Hindawi
Security and Communication Networks
Volume 2022, Article ID 7771621, 11 pages
https://doi.org/10.1155/2022/7771621

Research Article

Assessment of Addition-Chain-Based Masked S-Box Using Deep-Learning-Based Side-Channel Attacks

Huizhong Li,[1,2] **Jingdian Ming**,[1,2] **and Yongbin Zhou**[1,2,3]

[1]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*School of Cyber Security, Nanjing University of Science and Technology, Nanjing 210094, China*

Correspondence should be addressed to Yongbin Zhou; zhouyongbin@iie.ac.cn

Received 7 December 2021; Accepted 21 February 2022; Published 24 March 2022

Academic Editor: Dragan Peraković

Masking schemes are considered to be effective countermeasures to protect Internet-of-Things devices from side-channel attacks. Deep-learning-based side-channel attacks (DL-SCAs) have been demonstrated to be very effective targeting on masked implementations. In this paper, we investigate the resistance of a popular computation-based masking scheme against DL-SCAs, that is, the addition-chain-based one. We find that addition chain introduces computations of intermediate monomials over $\mathbb{F}_{2^n}$ with smaller output sizes, which decreases its resistance against DL-SCAs. Specifically, we first use mutual information metric to evaluate the side-channel resistance of different monomials from an information theory point of view. Next, we further propose the Kullback–Leibler divergence ratio as an evaluation metric to analyze the impact of monomial output size on DL-SCAs. The measurement values show that the monomial with smaller output size is less-resistant against DL-SCAs. Then we conduct simulated and practical experiments respectively to verify it. In simulated experiments, we perform DL-SCAs on first-order masked implementations with different noise levels and training trace numbers. The results demonstrate that monomials with smaller output size are more vulnerable. Moreover, with the increase (resp. decrease) in noise level (resp. training trace number), the resistance difference of these monomials becomes more significant. In addition, we obtain similar results through simulated experiments on second-order masked scenario. In practical experiments based on an ARM Cortex-M4 architecture, we collect power and electromagnetic traces in consideration of low and high noise levels. The results show that the number of required traces for targeting the S-Box output is at least three times as much that for targeting the weakest monomial.

1. Introduction

With the emergence and explosive development of the Internet of Things (IoT), the security issues of IoT devices have attracted extensive attention in both the academic and industrial communities [1, 2]. On one hand, attackers can facilitate unauthorized and malicious activities by identifying and exploiting vulnerabilities in IoT devices [3, 4]. On the other hand, IoT devices have been exploited to create botnet networks to generate distributed denial of service (DDoS) traffic [5, 6]. Furthermore, as numerous IoT devices gather and investigate private data, they become a gold mine for hostile actors [1]. The potential to diagnose compromised nodes, as well as the collection and preservation of testimony of an attack or illegal activity, have become top priorities. Therefore, it puts forward new requirements for digital forensics to uncover vital evidence. Actually, IoT devices have been shown to provide new kinds of evidence sources that were not available in traditional digital forensics primarily used for desktop or laptop computers [7–9]. However, unlike desktop and laptop computers, the bespoke hardware and software employed on most IoT devices obstruct the use of classical digital forensic evidence acquisition methods [10]. This situation demands alternative approaches to forensically inspect IoT devices. Among various techniques, side-channel attacks (SCAs) have been demonstrated to be a promising digital forensics approach [11–13].

SCAs exploit various physical leakages, for example, the running time [14], the power consumption [15], or the electromagnetic emanations [16], of a cryptosystem to recover its sensitive data. Generally speaking, side-channel attacks can be divided into two classes: nonprofiled attacks, such as differential power analysis (DPA) [17] and correlation power analysis (CPA) [18], and profiled attacks, such as template attacks (TAs) [15] and stochastic attacks (SAs) [19]. Among these SCAs, profiled attacks are recognized as more powerful ones and play a fundamental role of security evaluation of cryptographic algorithm implementations. Recently, a line of (since current research mainly focuses on using deep learning techniques for profiled attacks, we refer to deep-learning-based profiled attacks as DL-SCAs in this paper.) deep-learning-based side-channel attacks (DL-SCAs), raise SCA community's concern [20–23]. The practical results have demonstrated that these techniques are very effective to attack the embedded implementations even when some well-known countermeasures are involved.

Protecting cryptographic implementations from SCAs has been a challenging and longstanding issue for the embedded systems industry. Among countermeasures against SCAs, masking [24] is the most widely used because it is device-independent and provably secure. Generally speaking, masking aims to randomize the subtle dependency between sensitive intermediate and its corresponding side-channel leakages by splitting the sensitive values into $d + 1$ shares [25], where d is called the masking order. Any d among $d + 1$ shares are assigned uniform random and independent values. When protecting a cryptographic algorithm, it suffices to apply the operation individually to each share to perform a linear operation on masked data. In comparison, masked nonlinear operations are more difficult to implement [25]. There are mainly two ways at an acceptable level of cost to solve this problem: (1) implement by look-up tables 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 [26, 27] in higher-order masked implementations. As for implementations by computing over a finite field, addition chain has become a research hotpot [28–32]. Specifically, the nonlinear operation can be expressed as a sequence of squares and multiplications over $\mathbb{F}_{2^n}$. These nonlinear multiplications can be then implemented using previously known schemes, such as ISW [25]. Until now, there is still a lack of research on the resistance of addition-chain-based implementations against DL-SCAs because most DL-SCAs focus on look-up-table-based masked implementations [20, 21]. Therefore, we comprehensively investigate the performance of DL-SCAs on addition-chain-based masked S-Box implementations in this work. In addition, as Boolean masking is one of the most popular schemes which enables high performance when implemented on practical circuits [31], we focus on the analysis of Boolean masked S-Box implementations.

Our Contributions **.** In this paper, we investigate the side-channel resistance of addition-chain-based Boolean masked S-Box implementations when considering DL-SCAs. We find that in addition chains, the computations of intermediate monomials with smaller output sizes decrease the resistance of implementations against DL-SCAs. In order to analyze the impact of monomial output size on attacks, we use mutual information and propose the Kullback–Leibler (KL) divergence ratio as evaluation metrics, respectively. The measurement values show that the monomial with smaller output size is less resistant against DL-SCAs. Then we conduct simulated and practical experiments to verify it. In simulated experiments, we perform DL-SCAs on first- and second-order masked implementations respectively. The results show that monomials with smaller output size are more vulnerable. In addition, we evaluate the resistance of different monomials when the number of training traces is relatively small. With the decrease in training trace number, the resistance difference of these monomials becomes more significant, and the overall trend does not change. In practical experiments based on an ARM Cortex-M4 architecture, we collect power and electromagnetic traces in consideration of low and high noise levels. We obtain similar results to those of simulated experiments. Moreover, the practical results show that the number of required traces for targeting the S-Box output is at least three times as much that for targeting the weakest monomial.

The rest of our paper is organized as follows. Section 2 discusses the related work concerning addition-chain-based masked S-Box schemes and the analysis of side-channel security for these schemes. Section 3 reviews the notations and preliminaries. In section 4, we first use mutual information metric to evaluate the side-channel resistance of different monomials from an information theory point of view. Then, we propose the KL divergence ratio metric and analyze the relationship between the output size of monomials and their resistance against DL-SCAs. Next, in Section 5, we verify our analysis by simulated and practical experiments. Section 6 gives recommended addition chains with relatively high resistance against DL-SCAs. Finally, Section 7 concludes the paper.

2. Related Work

Rivain–Prouff masking scheme is the first provably secure higher-order masking for AES [31] using addition chain, as shown in Figure 1. In this way, the AES S-Box can be masked at any security order d. Later, it was extended to a generic method for higher-order masking by Carlet et al. [32]. Using the method they proposed, any n-bit S-Box can be expressed as a sequence of linear squares and nonlinear multiplications over $\mathbb{F}_{2^n}$. Then from a theoretical perspective, Roy and Vivek and Coron et al. [28, 29] further reduced the complexity of several well-known S-Boxes. The best-known method for fast polynomial evaluation was proposed by Carlet et al. [33]. From an implementation perspective, Coron et al. proposed to use common shares to further improve the addition chain in parallel implementations [30]. Actually, addition chains are widely used in various masking schemes, such as Boolean masking [31], mixed additive and multiplicative masking [34], and inner product masking [35].

The side-channel security of addition chain implementations has been studied by Prouff and Rivain [36] and Alexandre Duc et al. [37]. They assumed that the leakages in

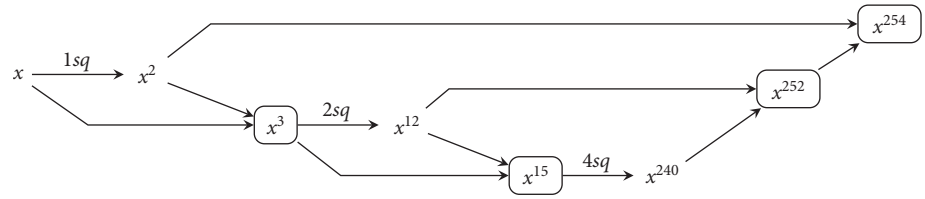


Figure 1: The computation of x^{254} used by Rivain and Prouff [31]. Monomial with border means that the computation for this monomial is a multiplication, and the number over the arrow represents the number of squares.

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. It can be seen as a simplified scenario, but the leakages from each operation are actually different and related to the function itself [38]. Then, Ming et al. [38] investigated how different operations impact the security of addition chain implementations. However, they mainly studied the resistance of addition chains against nonprofiled SCAs. For profiled attacks especially DL-SCAs, the experiments were only performed on the worst monomial computation (x^{85}) and the S-Box computation (output step) in AES. Therefore, there is still a lack of comprehensive study on the resistance of addition-chain-based implementations against DL-SCAs.

3. Preliminaries

3.1. Addition-Chain-Based Masked S-Boxes. The principle of masking is to split the sensitive variables into $d + 1$ shares, which satisfy the following relation:

$$x = x_0 \oplus x_1 \ldots \oplus x_d, \tag{1}$$

where x, x_i, and d denote the sensitive variable, the shares, and the masking order, respectively. Usually, the d shares $\{x_1, \ldots, x_d\}$, called masks, are assigned uniform random and independent values [25]. And the x_0, called the masked value, is processed such that it satisfies (1). In this paper, we consider the exclusive-OR (XOR) operation, so we use the notation $\oplus$.

It has been shown that any n-bit S-Box can be represented by a polynomial $\mathrm{S-Box}(x) = \sum u_i x^i$ over $\mathbb{F}_{2^n}$ [32], and the u_i can be obtained from the look-up table by applying Lagrange's Interpolation theorem. Thus, the common approach is to decompose each power function in terms of squares and nonlinear multiplications, where the nonlinear multiplications can be implemented, for example, using the ISW scheme [25]. The addition chain [39] is defined as follows:

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

$$a_0 = 1, a_1, a_2, \ldots, a_r = \alpha, \tag{2}$$

for every $i = 1, 2, \ldots, r$, there exist some $0 \le j, u \le i$ such that $a_i = a_j + a_u$.

In fact, the exponential computation for each monomial in an S-Box can be expressed as an addition chain. In this paper, we also use $\mathcal{F}$ to denote the whole processing required to compute an n-bit S-Box, where F_i denotes the ith intermediate monomial in $\mathcal{F}$. Thus, the processing in Figure 1 can be expressed as $\mathcal{F} = \langle x, x^2, x^3, x^6, x^{12}, \ldots, x^{254} \rangle$.

However, the number of intermediate computations gets increased when implementing S-Boxes through addition chain, which leads to more leakages [40]. Therefore, the adversary can use much fewer traces to attack the computation of certain monomials rather than S-Box outputs [38].

3.2. Deep-Learning-Based Side-Channel Attacks. A profiled side-channel attack consists of two phases: an offline profiling phase (training in deep learning context) and an online attack phase (testing respectively).

In profiling phase, the attacker has a device with knowledge about the secret key implemented and acquires a set of N_p side-channel traces $\mathcal{L}_{\text{profiling}} = \{\widetilde{l}[i] | i = 1, 2, \ldots, N_p\}$. Each trace $\widetilde{l}[i]$ is corresponding to $x[i] = f(m[i], k)$ in one encryption or decryption with known key k, where $f(\cdot)$ denotes the function of monomial in addition chain or AES S-Box in this work. $m[i]$ denotes the plaintext or ciphertext, and $x[i]$ denotes the sensitive intermediate value. Once the acquisition is done, the attacker builds suitable models and computes the estimation of probability $\Pr[\mathbf{L}|X = x]$ from a profiling set $\{(x[i], x[i])\}_{i=1,2,\ldots,N_p}$.

In the attack phase, the attacker acquires a small new set of traces $\mathcal{L}_{\text{attack}} = \{\mathbf{l}[i] | i = 1, 2, \ldots, N_p\}$ with a fixed unknown key k^*k^*, where N_a denotes the number of attack traces. With the help of the established models, the attacker can easily calculate the estimated posterior probabilities d_k among $|\mathcal{K}|$ guesses via the Bayes' Theorem, then select the key that maximizes it following the Maximum Likelihood strategy.

Recently, DL-SCAs have been shown to be a very efficient alternative to the state-of-the-art profiled attacks, and even outperform the traditional profiled attacks [20, 23]. Since convolutional neural networks (CNNs) have been shown to be good choices against most common countermeasures, we focus on using CNNs to attack addition chain implementations. Generally speaking, the CNNs consist of two main parts: feature extractor and classifier. Feature extractor is composed of stacked operations of convolution, pooling, and sometimes normalization layers. The classifier is composed of several fully connected layers. Each layer of the network receives the output from its immediate previous layer as its input, and passes its output as the input to the next layer, as it is called forward propagation. Higher-level

features are derived from features propagated from lower-level layers and finally calculate classification probabilities in the last output layer (for a classification task, usually the output layer is activated by Softmax function as detailed by Cagli et al. [20]).

In deep learning context, the purpose of the profiling phase is to train an neural network to estimate the correct label with high probability from the input traces in the following attack phase. The loss function is configured to compute the error generated by the model. During the training process, the attacker aims to minimize the loss function in order to get the best model possible. The most used loss function in the literature is the cross-entropy [41]. Let θ represent the model parameters of the neural network (e.g., the weights of a multilayer perceptron). Let $P_{k^*}(x)$ denote true labels with the correct key, and let $Q(x|\mathbf{l};\theta)$ be the output conditional probability represented by the neural network. Then, we define the cross-entropy of a deep leaning model as follows:

$$H(P_{k^*}, Q) = \mathbb{E}_{x\sim P_{k^*}} - (\log Q(x|\mathbf{l};\theta)), \tag{3}$$

where $\mathbb{E}$ denotes the expected value.

4. Analysis on Addition-Chain-Based Boolean Masked S-Box

4.1. The Output Size of Monomial Functions in Addition Chain. When analyzing side-channel security of addition-chain-based masked S-Box implementations, the leakages of each monomial computation in the addition chain can be utilized to perform attacks. And the output sizes of different monomials vary significantly. Since the profiling phase of DL-SCAs can be regarded as the training process of a classifier [42], the output size (i.e. the number of categories) will affect the classification performance. Then, during the attack phase, the probability of sensitive intermediate value is mapped to the probability of each key candidate $k \in \mathcal{K}$. Although the number of key candidates $|\mathcal{K}|$ (i.e. 256 in AES) is the same for monomials with different output sizes, the classification performance may still affect the attack results. Therefore, the output size of the target monomial may affect the difficulty for conducting a successful attack. In Table 1, we give the output sizes of AES S-Box and monomial functions on irreducible polynomials $x^8 + x^4 + x^3 + x + 1$ over $\mathbb{F}_{2^8}$. The # Output denotes the output size of functions, and the # Element denotes the number of functions whose output size is equal to a certain value. It can be observed that depending on the size of the output, functions can be divided into seven classes. It is clear that the exponent numbers of monomials in the same class have the same greatest common divisor with 255. So we also give the greatest common divisor of the monomial exponent number b and 255 in the column $\gcd(b, 255)$. Next, we first analyze the impact of output size from an information theory point of view.

4.2. Analysis Based on Information-Theoretic Metric. Information theoretic (IT) metrics measure the total information leakage irrespective of specific side-channel attacks. Mutual information (MI), as a well-known IT metric, has been widely used in side-channel analysis [43, 44]. Therefore, we use MI to evaluate the side-channel resistance of different monomials from an IT viewpoint.

Table 1: The output sizes of AES S-box and monomial functions on irreducible polynomials $x^8 + x^4 + x^3 + x + 1$ over $\mathbb{F}_{2^8}$.

Function	# element	$\gcd(b, 255)$	# output
AES S-box, x, x^2, x^4, ...	129	1	256
x^3, x^6, x^9, x^{12}, ...	64	3	86
x^5, x^{10}, x^{20}, x^{25}, ...	32	5	52
x^{15}, x^{30}, x^{45}, x^{60}, ...	16	15	18
x^{17}, x^{34}, x^{68}, x^{136}, ...	8	17	16
x^{51}, x^{102}, x^{153}, x^{204}	4	51	6
x^{85}, x^{170}	2	85	4

As illustrated by Standaert et al. [45], 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 $\mathcal{L}_i$ denotes the leakage of each share. We simulate the leakages $\mathcal{L}_0(x_0), \ldots, \mathcal{L}_d(x_d)$ as $\mathcal{L}_i(x_i) = HW(x_i) + \mathcal{N}_i$ in detail. Then the mutual information between the sensitive variable X and the leakage is denoted as $I(\mathcal{L}; X)$. As shown in Table 1, all monomials can be grouped into 7 classes in the sense of output size. And monomials with the same output size can be viewed as equivalent when using MI metric [38]. Considering $d = 1$ in the first-order Boolean masking, the MI results of AES S-Box and monomial functions are depicted in Figure 2. We take the first term from each class $(\text{AES } S-\text{Box}, x^3, x^5, x^{15}, x^{17}, x^{51}, x^{85})$ as representatives.

It can be observed that basically, the smaller the output size of the function, the higher the value of MI. This indicates that the function with smaller output size is likely to be less resistant against attacks. However, the results of MI metric do not exactly match the output size of functions. For example, the output size of x^5 is larger than that of x^{17}, but x^5 leaks more under MI metric. Besides, with the increase of the output size, the difference between the MI values of different functions becomes very small. Moreover, in Ming et al.'s study [38], it was also pointed out that the MI metric fails to work when evaluating the CPA resistance of different monomials. Therefore, to further investigate the impact of output size on DL-SCAs, we propose the KL divergence ratio as an evaluation metric.

4.3. Analysis Based on KL Divergence Ratio. The motivation for the use of the KL divergence is to quantitatively evaluate the difficulty of DL-SCAs directly from the viewpoint of the model's output distribution shape. Actually, the KL divergence between the probability distribution of true labels and the predicted distribution of the deep learning model is frequently used to evaluate the classification performance of neural networks [46].

In the case of SCAs, that is to calculate the KL divergence between the probability distribution of labels with the correct key $P_{k^*}(x)$ and the conditional probability distribution $Q(x|\mathbf{l};\theta)$ of a deep learning model with parameter θ, which is defined as follows:

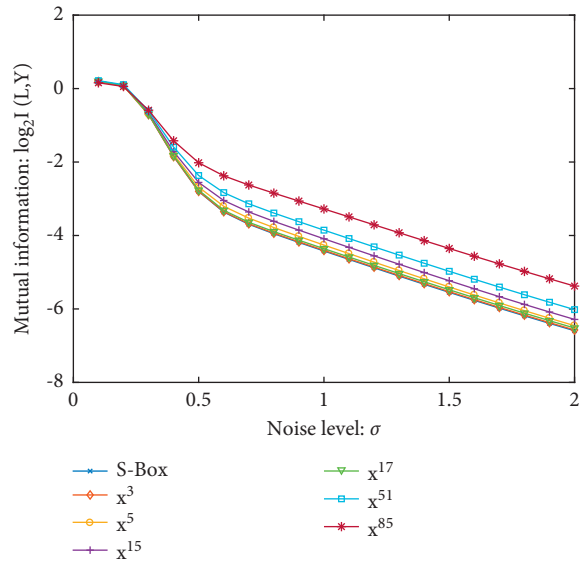


FIGURE 2: Mutual information of different output size functions for $d = 1$.

$$D_{\mathrm{KL}}\left(P_{k^*}\|Q\right)\mathbb{E}_{x\sim P_{k^*}}\left[\log\frac{P_{k^*}(x)}{Q(x|\mathbf{l};\theta)}\right]. \tag{4}$$

A lower KL divergence implies that the distribution of the deep learning model output is closer to the true label distribution. In other words, the model with smaller KL divergence has better classification performance. From (3) and (4), it can be observed that $D_{KL}(P_{k^*}\|Q) = H(P_{k^*}, Q) + (-\mathbb{E}_{x\sim P_{k^*}}\log P_{k^*}(x))$, where $-\mathbb{E}_{x\sim P_{k^*}}\log P_{k^*}(x)$ is the entropy of $P_{k^*}(x)$. In DL-SCAs, the true label is represented with a one-hot encoding, which means the entropy of $P_{k^*}(x)$ is equal to 0. Therefore, KL divergence is equivalent to cross-entropy in this scenario.

In the attack phase, the KL divergence can be estimated as follows:

$$\widehat{D}_{\mathrm{KL}}\left(P_{k^*}\|Q\right) = \frac{1}{N_a}\sum_{i=1}^{N_a}\sum_{x\in\mathcal{X}}P_{k^*}(x)\log\left(\frac{P_{k^*}(x)}{Q(x|\mathbf{l};\theta)}\right). \tag{5}$$

It can be observed that equations (4) and (5) only consider the probability distribution of labels with the correct key k^*. However, an adversary in SCAs needs to distinguish the correct key from other key hypotheses. Namely, the efficiency of the attack is also influenced by $\widehat{D}_{\mathrm{KL}}(P_{k^*}\|Q) = 1/N_a\sum_{i=1}^{N_a}\sum_{x\in\mathcal{X}}P_{k^*}(x)\log(P_{k^*}(x)/Q(x|\mathbf{l};\theta))$. for $k\neq k^*$. Therefore, we propose a novel metric KL divergence ratio (denoted as DR_{KL}) for deep learning model performance evaluation in side-channel scenario as follows:

$$DR_{\mathrm{KL}} = \frac{D_{\mathrm{KL}}\left(P_{k^*}\|Q\right)}{\mathbb{E}_{k\neq k^*}\left[D_{\mathrm{KL}}\left(P_{k^*}\|Q\right)\right]}. \tag{6}$$

The lower the KL divergence ratio of a deep learning model is, the more effective it is to attack the targeted function.

In order to analyze the impact of output size on attacks, one can first use leakages from monomials with different output sizes to train the deep learning model, and then calculate the KL divergence ratio in the attack phase respectively. In the attack phase, the KL divergence ratio is estimated as follows:

$$\widehat{DR}_{\mathrm{KL}} = \frac{\widehat{D}_{\mathrm{KL}}\left(P_{k^*}\|Q\right)}{\mathbb{E}_{k\neq k^*}\left[\widehat{D}_{\mathrm{KL}}\left(P_{k^*}\|Q\right)\right]}. \tag{7}$$

For fairly comparing the resistance of different monomials, all experimental settings (including the noise level of leakages, the number of traces used for training and attack, the network architecture, etc) should be the same except for the targeted function and its corresponding leakages.

Specifically, we also take the first term from each of the seven classes of functions in Table 1 as representatives, and analyze the impact of output size on their KL divergence ratio values through simulated experiments. We train CNNs and perform attacks on simulated leakages for first-order $(d = 1)$ and second-order $(d = 2)$ 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. For $d = 1$, we consider three different noise levels $\sigma = 0.5$, $\sigma = 1$, and $\sigma = 2$, where σ denotes the standard deviation of noise. For $d = 2$, we simulate traces with the noise level $\sigma = 1$. When $d = 1$ and $d = 2$, we use 10,000 and 30,000 traces in the profiling phase, respectively, and 50,000 traces are used to estimate the KL divergence ratio for each attack. The traces are labeled by the value of the SubByte output (e.g., 4 labels for x^{85} and 256 labels for S-Box) in the one-hot encoding representation. The CNN architecture and other more detailed experimental settings are given in the next section. And the experimental results are shown in Figure 3.

Overall, results based on the KL divergence ratio are easier to observe than those based on the MI metric, and more in line with our expectations. Broadly speaking, it can be observed that the smaller the output size of the function, the lower the value of KL divergence ratio, and vice versa. Therefore, the function with smaller output size is likely to be less resistant against DL-SCAs. We argue that a few exceptions are caused by the uncertainty of the model training or the high noise level.

5. Deep-Learning-Based Profiled Attacks on Masked S-Box Implementations

In this section, we verify our analysis by simulated and practical experiments, respectively. We perform DL-SCAs on different output size monomials and AES S-Box. The overall workflow of DL-SCAs is shown in Figure 4. In simulated experiments, the profiling, validation, and attack traces are generated by simulation. For each targeted implementation, we perform the following:

(1) Collect or generate traces for the profiling, validation, and attack phases, respectively

(2) Train CNN models

(3) Perform the key recovery and compute the success rate for each attack

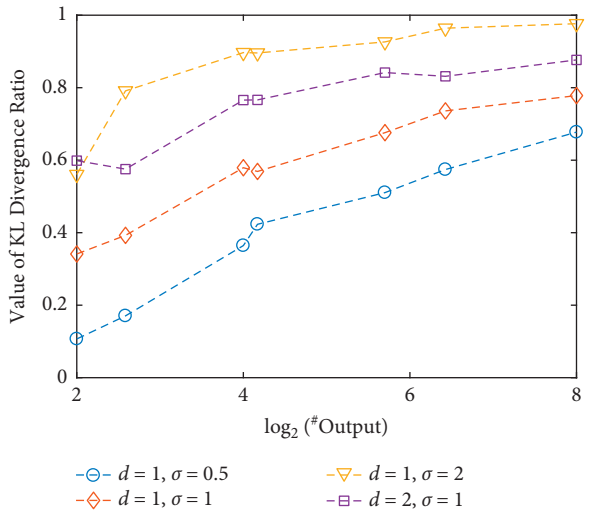


Figure 3: The KL divergence ratio values of different output size functions.

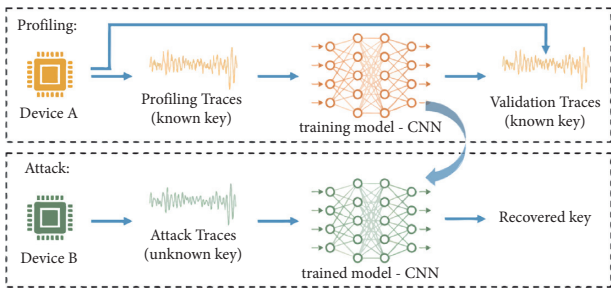


Figure 4: The overall workflow of DL-SCAs.

5.1. CNN Architecture. We refer to few previous works [21, 22] and use the hyperopt library [47] for designing a CNN model. The CNN is composed of five convolutional blocks followed by six fully connected layers. Each layer is activated by SeLU function and He Uniform initialization is used to improve the weight initialization. As for convolutional blocks, the kernel size is 11 and the number of kernels is $\{64, 128, 256, 512, 512\}$. The outputs of a convolutional layer are fed into the batch normalization layer and average pooling layer in each convolutional block. As for fully connected layers, each layer has 20 neurons except the last one. The number of nodes in the output layer is equal to the output size of functions (e.g., 4 neurons for attacking x^{85} and 256 neurons for attacking AES S-Box, respectively). The output layer is activated by Softmax function. The cross-entropy is used as loss function. In order to facilitate the comparison of the resistance of different monomials, 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 monomials in addition chains.

5.2. Simulated Experiments

5.2.1. Experimental Setup. We perform attacks on simulated leakages for first- and second-order masked implementations, respectively. The leakages from computations of set $\{\text{AES } S-\text{Box}, x^3, x^5, x^{15}, x^{17}, x^{51}, x^{85}\}$ are simulated using the Hamming weight model as in Section 4. Table 2 shows the simulated leakage settings and data splitting sizes in each attack scenario. The #PoIs denotes the points of interest (PoIs) for each share. Since we consider an evaluation scenario of DL-SCAs for masked S-Box addition chains, we assume that the attacker can exactly select PoIs. Specifically, for $d = 1$, each simulated leakage trace $\mathbf{t}$ contains 40 time samples, and the trace is generated as follows:

$$\mathbf{t}_{d=1}[i] = \begin{cases} HW(x_0) + \mathcal{N}(0,\sigma), & \text{for } i \text{ in } [1,20], \\ HW(x_0) + \mathcal{N}(0,\sigma), & \text{for } i \text{ in } [21,40], \end{cases} \quad (8)$$

where $\mathbf{t}_{d=1}[i]$ denotes the ith time sample of the trace. x_0 and x_1 denote the two shares of the sensitive variable x. And for $d = 2$, each simulated leakage trace $\mathbf{t}$ contains 45 time samples, and the trace is generated as follows:

$$\mathbf{t}_{d=2}[i] = \begin{cases} HW(x_0) + \mathcal{N}(0,\sigma), & \text{for } i \text{in} [1,15], \\ HW(x_1) + \mathcal{N}(0,\sigma), & \text{for } i \text{ in} [16,30], \\ HW(x_2) + \mathcal{N}(0,\sigma), & \text{for } i \text{ in} [31,45]. \end{cases} \quad (9)$$

where x_0, x_1 and x_2 denote the three shares of the sensitive variable x. As for $d = 1$, we consider the different noise levels and different training trace numbers. As for $d = 2$, we simulate traces with the noise level $\sigma = 1$.

A mini batch of 256 is employed. The learning rate is initially 0.009, and a technique called one cycle policy [48] is used to choose the right learning rate. We set 75 epochs for the training in $d = 1$ experiments and 150 epochs in $d = 2$ experiments. The traces are also labeled by the value of the SubByte output as in Section 4. 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 @3.20 GHz 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 CNNs.

5.2.2. Experimental Results. The success rate is used to evaluate the effectiveness of attacks. We run each attack 100 times with randomly selected subsamples of attack sets to find the average number of traces to achieve a success rate higher than 80%. The process of training and attack for each monomial are performed 10 times in our experiments. We observe that the training process of CNNs is unstable to a certain extent. In other words, one or two out of 10 experiments for each monomial failed to retrieve the correct secret key. Therefore, we show the best attack results for each monomial to compare their resistance.

Table 2: The simulated leakage settings and data splitting sizes in each attack scenario.

Attack scenario	Masking order d	Noise level σ	#PoIs	Training set size	Validation set size	Attack set size
$\mathscr{A}_1$	1	0.5, 1, 2	20	9000	1000	5000
$\mathscr{A}_2$	1	1	20	900	100	500
$\mathscr{A}_3$	2	1	15	27,000	3,000	5000

The results of first-order masked implementations with different noise levels (corresponding to $\mathscr{A}_1$) are shown in Figure 5. First of all, AES S-Box is more resistant against DL-SCAs than x^{85}, which is consistent with the results by Ming et al. [38]. Besides, we can see that the functions (including monomials and AES S-Box) with larger output size are generally more resistant against DL-SCAs than those functions with smaller output size. The results are substantially consistent with the theoretical analysis. For example, AES S-Box and x^3 require more traces for successful attacks than x^{51} and x^{85}. However, one may also note that when $\sigma = 0.5$ and $\sigma = 1$, x^3 requires more traces to perform successful attacks than AES S-Box. We argue that the main reason is the noise level is so low that the difference between the two functions is not significant. Besides, there is some randomness in the training of CNNs. Therefore, it is possible that x^3 requires a bit more attack traces. With the noise level increases, it is obvious that AES S-Box is much more resistant than x^3.

Moreover, the results of small training trace number (corresponding to $\mathscr{A}_2$) are shown in Figure 6. With the decrease in training trace number, the resistance difference of different functions becomes more obvious, and the overall trend does not change. Interestingly, for the attack on AES S-Box, the correct key cannot be successfully retrieved. We argue that the main reason is the lack of training data, while the classification problem is too complicated (256-classification).

Finally, Figure 7 shows the results of second-order masked implementations when $\sigma = 1$ (corresponding to $\mathscr{A}_3$). It can be observed that the results are basically in line with those of first-order masked implementations. The main reasons for the few inconsistencies are the uncertain training process of CNNs and high noise level. Overall, the results demonstrate that monomials with smaller output size are more vulnerable to DL-SCAs.

5.3. Practical Experiments. We also perform attacks on practical first-order masked addition-chain-based S-Box implementation. Specifically, we target the addition chain proposed by Rivain and Prouff [31], which is the first provably secure higher-order masking for AES using addition chain. We use the open-source code given by Rivain and Prouff [31]. From Figure 1 and Table 1, it can be obtained that in $\{x, x^2, x^{254}\}$, 256 inputs are mapped to 256 outputs; in $\{x^3, x^6, x^{12}, x^{252}\}$, inputs are mapped to 86 outputs; and in $\{x^{15}, x^{30}, x^{60}, x^{120}, x^{240}\}$, inputs are mapped to 18 outputs. That is to say, there are only three different output sizes of monomials in this addition chain implementation. Therefore, we perform power and electromagnetic analysis on x^3, x^{15} and AES S-Box as typical representatives.

5.3.1. Power Analysis. Our measurement setup for power analysis is shown in Figure 8(a). It consists of the ChipWhisperer-Lite board, the CW308 UFO board and CW308-TM32F4 target board. The target board contains a 32-bit ARM Cortex-M4 CPU with an STM32F405 device. The sampling rate is set to 29.5 MHz. It is a relatively ideal environment with low noise for power analysis because the highest signal-to-noise ratio (SNR) is close to 100. A total of 3000 traces and 24,400 points for each trace are recorded. Among them, 2250 traces are used, and 20 PoIs of each share are selected for training CNNs. As we consider an evaluation scenario of DL-SCAs for masked S-Box addition chains, we assume that the attacker can select PoIs with the highest Pearson correlation coefficients with the sensitive intermediates. A total of 250 traces are used for validation, and 500 traces are used for attack. The CNN architecture and other experimental settings are the same as those of simulated experiments. The results are shown in Figure 9. The results are basically consistent with those of simulated attacks. Among the three target functions, it is obvious that AES S-Box is the most resistant against DL-SCAs, while x^{15} is the weakest one.

5.3.2. Electromagnetic Analysis. Our experimental environment for electromagnetic analysis is shown in Figure 8(b). The addition-chain-based masked implementations are running on an STM32F407, which is also a Cortex-M4-based microcontroller. Its electromagnetic consumption is measured through an electromagnetic near field probe RS H 400-1 on the surface of the microcontroller. The traces are obtained through an Agilent DSO90404A Digital Storage Oscilloscope with a high impedance adapter, and the sampling rate is set to 1 GHz. A total of 40,000 traces and 25,000 points for each trace are recorded. The collected electromagnetic traces are with a higher noise, as the highest SNR of PoIs is lower than 2. Among them, 27,000 traces are used, and 20 PoIs of each share are selected for training CNNs. A total of 3000 traces are used for validation, and 10,000 traces are used for attack. The CNN architecture and other experimental settings are the same as mentioned earlier. The results are shown in Figure 10. It can be observed that the results are consistent with those of power analysis. For the attack on AES S-Box, the correct key cannot be successfully retrieved. We argue that an important reason is the SNR of our electromagnetic measurements is much less than that of power measurements. Besides, the classification problem is too complicated.

6. Discussion

In general, both theoretical analysis and attack experiments show that the monomial with smaller output size is less

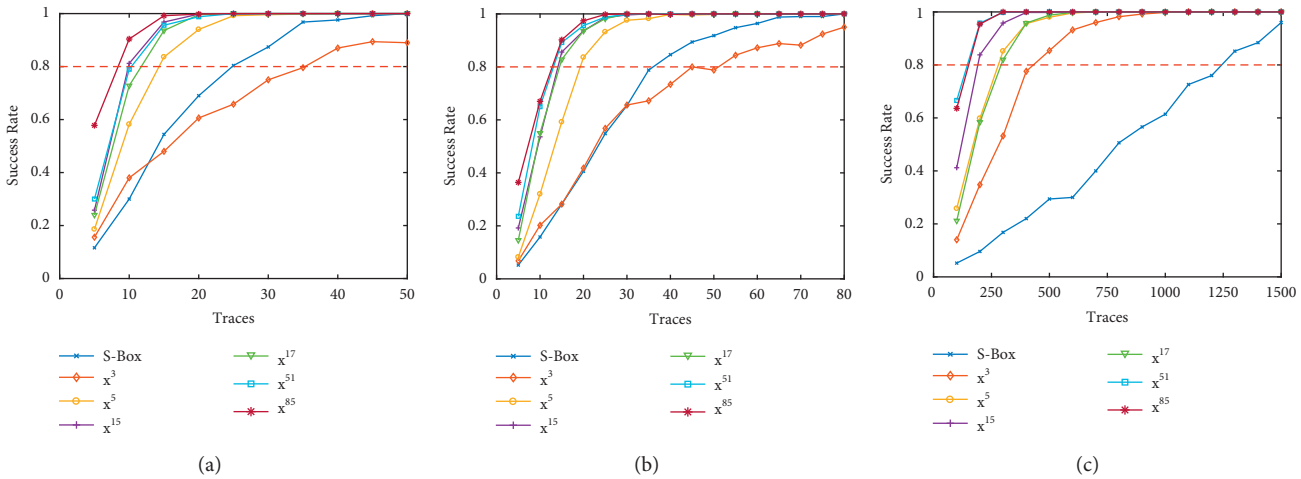


Figure 5: Attack scenario $\mathscr{A}_1$: the success rate of attacks on simulated leakages with different noise levels when $d = 1$ (a) $\sigma = 0.5$, (b) $\sigma = 1$, and (c) $\sigma = 2$.

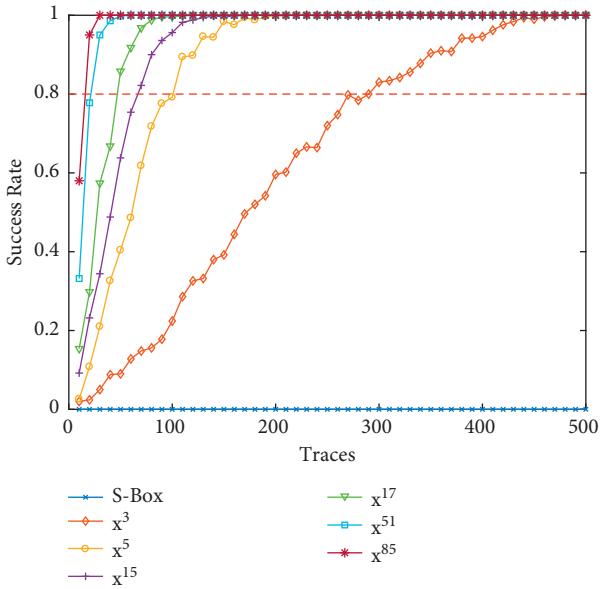


Figure 6: Attack scenario $\mathscr{A}_2$: the success rate of attacks on $\sigma = 1$ simulated leakages with small training trace number when $d = 1$.

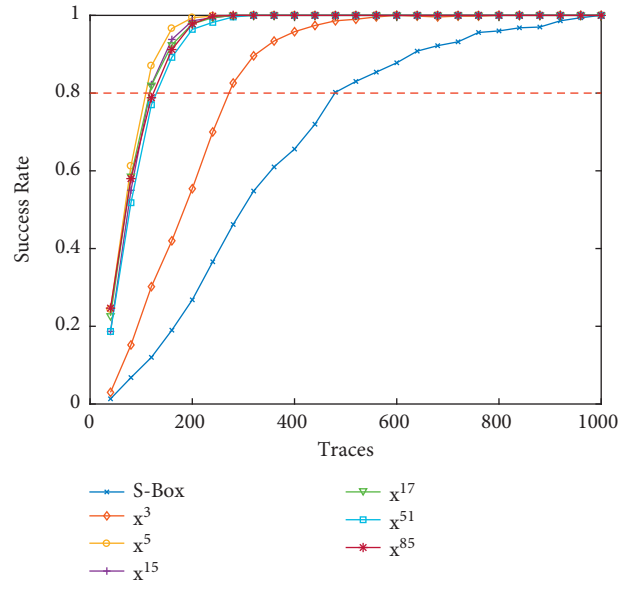
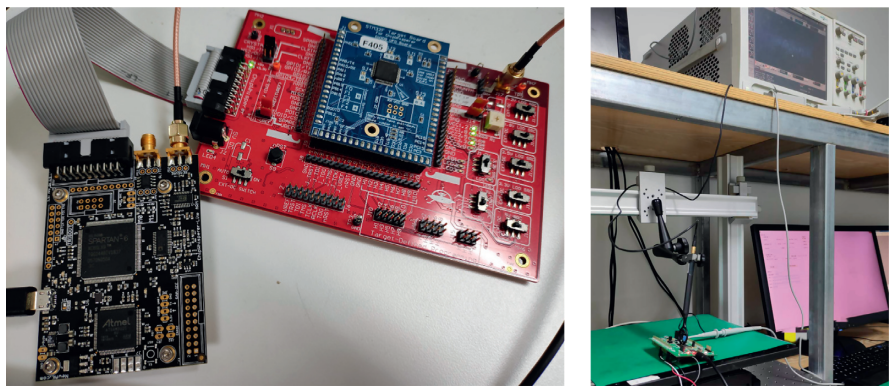


Figure 7: Attack scenario $\mathscr{A}_3$: the success rate of attacks on $\sigma = 1$ simulated leakages when $d = 2$.

resistant against DL-SCAs. Therefore, from the designer's point of view, in addition to efficiently implementing the addition chain, one should also choose those addition chains that avoid using small output size monomials. Actually, Carlet et at. [32] have proved that at least 7 squares and 4 multiplications are needed for an addition-chain-based AES S-Box. Ming et al. [38] recommended two strongest addition chains against CPA attacks for AES S-Box by enumerating the most efficient addition chains. As it is necessary to comprehensively consider different types of attacks when designing addition chains, we consider examining the resistance against DL-SCAs of addition chains recommended by Ming et al. [38].

For AES S-Box, the first recommended addition chain is $\mathscr{F} = \langle x, x^2, x^4, x^8, x^{16}, x^{18}, x^{32}, x^{64}, x^{82}, x^{86}, x^{172}, x^{254}\rangle$. It can be observed that x^{18} maps 256 inputs to 86 outputs, which is the monomial with the smallest output size in this chain. Except for x^{18}, all the other monomials in this addition chain map 256 inputs to 256 outputs. And the second recommended addition chain is $\mathscr{F} = \langle x, x^2, x^4, x^8, x^9, x^{18}, x^{19}, x^{27}, x^{54}, x^{108}, x^{127}, x^{254}\rangle$. It can be observed that $x^9, x^{18}, x^{27}, x^{54}$, and x^{108} map 256 inputs to 86 outputs, which are the monomials with the smallest output size in this chain. And all the other monomials in this addition chain map 256 inputs to 256 outputs. Overall, the aforementioned two addition chains are relatively strong against DL-SCAs.



: (a) (b)

Figure 8: Our experimental environment for collecting power and electromagnetic leakages: (a) collecting power leakages and (b) collecting electromagnetic leakages.

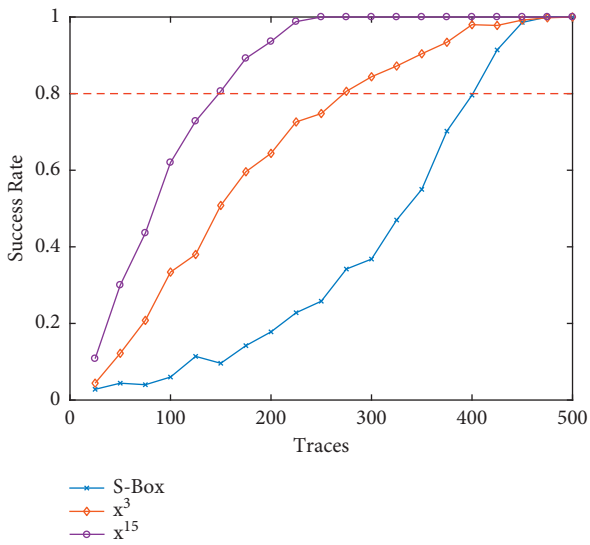


Figure 9: The success rate of attacks on practical power leakages when $d = 1$.

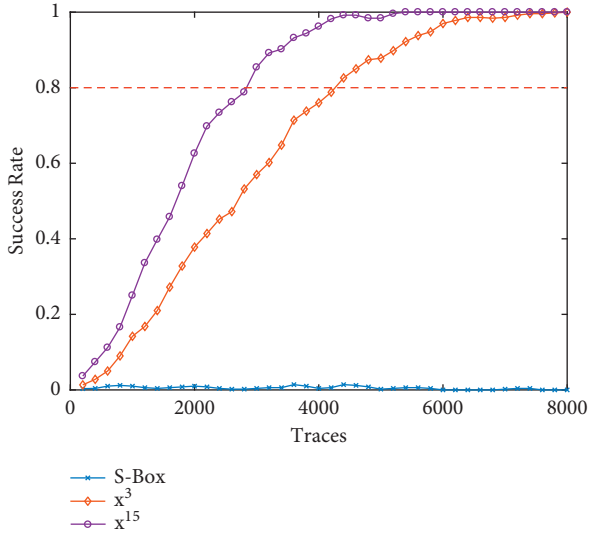


Figure 10: The success rate of attacks on practical electromagnetic leakages when $d = 1$.

However, there are five monomials in the second addition chain and only one monomial in the first chain with an output size of 86. Considering that the adversary may exploit the leakages of several monomials simultaneously, the first addition chain is the most recommended.

7. Conclusion

This paper investigates the performance of DL-SCAs on addition-chain-based Boolean masked S-Box implementations. We find that in addition chains, the computations of intermediate monomials with smaller output sizes decrease the resistance of implementations against DL-SCAs. First, we use MI metric to evaluate the side-channel resistance of different monomials from an IT viewpoint. Next, we further propose the KL divergence ratio metric to evaluate the impact of function output size on attacks. The measurement values show that the monomial with smaller output size is less resistant against DL-SCAs. Then we conduct simulated and practical experiments, respectively. The experimental results demonstrate that monomials with smaller output size are more vulnerable to DL-SCAs. Finally, we give some recommended guidelines on how to design addition chains with higher side-channel resistance according to the research results.

Actually, except Boolean masking, various masking schemes, such as mixed additive and multiplicative masking and inner product masking, are using addition chain to implement S-Boxes. It is our future work to evaluate the security of these masking schemes based on addition chain implementations.

Data Availability

All data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Nos. U1936209 and 62002353), China Postdoctoral Science Foundation (No. 2021M701726), and Yunnan Provincial Major Science and Technology Special Plan Projects (No. 202103AA080015).

References

[1] A. Khan and D. Peraković, "A survey on emerging security issues, challenges, and solutions for Internet of things (IoTs)," *Advances in Malware and Data-Driven Network Security*, pp. 148–175, 2022.

[2] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet-of-things networks," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4944–4956, 2021.

[3] H. HaddadPajouh, D. Ali, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on Internet of things security: requirements, challenges, and solutions," *Internet Things*, vol. 14, Article ID 100129, 2021.

[4] V. Adat and B. Brij, "Gupta. Security in Internet of things: issues, challenges, taxonomy, and architecture," *Telecommunication Systems*, vol. 67, no. 3, pp. 423–441, 2018.

[5] X. Zhang, O. Upton, N. Lang Beebe, and K.-K. Raymond Choo, "IoT botnet forensics: a comprehensive digital forensic case study on mirai botnet servers," vol. 32, Article ID 300926, 2020.

[6] I. Cvitic, D. Perakovic, B. B. Gupta, and K.-K. R. Choo, "Boosting-based DDoS detection in Internet of things systems," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2109–2123, 2022.

[7] R. K. Lomotey, J. C. Pry, and C. Chai, "Traceability and visual analytics for the internet-of-things (IoT) architecture," *World Wide Web*, vol. 21, no. 1, pp. 7–32, 2018.

[8] M. Chernyshev, S. Zeadally, and A. Zubair, "Baig, andrew woodward," *Internet of Things Forensics: The Need, Process Models, and Open Issues. IT Professional*, vol. 20, no. 3, pp. 40–49, 2018.

[9] I. Yaqoob, I. A. T. Hashem, T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, "Internet of things forensics: recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265–275, 2019.

[10] David Lillis, B. A. Becker, T. O'Sullivan, and M. Scanlon, "Current challenges and future research areas for digital forensic investigation," 2016, https://arxiv.org/abs/1604.03850.

[11] A. Sayakkara, *Electromagnetic Side-Channel Analysis Methods for Digital Forensics on Internet of Things*, School of Computer Science, University College Dublin, Dublin, Ireland, 2020.

[12] Le Quan, L. Miralles-Pechuán, A. Sayakkara, Nhien-An Le-Khac, and M. Scanlon, "Identifying Internet of things software activities using deep learning-based electromagnetic side-channel analysis," *Forensic Science International: Digital Investigation*, vol. 39, Article ID 301308, 2021.

[13] A. Sayakkara, Nhien-An Le-Khac, and M. Scanlon, "Facilitating electromagnetic side-channel analysis for IoT investigation: evaluating the EMvidence framework," *Forensic Science International: Digital Investigation*, vol. 33, Article ID 301003, 2020.

[14] C. Paul, "Kocher. Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems," *CRYPTO*, vol. 1109, pp. 104–113, 1996.

[15] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 13–28, CHES, Redwood Shores, CA, USA, August 2002.

[16] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier, "Electromagnetic side channels of an FPGA implementation of AES," *IACR Cryptol. ePrint Arch*, vol. 145, 2004.

[17] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the Annual International Cryptology ConferenceAdvances in Cryptology - CRYPTO' 99*, pp. 388–397, Santa Barbara, CA, USA, August 1999.

[18] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pp. 16–29, CHES, Cambridge, MA, USA, August 2004.

[19] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Proceedings of the Cryptographic Hardware and Embedded Systems - CHES 2005*, pp. 30–46, Edinburgh, UK, Sepember 2005.

[20] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," in *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pp. 45–68, CHES, Taipei, Taiwan, Sepember 2017.

[21] Z. Gabriel, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient CNN architectures in profiling attacks," *IACR Transaction Cryptography Hardware Embeded System*, vol. 2020, no. 1, pp. 1–36, 2020.

[22] L. Wouters, V. Arribas, B. Gierlichs, and B. Preneel, "Revisiting a methodology for efficient CNN architectures in profiling attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 3, pp. 147–168, 2020.

[23] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, pp. 3–26, Hyderabad, India, December 2016.

[24] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Proceedings of the Annual International Cryptology ConferenceAdvances in Cryptology - CRYPTO' 99*, pp. 398–412, Santa Barbara, CA, USA, December 1999.

[25] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: securing hardware against probing attacks," in *Proceedings of the Annual International Conference on Advances in Cryptology - CRYPTO 2003*, pp. 463–481, Santa Barbara, CA, USA, August 2003.

[26] J.-S. Coron, "Higher order masking of look-up tables," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic TechniquesAdvances in Cryptology - EUROCRYPT 2014*, pp. 441–458, Tallinn, Estonia, May 2014.

[27] J.-S. Coron, F. Rondepierre, and R. Zeitoun, "High order masking of look-up tables with common shares," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 1, pp. 40–72, 2018.

[28] A. Roy and S. Vivek, "Analysis and improvement of the generic higher-order masking scheme of FSE 2012," in *Proceedings of the International Coference on Cryptographic Hardware and Embedded Systems - CHES 2013*, pp. 417–434, Santa Barbara, CA, USA, January 2013.

[29] J.-S. Coron, A. Roy, and S. Vivek, "Fast evaluation of polynomials over binary finite fields and application to side-

channel countermeasures," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 73–83, 2015.

[30] J.-S. Coron, A. Greuet, E. Prouff, and R. Zeitoun, "Faster evaluation of SBoxes via common shares," in *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pp. 498–514, CHES, Santa Barbara, CA, USA, August 2016.

[31] M. Rivain and E. Prouff, "Provably secure higher-order masking of AES," in *Proceedings of the Cryptographic Hardware and Embedded Systems, CHES 2010*, pp. 413–427, Santa Barbara, CA, USA, August 2010.

[32] C. Carlet, L. Goubin, E. Prouff, M. Quisquater, and M. Rivain, "Higher-order masking schemes for S-boxes," *Fast Software Encryption*, pp. 366–384, 2012.

[33] C. Carlet, E. Prouff, M. Rivain, and T. Roche, "Algebraic decomposition for probing security," in *Proceedings of the Annual Cryptology Conference*, pp. 742–763, CHES, Santa Barbara, CA, USA, August 2015.

[34] A. Mathieu-Mahias and M. Quisquater, "Mixing additive and multiplicative masking for probing secure polynomial evaluation methods," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 1, pp. 175–208, 2018.

[35] W. Cheng, S. Guilley, C. Carlet, S. Mesnager, and J.-L. Danger, "Optimizing inner Product masking scheme by a coding theory approach," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 220–235, 2021.

[36] E. Prouff and M. Rivain, "Masking against side-channel attacks: a formal security proof." in *Proceedings of the Advances in Cryptology - EUROCRYPT 2013 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 142–159, Athens, Greece, May 2013.

[37] Alexandre Duc, S. Dziembowski, and S. Faust, "Unifying leakage models: from probing attacks to noisy leakage," *Journal of Cryptology*, vol. 32, no. 1, pp. 51–177, 2019.

[38] J. Ming, H. Li, Y. Zhou, W. Cheng, and Z. Qiao, "Revealing the weakness of addition chain based masked SBox implementations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 4, pp. 326–350, 2021.

[39] Donald Ervin Knuth, *The Art of Computer Programming, Volume I: Fundamental Algorithms*pp. 1–650, Addison-Wesley, Boston, MA, USA, 3rd edition, 1997.

[40] S. Micali and L. Reyzin, "Physically observable cryptography," in *Proceedings of the Theory of Cryptography*, pp. 278–296, Cambridge, MA, USA, February 2004.

[41] I. J. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning. Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA, USA, 2016.

[42] L. Masure, C. Dumas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*vol. 1, pp. 348–375, 2020.

[43] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic TechniquesAdvances in Cryptology - EUROCRYPT 2009*, pp. 443–461, Cologne, Germany, April 2009.

[44] G. Cassiers and F.-X. Standaert, "Towards globally optimized masking: from low randomness to low noise rate," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2, pp. 162–198, 2019.

[45] F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald et al., "The world is not enough: another look on second-order DPA," in *Proceedings of the Advances in Cryptology - ASIACRYPT 2010 16th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 112–129, Singapore, December, 2010.

[46] A. Ito, K. Saito, R. Ueno, and N. Homma, "Imbalanced data problems in deep learning-based side-channel attacks: analysis and solution," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3790–3802, 2021.

[47] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the of the 30th International Conference on Machine Learning (ICML 2013)*, pp. I-115–I-23, Atlana, GA, USA, June 2013.

[48] N. Leslie, *Smith. Cyclical Learning Rates for Training Neural Networks*, pp. 464–472, WACV, Snowmass Village, CO, USA, 2017.