1$ (b) $\sigma = 1$ (c) $\sigma = 2$

Figure 14: The success rate of deep learning based profiling attacks on simulated leakages with different noise levels.



(a) Power leakages (b) Electromagnetic leakages

Figure 15: The success rate of deep learning based profiling attacks on practical leakages.

6.2 Deep Learning based Profiled Attack

Recently, deep learning techniques gained substantial interest in the community of side-channel analysis. Previous results have shown that deep learning based attacks are a very efficient alternative to the state-of-the-art profiled attacks, and even outperform traditional profiled attacks in certain cases [CDP17]. We explore whether *PD* is still reasonable when measuring the resistance against such attacks. Given the robustness of convolutional neural networks (CNNs) to most common countermeasures [ZBHV20, CDP17], we analyze the performance of using CNNs to attack addition chain implementations.

CNN Architecture. We refer to recent work [ZBHV20] for designing a CNN model. The CNN is composed of two convolution blocks of 16 filters of size 3 followed by three fully-connected layers. Each layer is activated by SeLU function and He Uniform initialization is used to improve the weight initialization. In a convolution block, the outputs of a convolution layer are fed into the BatchNormalization layer and AveragePooling layer. The last fully-connected layer contains 4 or 256 neurons (corresponding to the network attacking x^{85} and SBox respectively) activated by softmax function. The cross-entropy is used as loss function. In order to facilitate the comparison of their resistance, the convolutional network remains unchanged except for the last fully-connected layer. As a remark, the network architectures used in this subsection are surely not optimal, as our goal is not to select the optimal parameters, but to compare different addition chains.

Experimental Setup. We perform simulated and practical attacks against x^{85} and SBox, respectively. For simulated experiments, we simulate 50,000 traces labeled by the value of the SubByte output (4 labels for x^{85} and 256 labels for SBox) with different noise

levels ($\sigma = 0.1$, $\sigma = 1$, and $\sigma = 2$). The data sets of power and electromagnetic traces used are the same as that of the template attack. Each trace consists of 40 PoIs. We fix the training set size to 38,250, validation set size to 6,750 and attack set size to 5,000. As for practical power analysis, there are 2,125 traces for training, 375 traces for validation and 500 traces for attack. As for electromagnetic analysis, there are 28,900 traces for training, 5,100 traces for validation and 6,000 traces for attack. A mini batch of 50 is employed. The learning rate is initially 0.005, and a technique called One Cycle Policy [Smi17] is used to choose the right learning rate. We set 100 epochs for the training in simulated experiments and 200 epochs in practical experiments. During the training, the network kernel weights are recorded for the best validation loss. Once the training is done, we reconstruct the neuron network with the best recorded weights. All experiments are conducted on an Intel(R) Xeon(R) CPU E5-2667 v4 @3.20GHz 32 core machine with two NVIDIA TITAN Xp GPUs. We use the Keras library (version 2.2.2) with the TensorFlow library (version 1.10.0) as the backend for CNN.

Experimental Results. The success rate is used to evaluate the effectiveness of attacks. We run each attack 100 times with randomly selected sub-samples of attack sets to find the average number of traces to achieve a success rate higher than 80%. The results of simulated and practical attacks are shown in Fig. 14 and Fig. 15, respectively. It can be observed that the results are basically consistent with those of template attacks. However, for the practical attack against SBox, the correct key cannot be successfully retrieved. We argue the main reason is that the classification problem is too complicated (256-classification) for the relatively simple CNN network. In addition, the training data might be insufficient.

7 Conclusion

Addition chain implementations enables more efficient and practical masked SBox designs, but the increased computations for intermediate monomials may cause more leakages related to the sensitive variables. These leakages have not been studied from the perspective of the induced functions, and our work fills this gap. In this paper, we introduce the notion of *PD* to quantify the information leakages caused by each monomial computation. Theoretically, *PD* is independent of the masking order for Boolean masking under higher-order CPA. Thus, we believe that it can also be used in some other SBox implementations, such as tower field based implementations. Regarding other masking schemes, we find that *PD* has a similar impact on the inner product masking (IPM) scheme. Specifically, monomials with lower *PD* values are also more resistant against higher-order CPA in our simulation. However, we have not formally proved that *PD* can be directly used as the predictor for IPM or other masking schemes. With the help of *PD*, we present how to quantify side-channel resistance of the whole processing step consisting of several intermediate computations. Eventually, we apply our method to the AES SBox, and study the practical side-channel resistance by both non-profiled attacks and profiled attacks. The results show that attacking intermediate results in the processing of an addition chain can be more efficient than targeting the SBox output.

Acknowledgements

This work is supported in part by National Natural Science Foundation of China (No.61632020, No.U1936209 and No.62002353) and Beijing Natural Science Foundation (No.4192067).

References

[CCDP04] Vincent Carlier, Hervé Chabanne, Emmanuelle Dottax, and Hervé Pelletier. Electromagnetic side channels of an FPGA implementation of AES. *IACR Cryptol. ePrint Arch.*, 2004:145, 2004.

[CDP17] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[CGC+21] Wei Cheng, Sylvain Guilley, Claude Carlet, Sihem Mesnager, and Jean-Luc Danger. Optimizing inner product masking scheme by a coding theory approach. *IEEE Trans. Inf. Forensics Secur.*, 16:220–235, 2021.

[CGP+12] Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-order masking schemes for s-boxes. In *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, pages 366–384, 2012.

[CGPZ16] Jean-Sébastien Coron, Aurélien Greuet, Emmanuel Prouff, and Rina Zeitoun. Faster evaluation of sboxes via common shares. In *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, pages 498–514, 2016.

[CK13] Omar Choudary and Markus G. Kuhn. Efficient template attacks. In *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, pages 253–270, 2013.

[Cor14] Jean-Sébastien Coron. Higher order masking of look-up tables. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 441–458, 2014.

[CPRR15] Claude Carlet, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Algebraic decomposition for probing security. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 742–763, 2015.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 13–28, 2002.

[CRV15] Jean-Sébastien Coron, Arnab Roy, and Srinivas Vivek. Fast evaluation of polynomials over binary finite fields and application to side-channel countermeasures. *J. Cryptogr. Eng.*, 5(2):73–83, 2015.

[CRZ18] Jean-Sébastien Coron, Franck Rondepierre, and Rina Zeitoun. High order masking of look-up tables with common shares. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):40–72, 2018.

[CS19] Gaëtan Cassiers and François-Xavier Standaert. Towards globally optimized masking: From low randomness to low noise rate or probe isolating multiplications with reduced randomness and security against horizontal attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):162–198, 2019.

[CSM+17] Kaushik Chakraborty, Sumanta Sarkar, Subhamoy Maitra, Bodhisatwa Mazumdar, Debdeep Mukhopadhyay, and Emmanuel Prouff. Redefining the transparency order. *Des. Codes Cryptogr.*, 82(1-2):95–115, 2017.

[DDF19] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. *J. Cryptol.*, 32(1):151–177, 2019.

[DFS19] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version. *J. Cryptol.*, 32(4):1263–1297, 2019.

[DPRS11] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *J. Cryptogr. Eng.*, 1(2):123–144, 2011.

[FLD12] Yunsi Fei, Qiasi Luo, and A. Adam Ding. A statistical model for DPA with novel algorithmic confusion analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, pages 233–250, 2012.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 463–481, 2003.

[Knu97] Donald Ervin Knuth. *The art of computer programming, Volume I: Fundamental Algorithms, 3rd Edition*. Addison-Wesley, 1997.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 104–113, 1996.

[LZM+20] Huizhong Li, Yongbin Zhou, Jingdian Ming, Guang Yang, and Chengbin Jin. The notion of transparency order, revisited. *Comput. J.*, 63(12):1915–1938, 2020.

[MQ18] Axel Mathieu-Mahias and Michaël Quisquater. Mixing additive and multiplicative masking for probing secure polynomial evaluation methods. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):175–208, 2018.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 278–296, 2004.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 142–159, 2013.

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[Pro05] Emmanuel Prouff. DPA attacks and s-boxes. In *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, pages 424–441, 2005.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, pages 413–427, 2010.

[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-order masking and shuffling for software implementations of block ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, pages 171–188, 2009.

[RV13] Arnab Roy and Srinivas Vivek. Analysis and improvement of the generic higher-order masking scheme of FSE 2012. In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, pages 417–434, 2013.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Smi17] Leslie N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Santa Rosa, CA, USA, March 24-31, 2017*, pages 464–472. IEEE Computer Society, 2017.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

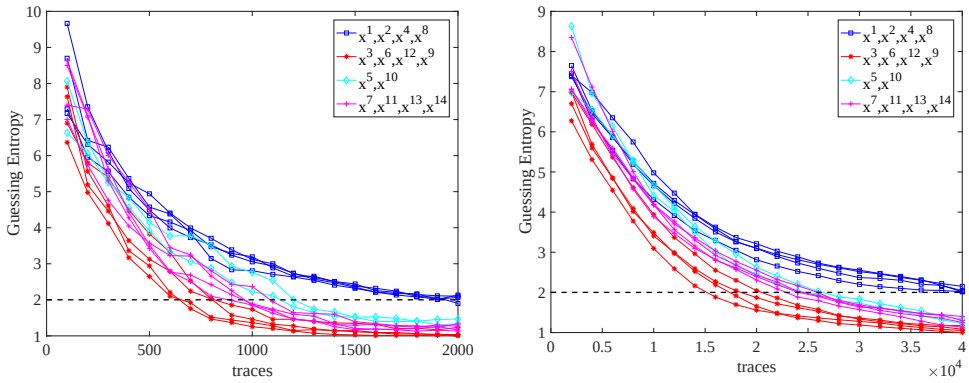
[SVO+10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

[WO11] Carolyn Whitnall and Elisabeth Oswald. A comprehensive evaluation of mutual information analysis using a fair evaluation framework. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 316–334, 2011.

[ZBHV20] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Methodology for efficient CNN architectures in profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):1–36, 2020.

A Simulations over $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^6}$ with High Noise Level

The simulated results on monomial functions when $n = 4$ and $\sigma = 2$ are shown as Fig. 16. And the results when $n = 6$ and $\sigma = 2$ is shown as Fig. 17.



(a) $d = 1$ (b) $d = 2$

Figure 16: The results of GE for $n = 4$ and $\sigma = 2$.

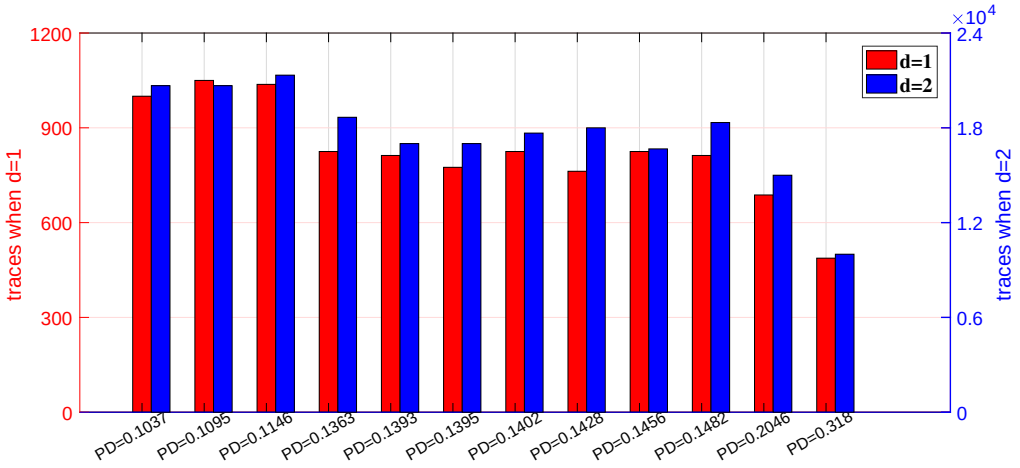


Figure 17: Number of traces for GE to be below 4 (in y-axis) versus the different PD (in x-axis) for $n = 6$ and $\sigma = 2$.

B *PD* Values for $n = 8$

The PD values for $n = 8$ based on the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ are shown as Table. 4.

Table 4: The PD values of different classes for $n = 8$ based on $x^8 + x^4 + x^3 + x + 1$

$n = 8$	
classes	PD
$x, x^2, x^4, x^8, x^{16}, x^{32}, x^{64}, x^{128}$	0.0908
$x^3, x^6, x^{12}, x^{24}, x^{48}, x^{96}, x^{129}, x^{192}$	0.1193
$x^5, x^{10}, x^{20}, x^{40}, x^{65}, x^{80}, x^{130}, x^{160}$	0.1267
$x^7, x^{14}, x^{28}, x^{56}, x^{112}, x^{131}, x^{193}, x^{224}$	0.1159
$x^9, x^{18}, x^{33}, x^{66}, x^{36}, x^{72}, x^{132}, x^{144}$	0.1202
$x^{11}, x^{22}, x^{44}, x^{88}, x^{97}, x^{133}, x^{176}, x^{194}$	0.1157
$x^{13}, x^{26}, x^{52}, x^{67}, x^{104}, x^{134}, x^{161}, x^{208}$	0.1162
$x^{15}, x^{30}, x^{60}, x^{120}, x^{135}, x^{195}, x^{225}, x^{240}$	0.1408
$x^{17}, x^{34}, x^{68}, x^{136}$	0.1216
$x^{19}, x^{38}, x^{49}, x^{76}, x^{98}, x^{137}, x^{152}, x^{196}$	0.1152
$x^{21}, x^{42}, x^{69}, x^{81}, x^{84}, x^{138}, x^{162}, x^{168}$	0.1169
$x^{23}, x^{46}, x^{92}, x^{113}, x^{139}, x^{184}, x^{197}, x^{226}$	0.1155
$x^{25}, x^{35}, x^{50}, x^{70}, x^{100}, x^{140}, x^{145}, x^{200}$	0.1247
$x^{27}, x^{54}, x^{99}, x^{108}, x^{141}, x^{177}, x^{198}, x^{216}$	0.1134
$x^{29}, x^{58}, x^{71}, x^{116}, x^{142}, x^{163}, x^{209}, x^{232}$	0.1147
$x^{31}, x^{62}, x^{124}, x^{143}, x^{199}, x^{227}, x^{241}, x^{248}$	0.1161
$x^{37}, x^{74}, x^{148}, x^{41}, x^{82}, x^{164}, x^{73}, x^{146}$	0.1157
$x^{39}, x^{57}, x^{78}, x^{114}, x^{147}, x^{156}, x^{201}, x^{228}$	0.1194
$x^{43}, x^{86}, x^{89}, x^{101}, x^{149}, x^{172}, x^{178}, x^{202}$	0.1140
$x^{45}, x^{75}, x^{90}, x^{105}, x^{150}, x^{165}, x^{180}, x^{210}$	0.1484
$x^{47}, x^{94}, x^{121}, x^{151}, x^{188}, x^{203}, x^{229}, x^{242}$	0.1129
$x^{51}, x^{102}, x^{153}, x^{204}$	0.1803
$x^{53}, x^{77}, x^{83}, x^{106}, x^{154}, x^{166}, x^{169}, x^{212}$	0.1146
$x^{55}, x^{110}, x^{115}, x^{155}, x^{185}, x^{205}, x^{220}, x^{230}$	0.1251
$x^{59}, x^{103}, x^{118}, x^{157}, x^{179}, x^{206}, x^{217}, x^{236}$	0.1156
$x^{61}, x^{79}, x^{122}, x^{158}, x^{157}, x^{211}, x^{233}, x^{244}$	0.1161
$x^{63}, x^{126}, x^{159}, x^{207}, x^{231}, x^{243}, x^{249}, x^{252}$	0.1192
x^{85}, x^{170}	0.3190
$x^{87}, x^{93}, x^{117}, x^{171}, x^{174}, x^{186}, x^{213}, x^{234}$	0.1146
$x^{91}, x^{107}, x^{109}, x^{173}, x^{181}, x^{182}, x^{214}, x^{218}$	0.1155
$x^{95}, x^{125}, x^{175}, x^{190}, x^{215}, x^{235}, x^{245}, x^{250}$	0.1283
$x^{111}, x^{123}, x^{183}, x^{189}, x^{219}, x^{222}, x^{237}, x^{246}$	0.1191
$x^{119}, x^{187}, x^{221}, x^{238}$	0.1243
$x^{127}, x^{191}, x^{223}, x^{239}, x^{247}, x^{251}, x^{253}, x^{254}$	0.1174