SNR-Centric Power Trace Extractors for Side-Channel Attacks

Changhai Ou[1], Degang Sun[2], Siew-Kei Lam[1], Xinping Zhou[3], Kexin Qiao[3] and Qu Wang[4]

[1] School of Computer Science and Engineering, Nanyang Technological University, chou@ntu.edu.sg.
[2] Institute of Information Engineering, Chinese Academy of Sciences, sundegang@iie.ac.cn.
[3] Beijing Unionpay Card Technology Co., Ltd, China, zhouxinping@bctest.com,qiaokexin@bctest.com.
[4] School of Information and Communication Engineering, Beijing University of Posts and Telecommunication, wangqu@ict.ac.cn

Abstract. The existing power trace extractors consider the case that the number of power traces owned by the attacker is sufficient to guarantee his successful attacks, and the goal of power trace extraction is to lower the complexity rather than increase the success rates. Although having strict theoretical proofs, they are too simple and leakage characteristics of POIs have not been thoroughly analyzed. They only maximize the variance of data-dependent power consumption component and ignore the noise component, which results in very limited SNR to improve and seriously affects the performance of extractors. In this paper, we provide a rigorous theoretical analysis of SNR of power traces, and propose a novel SNR-centric extractor, named Shortest Distance First (SDF), to extract power traces with smallest the estimated noise by taking advantage of known plaintexts. In addition, to maximize the variance of the exploitable component while minimizing the noise, we refer to the SNR estimation model and propose another novel extractor named Maximizing Estimated SNR First (MESF). Finally, we further propose an advanced extractor called Mean optimized MESF (MMESF) that exploits the mean power consumption of each plaintext byte value to more accurately and reasonably estimate the data-dependent power consumption of the corresponding samples. Experiments on both simulated power traces and measurements from an ATmega328p micro-controller demonstrate the superiority of our new extractors.

Keywords: shortest distance first · SDF · MESF · signal-to-noise ratio · SNR · power trace extractor · side-channel attack.

1 Introduction

Secret information may leak from devices through side-channels such as electromagnetic radiation [2], cache patterns [22], acoustic [11,12], timing [3,18] and power consumption [19] during the implementation of cryptographic algorithms. These leakages are usually unconscious and difficult to be discovered. By applying statistical analysis on assumed power consumption of intermediate values and side-channel leakages, an attacker can recover sensitive information (e.g. encryption key) in the targeted devices. Side-channel attacks on power consumption channel, such as Differential Power Analysis (DPA) [19], Correlation Power Analysis (CPA) [4], Template Attacks (TA) [7] and Mutual Information

Analysis [13], pose serious threats to the security of cryptographic implementations. To improve the attack efficiency, an attacker always tries his best to construct optimal distinguishers and profile more accurate leakage models to better exploit leaky informations.

The Signal-to-Noise Ratio (SNR) becomes one of the key factors that affect the attack performance. They varies with algorithm implementation, hardware devices, sampling equipment, etc. For example, the probe is not placed on a reasonable position in electromagnetic attacks, and the implementation is in highly parallel. In this case, classic pre-processing schemes such as power traces alignment [35], averaging [23, 25, 32], higher-order cumulant [21], dimensionality reduction [6,31] and Points-Of-Interest (POIs) extraction [8], are usually performed before side-channel attacks to "de-noise" [14]. The attacker can obtain power traces with a higher SNR after pre-processing, thus obtaining higher attack performance. This paper considers the attack scenario that the attacker has enough power traces to guarantee his successful attacks. We focus on another powerful pre-processing scheme that recognizes and extracts a part of power traces with high SNR from the original set to lower the computing complexity and make the attacks more efficient. The related works will be given in the next section before introducing our contributions.

1.1 Related Works

Extracting power traces with high SNR to enhance attacks was firstly exploited by Kris et al. in [34]. The SNR of a time sample on power traces is the ratio of the variance of the exploitable power consumption component to the variance of noise (see Section 2.1). The power consumption of a POI of the outputs of an S-box could be approximated by a normal distribution. Parts of samples were extracted from both two tails of the distribution to enlarge the variance of the exploitable power consumption component [17]. Therefore, this scheme does not change the variance of noise. For simplicity, we name a power trace extraction scheme as an extractor.

Noura et al. exploited a new leakage model to extract power traces to improve CPA in [27]. Hu et al. proposed an Adaptive Chosen Plaintext Correlation Power Analysis (ACPCPA) in [15]. They tried to solve the problem of discarding too many power traces in the extractor proposed by Yongade et al. in [17]. Specifically, they analyzed the correlation between the Hamming weights of S-box outputs and power consumption of POIs, and got a conclusion that Hamming weights 0, 1, 7 and 8 corresponded to power traces with high SNR. They selected the best two candidates of a sub-key with the largest correlation coefficients in CPA, and encrypted plaintexts having Hamming weights 0, 1, 7 and 8 under them, respectively. Then, the newly acquired power traces were combined together with the original power trace set independently to perform CPA. This extractor was similar to the one given by Haruki et al. in [30] and the improved one given by us in [28], since they also aimed to enlarge the variance of the exploitable power consumption component rather than smaller the noise to improve SNR. The SNR improved by these extractors was very limited.

It is worth mentioning that it is difficult for the existing extractors to obtain a higher success rate [33] than the attack directly performed on the original power trace set. Therefore, we only consider the case that the number of power traces owned by the attacker is sufficient to guarantee his successful attacks, and the goal of his extraction is to lower the complexity rather than increase the success rate. For example, the attacker has very enough power traces to obtain a success rate of 1.00, but the samples are with too large noise and too many samples on power traces, so that he can extract a very small part from them to significantly reduce the computational complexity while achieve the same success rate. This is practical and meaningful.

Recently, other schemes such as Principal Component Analysis (PCA) [24], were also exploited to extract power traces [16]. The above-mentioned extractors can be able to accurately extract power traces with high SNR if the noise level is low. However, if the

noise level is high, the distribution of power consumption is seriously affected. In this case, accuracy of recognition of power traces with high SNR greatly reduces. Moreover, the variance of exploitable power consumption component could be improved is also very small compared to variance of noise in many cases. This also indicates that it is far from enough to increase SNR by enlarging the variance of exploitable power consumption component only. How to accurately recognize power traces with high SNR to improve the attack efficiency is still a very important but challenging issue.

1.2 Our Contributions

We aim to build more advanced extractors to recognize power traces with higher SNR, thus enhancing their practical significance and enabling the launch of more efficient attacks. The main contributions of this paper are as follows:

(i) We provide rigorous theoretical analysis of SNR in power traces, and propose a novel extractor named Shortest Distance First (SDF) to extract the power traces with smallest estimated noise rather than enlarging variance of exploitable power consumption. Samples with small noise occupy the majority of the whole signal population, thus providing ample opportunities to improve the SNR. We will show that SDF achieves significantly better performance than the existing extractors. The proposed method paves the way for new directions in power trace extraction.

(ii) To maximize the exploitable component while minimize the estimated noise, we analyze the SNR model of the extracted samples. We further introduce the data-dependent power consumption component exploited in the existing extractors into SDF, and propose a more reasonable extractor called Maximizing Estimated SNR First (MESF) according to this model.

(iii) It is not very good to directly use samples with noise to estimate their exploitable component. Supported by theoretical analysis, it is estimated by utilizing the mean power consumption of each plaintext byte value. Based on this scheme, we further build an extractor called Mean optimized MESF (MMESF), which takes advantage of known plaintexts in the side-channel community.

Experiments show that our new extractors only need a very small part of the original power trace set (e.g., one-tenth) to achieve a similar success rate, which demonstrates their superiority in blind recognition of power traces with high SNR.

1.3 Organization

The rest of this paper is organized as follows: leakage characteristics of POIs, CPA and the existing power trace extractors, which we collectively name TAILS, are introduced in Section 2. Our extractor SDF to minimize the estimated noise on power traces is given in Section 3. To further maximize the variance of the data-dependent component, another two extractors MESF and MMESF are introduced in detail in Section 4. Experiments on both simulated power traces and an ATmega328p micro-controller are presented in Section 5 to illustrate the superiority of our new extractors. Finally, we conclude this paper in Section 6.

2 Preliminaries

2.1 Leakage Characteristics of POIs

Let $x = x_0^j||x_1^j||\cdots||x_{15}^j$ denote the j-th encrypted plain-text in AES-128 algorithm, $\kappa = \kappa_0||\kappa_1||\cdots||\kappa_{15}$ denote the 16-byte key used in cryptographic device, $z_i^j = \mathsf{Sbox}\left(x_i^j \oplus \kappa_i\right)$

denote the look-up table operation and $l(\tau)$ denote the corresponding sample leaked at time τ. The attacker encrypts n plaintexts $\mathcal{X} = \left(x^1, x^2, \ldots, x^n\right)$ and acquires n power traces $\mathcal{L} = \left(l^1, l^2, \ldots, l^n\right)$. According to [23], the power consumption of a single sample $l(\tau)$ can be modeled as the sum of an operation-dependent component $l_o(\tau)$, a data-dependent component $l_d(\tau)$, electronic noise $l_{el.n}(\tau)$, switching noise $l_{sw.n}(\tau)$, and the constant component $l_c(\tau)$:

$$l(\tau) = l_o(\tau) + l_d(\tau) + l_{el.n}(\tau) + l_{sw.n}(\tau) + l_c(\tau). \tag{1}$$

These 5 components are independent of each other, and the exploitable component $l_e(\tau)$ consists of the operation-dependent component $l_o(\tau)$ and data-dependent component $l_d(\tau)$:

$$l_e(\tau) = l_o(\tau) + l_d(\tau). \tag{2}$$

For a time sample $\mathcal{L}(\tau)$, the variance of the constant component $\sigma^2(\mathcal{L}_c(\tau)) = 0$. For classic DPA, the attacker only considers one of the 8-bit intermediate values (i.e. the outputs of an Sbox), the power consumption of the other 7 bits is switching noise (i.e. algorithm noise). The variance of switching noise here is larger than 0. For classic CPA considering all bits of intermediate values, the variance of switching noise $\sigma^2(\mathcal{L}_{sw.n}(\tau)) = 0$.

Let $\mathcal{L}_n(\tau)$ denote the noise component including $\mathcal{L}_{el.n}(\tau)$ and $\mathcal{L}_{sw.n}(\tau)$. The electronic noise is normal distributed. The SNR of the time sample $\mathcal{L}(\tau)$ is the ratio of variance of exploitable power consumption component $\mathcal{L}_e(\tau)$ and the variance of noise component. Thus, SNR can be simplified as:

$$\hat{\mathsf{SNR}}(\tau) = \frac{\sigma^2(\mathcal{L}_e(\tau))}{\sigma^2(\mathcal{L}_n(\tau))}. \tag{3}$$

The same conclusion were drawn in Eq. 4.10 in [23]. For simplicity, the variance of 4 components $\sigma^2(\mathcal{L}_o(\tau))$, $\sigma^2(\mathcal{L}_d(\tau))$, $\sigma^2(\mathcal{L}_c(\tau))$ and $\sigma^2(\mathcal{L}_n(\tau))$ are expressed as $\sigma_o^2(\tau)$, $\sigma_d^2(\tau)$, $\sigma_c^2(\tau)$ and $\sigma_n^2(\tau)$.

2.2 Correlation Power Analysis

For all components of power consumption, $\sigma_d^2(\tau)$ is the only component correlating to the leakage model (e.g. Hamming weight model). Let $\mathcal{M}(\tau)$ denote the assumed leakage model of the intermediate values, the correlation coefficient between it and the total power consumption $\mathcal{L}(\tau)$ is:

$$\hat{\rho}(\mathcal{M}(\tau), \mathcal{L}(\tau)) = \frac{\mathsf{cov}(\mathcal{M}(\tau), \mathcal{L}(\tau))}{\sigma(\mathcal{M}(\tau)) \cdot \sigma(\mathcal{L}(\tau))}. \tag{4}$$

"cov" here is the covariance matrix operator. Mangard et al. further analyzed the correlation between power consumption and $\mathcal{M}(\tau)$ in [23]. The important Formula 6.5 given in their paper can be expressed as:

$$\hat{\rho}(\mathcal{M}(\tau), \mathcal{L}(\tau)) = \frac{\hat{\rho}(\mathcal{M}(\tau), \mathcal{L}_d(\tau))}{\sqrt{1 + \frac{1}{\hat{\mathsf{SNR}}(\tau)}}}. \tag{5}$$

Here $\hat{\rho}(\cdot, \cdot)$ is Pearson correlation coefficient operator. For a classic CPA attack, $\hat{\rho}(\mathcal{M}(\tau), \mathcal{L}_d(\tau))$ is a constant for a time sample, SNR determines the correlation coefficient $\hat{\rho}(\mathcal{M}(\tau), \mathcal{L}(\tau))$. Therefore, $\hat{\rho}$ approaches a constant when the number of power traces exploited in an attack is large enough. It can also be seen from Eqs. 3 and 5 that SNR plays a very important role in attacks, and its improvement will bring a higher success rate to the attacker.

2.3 Existing Power Trace Extractors

Let R denote the Gaussian distributed noise component $\mathcal{L}_n$ with mean 0 and variance σ_n^2, C denote the sum of constant component $\mathcal{L}_c$ and operation-dependent component $\mathcal{L}_o$. The leakage of the look-up table operation can be modeled as:

$$\mathcal{L} = \mathsf{HW}(\mathcal{Z}) + C + R. \tag{6}$$

Here $\mathsf{HW}(\cdot)$ is the Hamming weight function, and $\mathcal{Z} = \left(z^1, z^2, \ldots, z^n\right)$ are intermediate values. We simulate 25, 600 power traces and exploit the existing extractors to extract 12, 800 from them. Here C in Eq. 6 is set to 100, noise level σ_n^2 is set to 0.09, thus the samples of each Hamming weight in $\mathsf{HW}(\mathcal{Z})$ approximately follow normal distribution $\mathcal{N}(\mathsf{HW}(z), 0.09)$. This small noise is convenient for readers to observe the power consumption distribution of each Hamming weight. The principles of classic power trace extractors are the same. They improve SNR by enlarging the variance of component $\mathcal{L}_d$. Here we define an evaluation function:

$$\mathcal{D}^i(\tau) = \left| l^i(\tau) - \bar{\mathcal{M}}(\tau) \right| \tag{7}$$

to quantify this, $l^i(\tau)$ denotes the i-th trace in $\mathcal{L}$ and $\bar{\mathcal{M}}(\tau)$ denotes the mean power consumption of all traces, which satisfies:

$$\bar{\mathcal{M}}(\tau) = \frac{1}{n} \sum_{i=1}^{n} l^i(\tau). \tag{8}$$

Here "$|\cdot|$" denotes the absolute value operator. The extracted power traces from both two tails of the overall normal distribution are shown in Fig. 1. Due to the small noise, the leakage distribution of intermediate values can be well distinguished. For simplicity, we collectively name these power trace extractors as "TAILS". The extracted values occupy four-fifths of the power consumption region in Fig. 1. They have the largest distances from the mean value $\bar{\mathcal{M}}(\tau)$ when extracting half of the samples. According to the variance of the estimated data-dependent component:

$$\sigma_d^2(\tau) = \frac{1}{n-1} \sum_{i=1}^{n} \left(l^i(\tau) - \bar{\mathcal{M}}(\tau) \right)^2, \tag{9}$$

they are also with the largest $\sigma_d^2(\tau)$. It is worth noting that the effect of the noise component is ignored in the estimation, thus it is biased. In other words, TAILS improves SNR through enlarging $\sigma_d^2(\tau)$.

TAILS is a blind recognition algorithm based on distribution of power consumption, while the extractor given by Hu et al. in [15] was based on the distribution of Hamming weights. They selected two candidates of a sub-key with the largest correlation coefficients in CPA, and exploited them to encrypt the plaintexts having intermediate values 0, 1, 7 and 8. They then launched attacks on the newly captured power traces together with the original power trace set independently and achieved a higher success rate that TAILS. In fact, their extractor is to extract the Hamming weights with largest variance from both two tails of Hamming weight distribution. We can simply draw a conclusion that σ_d^2 linearly depends on $\sigma^2(\mathsf{HW}(\mathcal{Z}))$ (i.e., $\sigma_d^2 \propto \sigma^2(\mathsf{HW}(\mathcal{Z}))$ under very small noise. If these traces can be well recognized, the variance of data-dependent component $\sigma^2(\mathsf{HW}(\mathcal{Z}))$ improves from 2.0078 to 10.3529 in leakage model provided by Eq. 6.

Noisy samples corresponding to Hamming weights close to 4 also appear at both two tails of the distribution if SNR is low. Similarly, samples corresponding to smaller or larger Hamming weights may also appear in the middle of the distribution. This will seriously affect the power trace extraction of TAILS. Obviously, Hamming weight based extractor is

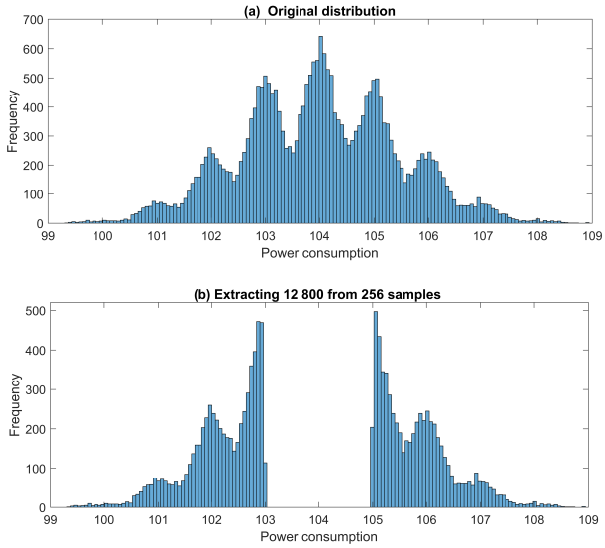

Figure 1: Exploiting the scheme of Kim et al. to extract 12, 800 from 25, 600 power traces.

more accurate in this case. It is worth noting that the purposes of maximizing the variance of Hamming weights and maximizing the variance of power consumption $\sigma^2(\mathcal{L}(\tau))$ are both to maximize the variance of data-dependent power consumption $\sigma_d^2(\tau)$. In other words, the principles of the existing extractors are the same, and we will not strictly distinguish them in this paper. However, the Hamming weight based extractor given by Hu et al. in [15] introduces additional power traces with high SNR to improve the SNR of the combined power trace set, and is orthogonal to our original goal. Thus, we do not consider it in our paper.

3 Blind Recognition of Noise

3.1 More accurate SNR Description

This paper considers extracting the power traces with high SNR for more general differential power attacks [20] (e.g. CPA introduced in Section 2.2), each time sample is considered separately when extracting power traces. Therefore, they have a mark τ. Here we push the Eq. 3 provided in [23] a step further. Since the data-dependent component $\mathcal{L}_d(\tau)$ and operation-dependent component $\mathcal{L}_o(\tau)$ are independent, Eq. 3 can be further expressed as:

$$\hat{\mathsf{SNR}}(\tau) = \frac{\sigma^2(\mathcal{L}_o(\tau)) + \sigma^2(\mathcal{L}_d(\tau))}{\sigma^2(\mathcal{L}_n(\tau))}. \tag{10}$$

For a time sample $\mathcal{L}(\tau)$, its operation on all power traces is usually the same. In this case, the variance of operation-dependent power consumption $\sigma^2(\mathcal{L}_o(\tau)) = 0$. SNR can be further simplified as:

$$\hat{\mathsf{SNR}}(\tau) = \frac{\sigma^2(\mathcal{L}_d(\tau))}{\sigma^2(\mathcal{L}_n(\tau))} \tag{11}$$

(i.e., $\sigma_d^2(\tau)/\sigma_n^2(\tau)$). The Eq. 11 indicates that there are two ways to improve SNR, one is noise reduction, and the other is to enlarge $\sigma_d^2(\tau)$ as we introduced in Section 2.3. The

existing works adopt the second approach. In this section, we adopt the first approach to improve SNR, and show that the new extractor is more superior.

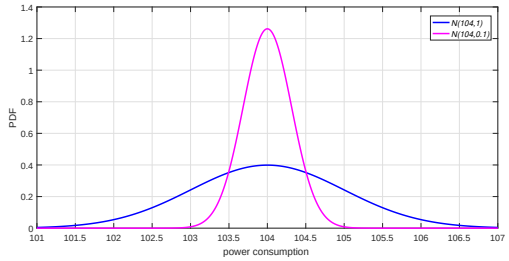

Figure 2: Probability density function of $\mathcal{N}(104, 1)$ and $\mathcal{N}(104, 0.10)$.

Taking two normal distributions with mean $\mu = 104$ and variance $\sigma_n^2 = 1$ and $\sigma_n^2 = 0.1$ shown in Fig. 2 for an example, most of samples in distribution $\mathcal{N}(104, 0.1)$ lie closer to μ compared with $\mathcal{N}(104, 1)$. Since the samples with the largest variance are at both tails, if the same number of samples are extracted from the middle of distributions, the samples corresponding to $\mathcal{N}(104, 0.1)$ have a smaller variance. Let's give an intuitive example using the 25, 600 samples in our simulation with $\sigma_n^2 = 0.09$. We rank the samples according to their noise component, and select 12, 800 samples with minimum noise from them (as shown in Fig. 3). These time samples are very close to the mean power consumption of their corresponding Hamming weights. There are many more samples with low noise in the middle than two tails. Therefore, noise can be reduced by extracting samples close to their mean values. Moreover, reducing σ_n^2 has more potential to improve SNR than enlarging σ_d^2. The SNR of the power traces extracted by this is increased by more than 7 times, compared with less than 2 times in TAILS. The noise of the extracted power traces can be infinitely small in theory if there are enough samples to extract. This will be further introduced in Section 4.3.

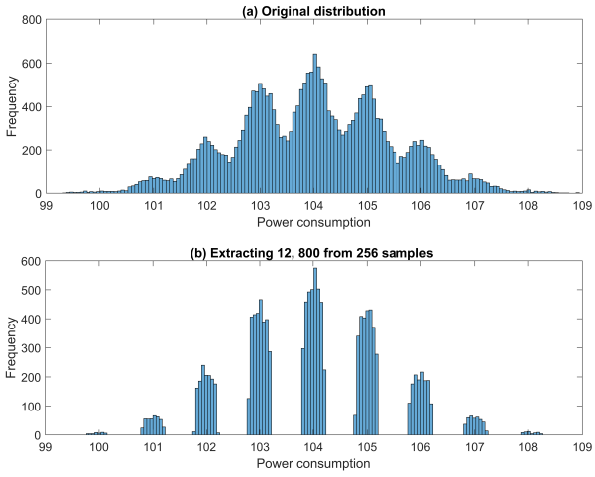

Figure 3: Power traces extracted from the middle of each Hamming weight distribution.

3.2 Optimal Extractor

The optimal extractor if only the noise components $\mathcal{L}_n(\tau)$ is taken into consideration is that, the power traces are ranked according to their noise from small to large. This requires the knowledge of key and is only exploitable as an important reference for power trace extraction. Mean power consumption $\bar{\mathcal{T}}^{hw_{x^i}}$ of the corresponding Hamming weight of a plaintext byte value x^i $(1 \leq i \leq n)$ is also required in this case, extracting $n^{'}$ power traces can ensure that their noise variance is minimum. In this case, the extractor is a profiled attack and we name it IDEAL. We randomly generate 100 samples from the leakage model provided in Eq. 6 with $\sigma_n^2 = 30$, and extract parts of samples with the smallest noise from them. Most papers only exploited the result of a single CPA attack to compare the correlation coefficients of different sub-key candidates, which was not accurate enough. Since the correlation coefficient of each candidate under different experiments may be very different, we exploit the average correlation coefficients to measure their variations with the increasing number of power traces (as shown in Fig. 4(a)). The corresponding results of CPA are also given, all samples are randomly extracted except for the optimal extractor.

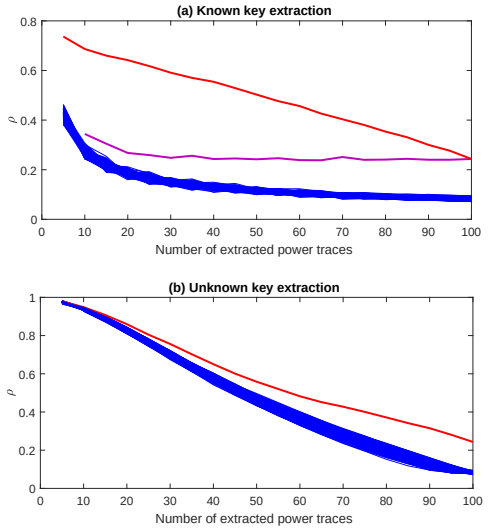

Figure 4: Correlation coefficients under random and optimal extractors.

The mean power consumption $\bar{\mathcal{T}}^{hw_{x^i}}$ linearly correlates to the power consumption. The correlation coefficient will be very high if we perform CPA on these traces and the variance of noise approaches 0. The one corresponding to the correct sub-key in CPA with random power trace extraction is significantly higher than others, but still much lower than the one corresponding to samples with the smallest noise (the upper two curves in Fig. 4(a)). The correlation coefficient of IDEAL (red curve) decreases gradually with the increasing number of power traces extracted. Finally, it is consistent with the one corresponding to classic CPA when exploiting all traces. In this case, the SNR of the extracted samples monotonically decreases (as shown in Fig. 5(a)).

With the knowledge of key, the noise of power traces can be accurately estimated according to the means $\bar{\mathcal{T}}^{hw_{x^i}}$, and the ones with the highest SNR can be extracted. In this case, the optimal extractor IDEAL can be a profiled attack and these means can be exploited from the templates to achieve higher accuracy. However, this is impossible in the case of unknown key. In other words, it is impossible to exploit the above optimal power trace extractor IDEAL in attacks as we have explained before. The power traces

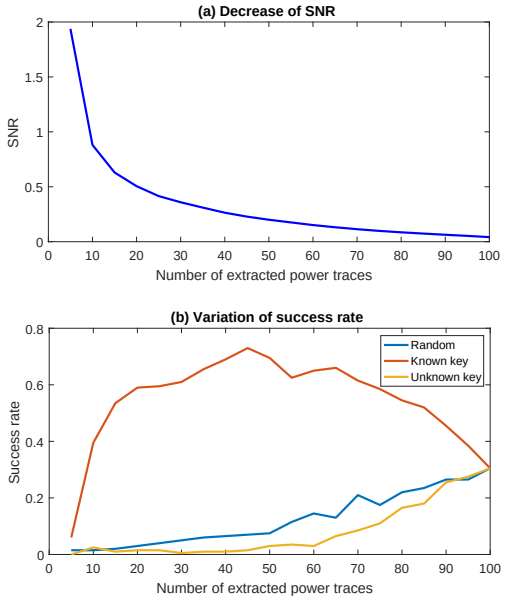

Figure 5: SNR and success rate under random and optimal extractors.

close to these "templates" may still have large noise under the known key. However, as long as a sample has power consumption close enough to a "template" under the case of unknown key, it will be extracted to launch attacks even if it is wrongly matched. In other words, each guessing sub-key will "frantically" extract the samples that are most beneficial to itself. This will result in very high linear correlation between the means (or templates) and the power traces they extract, and obtaining correlation coefficients even significantly higher than the one corresponding to the correct sub-key under the known key case (as shown in Fig. 4(b)). Thus, it is impossible to distinguish the correct key.

The success rate may be even worse than that of random extraction if IDEAL is exploited to extract power traces for each candidate in the case of unknown key (as shown in Fig. 5(b)). However, we only achieve a success rate about 0.31 under 100 power traces, the optimal extractor IDEAL achieves the same success rate by only extracting 10 power traces. Although it is impossible to build such a good extractor in reality, IDEAL tells us that with more accurate SNR estimation, success rate can be significantly improved. In other words, it provides a reference for us to build a highly accurate extractor. In the next sections, we will explore how to build such an efficient extractor without knowing the key and make its performance close to IDEAL in the known-key case.

3.3 Shortest Distance First Extractor

For evaluators, leakage detection [26] and assessment [29] are to determine whether a cryptographic device leaks information and evaluate the corresponding security level, respectively. Compared with the attackers, they have more advantages such as the knowledge of the key, and the ability to capture a very large number of power traces. In this case, the power traces with small noise are easy to accurately estimate and extract. However, the correct key is unknown in the actual attacks, how to improve the accuracy of power trace extraction to approach the success rate limit of optimal extractor IDEAL shown in Fig. 5(b) has become an important problem to be solved.

The existing power trace extractors only consider the case that the number of power

traces owned by the attacker is sufficient to guarantee his successful attacks, and the goal of his extraction is to lower the complexity rather than increase the success rate. Since it is difficult for them to achieve a success rate higher than the attack directly performed on the original power trace set. We have also explained in Section 3.2 that if we exploit profiled Hamming-weight templates to estimate the noise component of samples and extract a part that best match templates from them, the correlation coefficients of all the candidates are very high, so that profiled attacks cannot be exploited. Here we focus on the most common implicit assumption of the known plaintexts in side-channel society, and take the second-best to compute the mean power consumption of each plaintext byte value for power trace extraction. In this case, although these means may not be so accurately estimated when the power traces are limited, they are a good reference.

Our Shortest Distance First (SDF) extractor is very simple (see Algorithm 1). Suppose that we employ it to extract $n^{'}$ power traces with minimum estimated noise from the n traces $\mathcal{L}(\tau) = \left(l^1(\tau), l^2(\tau), \ldots, l^n(\tau)\right)$ leaked at time τ. SDF keeps a counter $\mathcal{C}$ to record the current number of power traces of 256 plain-text byte values (Step 2). The corresponding power consumption $l^i(\tau)$ is added to $\mathcal{M}^{x^i}$ (Step 3). Here $\mathcal{M}^{x^i}$ denotes the mean power consumption of plaintext byte value x^i. We then average $\mathcal{M}$ and get the mean power consumption vector of all 256 plain-text byte values after traversing all power traces (Step 5).

Here, we need to clarify the difference between profiled and non-profiled attacks. The former means that the attacker has the same device as the targeted one. He can collect as many power traces as he wishes to profile accurate enough templates and exploits them to launch attacks on the targeted device. The classic profiled attack is TA as we introduced in Section **??**. Non-profiled attacks such as classic CPA and DPA, do not have such a profiling stage. The power traces are employed to not only estimate the mean power consumption but also launch attacks in SDF. We can also see the same power traces from $\mathcal{L}(\tau)$ employed in Steps 3 and 7 of Algorithm 1. They are all from the targeted device, and no additional power traces are introduced when estimating the means. In this case, SDF is still a non-profiled attack.

Algorithm 1: Shortest Distance First (SDF) extractor.

Input: Power traces $\mathcal{L}(\tau)$, plain-texts $\mathcal{X}$ and the number of traces to be extracted $n^{'}$.

Output: The new power trace set $\mathcal{L}^{'}(\tau)$ and their plain-texts $\mathcal{X}^{'}$.

1 **for** *i from 1 to n* **do**
2 $\quad \mathcal{C}^{x^i} \leftarrow \mathcal{C}^{x^i} + 1$;
3 $\quad \mathcal{M}^{x^i} \leftarrow \mathcal{M}^{x^i} + l^i(\tau)$;
4 **end**
5 $\mathcal{M} \leftarrow \mathcal{M}/\mathcal{C}$;
6 **for** *i from 1 to n* **do**
7 $\quad \mathcal{D}^i \leftarrow \left| l^i(\tau) - \mathcal{M}^{x^i} \right|$;
8 **end**
9 $(\mathcal{T}, \mathcal{P}) \leftarrow \mathsf{AscendSort}(\mathcal{L}, \mathcal{X}, \mathcal{D})$;
10 $\mathcal{L}^{'}(\tau) = \mathcal{T}^{1\ldots n^{'}}(\tau)\,; \mathcal{X}^{'} = \mathcal{P}^{1\ldots n^{'}}$;

The noise component of a sample $l^i(\tau)\,(1 \le i \le n)$ in power trace set $\mathcal{L}$ can be blindly estimated as:

$$\mathcal{D}^i = \left| l^i(\tau) - \mathcal{M}^{x^i} \right| \tag{12}$$

rather than the mean $\bar{\mathcal{M}}(\tau)$ of all samples in Eq. 7 (see Steps 6 $\sim$ 8). Here vector $\mathcal{D}$ is employed to save the distances. We sort the estimated noise components in ascending

order and get the corresponding re-ranked power trace set $\mathcal{T}$ and plaintext byte values $\mathcal{P}$ (Step 9). Finally, $n^{'}$ traces with smallest distances (i.e. smallest estimated noise component) and their corresponding plaintexts are extracted from the first $n^{'}$ elements of $\mathcal{T}$ and $\mathcal{P}$ (Step 10). In this case, the estimated SNR of the extracted power traces in $\mathcal{L}^{'}$ can be expressed as:

$$\hat{\mathsf{SNR}}(\tau) = \frac{\sum_{i=1}^{n^{'}} \left(\bar{\mathcal{T}}^{hw_{x^i}} - \bar{\mathcal{M}}(\tau)\right)^2}{\sum_{i=1}^{n^{'}} \left(l^i(\tau) - \bar{\mathcal{T}}^{hw_{x^i}}\right)^2}, \tag{13}$$

since SNR estimation can exploit the knowledge of key and the well profiled templates. Taking the Hamming weight model given in Eq. 6 for an example, the values of templates can use 0, 1, ..., 7 and 8 directly. Since infinite power traces can be exploited to profile sufficiently accurate templates in theory. However, the leakage model is unknown in the real leakage, so we can only employ a large number of power traces in profiling to make the templates as accurate as possible. More advanced profiling technology like leakage certification [5, 9], can also be exploited in this case.

It is worth mentioning that the extraction is performed on all rather than only one time sample on power traces, and its target is to launch more efficient attacks. In other words, Algorithm 1 implements on each time sample. The same extractor may extract very different power traces on different time samples. However, this does not go against the purpose of the extractors. Since they aim to maximize the SNR of the power traces extracted from each time sample to achieve the optimal attack from them. The overall optimal attack performance is achieved by combining these single attacks. Therefore, we use time stamp τ to indicate the location of leakage in almost all expressions.

4 Bind Estimation of SNR

There have two ways to improve SNR according to Eq. 11: one is noise reduction as we introduced in Section 3, and the other is to enlarge the variance of data-dependent power consumption $\sigma^2(\mathcal{L}_d(\tau))$. Here we further attempt to enlarge $\sigma^2(\mathcal{L}_d(\tau))$ of the extracted power traces on the basis of the noise reduction introduced in Section 3. Compared to only consider the noise component, blindly estimating the SNR requires the estimation of data-dependent component (see Eq. 11). Therefore, SNR estimation will be more accurate than only considering noise estimation.

4.1 Maximizing Estimated SNR

The data-dependent power consumption of a sample $l^i(\tau)$ can be roughly quantified using Eq. 7. Maximizing the variance of data-dependent power consumption, is equivalent to maximizing $\mathcal{D}^i(\tau)$. In other words, the larger it is, the greater the data-dependent power consumption of the sample. This corresponds to TAILS that extracts samples from two tails of the total distribution introduced in Section 2.3.

We aim to maximize the SNR, that is, to minimize the noise while maximizing the variance of the data-dependent component. In this case, the evaluation function given in Step 7 of Algorithm 1 can be expressed as:

$$\mathcal{D}^i(\tau) = \frac{\left|l^i(\tau) - \bar{\mathcal{M}}(\tau)\right|}{\left|l^i(\tau) - \mathcal{M}^{x^i}\right|}. \tag{14}$$

We can see that Eq. 14 is an important model for us to estimate SNR. Since the variance of data-dependent power consumption can be estimated as $\sigma^2(\mathcal{L}_d(\tau)) = \frac{1}{n-1}\sum_{i=1}^{n}\left(l^i(\tau) - \bar{\mathcal{M}}(\tau)\right)^2$ as explained in TAILS, and variance of noise can be estimated as $\sigma^2(\mathcal{L}_n(\tau)) = \frac{1}{n-1}\sum_{i=1}^{n}\left(l^i(\tau) - \mathcal{M}^{x^i}\right)^2$,

maximizing σ_d^2 while minimizing σ_n^2 can be achieved by maximizing $\mathcal{D}(\tau)$. We name this new power trace extractor as Maximizing Estimated SNR First (MESF).

4.2 Mean Optimized MESF

This paper considers the case that the number of power traces is enough to guarantee the successful attack. On one hand, it is not very good to directly exploit $l^i(\tau)$ containing noise to estimate data-dependent power consumption, which will introduce significant but unnecessary estimation error caused by noise. On the other hand, the attacker may do not have enough power traces to accurately measure mean power consumption for each plain-text byte value. In this case, the accuracy of estimation of the data-dependent component and noise component will also be significantly affected. This estimation error aggravates in the case of very large noise. However, compared with $l^i(\tau)$, the mean power consumption of each plaintext byte value is still more accurate and theoretically more reasonable in measuring data-dependent power consumption.

Overall, a better strategy is to exploit the mean power consumption of each plaintext byte value when measuring their data-dependent component. In this case, Eq. 14 can be further optimized as:

$$\mathcal{D}^i(\tau) = \frac{\left|\mathcal{M}^{x^i} - \bar{\mathcal{M}}(\tau)\right|}{\left|l^i(\tau) - \mathcal{M}^{x^i}\right|}. \tag{15}$$

We name this new power trace extractor as Mean optimized MESF (MMESF). In fact, all estimates of noise and data-dependent power consumption in Eqs. 7, 14 and 15 are in a blind way. Since these extractors rely on different mathematical principles, their accuracy may vary with a different number of power traces extracted and different noise levels, which will be illustrated in our experiments.

4.3 Comparison

Our three extractors SDF, MESF and MMESF, have great differences in the extracted power traces. Their results of extracting 12, 800 from 25, 600 power traces in Fig. 1 are shown in Fig. 6. The SNR of the 25, 600 samples generated by the leakage model in Eq. 6 is estimated to be 22.0781 when the noise level $\sigma_n^2 = 0.09$, approaching the theoretical value 22.3049. SDF is designed to minimize the estimated noise component, and its SNR is about 146.9011, approaching the upper limit 155.8411 of the optimal extractor IDEAL under known key. It can also be seen from Figs. 3 and 6 that the traces extracted by them are similar to the optimally extracted ones, which fully illustrates their advantages under small noise.

The power traces extracted by MESF and MMSEF are quite different from those extracted by SDF. The distribution of samples extracted by MESF appears to be small in the middle and large at both two ends. These samples are not concentrated at both two tails like TAILS, since MESF benefits from the advantages of both SDF and TAILS. The SNR of the extracted samples corresponding to MESF is 40.8126. Compared with MESF, MMESF exploits the mean power consumption of the corresponding intermediate value of each plaintext byte value to measure the data-dependent component, which is more accurate and more reasonable in theory. Both MESF and MMESF benefit from the data-dependent power consumption component, but MMESF benefits more, and the samples it extracts mainly distribute at both two tails. Moreover, these samples first concentrate on the mean power consumption of Hamming weights, and then spread to both ends. The SNR of the samples extracted by it is 88.5545, which is more than double that of MESF.

The SNR of the samples extracted by the above three extractors is significantly higher than that of TAILS, which is only about 38.6273 in this experiment. This fully illustrates that they have effectively recognized the power traces with high SNR in different

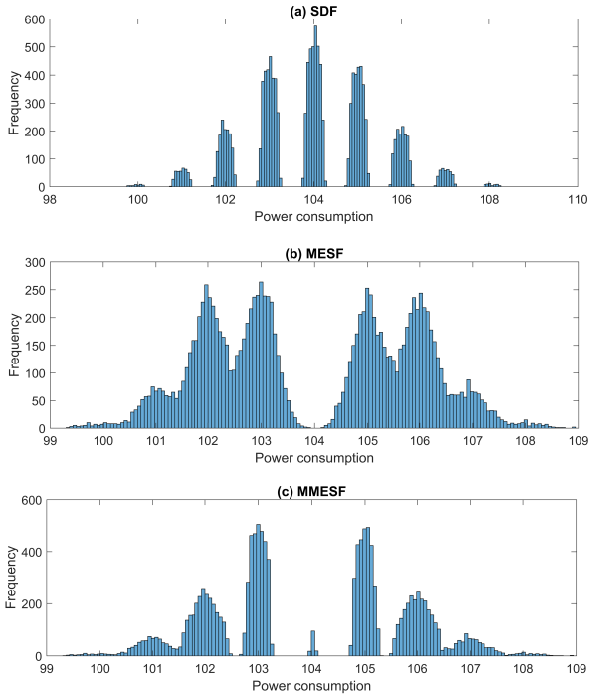

Figure 6: The samples extracted by different extractors.

degrees, thus improving the success rates. MESF and MMESF achive SNR lower than SDF, but this does not mean that they are worse than SDF. These extractors will be further discussed in Section 5.

5 Experimental Results

5.1 Simulations

To compare the performance of extractors under different noise levels, our first experiment is performed on simulated power traces. They are randomly generated from the leakage model provided in Eq. 6. We firstly compare the performance of extractors under a different number of power traces under the noise level $\sigma_n^2 = 400$. Due to the large noise, too many traces are required for attacks, which makes the evaluation time-consuming. Therefore, we only repeat the above experiments 200 times, and the corresponding experimental results are shown in Fig. 7. To guarantee a stable success rate (SR) about 1.00 for classic CPA, nearly 6 000 traces are required in the original sets. All extractors TAILS, SDF, MESF and MMESF achieve higher success rates than random extraction and even close to the ideal extraction in Figs. 7(b) and 7(c), which correspond to the case that the attacker has sufficient traces. SDF and MMESF achieve success rates 1.00 even under the poor condition that only 300 traces are extracted, which fully demonstrates their superiority.

The SNR corresponding to MESF and TAILS is the lowest in Fig. 7, and only a tiny change occurs when extracting more power traces. They are designed to maximize the variance of the data-dependent power consumption component rather than reducing the noise on the samples. The SNR of power traces extracted by SDF and MMESF decreases monotonically, both success rate and it are higher when the original sets have more power traces (see Figs. 7(a), 7(b) and 7(c)). This indicates: 1) they can still recognize the power

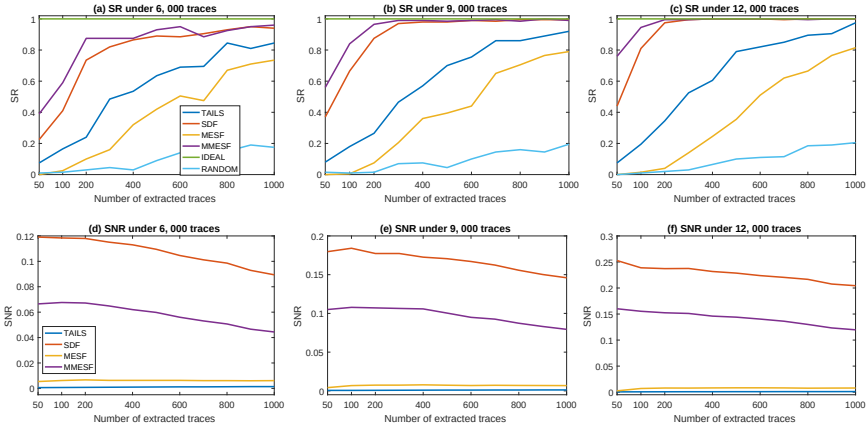

Figure 7: Success rates and SNR (of the extracted traces) under different number of power traces.

traces with high SNR even in the case of very large noise, and 2) SNR has become an important factor determining the success rate. The performance of our extractors SDF and MMESF is also more stable, and the areas with high success rates are significantly wider than that of TAILS. This makes it much easier for attackers to set a reasonable parameter of the number of power traces to extract.

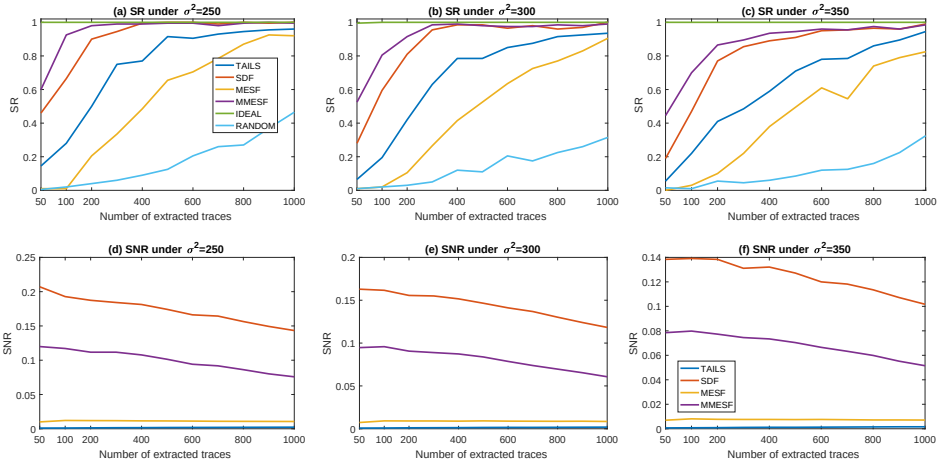

Figure 8: Success rates and SNR (of the extracted traces) under different noise levels.

Noise has a great influence on the estimation of mean power consumption of each plaintext byte value, and reduces the accuracy of power trace extraction. Therefore, it is necessary to further discuss its impact on the success rate. In this case, we also consider the success rate and the SNR of the extracted power traces under different noise levels $\sigma_n^2 = 350$, $\sigma_n^2 = 300$ and $\sigma_n^2 = 250$, the corresponding experimental results are shown in Fig. 8. Here, 6000 power traces are randomly generated for extraction, and each experiment is repeated 200 times. With the decrease of noise level, both success rates and the accuracy of all extractors increase. SDF and MMESF still achieve the highest

success rates. Although the success rates of TAILS and MESF do not reach such a height when they extract a small number of samples, they increase rapidly when more power traces are extracted. MESF and TAILS not only change the SNR, but also increase the $\hat{\rho}\left(\mathcal{M}\left(\tau\right),\mathcal{L}_d\left(\tau\right)\right)$ in Eq. 5. Their success rates benefit from at least these two factors. It is worth mentioning that the relationship between success rate and the noise is much more complex than this, more details can be seen in [10, 36].

The success rate of classic CPA without extraction is about 1.00 under $\sigma_n^2 = 250$ when randomly generating 4000 power traces to perform attacks. SDF and MMESF achieve success rates even about 1.00 when only extracting 400 traces. If we launch the attacks on more traces, the success rates will be stable thus guaranteeing the key recovery. However, the extractors significantly reduce the number of traces used for attacks thus significantly lowering the computational complexity and highlighting their significance.

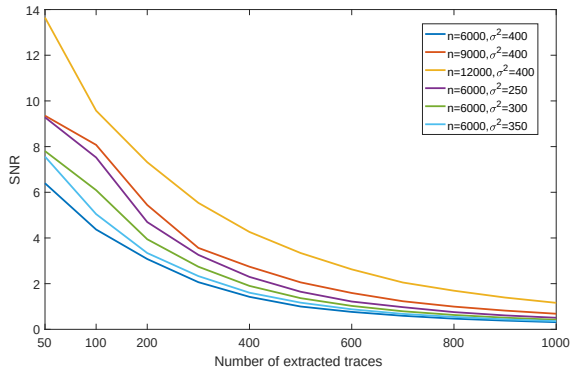

Figure 9: SNR of the traces extracted by IDEAL under different number of power traces and different noise levels.

SNR of the traces extracted by IDEAL under different number of power traces and different noise levels is also shown in Fig. 9. With the increase in the total number of power traces, we can obtain higher SNR if extracting the same proportion from them. On one hand, more traces make the estimation of mean power consumption more accurate. On the other hand, the number of traces with small noise also increases. The decrease of noise level from $\sigma_n^2 = 400$ to $\sigma_n^2 = 350$, $\sigma_n^2 = 300$ and $\sigma_n^2 = 250$ also makes the mean power consumption more accurate under fewer power traces. Unlike the results given in Section 4.3, IDEAL achieves significantly higher SNR than all other extractors. However, SDF and MMESF achieve success rates close to it. They are more accurate compared with other extractors.

5.2 Experiments on ATmega328p Micro-controller

Our second experiment is performed on an ATMega328p micro-controller implementing the unprotected AES-128 algorithm provided by [1]. The clock operating frequency of this micro-controller is 16 MHz. We encrypt 300, 000 plaintexts and exploit a WaveRunner 8104 oscilloscope to acquire the power traces. The sampling rate is set to 1 GS/s. We perform classic CPA on 10, 000 power traces to extract a time sample with SNR about 0.0016 and correlation coefficient 0.09399 rather than the best POIs of the first S-box in the first round to simulate the very low SNR scenario. Actually, we will achieve better performance if extractions are directly performed on better POIs. Although the components of power consumption are more complex than those simulated in actual attacks, we can still draw conclusions similar to these in Section 5.1 (as shown in Fig. 10). However,

all extractors still have monotonically increasing or decreasing SNR as shown in Fig. 7 and 8.

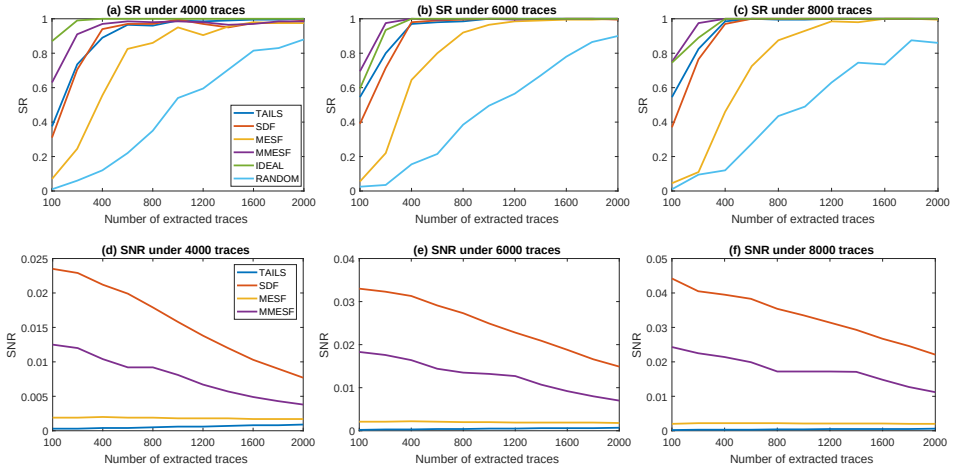

Figure 10: Success rates and SNR (of the extracted traces) under different number of traces.

We exploit 1 000 power traces acquired to profile the mean power consumption of each Hamming weight to estimate the SNR more accurately. We then randomly select 4, 000, 6, 000 and 8, 000 power traces from them for extraction. This operation is repeated 200 times, and the corresponding success rates are about 0.990, 1.00 and 1.00. In this case, Fig. 10(a) corresponds to the case of insufficient power traces, and Figs. 10(b) and 10(c) correspond to the case of sufficient power traces. Extracting a small number of power traces can significantly improve the SNR. Almost all extractors achieve a success rate of 1.00 when extracting 400 power traces. Therefore, they significantly lower the complexity compared with the attacks directly performed on the original set.

The experimental results vividly show us that all extractors achieve success rates significantly higher than random extraction. Our SDF almost achieves the same success rates compared with TAILS, while MMESF achieves success rate even higher than IDEAL. This fully illustrates the accuracy of MMESF to exploit the mean power consumption of each plaintext byte value when estimating the data-dependent component. MMESF outperforms SDF because of this component. These means $\mathcal{M}^{x^i}\left(x^i = 0, 1, \ldots, 255\right)$ are more accurately estimated with the increasing number of power traces, which makes the extraction performance better. They can extract better power traces than other extractors and achieve the highest success rates in most cases. Similar conclusions can also be drawn from Section 5.1.

From the aspect of SNR, SDF is superior to MMESF. This indicates that it still accurately extracts the power traces with high SNR even under very large noise. However, SDF ignores the data-dependent power consumption component, thus making its $\hat{\rho}\left(\mathcal{M}\left(\tau\right), \mathcal{L}_d\left(\tau\right)\right)$ smaller than MMESF's. Therefore, it achieves a success rate lower than MMESF's. The SNR of MESF and TAILS is always much lower than that of MMESF. It is worth noting that the SNR of MMESF and SDF decreases when more traces are extracted, just contrary to the SNR of MESF and TAILS. Although this growth is not obvious and even difficult to observe. Similar conclusions can also be drawn from Figs. 7 and 8. On one hand, compared with the extraction enlarging the data-dependent power consumption component, reducing the noise can improve the SNR of the extracted power traces to a greater extent. On the other hand, this illustrates that SNR is only one of the

important factors to determine the success rate. For example, the SNR improves with the increasing number of power traces in the original set. This is also an important factor.

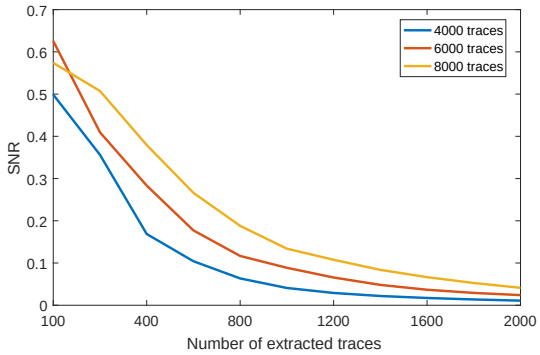

Figure 11: SNR of the traces extracted by IDEAL under different number of power traces and different noise levels.

SNR of traces extracted from the original set with 4, 000, 6, 000 and 8, 000 traces using IDEAL is shown in Fig. 11. Compared with the simulated power traces, the composition of real leakage is much more complicated. The extractors achieve SNR about $10 \sim 15$ times lower than the ideal extraction (as shown in Figs. 10 and 11). This gap decreases rapidly with a larger proportion of power traces extracted. In terms of success rate, these extractors have achieved outstanding results, but there is still a lot of room for improvement.

6 Conclusions

The power trace extractors are suitable to reduce the computational complexity when the power traces are sufficient, and the existing ones simply extract samples with the largest estimated variance of exploitable power consumption to improve the SNR. Although having strict theoretical proof, they are too simple and leakage characteristics of POIs have not been thoroughly analyzed. These limits their accuracy on power trace extraction. In this paper, we deeply analyze the SNR of power traces in theory, and propose a novel extractor named SDF to extract the power traces with the smallest estimated noise. SDF can achieve SNR and success rate significantly higher than the existing extractors when the same number of power traces are extracted. Thus, it has wider application sceneries, and opens up a new way for power trace extraction.

To maximize the exploitable component while minimizing the noise, we further propose a novel extractor named MESF. TAILS directly exploits the distance between single sample and population mean as a reference of its data-dependent component. This measurement is with noise and not very good. Therefore, we further exploit the mean power consumption of each plaintext byte value to estimate the exploitable power consumption of the corresponding samples, and propose a more advanced extractor named MMESF, which is also more reasonable in theory. The above three extractors show their higher accuracy in extraction. They significantly reduce the computing complexity compared with the existing extractors, which fully illustrates their superiority.

All estimates in this paper are based on the estimated mean power consumption of plaintext byte values. This limits our extractors to the scenarios where the attacker has a large number of power traces to estimate the means. We leave the introduction of profiled attacks into extractors as an open problem, and believe this will significantly improve

their performance. We will also theoretically analyze relationship between success rate and the noise given in [10,36], and introduce it into the extractors to enhance power trace extraction in our future work.

References

[1] Avr-Crypto-Lib. https://github.com/DavyLandman/AESLib.

[2] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side-Channel(s). In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 29–45, 2002.

[3] A. C. Aldaya, C. P. García, L. M. A. Tapia, and B. B. Brumley. Cache-Timing Attacks on RSA Key Generation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(4):213–242, 2019.

[4] E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with A Leakage Model. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 16–29, 2004.

[5] O. Bronchain, J. M. Hendrickx, C. Massart, A. Olshevsky, and F. Standaert. Leakage Certification Revisited: Bounding Model Rrrors in Side-Channel Security Evaluations. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, pages 713–737, 2019.

[6] E. Cagli, C. Dumas, and E. Prouff. Kernel Discriminant Analysis for Information Extraction in the Presence of Masking. In *Smart Card Research and Advanced Applications - 15th International Conference, CARDIS 2016, Cannes, France, November 7-9, 2016, Revised Selected Papers*, pages 1–22, 2016.

[7] S. Chari, J. R. Rao, and P. 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.

[8] F. Durvaux and F. Standaert. From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces. In *35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2016. Proceedings, Part I*, pages 240–262, 2016.

[9] F. Durvaux, F. Standaert, and S. M. D. Pozo. Towards Easy Leakage Certification. In *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, pages 40–60, 2016.

[10] Y. Fei, A. A. Ding, J. Lao, and L. Zhang. A Statistics-Based Success Rate Model for DPA and CPA. *J. Cryptographic Engineering*, 5(4):227–243, 2015.

[11] D. Genkin, I. Pipman, and E. Tromer. Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs. In *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, pages 242–260, 2014.

[12] D. Genkin, A. Shamir, and E. Tromer. RSA Key Extraction Via Low-Bandwidth Acoustic Cryptanalysis. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 444–461, 2014.

[13] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual Information Analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, pages 426–442, 2008.

[14] S. Hajra and D. Mukhopadhyay. On the Optimal Pre-processing for Non-Profiling Differential Power Analysis. In *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, pages 161–178, 2014.

[15] W. Hu, L. Wu, A. Wang, X. Xie, Z. Zhu, and S. Luo. Adaptive Chosen-Plaintext Correlation Power Analysis. In *Tenth International Conference on Computational Intelligence and Security, CIS 2014, Kunming, Yunnan, China, November 15-16, 2014*, pages 494–498, 2014.

[16] Y. Kim and H. Ko. Using Principal Component Analysis for Practical Biasing of Power Rraces to Improve Power Analysis Attacks. In *Information Security and Cryptology - ICISC 2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers*, pages 109–120, 2013.

[17] Y. Kim, T. Sugawara, N. Homma, T. Aoki, and A. Satoh. Biasing Power Traces to Improve Correlation Power Analysis Attacks. In *Constructive Side-Channel Analysis and Secure Design - COSADE 2010 - First International Workshop, Daemstadt, Germany, February 4-5, 2010*, pages 77–80, 2010.

[18] P. 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.

[19] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 388–397, 1999.

[20] P. C. Kocher, J. Jaffe, B. Jun, and P. Rohatgi. Introduction to Differential Power Analysis. *J. Cryptographic Engineering*, 1(1):5–27, 2011.

[21] T. Le, J. Clédière, C. Servière, and J. Lacoume. Noise Reduction in Side Channel Attack Using Fourth-Order Cumulant. *IEEE Trans. Information Forensics and Security*, 2(4):710–720, 2007.

[22] F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee. Last-Level Cache Side-Channel Attacks Are Practical. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 605–622, 2015.

[23] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Springer, 2007.

[24] D. Mavroeidis, L. Batina, T. van Laarhoven, and E. Marchiori. PCA, Eigenvector Localization and Clustering for Side-Channel Attacks on Cryptographic Hardware Devices. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I*, pages 253–268, 2012.

[25] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Trans. Computers*, 51(5):541–552, 2002.

[26] A. Moradi, B. Richter, T. Schneider, and F. Standaert. Leakage Detection with the χ^2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.

[27] B. Noura, M. Mohsen, and T. Rached. Optimized Power Trace Numbers in CPA Attacks. In *The 8-th International Multi-Conference on Systems, Signals and Devices, Sousse, Tunisia, March 22-25, 2011. Proceedings*, pages 1–5, 2011.

[28] C. Ou, Z. Wang, D. Sun, X. Zhou, J. Ai, and N. Pang. Enhanced Correlation Power Analysis by Biasing Power Traces. In *Information Security - 19th International Conference, ISC 2016, Honolulu, HI, USA, September 3-6, 2016, Proceedings*, pages 59–72, 2016.

[29] O. Reparaz, B. Gierlichs, and I. Verbauwhede. Fast Leakage Assessment. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 387–399, 2017.

[30] H. Shimada, Y.-i. Hayashi, N. Homma, T. Mizuki, T. Aoki, and H. Sone. Using Selected-Plaintext Sets for Efficient Evaluation of EM Information Leakage from Cryptographic Devices. In *2012 Proceedings of SICE Annual Conference, SICE 2014, Akita University, Akita, Japan, August 20-23, 2012. Proceedings*, pages 64–67. IEEE, 2012.

[31] Y. Souissi, M. Nassar, S. Guilley, J. Danger, and F. Flament. First Principal Components Analysis: A New Side Channel Distinguisher. In *Information Security and Cryptology - ICISC 2010 - 13th International Conference, Seoul, Korea, December 1-3, 2010, Revised Selected Papers*, pages 407–419, 2010.

[32] F. Standaert. How (not) to Use Welch's t-test in Side-Channel Security Evaluations. *IACR Cryptology ePrint Archive*, 2017:138, 2017.

[33] F. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 443–461, 2009.

[34] K. Tiri and P. Schaumont. Changing the Odds Against Masked Logic. In *Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers*, pages 134–146, 2006.

[35] J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker. Improving Differential Power Analysis by Elastic Alignment. In *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, pages 104–119, 2011.

[36] H. Zhang. On the Exact Relationship between the Success Rate of Template Attack and Different Parameters. *IEEE Trans. Information Forensics and Security*, 15:681–694, 2020.