SNR to Success Rate: Reaching the Limit of Non-Profiling DPA

Suvadeep Hajra
Dept. of Computer Science & Engg.
Indian Institute of Technology, Kharagpur, India
suvadeep.hajra@gmail.com

Debdeep Mukhopadhyay
Dept. of Computer Science & Engg.
Indian Institute of Technology, Kharagpur, India.
debdeep.mukhopadhyay@gmail.com

***Abstract*—Many profiling power analysis attacks estimate the multivariate probability distribution using a profiling step, and thus, can optimally combine the leakages of multiple sample points. Though there exists several approaches like filtering, Principal Component Analysis for combining the leakages of multiple sample points in non-profiling DPA, their optimality has been been rarely studied. We study the issue of optimally combining the leakages of multiple sample points in non-profiling DPA attacks using a linear function. In this work, our contributions are three-fold: 1) we first derive a relation between the success rate of a CPA attack and the SNR of the power traces, 2) we introduce a multivariate leakage model for Virtex-5 FPGA device, and 3) using the proposed multivariate leakage model, we devise linear filters to maximize the SNR of the output leakage which, in turn, optimizes the success rate of the CPA attacks in a non-profiling setup.**

I. Introduction

Differential Power Analysis (DPA) [17] has been proven to be an extremely lethal tool for side-channel analysis. It is highly effective in finding the secret key of a secure device by analysing the power traces of the device, even without knowing the implementation details. One of its strengths comes from its ability to exploit minute data-dependency of leakage by accumulating them over a large number of power traces. Since power traces are the scarce resource, reducing the number of required power traces for a successful DPA attack, or increasing the success rate of a DPA attack using a limited number of power traces has been in the focus of DPA literature since its introduction.

The success rate of the DPA [20] attacks is largely influenced by the Signal-to-Noise Ratio (SNR) [20] of the power traces. As a consequence, in many applications, Power Analysis attacks are preceded by various pre-processing techniques like integration [20], PCA [3], filtering [7], [21] for the reduction of noise in the power traces. These techniques attempt to improve the performance of the DPA attacks directly or indirectly by extracting information from multiple sample points. Some of these techniques like PCA are based on some implicit assumptions, thus optimally applicable to some specific scenarios only, while others deploy some heuristic methods (please refer to Section II-D).

Various profiling attacks like Template attack [6] and Stochastic attack [29] provide optimal performance by jointly evaluating the leakages at multiple sample points. However, they use a separate profiling step for approximating the multivariate leakage distribution [32] of the power traces. The profiling step requires a large number of power traces to estimate the multivariate leakage distribution with sufficient accuracy. Moreover, in most of the cases, it needs the knowledge of the secret key which may not be available in many attacking scenarios.

Principal Component Analysis (PCA) has been introduced as a tool to reduce the size of the sample points in Template attacks [2]. Later in [31], PCA is used as a distinguisher. Recently in [3], Batina et al. have presented it as a pre-processing tool for the reduction of noise in a non-profiling setup. However, it performs better under the assumption that the data-dependent variations is larger than the noise variations. Unfortunately, in side-channel analysis, this assumption does not hold always. Though in [3], Batina et al. have proposed a new distinguisher based on some empirical observation, the performance of such distinguisher is far from being optimal.

Contributions: In this paper, we have studied how to maximize the success rate of a DPA attack by combining the leakages of multiple sample points. We have explored two possible ways of combining: a) combine the leakages of multiple sample points first and then apply a univariate distinguisher on the combined leakage, and b) apply a univariate distinguisher on multiple sample points independently and then combine their outputs. We have further shown that in certain cases both the approaches are equivalent in terms of the success rate of the attack. Next, we have devised an optimal way of combining the leakages of multiple sample points using the following three steps:

1) We derive an exact relation between the SNR of the power traces and the success rate of univariate Correlation Power Analysis (CPA) for arbitrary distribution of plaintext. Thanks to the relation, maximization of the success rate by combining leakages of multiple sample points becomes equivalent to the maximization of the effective SNR by combining the leakages.
2) We introduce a multivariate leakage model for Xilinx Virtex-5 FPGA device by extending the conventional leakage model for multiple sample points. The proposed multivariate leakage model enables us to determine the variance of the data dependant signal of a sample point without knowing the correct key, thus the SNR of the

sample points also.

3) We derive a linear FIR filter which, when applied to the power traces, maximizes the SNR of its output. The derivation does not require the knowledge of the secret key, thus can be used in non-profiling DPA attacks. We also study how the derived linear FIR filter can be made more resistant to the estimation error and computationally more efficient in practice.

We have supported our theoretical study by experimental evaluation.

Rest of the paper is organized in this way: Section II describes the background of DPA along with the necessary notations used in the work. In Section III, a relation between the success rate of CPA and the SNR of the power traces has been derived. Section IV has extended the conventional leakage model over multiple sample points which results into a multivariate leakage model. In Section V, an expression has been derived to compute the coefficients of the linear FIR filter which optimizes the SNR of its output. Section VI has approximated the linear FIR filter for making it more resistant to estimation error and computationally efficient. In Section VII, the improvements in the performance of CPA using the proposed filtering techniques have been experimentally verified for various scenarios. Section VIII verifies the optimality of the proposed pre-processing techniques. Finally, conclusion has been drawn in Section IX.

II. Preliminaries

A. Notations

For the rest of the paper, we will use a calligraphic letter like $\mathcal{X}$ to denote a finite set. The corresponding capital and small letter, X and x, are used to denote a random variable over the set and a particular element of it respectively. $E[X]$, σ_X and $Var(X)$ are used to denote mean, standard deviation and variance of the random variable X respectively. We also denote by $Cov(X, Y)$ and $Corr(X, Y)$, the covariance and the Pearson's correlation coefficient between the random variables X and Y respectively. The vector $\{x_0, \cdots, x_{r-1}\}$ is denoted by $\{x_i\}_{0 \leq i \leq r}$. Alternatively, it is also denoted by a letter in bold like $\mathbf{x}$. For convenience, sometimes we use μ_X to denote the mean of the random variable X. Gaussian distribution with mean m and standard deviation σ is represented by $\mathrm{N}(m, \sigma)$. $\mathbf{x}'$ denotes the transpose of the vector or matrix $\mathbf{x}$.

B. Differential Power Analysis

We will mainly follow the formalisation of Differential Power Analysis by Standaert et al. in [32]. It is briefly described below.

The DPA attacks have two parts. In the first part, a Device Under Test (DUT) is under the control of the attacker. The attacker collects the leakage L_{t^*} at sample point t^* due to the manipulation of some intermediate key-dependent variable $S = F_{k^*}(X)$ by executing the DUT repeatedly, say q times, for q different inputs. S is commonly referred to as *target* and $F_{k^*} : \mathcal{X} \rightarrow \mathcal{S}$ be a function of a known part of the plaintext $x \in \mathcal{X}$. F_{k^*} is determined by both the algorithm and a small part of the secret key, referred to as the subkey, $k^* \in \mathcal{K}$. The leakage L_{t^*} satisfies

$$L_{t^*} = \tilde{\Psi}_{t^*}(S) + N_{t^*} \tag{1}$$

where the function $\tilde{\Psi}_{t^*} : \mathcal{S} \rightarrow \mathbb{R}$ maps the target S to the deterministic part of the leakage and $N_{t^*} \sim \mathrm{N}(\mu_{N_{t^*}}, \sigma_{N_{t^*}})$ accounts for the independent Gaussian noise. At the end, the attacker collects q measurement curves $\mathbf{l}_{t^*} = \{l^0_{t^*}, \cdots, l^{q-1}_{t^*}\}$ corresponding to the execution of q plaintexts $\mathbf{x} = \{x_0, \cdots, x_{q-1}\}$.

In the second part, the attacker chooses a suitable prediction model $\Psi : \mathcal{S} \rightarrow \mathbb{R}$ and compute the predicted leakage represented by the random variable P_k using $P_k = \Psi(S_k) = \Psi(F_k(X))$, where $S_k = F_k(X)$, for each key hypothesis $k \in \mathcal{K}$. It should be noted that $S = F_{k^*}(X) = S_{k^*}$. If Ψ is a good approximation for $\tilde{\Psi}_{t^*}$, the leakage L_{t^*} is strongly dependent on the correct predicted leakage P_{k^*}. However, since $F_{k^*}(X)$ and $F_k(X)$ are almost independent for $k^* \neq k$, L_{t^*} is independent of the prediction variable P_k. Then, a statistical tool D is used to detect this dependence between the actual leakage and the predicted leakage for the correct key. The theoretical distinguisher is given by $\mathbf{D}(t^*) = \{d_k(t^*)\}_{k \in \mathcal{K}} = \{\mathrm{D}(L_{t^*}, P_k)\}_{k \in \mathcal{K}} = \{\mathrm{D}(\tilde{\Psi}_{t^*}(F_{k^*}(X)) + N_{t^*}, \Psi(F_k(X)))\}_{k \in \mathcal{K}}$. The theoretical first order success rate (1-OSR) [32] of the attack is given by $Pr(k^* = argmax_{k \in \mathcal{K}}\, d_k(t^*))$. However, in practice, the random variables X, L_{t^*}, N_{t^*} and P_k are estimated by the vector $\mathbf{x}$, $\mathbf{l}_{t^*}$, $\mathbf{n}_{t^*} = \{n^0_{t^*}, \cdots, n^{q-1}_{t^*}\}$ and $\mathbf{p}_k = \{\Psi(F_k(x_j))\}_{0 \leq j < q}$ respectively. Thus, the practical distinguisher is given by $\hat{\mathbf{D}}(t^*) = \{\hat{d}_k(t^*)\}_{k \in \mathcal{K}} = \{\hat{\mathrm{D}}(\mathbf{l}_{t^*}, \mathbf{p}_k)\}_{k \in \mathcal{K}}$ and the practical 1-OSR of the attack is given by $Pr(k^* = argmax_{k \in \mathcal{K}}\, \hat{d}_k(t^*))$.

C. Correlation Power Analysis with a model

When the hardware leakage behavior follows an well known leakage model like Hamming weight model or Hamming distance model, some known prediction model Ψ closely approximates $\tilde{\Psi}$ i.e. $\tilde{\Psi}(s) \approx a_{t^*}\Psi(s)$ holds for some real constant a_{t^*} and for all $s \in \mathcal{S}$. Then, Eq. (1) can be approximated [5] as

$$L_{t^*} = a_{t^*}\Psi(S) + N_{t^*} \tag{2}$$

Under the above equation, the relation between the actual leakage L_{t^*} and the predicted leakage for the correct key $P_{k^*} = \Psi(S)$ (since $S = S_{k^*}$) becomes linear. In Correlation Power Analysis (CPA) [5], Pearson's correlation is used to detect the linearity by computing

$$\rho_k(t^*) = \frac{1}{q\hat{\sigma}_{\mathbf{l}_{t^*}}\hat{\sigma}_k} \sum_{j=0}^{q-1} (l^j_{t^*} - \hat{E}[\mathbf{l}_{t^*}])(p^j_k - \hat{E}[\mathbf{p}_k]) \tag{3}$$

$$= \frac{\widehat{Cov}(\mathbf{l}_{t^*}, \mathbf{p}_k)}{\hat{\sigma}_{\mathbf{l}_{t^*}}\hat{\sigma}_k} \tag{4}$$

for all $k \in \mathcal{K}$ where $\mathbf{p}_k$ is the vector $\{p^0_k, \ldots, p^{q-1}_k\} = \{\Psi(F_k(x_0)), \ldots, \Psi(F_k(x_{q-1}))\}$, $\hat{E}[\mathbf{u}]$ is the mean of the elements of the vector $\mathbf{u}$, $\hat{\sigma}_k = \hat{\sigma}_{\mathbf{p}_k}$ and $\widehat{Cov}(\mathbf{u}, \mathbf{v})$ be the maximum likelihood estimator of the covariance between $\mathbf{u}$

and $\mathbf{v}$. Since, Pearson's correlation detects the linear relation between two variables, it performs better than other attacks like Mutual Information Analysis (MIA) [12], Difference of Mean (DoM) [17] when the leakage follows a well known leakage model. When the hardware leakage model is not sufficiently known, 'generic' attacks like MIA perform better than CPA. In the rest of the paper, we will consider only the scenarios where the hardware closely follows a well known leakage behavior.

D. Multivariate DPA

In practical attacks, multiple leakage samples at discrete sample points are collected during the encryptions or decryptions. As a result, the leakage $\mathbf{L}$ is a T-dimensional random variable $\{L_0, \cdots, L_{T-1}\}$ where L_t represents the leakage of sample point t for $0 \leq t < T$. In that case, a univariate distinguisher is applied on each of the sample points independently and then the attacker chooses the best result among those.

While all the profiling attacks like Template attack [6] and Stochastic attack [29] optimizes their performance by considering the multivariate leakage distribution of the power traces, combining the leakages of multiple sample points is rare in non-profiling DPA. Though a few distinguishers like MIA can be extended as a multivariate distinguisher, most of them are not easily extendable for multivariate DPA. Even though they can be extended for multivariate DPA, they do not always improve the performance of the attacks. Instead, such multivariate approaches mainly exist in the forms of various pre-processing techniques like PCA [3], [31], integration [20] and filtering [22]. However, they are either heuristic in nature or based on some assumption. Moreover, to the best of the authors knowledge, there is no such techniques which optimally combines the leakages of multiple sample points. In this paper, we investigate the possibility of combining leakages of multiple sample points in a way that optimizes the success rate.

As shown in Fig. 1, there are two alternative approaches to combine the leakages of multiple sample points: 1) combine the leakages of multiple sample points first using a function $g : \mathbb{R}^T \to \mathbb{R}$ and then apply a univariate distinguisher on the resultant leakage $g(L_0, \cdots, L_{T-1})$ (as shown in Fig. 1a), and 2) apply the univariate distinguisher D on all the sample points independently resulting in $|\mathcal{K}|$ vectors $\{d_k(t)\}_{0 \leq t < T}$ for each $k \in \mathcal{K}$ and then apply the function g to generate the final distinguisher $\{\tilde{d}_k\}_{k \in \mathcal{K}}$ having $\tilde{d}_k = g(d_k(0), \cdots, d_k(T-1))$ (as shown in Fig. 1b).

Interestingly, if we consider Pearson's correlation (as in CPA) as the univariate distinguisher and restrict the function g to the space of linear functions, then the above two approaches are equivalent. To see it, let us denote the Pearson's correlation at sample point t for key guess k by $\rho_k(t)$. Since, g is a $T \times 1$ linear mapping, $g(y_0, \ldots, y_{T-1})$ can be represented as an inner product of the vector $\{y_0, \ldots, y_{T-1}\}$ and the real coefficient vector $\{h_0, \ldots, h_{T-1}\}$. Hence, the output for the key guess k obtained in the second approach

$$\begin{aligned}
\tilde{d}_k &= g(\hat{d}_k(0), \ldots, \hat{d}_k(T-1)) \\
&= \sum_{t=0}^{T-1} h_t \rho_k(t) \\
&= \sum_{t=0}^{T-1} \frac{h_t \widehat{Cov}(\mathbf{l}_t, \mathbf{p}_k)}{\hat{\sigma}_{\mathbf{l}_t} \hat{\sigma}_k} \\
&= \sum_{t=0}^{T-1} \frac{\widehat{Cov}(h_t \mathbf{l}_t / \hat{\sigma}_{\mathbf{l}_t}, \mathbf{p}_k)}{\hat{\sigma}_k} \\
&= \frac{\widehat{Cov}(\sum_{t=0}^{T-1} h_t \mathbf{l}_t / \hat{\sigma}_{\mathbf{l}_t}, \mathbf{p}_k)}{\hat{\sigma}_k} \\
&= \frac{\widehat{Cov}(\tilde{g}(\mathbf{l}_0, \ldots, \mathbf{l}_{q-1}), \mathbf{p}_k)}{\hat{\sigma}_k} \\
&= \frac{\widehat{Cov}(\mathbf{l}_o, \mathbf{p}_k)}{\hat{\sigma}_{\mathbf{l}_o} \hat{\sigma}_k} \hat{\sigma}_{\mathbf{l}_o}
\end{aligned}$$

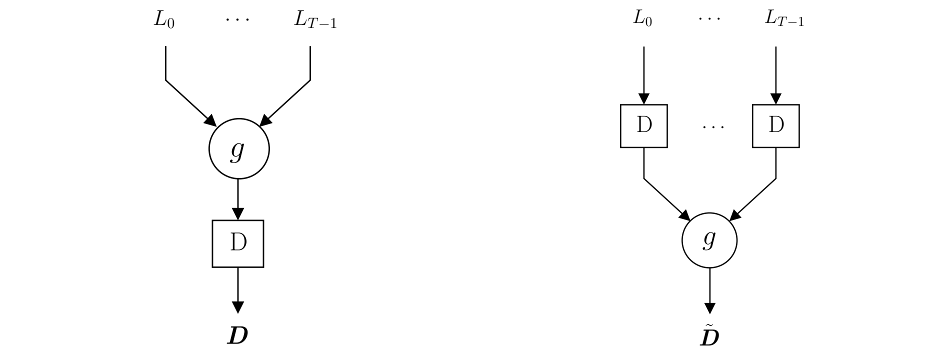
where $\tilde{g}$ be a $T \times 1$ linear mapping with coefficient vector $\{h_t / \hat{\sigma}_{\mathbf{l}_t}\}_{t=0}^{T-1}$ and $\mathbf{l}_o = \tilde{g}(\mathbf{l}_0, \ldots, \mathbf{l}_{q-1})$. Since $\hat{\sigma}_{\mathbf{l}_o}$ does not influence the success rate of a univariate distinguisher, for each linear function g in the second approach, there exists a linear function $\tilde{g}$ in the first approach which results in the same success rate. In other words, optimization of the success rate in the first approach also optimizes the success rate in the second approach and vice-versa. Thus, it is enough to study only one approach. We choose to study the first approach.

In the next section, we derive a relation between the success rate of CPA and the SNR of the power traces.

III. Relation between Success Rate and SNR

The first attempt to estimate the number of traces required to achieve a level of success rate from the value of correlation coefficient was made in [19] for the restricted case of only two subkeys. This relation was extended for arbitrary key set in [33]. In both the works, the authors assumed the correlation for a wrong key is asymptotically null [34]. The later work also considered the distributions of the correlation coefficients for different keys as independent from one another. Later in [28], Rivain made the relation more precise by considering the linear dependency between the correlation coefficients of two different keys. In [10], Fei et al. established a relationship among the success rate of a mono-bit DPA attack, the side channel characteristic and the algorithm dependent parameters called *confusion coefficients*. Recently in [34], the idea of confusion coefficients has been extended for multi-bit CPA under uniform setting hypothesis [28].

The *uniform setting hypothesis* does not hold in many practical applications. The *uniform setting hypothesis* considers a simpler leakage model like Hamming weight model. In a more general model like Hamming distance model [5], the hypothesis may not hold (for example $P_k = HW(sbox(X \oplus k) \oplus X)$). Additionally, the hypothesis assumes the plaintext distribution to be uniform which also may not be true in many chosen plaintext attack setting [18], [35]. In this chapter, we have



(a) Approach 1: Combining is done before applying the distinguisher.

(b) Approach 2: Combining is done after applying the distinguisher.

Fig. 1: Alternative approaches for combining leakages of multiple sample points.

derived the 1-OSR of univariate CPA by relaxing the *uniform setting hypothesis* assumption. Thus, our model only uses the fact that the noise follows a Gaussian distribution.

In this section, we consider a univariate CPA on the leakage of interesting sample point, L_{t^*}, which follows Eq. (2). In the rest of the section, without loss of generality, we neglect the sub-script t^* from the terms in Eq. (2).

In [28], the occurrence ratio of $x \in \mathcal{X}$ in the input vector $\mathbf{x}$ has been defined as

$$r_x = \frac{|\{i|x_i = x\}|}{q}. \tag{5}$$

Then, $\hat{E}[\mathbf{p}_k]$ can be given by $\sum_{x\in\mathcal{X}} r_x P_k(x) = \sum_{x\in\mathcal{X}} r_x \Psi(F_k(x))$ where $P_k(x)$ represents the value of P_k given $X = x$. [28] has defined

$$\begin{aligned}\dot{\rho}_k &= \frac{1}{q\hat{\sigma}_k}\sum_{i=0}^{q-1}(p_k^i - \hat{E}[\mathbf{p}_k])l_{t^*}^i \\ &= \frac{1}{q\hat{\sigma}_k}\sum_{i=0}^{q-1}(P_k(x_i) - \hat{E}[\mathbf{p}_k])l_{t^*}^i \end{aligned} \tag{6}$$

Replacing ρ_k by $\dot{\rho}_k$, success rate remains unchanged in a univariate attack. The distribution of $\dot{\rho}_k$ is given by the following proposition.

Proposition 1: [28] The vector of coefficients $\{\dot{\rho}_k\}_{k\in\mathcal{K}}$ has multivariate Gaussian distribution with mean vector $\mu_{\dot{\rho}}$ having elements

$$E[\dot{\rho}_k] = \frac{1}{\hat{\sigma}_k}\sum_{x\in\mathcal{X}} r_x(P_k(x) - \hat{E}[\mathbf{p}_k])E[L|x] \tag{7}$$

for all $k \in \mathcal{K}$ and with covariance matrix $\mathbf{\Sigma}_{\dot{\rho}}$ having elements

$$\begin{aligned}Cov(\dot{\rho}_{k_1}, \dot{\rho}_{k_2}) =& \frac{1}{q\hat{\sigma}_{k_1}\hat{\sigma}_{k_1}}\sum_{x\in\mathcal{X}} r_x(P_{k_1}(x) - \hat{E}[\mathbf{p}_{k_1}])\times \\ & (P_{k_2}(x) - \hat{E}[\mathbf{p}_{k_2}])Var(L|x)\end{aligned} \tag{8}$$

for all $(k_1, k_2) \in \mathcal{K}^2$.

Applying the above proposition on the leakage L which follows the leakage model given in Eq. (2), we state and prove the following result about the distribution of the comparison vector (as defined in [28]) $\{\Delta\dot{\rho}_k\}_{k\in\mathcal{K}\setminus\{k^*\}} = \{\dot{\rho}_{k^*} - \dot{\rho}_k\}_{k\in\mathcal{K}\setminus\{k^*\}} = \mathbf{\Delta}\dot{\rho}$:

Corollary 1: The comparison vector $\mathbf{\Delta}\dot{\rho}$ has a multivariate Gaussian distribution with mean vector $\mu_{\mathbf{\Delta}\dot{\rho}}$ having elements

$$E[\Delta\dot{\rho}_k] = a \cdot \widehat{Cov}(\mathbf{\Delta p}_k, \mathbf{p}_{k^*}) \tag{9}$$

for all $k \in \mathcal{K} \setminus \{k^*\}$ where $\mathbf{\Delta p}_k = \frac{\mathbf{p}_{k^*}}{\hat{\sigma}_{k^*}} - \frac{\mathbf{p}_k}{\hat{\sigma}_k}$. The distribution of the vector has a covariance matrix $\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}$ having elements

$$Cov(\Delta\dot{\rho}_{k_1}, \Delta\dot{\rho}_{k_2}) = \frac{\sigma_N^2}{q}\widehat{Cov}(\mathbf{\Delta p}_{k_1}, \mathbf{\Delta p}_{k_2}) \tag{10}$$

for all $(k_1, k_2) \in (\mathcal{K} \setminus \{k^*\})^2$ where $\mathbf{\Delta p}_k$ is defined as before.

Proof: From the definition of $\Delta\dot{\rho}_k$ and Eq. (6), we get

$$\begin{aligned}\Delta\dot{\rho}_k =& \dot{\rho}_{k^*} - \dot{\rho}_k \\ =& \frac{1}{q\hat{\sigma}_{k^*}}\sum_{i=0}^{q-1}(P_{k^*}(x_i) - \hat{E}[\mathbf{p_{k^*}}])l_{t^*}^i \\ & - \frac{1}{q\hat{\sigma}_k}\sum_{i=0}^{q-1}(P_k(x_i) - \hat{E}[\mathbf{p_k}])l_{t^*}^i \\ =& \frac{1}{q}\sum_{i=0}^{q-1}(\Delta P_k(x_i) - \hat{E}[\mathbf{\Delta p_k}])l_{t^*}^i\end{aligned}$$

where $\Delta P_k(x_i)$ denotes $\frac{P_{k^*}(x_i)}{\hat{\sigma}_{k^*}} - \frac{P_k(x_i)}{\hat{\sigma}_k}$. Replacing $\dot{\rho}_k$ by $\Delta\dot{\rho}_k$ in Proposition 1 and using Eq. (2), we get

$$\begin{aligned}E[\Delta\dot{\rho}_k] &= \frac{1}{q}\sum_{x\in\mathcal{X}} r_x(\Delta P_k(x) - \hat{E}[\mathbf{\Delta p_k}])E[L|x] \\ &= \frac{1}{q}\sum_{x\in\mathcal{X}} r_x(\Delta P_k(x) - \hat{E}[\mathbf{\Delta p_k}])(a \cdot P_{k^*}(x) + \mu_N) \\ &= \frac{1}{q}\sum_{x\in\mathcal{X}} r_x(\Delta P_k(x) - \hat{E}[\mathbf{\Delta p_k}])(a \cdot P_{k^*}(x)) \\ &= a \cdot \widehat{Cov}(\mathbf{\Delta p}_k, \mathbf{p}_{k^*})\end{aligned}$$

Similarly,

$$\begin{aligned}Cov(\Delta\dot{\rho}_{k_1}, \Delta\dot{\rho}_{k_2}) =& \frac{1}{q}\sum_{x\in\mathcal{X}} r_x(\Delta P_{k_1}(x) - \hat{E}[\mathbf{\Delta p_{k_1}}])\times \\ & (\Delta P_{k_2}(x) - \hat{E}[\mathbf{\Delta p_{k_2}}])Var(L|x) \\ =& \frac{\sigma_N^2}{q}\widehat{Cov}(\mathbf{\Delta p}_{k_1}, \mathbf{\Delta p}_{k_2})\end{aligned}$$

■

For a successful attack, the condition $\{\Delta\dot{\rho}_k\}_{k\in\mathcal{K}\setminus\{k^*\}} > \mathbf{0}$ holds where $\mathbf{0}$ is a zero vector of size $|\mathcal{K}|-1$ and $\mathbf{v}_1 > \mathbf{v}_2$ implies each element of $\mathbf{v}_1$ is greater than the corresponding element of $\mathbf{v}_2$. Thus the first order success rate can be given by the term $Pr(\{\Delta\dot{\rho}_k\}_{k\in\mathcal{K}\setminus\{k^*\}} > \mathbf{0})$. We mention by passing that for a negative value of a in Eq. (2), one would expect a negative correlation for the correct key and thus the definition of success rate should be changed accordingly. For the time being we assume a positive correlation for the correct key and state Proposition 2. Without loss of generality, we also assume that the distribution of $\mathbf{\Delta}\dot{\rho} = \{\Delta\dot{\rho}_k\}_{k\in\mathcal{K}\setminus\{k^*\}}$ is non-degenerative [26] (see Appendix B for the degenerative case).

Proposition 2: The first order success rate (1-OSR$_{CPA}(\mathbf{x})$) of CPA for the input plaintext vector $\mathbf{x}$ is given by

$$\text{1-OSR}_{CPA}(\mathbf{x}) = \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}}(\mu_{\mathbf{\Delta}\dot{\rho}}) \tag{11}$$

where $\Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}}$ be the cdf of a multivariate normal distribution with $(|\mathcal{K}|-1)$-dimensional zero mean vector and covariance matrix $\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}$.

Proof: Since, $\mathbf{\Delta}\dot{\rho}$ follows the multivariate normal distribution with mean $\mu_{\mathbf{\Delta}\dot{\rho}}$ and covariance matrix $\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}$, the first order success rate is given by

$$\begin{aligned}\text{1-OSR}_{CPA}(\mathbf{x}) &= Pr(\mathbf{\Delta}\dot{\rho} > \mathbf{0}) \\ &= f_{\mathbf{\Delta}\dot{\rho}}(\mathbf{0} < \mathbf{\Delta}\dot{\rho} < \infty) \\ &= f_{\mathbf{\Delta}\dot{\rho}}(-\mu_{\mathbf{\Delta}\dot{\rho}} < \mathbf{\Delta}\dot{\rho} - \mu_{\mathbf{\Delta}\dot{\rho}} < \infty) \\ &= f_{\mathbf{\Delta}\dot{\rho}}(-\infty < \mathbf{\Delta}\dot{\rho} - \mu_{\mathbf{\Delta}\dot{\rho}} < \mu_{\mathbf{\Delta}\dot{\rho}}) \\ &= \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}}(\mu_{\mathbf{\Delta}\dot{\rho}})\end{aligned}$$

where $f_{\mathbf{\Delta}\dot{\rho}}$ denotes the pdf of the distribution of $\mathbf{\Delta}\dot{\rho}$, and ∞ and $-\infty$ are the $(|\mathcal{K}|-1)$-dimensional vector of which all the elements are ∞ and $-\infty$ respectively.

■

To analyse the first order success rate further, from Corollary 1, we note that $\mu_{\mathbf{\Delta}\dot{\rho}} = a\mu_{\mathbf{\Delta P}}(\mathbf{x})$ and $\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}} = \frac{\sigma_N^2}{q}\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ where $\mu_{\mathbf{\Delta P}}(\mathbf{x})$ be the vector $\{\widehat{Cov}(\mathbf{\Delta p}_k, \mathbf{p}_{k^*})\}_{k\in\mathcal{K}\setminus\{k^*\}}$ and $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ be the $(|\mathcal{K}|-1)\times(|\mathcal{K}|-1)$ matrix with elements $\widehat{Cov}(\mathbf{\Delta p}_{k_1}, \mathbf{\Delta p}_{k_2})$ for all $(k_1,k_2)\in(\mathcal{K}\setminus\{k^*\})^2$. Let us also define the signal-to-noise ratio [20] of traces as

$$SNR = \frac{Var(E[L|P_{k^*}])}{Var(L - E[L|P_{k^*}])} = \frac{a^2\sigma_{k^*}^2}{\sigma_N^2} \tag{12}$$

Then we state the following result. Again we assume that $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ is a positive definite matrix. Please refer to Appendix B for the situation when $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ is not a positive definite matrix.

Corollary 2: The first order success rate (1-OSR$_{CPA}(\mathbf{x})$) of CPA for the input plaintext vector $\mathbf{x}$ is given by

$$\text{1-OSR}_{CPA}(\mathbf{x}) = \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})}(\sqrt{q}\sqrt{SNR}\sigma_{k^*}^{-1}\mu_{\mathbf{\Delta P}}(\mathbf{x})) \tag{13}$$

where Φ, $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ and $\mu_{\mathbf{\Delta P}}(\mathbf{x})$ is defined as before.

Proof: Let us first denote the multi-dimensional intigration $\int_{ll_0}^{ul_0}\cdots\int_{ll_{|\mathcal{K}|-1}}^{ul_{|\mathcal{K}|-1}} f(y_0,\ldots,y_{|\mathcal{K}|-1})dy_{|\mathcal{K}|-1}\cdots dy_0$ as $\int_{\mathbf{ll}}^{\mathbf{ul}} f(y_0,\cdots,y_{|\mathcal{K}|-1})d\mathbf{y}$ where $\mathbf{ll} = \{ll_0,\ldots,ll_{|\mathcal{K}|-1}\}$, $\mathbf{ul} = \{ul_0,\ldots,ul_{|\mathcal{K}|-1}\}$ and $\mathbf{y} = \{y_0,\ldots,y_{|\mathcal{K}|-1}\}$. From Proposition 2, we get

$$\begin{aligned}\text{1-OSR}_{CPA}(\mathbf{x}) &= \int_{-\infty}^{\mu_{\mathbf{\Delta}\dot{\rho}}} \frac{e^{-\frac{1}{2}\mathbf{y}'\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}^{-1}\mathbf{y}}}{\sqrt{(2\pi)^{k-1}|\mathbf{\Sigma}_{\mathbf{\Delta}\dot{\rho}}|}}d\mathbf{y} \\ &= \int_{-\infty}^{\frac{\sqrt{q}}{\sigma_N}\mu_{\mathbf{\Delta}\dot{\rho}}} \frac{e^{-\frac{1}{2}\tilde{\mathbf{y}}'(\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x}))^{-1}\tilde{\mathbf{y}}}}{\sqrt{(2\pi)^{k-1}|\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})|}}d\tilde{\mathbf{y}}, \\ &\qquad\qquad\qquad \text{where } \tilde{\mathbf{y}} = \frac{\sqrt{q}}{\sigma_N}\mathbf{y} \\ &= \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})}\left(\frac{\sqrt{q}}{\sigma_N}\mu_{\mathbf{\Delta}\dot{\rho}}\right) \\ &= \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})}\left(\frac{\sqrt{q}a}{\sigma_N}\mu_{\mathbf{\Delta P}}(\mathbf{x})\right) \\ &= \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})}\left(\sqrt{q}\sqrt{\frac{a^2\sigma_{k^*}^2}{\sigma_N^2}}\sigma_{k^*}^{-1}\mu_{\mathbf{\Delta P}}(\mathbf{x})\right) \\ &= \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})}\left(\sqrt{q}\sqrt{SNR}\sigma_{k^*}^{-1}\mu_{\mathbf{\Delta P}}(\mathbf{x})\right)\end{aligned}$$

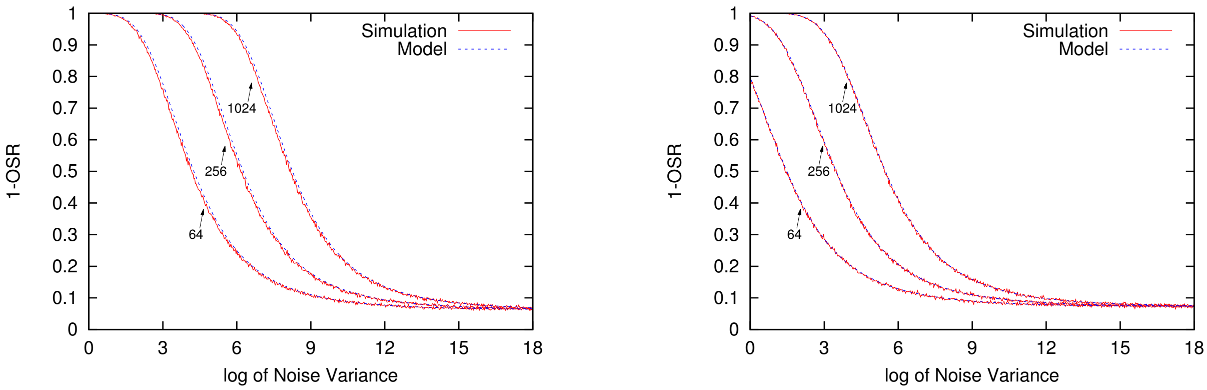
■

The parameters $\mu_{\mathbf{\Delta P}}(\mathbf{x})$ and $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ depend on the frequency distribution of the elements of $\mathbf{x}$. However, for a fixed set of power traces, the values of the parameters are fixed. Moreover, as the number of power traces q increases, the parameters $\mu_{\mathbf{\Delta P}}(\mathbf{x})$ and $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ evolve. For a sufficiently large value of q, the frequency distribution of X converges to some distribution f_X, and hence, the parameters $\mu_{\mathbf{\Delta P}}(\mathbf{x})$ and $\mathbf{\Sigma}_{\mathbf{\Delta P}}(\mathbf{x})$ to some values, denoted by $\mu_{\mathbf{\Delta P}}^{f_X}$ and $\mathbf{\Sigma}_{\mathbf{\Delta P}}^{f_X}$ respectively, which are independent of q. In that case, Eq. (13) can be written as

$$\text{1-OSR}_{CPA}^{f_X}(q) = \Phi_{\mathbf{0},\mathbf{\Sigma}_{\mathbf{\Delta P}}^{\mathbf{f_X}}}(\sqrt{q}\sqrt{SNR}\sigma_{k^*}^{-1}\mu_{\mathbf{\Delta P}}^{\mathbf{f_X}}) \tag{14}$$

To experimentally validate Eq. (14), we computed practical 1-OSR by simulation. For the simulation, we generated power traces by adding Gaussian noise to the Hamming weight of the output of PRESENT S-box. The success rate is computed by repeating CPA on the simulated power traces 10000 times. On the other hand, we estimated 1-OSR using the model given by Eq. (14). Both the results are plotted in Fig. (2) with the increasing variance of Gaussian noise.

Similar relation between more general o^{th} order success rate with SNR can be found. Thus, for a given algorithm and a fixed set of traces, maximization of the success rate requires the maximization of SNR. In this work, we combine the leakages $L_0,\cdots,L_{T-1}$ using a linear function g in such a way that it maximizes the SNR of the resultant leakage



(a) Plaintext distribution is uniform.

(b) Plaintext distribution is non-uniform. Each plaintext values 0 to 5 has probability $1/6$ and the rest of the values have probability 0.

Fig. 2: Plots of the practical and theoretical 1-OSR (1st order success rate) for CPA on the output of PRESENT S-box using HW model. The 1-OSR is estimated using number of traces q equals to 64, 256 and 1024 respectively.

$g(L_0, \cdots, L_{T-1})$. However, such combining is not possible in non-profiling setup without any estimation of the information contained in each sample point. Thus, in the following sections, we try to estimate the information at each sample point using some parameters which can be computed without knowing the correct key.

IV. Multivariate Leakage Model: Extending the leakage model over multiple time samples

A. Profiling the Power Traces of AES

In this section, our objective is to investigate how leakage due to a known computation varies over a range of sample points. The nature of leakages at several sample points have been investigated with respect to the predicted leakage for the correct key $P_{k^*} = \Psi(S)$ using the following metrics.

1) *Squared Pearson's Correlation between Data Dependent Leakage and Predicted Leakage (SCDP)*: It is defined as follows:

$$SCDP_t = Corr^2(E[L_t|P_{k^*}], P_{k^*})$$

Since, Pearson's correlation detects the linear relation between two variables [8], $SCDP_t = Corr^2(E[L_t|P_{k^*}], P_{k^*})$ reveals the linear dependency between the deterministic leakage at t and the predicted leakage P_{k^*}. It should be noted that if the leakage of a sample point t follows Eq. (2), then the value of $SCDP$ at t is almost one. On the other hand, if L_t and P_{k^*} are almost independent, $E[L_t|p]$ will be almost constant for all $p \in \mathcal{P}$, resulting to $SCDP_t$ almost zero.

2) *Variation of Data Dependent Leakage (VDL)*:

$$VDL_t = Var(E[L_t|P_{k^*}])$$

It reveals the variations in leakage caused by the predicted leakage P_{k^*} at sample point t. Sometimes, it is used to quantify the signal in the leakage. On the other hand, noise is quantified by $Var(L_t - E[L_t|P_{k^*}])$.

3) *Squared Mean Leakage (SML)*:

$$SML_t = E^2[L_t]$$

It quantifies the magnitude or the strength of the leakage at a sample point.

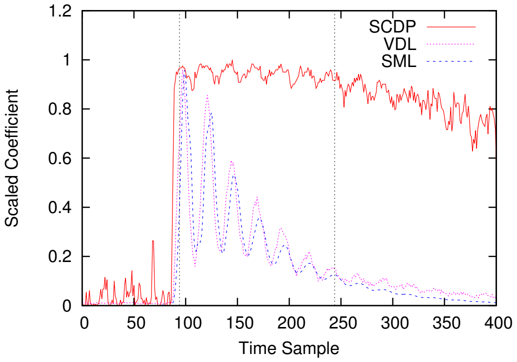


Fig. 3: Plots of the chosen metrics in the last round of unprotected implementation of AES.

Fig. 3 shows the plot for the above three metrics which are estimated over 20000 traces of AES encryptions. The AES is implemented using parallel iterative hardware architecture on the setup described in Appendix A. The correct predicted leakage P_{k^*} is taken as the Hamming distance between the ciphertext and the input to the last round. The metrics are plotted only for 400 sample points around the last round register update.

The figure shows that as the cycle begins, with the mean leakage (SML), SCDP also rises rapidly, remains almost constant for about 150 sample points and then it decreases slowly. The slight fluctuations in the curve are due to the presence of small amount of noise after averaging a limited number of power traces. This leads us to the following observation:

Observation 1: The deterministic leakage at a large number of sample points show high linear dependencies with the correct predicted leakage P_{k^*}.

In other words, a large number of sample points contain information about the correct predicted leakage P_{k^*}. It should be noted that various profiling attacks optimally extract the information from multiple sample points by estimating the multivariate leakage distribution of the sample points using a profiling step.

From the figure, we can also see that VDL almost superimposes on SML i.e. VDL is highly correlated to SML. This leads us to the following observation:

Observation 2: The variation in deterministic leakage of a sample point is correlated to the square of the mean leakage of the sample point.

In other words, the second observation states that the magnitude of the variation in leakage at a sample point due to some computation is proportional to the mean value (strength) of the leakage at that sample point. It should be noted that similar kinds of observation can be found in Chapter 4.3.2 of [20] for the leakages of a micro-controller. The authors have also suggested several trace compression techniques based on the observation and have shown their usefulness to attack software implementation of AES. However, to the best of our knowledge, no attempt has been made to incorporate these observations into the leakage model.

In the next part of the section, we extend the conventional leakage model over multiple sample points using the above two observations.

B. Modeling the Leakage over Multiple Time Samples

Observation 1 and 2 immediately extend the conventional leakage model given by Eq. (2), into the following multivariate leakage model:

$$L_t = a_t\Psi(S) + N_t = a_tP_{k^*} + N_t \tag{15}$$

for $t_0 \leq t < t_0 + \tau$ where $a_t \in \mathbb{R}$ and the random vector $\mathbf{N} = \{N_{t_0}, \ldots, N_{t_0+\tau-1}\}$ follows a multivariate Gaussian distribution with covariance matrix $\mathbf{\Sigma_N}$. It should be noted that the linear relation in Eq. (15) is a consequence of Observation 1 while Observation 2 enforces the mean vector of $\mathbf{N}$ to be either a zero vector or a multiple of the vector $\mathbf{a} = \{a_{t_0}, \cdots, a_{t_0+\tau-1}\}$.

In a parallel iterative hardware architecture, a single round consists of several parallel S-boxes and the attacker targets only a part of it (usually a single S-box). Thus, in addition to the predicted leakage P_{k^*} due to the computation of the target $S = F_{k^*}(X)$, leakage due to the computation of the other parallel bits adds to it. This is known as algorithmic noise and we denote it by U. It should be noted that for a fully serialized architecture, U takes the value zero. Leakages due to the key bits and the control bits is denoted by c. Since key scheduling and the controlling operations are fixed for a specific round in all the encryptions, c is constant for all the inputs.

Thus, we can adopt Eq. (15) to incorporate these new variables as follows:

$$\begin{aligned} L_t &= a_t(P_{k^*} + U + c) + N_t && (16)\\ &= a_t(I + c) + N_t, \qquad t_0 \leq t < t_0 + \tau && (17)\end{aligned}$$

where $I = P_{k^*} + U$ and the noise $\mathbf{N} = \{N_{t_0}, \cdots, N_{t_0+\tau-1}\}$ follows a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix $\Sigma_{\mathbf{N}}$. We are interested in the leakages of the above window namely $\{t_0, \ldots, t_0 + \tau - 1\}$ that can be roughly determined by the clock cycle in which the target operation is being performed (see Paragraph *Determination of Window*). We denote this time span by $\{0, \ldots, \tau - 1\}$ and in the rest of the paper, power trace is referred by the sample points of this time span only.

CPA and the Multivariate Leakage Model: In classical CPA, Pearson's correlation is applied to each of the sample points independently i.e. $Corr(L_t, P_{k^*})$ is computed for all $0 \leq t < T$. From the relation between Pearson's correlation and linear regression [25], each L_t can be written as

$$L_t = a_tP_{k^*} + b_t + N_t \tag{18}$$

where a_t is given by $Corr(L_t, P_{k^*})\sigma_{L_t}/\sigma_{P_{k^*}}$ and $b_t = E[L_t] - a_tE[P_{k^*}]$. The random variable N_t is independent of P_{k^*} and has zero mean. Thus, CPA also assumes a linear leakage model for all the sample points. However, comparing Eq. (16) and (18), we note that Eq. (16) additionally assumes the constant bias b_t to be proportional to a_t in the window $\{t_0, \ldots, t_0 + \tau - 1\}$. In practice, b_t may not be exactly proportional to a_t. However, for our practical experiments, the approximation provides better results.

Determination of Window: The model is valid only in the clock cycle in which the target operation is being performed (called the *target clock cycle*). If implementation details are available, then the target clock cycle can be easily determined. When implementation details are not available, other methods such as in [23] can be incorporated. In [23], the authors suggested to use Inter-Class Variance (ICV), $Var(E[L_t|X])$, for the selection of correct time-window in collision attack. Recently in [4], Normalized Inter-Class Variance (NICV) has been introduced as a metric for window selection. NICV takes the ratio of Inter-Class Variance, $Var(E[L_t|X])$, and the leakage variance, $Var(L_t)$, to determine the relevant sample points. We have found better result using Corrected Inter-Class Variance (CICV) which has been computed as follows. Since the expectation $E[L_t|X]$ is estimated using a finite number of power traces, say q, the estimated value for ICV at sample point t is actually $\widehat{ICV}_t \approx Var(E[L_t|X]) + \sigma^2_{N_t}/f$ where $f = q/|\mathcal{X}|$ is the average frequency of the elements of $\mathcal{X}$. On the other hand, we can write the estimated leakage variance at t as $\hat{\sigma}^2_{L_t} \approx Var(E[L_t|X]) + \sigma^2_{N_t}$. Thus, we computed the

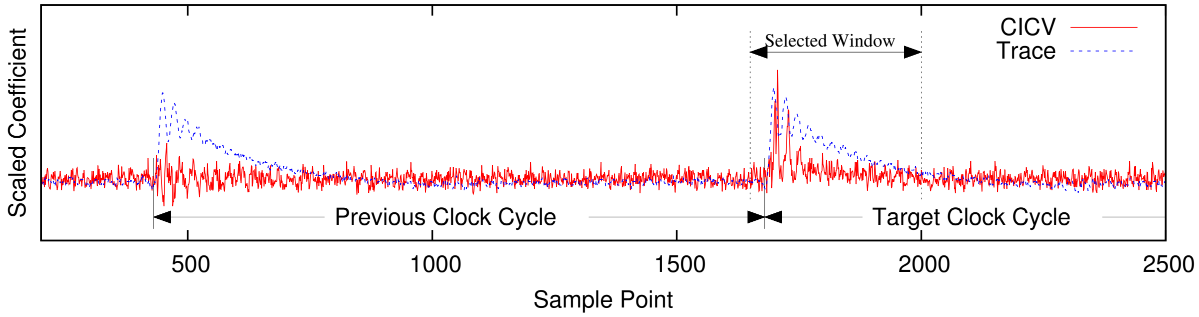


Fig. 4: The dashed line plots a trace for last two clock cycles of an AES encryption. The solid line is the plot for CICV computed for the two rounds using 1000 traces. The figure shows higher peak in the CICV curve for the correct clock cycle.

value of CICV at t as

$$\begin{aligned} CICV_t &= \widehat{ICV}_t - \frac{1}{f}\sigma^2_{N_t} \\ &= \widehat{ICV}_t - \frac{1}{f-1}(\sigma^2_{N_t} - \frac{\sigma^2_{N_t}}{f}) \\ &\approx \widehat{ICV}_t - \frac{1}{f-1}(\hat{\sigma}^2_{L_t} - \widehat{ICV}_t) \\ &= \frac{f}{f-1}\widehat{ICV}_t - \frac{1}{f-1}\hat{\sigma}^2_{L_t}. \end{aligned}$$

Fig. 4 shows the CICV computed for the last two rounds of an AES encryptions using 1000 traces. For the correct clock cycle, CICV shows a higher peak. Once the target clock cycle has been identified using the CICV, a window of sample points for which the mean leakages $E[L_t]$ are significantly higher can be chosen as the target window.

C. Experimental Validation of the Multivariate Leakage Model

We experimentally validated Eq. (16) and (17) on the side-channel evaluation board SASEBO-GII. The validation is carried over the following steps. We first classify all the traces according to the values of I. Then we estimate the deterministic leakage $\mathbf{d}^i = \{E[L_t|I = i]\}_{0 \le t < \tau}$ for all $i \in \mathcal{I}$ by computing the mean leakage curve of each class. Lastly, we verify the linear equation $E[L_t|I = i] - E[L_t|I = 0] = a_t \cdot i$ for all $i \in \mathcal{I} \setminus \{0\}$ and $0 \le t < \tau$ using linear regression. However, we do not know the values of a_t, $0 \le t < \tau$. Thus, we infer the values of a_t from the deterministic leakage curves $\mathbf{d}^i$s.

We implemented an iterative structure of 32 parallel 10×4 S-boxes using distributed ROM in the setup described in Appendix A. All of the S-boxes were connected to the same input to increase the SNR of the power traces by the synchronous computations of the S-boxes. It should be noted that though the duplication of a single S-box increases the SNR of all the sample points, their relative SNR remains same. We collected 1600 power traces each having 200 sample points with random inputs. The values of the target variable S is taken to be the output of the S-box. We have also considered the Hamming distance model i.e. $\Psi(s)$ is taken to be the Hamming distance between s and the least significant 4 bit of the S-box input for all $s \in \mathcal{S}$. Since all the parallel S-boxes have the same input and the output, the algorithmic noise U is zero i.e. $I = P_{k^*} = \Psi(S)$.

The classification involves partitioning all the 1600 traces into five HD classes for $I = 0$ to 4. Fig. 5 shows the deterministic leakage curve $\mathbf{d}^i = \{E[L_t|I = i]\}_{0 \le t < \tau}$ for $0 \le i \le 4$ i.e. for each of the five classes. It is seen in the figure that the deterministic leakage for different HD classes i.e. different values of I are following almost same pattern. However, the non-zero leakage for HD class 0 is caused by the switching activities of the control bits and the DC power consumption which is also present in the leakages of all other classes. To remove this factor, we computed absolute deterministic leakage curves as $\bar{\mathbf{d}}^i = \mathbf{d}^i - \mathbf{d}^0 = \{E[L_t|I = i] - E[L_t|I = 0]\}_{t=0}^{\tau-1} = \{a_t \cdot i\}_{t=0}^{\tau-1}$ (from Eq. (17)) for $i = 1, \cdots, 4$. Table I shows the correlation between $\bar{\mathbf{d}}^{i_1}$ and

Correlation	$\bar{\mathbf{d}}^1$	$\bar{\mathbf{d}}^2$	$\bar{\mathbf{d}}^3$	$\bar{\mathbf{d}}^4$
$\bar{\mathbf{d}}^1$	1	0.9991	0.9981	0.9978
$\bar{\mathbf{d}}^2$	0.9991	1	0.9995	0.9992
$\bar{\mathbf{d}}^3$	0.9981	0.9995	1	0.9997

TABLE I: Pearson's correlation between absolute deterministic leakage curves of different pairs of HD Classes

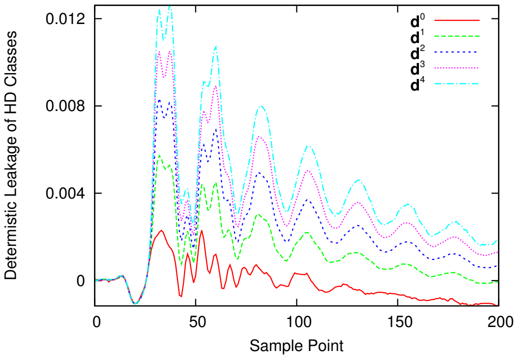


Fig. 5: Mean Leakage for the five Hamming distance classes across the 200 sample points.

$\bar{\mathbf{d}}^{i_2}$ for all $i_1, i_2 \in \mathcal{I} \backslash \{0\}$. The values of these correlations are close to one which ensure that all of these vectors follow linear relations with a common vector namely $\mathbf{a} = \{a_0, \cdots, a_{\tau-1}\}$. We estimate $\mathbf{a}$ by $\frac{\sum_{i=1}^{4} \bar{\mathbf{d}}^i}{\sum_{i=1}^{4} i}$.

Next, we plot the vectors $\bar{\mathbf{d}}^i$ for all $i \in \mathcal{I} \setminus \{0\}$ against the estimated $\mathbf{a}$. The plot is shown in Fig. 6. The figure shows the linear relationships of $\bar{\mathbf{d}}^i$'s with the estimated $\mathbf{a}$. So, we have further used linear regression to find the closest linear models of the relation between each of $\bar{\mathbf{d}}_1$, $\bar{\mathbf{d}}_2$, $\bar{\mathbf{d}}_3$ and $\bar{\mathbf{d}}_4$ and the estimated $\mathbf{a} = \{a_t\}_{t=0}^{\tau-1}$. The relations obtained using linear regression are sufficiently close to the expected relation which are shown in Table II. This provides an evidence of the validity of Eq. (17).

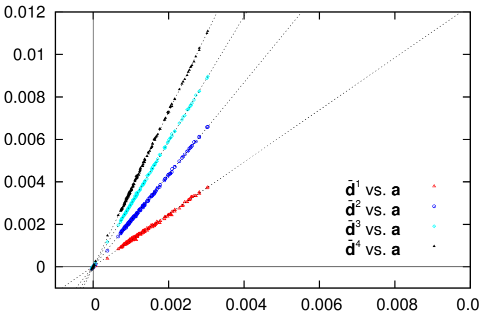


Fig. 6: Scatter Plots of $\bar{\mathbf{d}}^1$, $\bar{\mathbf{d}}^2$, $\bar{\mathbf{d}}^3$ and $\bar{\mathbf{d}}^4$ against $\mathbf{a}$.

Variable	Obtained Relation	Expected Relation
$E[L_t \| I = 1] - E[L_t \| I = 0]$	$a_t \times 1.23 - 1.60 \times 10^{-5}$	$a_t \times 1$
$E[L_t \| I = 2] - E[L_t \| I = 0]$	$a_t \times 2.17 - 7.26 \times 10^{-8}$	$a_t \times 2$
$E[L_t \| I = 3] - E[L_t \| I = 0]$	$a_t \times 2.95 - 1.41 \times 10^{-6}$	$a_t \times 3$
$E[L_t \| I = 4] - E[L_t \| I = 0]$	$a_t \times 3.65 - 1.75 \times 10^{-5}$	$a_t \times 4$

TABLE II: Relations of $\bar{\mathbf{d}}_1$, $\bar{\mathbf{d}}_2$, $\bar{\mathbf{d}}_3$ and $\bar{\mathbf{d}}_4$ with $\mathbf{a} = \{a_t\}_{t=0}^{\tau-1}$.

In the next section, we explore optimal pre-processing of power traces using linear FIR filter for non-profiling DPA attacks.

V. Optimum Filtering in Non-profiling DPA

Matched filter is commonly used to maximize the SNR of a noisy signal. In the next part of the section, we derive the impulse response of the matched filter for the application in SCA.

A. *Finding the Impulse Response of the Matched Filter*

In SCA, the output of a filter with impulse response $\mathbf{h} = \{h_0, \cdots, h_{\tau-1}\}$ applied on the leakage $\mathbf{L} = \{L_0, \cdots, L_{\tau-1}\}$ can be given by the convolution between the impulse response $\mathbf{h}$ and the leakage $\mathbf{L}$ i.e. $L_o = \sum_{t=0}^{\tau-1} h_{\tau-t-1} L_t$. However, for convenience, we use the inner product form. Thus, using the inner product form, we express the output leakage L_o by the following equation.

$$L_o = \sum_{t=0}^{\tau-1} h_t L_t. \tag{19}$$

The impulse response $\mathbf{h}$ of the matched filter is derived such that the SNR of the output leakage L_o is maximized. Using Eq. (12), we compute the SNR of L_o.

$$\mathrm{SNR}_{L_o} = \frac{Var\left(E\left[L_o | P_{k^*}\right]\right)}{Var\left(L_o - E\left[L_o | P_{k^*}\right]\right)} \tag{20}$$

Putting $L_o = \sum_{t=0}^{\tau-1} h_t L_t$, we simplify the numerator of RHS of the above equation as

$$\begin{aligned}
&Var\left(E\left[L_o | P_{k^*}\right]\right)\\
&= Var\left(E\left[\left(\sum_{t=0}^{\tau-1} h_t L_t\right) | P_{k^*}\right]\right)\\
&= Var\left(E\left[\left(\sum_{t=0}^{\tau-1} h_t (a_t(P_{k^*} + U + c) + N_t)\right) | P_{k^*}\right]\right)\\
&= Var\left(\sum_{t=0}^{\tau-1} h_t a_t P_{k^*}\right)\\
&= |\mathbf{h}'\mathbf{a}|^2 \sigma_{k^*}^2
\end{aligned} \tag{21}$$

where σ_{k^*} denotes $\sigma_{P_{k^*}}$. Similarly, we simplify the denominator of RHS of Eq. (20) as

$$\begin{aligned}
&Var\left(L_o - E\left[L_o | P_{k^*}\right]\right)\\
&= Var\left(\sum_{t=0}^{\tau-1} h_t (a_t(U + c) + N_t)\right)\\
&= Var\left(\sum_{t=0}^{\tau-1} h_t \tilde{N}_t\right),\\
&= Var\left(\mathbf{h}'\tilde{\mathbf{N}}\right),\\
&= E\left[\left(\mathbf{h}'\left(\tilde{\mathbf{N}} - E[\tilde{\mathbf{N}}]\right)\right)\left(\mathbf{h}'\left(\tilde{\mathbf{N}} - E[\tilde{\mathbf{N}}]\right)\right)'\right]\\
&= \mathbf{h}' E\left[\left(\tilde{\mathbf{N}} - E[\tilde{\mathbf{N}}]\right)\left(\tilde{\mathbf{N}} - E[\tilde{\mathbf{N}}]\right)'\right] \mathbf{h}\\
&= \mathbf{h}' \Sigma_{\tilde{\mathbf{N}}} \mathbf{h}
\end{aligned} \tag{22}$$

where $\tilde{N}_t = a_t(U + c) + N_t$, $\tilde{\mathbf{N}} = \{\tilde{N}_0, \cdots, \tilde{N}_{\tau-1}\}$ and $\Sigma_{\tilde{\mathbf{N}}}$ is the covariance matrix of $\tilde{\mathbf{N}}$. Using Eq. (21) and (22), we write Eq. (20):

$$\mathrm{SNR}_{L_o} = \frac{|\mathbf{h}'\mathbf{a}|^2 \sigma_{k^*}^2}{\mathbf{h}' \Sigma_{\tilde{\mathbf{N}}} \mathbf{h}} \tag{23}$$

The optimization of the above expression is an well studied problem in DSP. The following theorem provides the value of $\mathbf{h}$ which maximizes the above expression.

Theorem 1: [30], [36] The impulse response $\mathbf{h}_{\text{MF}}$ of the matched filter, the linear filter which maximizes the SNR of the output leakage L_o, can be given by

$$\mathbf{h}_{\text{MF}} = \Sigma_{\tilde{\mathbf{N}}}^{-1} \mathbf{a} \tag{24}$$

where $\mathbf{a}$ and $\Sigma_{\tilde{\mathbf{N}}}$ are defined as before.

The proof of Theorem 1 is given in Appendix C. To compute $\mathbf{h}_{\text{MF}}$, we need the covariance matrix $\Sigma_{\tilde{\mathbf{N}}}$ and the vector $\mathbf{a}$. Both require the knowledge of the secret key to estimate. Thus, the impulse response of the matched filter cannot be estimated in non-profiling DPA attacks. In the next section, we introduce an optimum linear filter which maximizes the SNR of the output leakage and can also be estimated without the knowledge of the secret key.

B. Optimum Linear Filter for Non-profiling DPA

To overcome the issue of estimating the noise covariance matrix for computing the impulse response of the matched filter, we introduce Signal Ratio (SR) of the output leakage L_o as follows.

$$\text{SR}_{L_o} = \frac{Var\left(E\left[L_o|P_{k^*}\right]\right)}{Var\left(L_o\right)}. \tag{25}$$

From Eq. (21), we get that the numerator of RHS of the above expression equals to $|\mathbf{h}'\mathbf{a}|^2\sigma_{k^*}^2$. The denominator can be simplified as

$$\begin{aligned}
Var\left(L_o\right) &= Var\left(\sum_{t=0}^{\tau-1} h_t L_t\right) \\
&= Var\left(\mathbf{h}'\mathbf{L}\right) \\
&= E\left[(\mathbf{h}'(\mathbf{L}-E\left[\mathbf{L}\right]))(\mathbf{h}'(\mathbf{L}-E\left[\mathbf{L}\right]))'\right] \\
&= \mathbf{h}'E\left[(\mathbf{L}-E\left[\mathbf{L}\right])(\mathbf{L}-E\left[\mathbf{L}\right])'\right]\mathbf{h} \\
&= \mathbf{h}'\Sigma_{\mathbf{L}}\mathbf{h}
\end{aligned} \tag{26}$$

where $\Sigma_{\mathbf{L}}$ be the covariance matrix of the multivariate leakage $\mathbf{L}$. Putting the values of the numerator and denominator of RHS of Eq. (25) from Eq. (21) and Eq. (26), we get

$$\text{SR}_{L_o} = \frac{|\mathbf{h}'\mathbf{a}|^2\sigma_P^2}{\mathbf{h}'\Sigma_{\mathbf{L}}\mathbf{h}}. \tag{27}$$

The following lemma provides a relationship between SNR and SR

Lemma 1: The SR of the output leakage L_o reaches its maximum if and only if SNR of that also reaches its maximum.

Proof: The denominator of the RHS of Eq. (25) can be simplified as

$$\begin{aligned}
Var\left(L_o\right) &= Var\left(\sum_{t=0}^{\tau-1} h_t L_t\right) \\
&= Var\left(\sum_{t=0}^{\tau-1} h_t(a_t P_{k^*} + a_t(I+c) + N_t)\right) \\
&= Var\left(\sum_{t=0}^{\tau-1} h_t(a_t P_{k^*} + \tilde{N}_t)\right) \\
&= Var\left(\mathbf{h}'\mathbf{a}P_{k^*}\right) + Var\left(\mathbf{h}'\tilde{\mathbf{N}}\right) \\
&= |\mathbf{h}'\mathbf{a}|^2\sigma_{k^*}^2 + \mathbf{h}'\Sigma_{\tilde{\mathbf{N}}}\mathbf{h}
\end{aligned} \tag{28}$$

where, as defined before, $\tilde{N}_t = a_t(I+c) + N_t$ and $\tilde{\mathbf{N}} = \{\tilde{N}_0, \cdots, \tilde{N}_{\tau-1}\}$. Putting the values of numerator and denominator of RHS of Eq. 25 from Eq. (21) and (28), we get

$$\begin{aligned}
\text{SR}_{L_o} &= \frac{|\mathbf{h}'\mathbf{a}|^2\sigma_{k^*}^2}{|\mathbf{h}'\mathbf{a}|^2\sigma_{k^*}^2 + \mathbf{h}'\Sigma_{\tilde{\mathbf{N}}}\mathbf{h}} \\
&= \frac{1}{1+\frac{\mathbf{h}'\Sigma_{\tilde{\mathbf{N}}}\mathbf{h}}{|\mathbf{h}'\mathbf{a}|^2\sigma_{k^*}^2}} = \frac{1}{1+\frac{1}{\text{SNR}_{L_o}}}
\end{aligned}$$

We rewrite the above equation as,

$$\frac{1}{\text{SR}_{L_o}} = 1 + \frac{1}{\text{SNR}_{L_o}}$$

Hence, the conclusion follows. ■

Thus, the optimization of SNR of the output leakage L_o is equivalent to the optimization of the SR. Comparing Eq. (23) and (27), we note that $\Sigma_{\tilde{\mathbf{N}}}$ in the denominator of the definition of SNR is replaced by $\Sigma_{\mathbf{L}}$ in the definition of SR. Thus, replacing $\Sigma_{\tilde{\mathbf{N}}}$ by $\Sigma_{\mathbf{L}}$ in Theorem 1, we get the following lemma.

Lemma 2: The impulse response $\mathbf{h}$ of a linear filter which maximizes the SR of the output leakage L_o can be given by $\Sigma_{\mathbf{L}}^{-1}\mathbf{a}$.

We, now, state and prove our final result in Theorem 2. Before that let us denote by $\mu_{\mathbf{L}}$ the mean leakage vector $E[\mathbf{L}] = \{E[L_0], \cdots, E[L_{\tau-1}]\}$.

Theorem 2: Let the leakage $\mathbf{L}$ follows Eq. (17). The linear FIR filter with impulse response $\mathbf{h}_{\text{opt}} = \Sigma_{\mathbf{L}}^{-1}\mu_{\mathbf{L}}$ maximizes the SNR of the output leakage $L_o = \mathbf{h}_{\text{opt}}'\mathbf{L}$.

Proof: If we let $\mathbf{h}_{\text{opt}} = \Sigma_{\mathbf{L}}^{-1}\mathbf{a}$, according to Lemma 2, $\mathbf{h}_{\text{opt}}$ optimizes the SR of L_o. Thus, according to Lemma 1, $\mathbf{h}_{\text{opt}}$ also optimizes the the SNR of L_o. Taking the expectation of both sides of Eq. (17), we get

$$\begin{aligned}
\mu_{\mathbf{L}} &= \mathbf{a}(E\left[I\right]+c) \\
\text{or,}\quad \mathbf{a} &= \mu_{\mathbf{L}}/(E\left[I\right]+c).
\end{aligned}$$

Putting this value of $\mathbf{a}$ into $\mathbf{h}_{\text{opt}} = \Sigma_{\mathbf{L}}^{-1}\mathbf{a}$, we get $\mathbf{h}_{\text{opt}} = \Sigma_{\mathbf{L}}^{-1}\mu_{\mathbf{L}}/(E[I]+c)$. Since a constant factor in the impulse response of a filter does not have any affect on the SNR of the output, by neglecting the constant factor, we get $\mathbf{h}_{\text{opt}} = \Sigma_{\mathbf{L}}^{-1}\mu_{\mathbf{L}}$. ■

Thus, the impulse response of an optimum linear filter can be computed using the expression $\Sigma_{\mathbf{L}}^{-1}\mu_{\mathbf{L}}$. It should be noted that both $\Sigma_{\mathbf{L}}$ and $\mu_{\mathbf{L}}$ can be estimated without using the correct key. Hence, the filter can be useful in non-profiling DPA also.

VI. Approximating the Optimum Linear FIR Filter

Computation of $\Sigma_{\mathbf{L}}^{-1}\mu_{\mathbf{L}}$ involves the computation of the inverse of a $\tau \times \tau$ matrix which has a computational complexity $\mathcal{O}(\tau^3)$. Moreover, the inverse operation is highly susceptible to the error in the estimation of the covariance matrix. To avoid this operation, we note that the diagonal elements $c_{t,t}$ of the matrix $\Sigma_{\mathbf{L}} = \{c_{t_1,t_2}\}_{0 \le t_1,t_2 < \tau}$ are the variance of the leakage L_t and the off-diagonal elements c_{t_1,t_2}, where $t_1 \neq t_2$, are the covariance between L_{t_1} and L_{t_2}. Let us approximate the leakage covariance matrix $\Sigma_{\mathbf{L}}$ by setting all of its off-diagonal elements to zero. Thus, the approximated covariance matrix $\tilde{\Sigma}_{\mathbf{L}}$ is a diagonal matrix having the diagonal elements $\{c_{0,0}, \cdots, c_{\tau-1,\tau-1}\} = \{\sigma_{L_0}^2, \cdots, \sigma_{L_{\tau-1}}^2\}$. Consequently, the impulse response of the approximation of the optimum linear filter is given by

$$\mathbf{h}_{\text{appr}} = \tilde{\Sigma}_{\mathbf{L}}^{-1}\mu_{\mathbf{L}} = \left\{\frac{E\left[L_0\right]}{\sigma_{L_0}^2}, \cdots, \frac{E\left[L_{\tau-1}\right]}{\sigma_{L_{\tau-1}}^2}\right\} \tag{29}$$

It should be noted that the approximate optimum filter given by Eq. (29) neglects the correlation between the leakages of two different sample points. Thus, it is more suitable in scenarios where leakages of different sample points are loosely correlated.

When the leakages of different sample points are significantly correlated, the approximation of Eq. (29) might result into sub-optimal pre-processing. To avoid this, the leakage $\mathbf{L} = \{L_0, \cdots, L_{\tau-1}\}$ can be transformed into a new basis system $\tilde{\mathbf{L}} = \{\tilde{L}_0, \cdots, \tilde{L}_{\tau-1}\}$ by some linear transformation such that the leakage components along two different axes $\tilde{L}_{t_1}$ and $\tilde{L}_{t_2}$ become uncorrelated. Here, we discuss two such basis conversions.

Eigenvector Domain: In eigenvector domain, the basis is given by the set of eigenvectors of the covariance matrix of the original data-set. In this new basis, components along different eigenvectors (referred to as Principal Components or PCs) are uncorrelated to each other. Principal Component Analysis (PCA) [9] is a means to convert a data set into the basis of eigenvectors. PCA also sorts the PCs by their variance i.e. the first PC has maximum variance, the second PC has second maximum variance, and so on. Thus, in low noise scenario, where most of variations in traces is due to the target S, PCA projects the data dependent variations (signal) into the first PC while variations in all other PCs are mainly caused by noise. As a result, performing DPA on the first PC greatly increases performance of a DPA attack [2], [3], [31]. However in high noise scenario, data dependent variations are rather scattered among all the PCs [3], [15]. Since, PCA is a linear transformation [9], Eq. (17) is valid in the domain of eigenvector also. Consequently, we can apply the approximate optimal linear filter given by Eq. (29) on this domain i.e. on the PCs.

Frequency Domain: Other alternative is to use Discrete Fourier Transform (DFT) to convert the leakage samples into frequency domain which can be achieved using only $\mathcal{O}(\tau log(\tau))$ operations. In frequency domain, the absolute value of the complex coefficients obtained from the DFT is commonly used to attack [11], [24]. By taking only the absolute value, phase component is ignored which is useful to attack misaligned traces. However, we do not use it since the absolute operation is not a linear operation. Rather, we keep both the real part (cosine coefficient) and the imaginary part (sine coefficient) as separate sample points. Since, both the real and the imaginary parts are obtained using linear transformations, the resulting DFT traces also follows Eq. (16) and (17). Moreover, even if there exist significant correlations among different sample points in time domain, we can assume the covariance matrix of the leakages in frequency domain as sparse. Hence, we can apply the approximate optimal filter given by Eq. (29) for optimal pre-processing in this domain.

Computational Complexity: To estimate the impulse response of the optimum linear filter, $\mathbf{h}_{\text{opt}} = \mathbf{\Sigma}_{\mathbf{L}}^{-1}\mu_{\mathbf{L}}$, using q traces of size τ, one needs $\mathcal{O}(q\tau^2)$ operations for estimating $\mathbf{\Sigma}_{\mathbf{L}}$ and $\mathcal{O}(\tau^3)$ operations for computing the inverse. Multiplication of $\mathbf{\Sigma}_{\mathbf{L}}^{-1}$ and $\mu_{\mathbf{L}}$ requires $\mathcal{O}(\tau^2)$ operations. Thus in total, it requires $\mathcal{O}(\tau^2(\tau + q))$ operations to compute. On the other hand, to compute $\mathbf{h}_{\text{appr}}$, one needs only $\mathcal{O}(q\tau)$ operations.

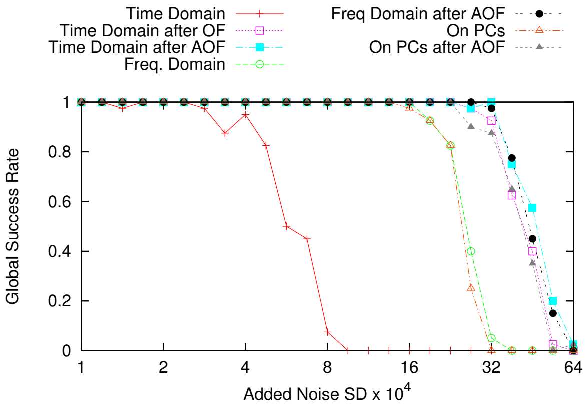


Fig. 7: Plots of the global success rate (GSR) of CPA after applying various pre-processing techniques on real traces of AES encryption. The GSR is computed over 40 sets of 3000 power traces.

VII. Experimental Results

For experimental evaluation, we have collected 40 sets of 3000 traces of AES encryptions. The cipher is implemented using parallel iterative hardware architecture on SASEBO-GII using the setup described in Appendix A. The S-boxes are implemented using Xilinx device primitive: distributed ROM. The setup is properly calibrated to reduce the quantization noise.

The effectiveness of the two proposed pre-processing techniques has been evaluated by comparing the success rate of CPA performed on the pre-processed traces by (1) the optimum filter (OF) and (2) the approximate optimum filter (AOF) with the success rate of CPA performed on (3) all the sample points independently. The attacks are performed on the power traces in three domains: time domain, frequency domain and eigenvector domain i.e. on the PCs. Fig. 7 shows global success rate [1] of CPA after applying all the above pre-processing. For parallel implementation of AES, global success rate is defined by the probability of getting the correct subkey for all the 16 bytes simultaneously. The figure shows that CPA performs better on the pre-processed traces in each of the three domains.

We have further evaluated the pre-processing techniques by adding a constant noise to each of the sample points of the traces. Such noise may be caused by vertical misalignment of the power traces. In the presence of constant noise, leakages of the different sample points get positively correlated. Thus filtering using AOF, which neglects the correlation between the leakages of two different sample points, becomes sub-optimal. This can be seen in Fig. 8. The figure shows the GSR of CPA on the output of AOF in time domain is badly affected by the constant noise. However, AOF in frequency domain and on the PCs performs almost optimally since, in the new basis, the sample points get sparsely correlated.

To test the effect of error in the estimation of covariance matrix $\mathbf{\Sigma}_{\mathbf{L}}$ when the number of power traces is small, we performed the attacks with the increasing number of power

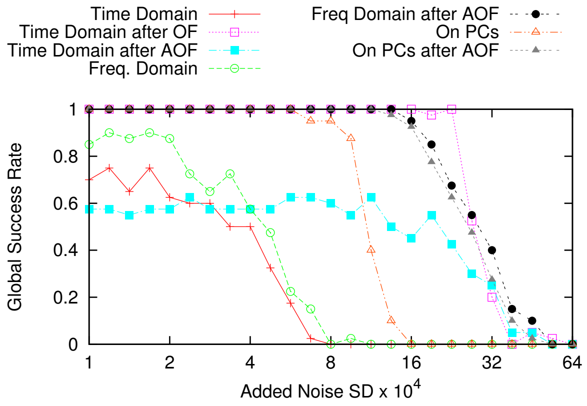


Fig. 8: Plots of the global success rate (GSR) of CPA after applying various pre-processing techniques on AES encryption traces. A constant noise is added to each sample point of the traces and then GSR is computed by adding independent Gaussian noise to each sample points of increasing variance.

traces on the original acquired traces. As shown in Fig. 9, CPA on the output of OF does not performs well for lesser number of power traces. The performance of CPA on the output of AOF on the PCs is also slightly effected when the number of power traces is small. However, the success rate of the CPA on the output of AOF in both time and frequency domains reaches to one faster than all other attacks.

In conclusion, we can say that CPA performs better on the output of OF when the number of power traces is sufficiently large. But, it performs worse when the number of power traces is less due to erroneous estimation of the covariance matrix $\mathbf{\Sigma_L}$. On the other hand, AOF provides a computationally efficient alternative to OF which also performs well when the number of power traces is small. However in the presence of highly correlated noise, AOF in time domain is not a good choice since it neglects the correlations between the leakages of two different sample points. This shortcoming of AOF can be circumvented by transforming the power traces into a new domain like frequency domain where the leakages of multiple sample points become sparsely correlated.

VIII. Optimality of OF

In this section, our objective is to experimentally verify the optimality of OF as a technique for combining the leakages of multiple sample points. We assume that a multivariate profiling attack can exploit the leakages of multiple sample points optimally, thus, gives the limit of the improvement in the success rate which can be achieved by combining the leakages of multiple sample points. Hence, the optimality of OF has been verified by comparing the success rate of a univariate profiling attack performed on the output of OF with the success rate of the multivariate profiling attack performed on the unprocessed power traces. We choose the Stochastic attack [29] as the profiling attack since it can "learn" quickly using smaller number of traces [13].

A. Experimental Evaluation on Simulated Traces

For the experimental evaluation of OF, we generated simulated leakage using HW model. The leakage of 300-dimension, $\mathbf{L}$, has been generated as $\mathbf{L} = \mathrm{HW}(\mathrm{sbox}(X \oplus k^*))\mathbf{a} + \mathbf{N}$ where $\mathbf{a}$ is a chosen vector of 300 elements. The attacks are performed with increasing variance of the noise. In the profiling phase, 100000 traces have been used and in the attack phase, a different set of 100000 traces has been used.

Fig.10 plots the success rate of Stochastic attack performed (1) on all sample points, (2) on the output of MF (matched filter) (3) on the output of OF. The figure shows that the success rate obtained in the three attacks are same.

B. Experimental Evaluation on Real Traces

For experimental evaluation of the optimality of OF on real traces, we performed the profiling Stochastic attack on the same set of 120000 traces used in Section VII. Three variants of the attack have been performed: (1) multivariate Stochastic attack on all the sample points, (2) univariate Stochastic attack at the sample point where leakage is maximum, and (3) univariate Stochastic attack on the output of OF. For all the

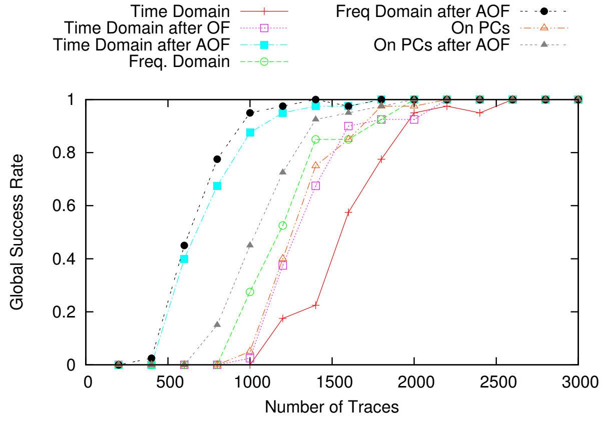


Fig. 9: Global success rate (GSR) is computed with the increasing number of power traces without adding any noise.

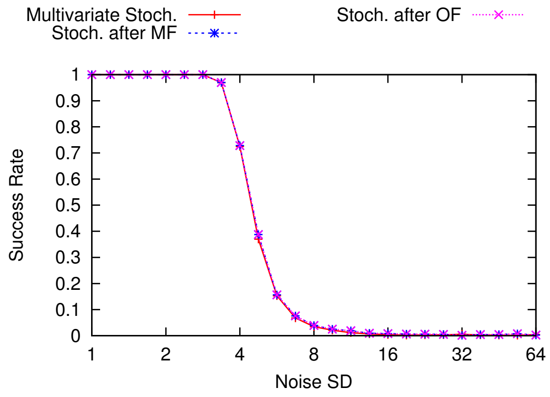


Fig. 10: Plots of the success rate of profiling Stochastic attack on simulated power traces.

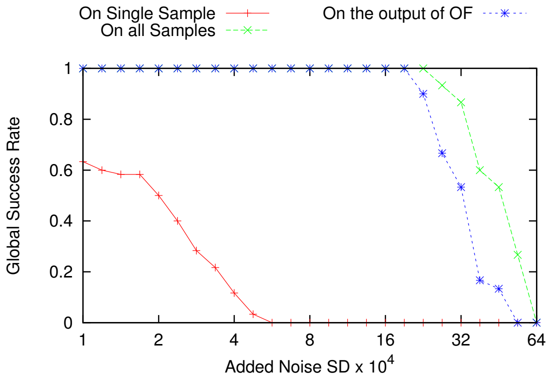


Fig. 11: Plots of the global success rate (GSR) of profiling Stochastic attack on real traces.

three attacks, Hamming distance model has been used and the same set of 120000 power traces has been used for both the profiling and attack phase. Fig.11 plots the global success rate (GSR) obtained using the three attacks. The plots of the GSR obtained using the univariate Stochastic attack on the sample point having maximum leakage shows the limit of the GSR which can be obtained by exploiting the leakage of a single sample point. The plots for the multivariate Stochastic attack on all the sample points shows the limit of the improvement in the GSR which can be obtained by optimally combining the leakages of all the sample points. The plots for the univariate Stochastic attack on the output of OF shows that the GSR of the attack has been improved by the pre-processing using OF and reaches very close to the GSR of multivariate Stochastic attack.

IX. Conclusion

The paper has investigated the optimization of non-profiling multivariate DPA attacks by linearly combining the leakages of multiple sample points. The investigation has been carried over in three parts. In the first part, a theoretical study of the factors influencing the DPA attacks has been conducted. The study has established a theoretical relationship among the success rate of CPA, number of power traces used in the attack, the SNR of the power traces, and other algorithm and model dependent parameters. In the second part, a multivariate leakage model for Xilinx Virtex-5 FPGA device has been proposed. The proposed model has also been experimentally verified. In the third part, optimum pre-processing of the power traces using linear filter has been studied. We have proposed two pre-processing techniques for the non-profiling DPA attacks based the proposed multivariate leakage model. Optimality of one pre-processing techniques has been empirically evaluated on both simulated and real traces.

References

[1] Dpa contest/v2/, http://www.dpacontest.org/v2/index.php, 2012.

[2] C. Archambeau, E. Peeters, F.-X. Standaert, and J.-J. Quisquater. Template Attacks in Principal Subspaces. In Goubin and Matsui [14], pages 1–14.

[3] L. Batina, J. Hogenboom, and J. G. J. van Woudenberg. Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis. In O. Dunkelman, editor, *CT-RSA*, volume 7178 of *Lecture Notes in Computer Science*, pages 383–397. Springer, 2012.

[4] S. Bhasin, J.-L. Danger, S. Guilley, and Z. Najm. NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage. Cryptology ePrint Archive, Report 2013/717, 2013.

[5] E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In M. Joye and J.-J. Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[6] S. Chari, J. R. Rao, and P. Rohatgi. Template Attacks. In B. S. K. Jr., Çetin Kaya Koç, and C. Paar, editors, *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

[7] C. Clavier, J.-S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Çetin Kaya Koç and C. Paar, editors, *CHES*, volume 1965 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2000.

[8] S. Dowdy, S. Wearden, and D. Chilko. *Statistics for research*. John Wiley & Sons, third edition, 2004.

[9] R. O. Duda, P. E. P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, pub-WILEY:adr, second edition, 2001.

[10] Y. Fei, Q. Luo, and A. A. Ding. A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In E. Prouff and P. Schaumont, editors, *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 233–250. Springer, 2012.

[11] C. H. Gebotys, S. Ho, and C. C. Tiu. EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In Rao and Sunar [27], pages 250–264.

[12] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual Information Analysis. In E. Oswald and P. Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.

[13] B. Gierlichs, K. Lemke-Rust, and C. Paar. Templates vs. Stochastic Methods. In Goubin and Matsui [14], pages 15–29.

[14] L. Goubin and M. Matsui, editors. *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*. Springer, 2006.

[15] S. Hajra and D. Mukhopadhyay. Pushing the Limit of Non-Profiling DPA using Multivariate Leakage Model. Cryptology ePrint Archive, Report 2013/849, 2013.

[16] T. Katashita, A. Satoh, T. Sugawara, N. Homma, and T. Aoki. Development of side-channel attack standard evaluation environment. In *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*, pages 403–408, 2009.

[17] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[18] B. Köpf and D. A. Basin. An Information-Theoretic Model for Adaptive Side-Channel Attacks. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 286–296. ACM, 2007.

[19] S. Mangard. Hardware Countermeasures against DPA ? A Statistical Analysis of Their Effectiveness. In T. Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

[20] S. Mangard, E. Oswald, and T. Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[21] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *In USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.

[22] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *In USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.

[23] A. Moradi, O. Mischke, and T. Eisenbarth. Correlation-Enhanced Power Analysis Collision Attack. In S. Mangard and F.-X. Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2010.

[24] D. Oswald and C. Paar. Improving Side-Channel Analysis with Optimal Linear Transforms. In S. Mangard, editor, *CARDIS*, volume 7771 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2012.

[25] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 3rd edition, Feb. 1991.

[26] C. R. Rao. *Linear statistical inference and its applications*. Wiley, 1973.

[27] J. R. Rao and B. Sunar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*. Springer, 2005.

[28] M. Rivain. On the Exact Success Rate of Side Channel Analysis in the Gaussian Model. In R. M. Avanzi, L. Keliher, and F. Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 165–183. Springer, 2008.

[29] W. Schindler, K. Lemke, and C. Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In Rao and Sunar [27], pages 30–46.

[30] J. Sills and E. Kamen. Time-varying matched filters. *Circuits, Systems and Signal Processing*, 15(5):609–630, 1996.

[31] Y. Souissi, M. Nassar, S. Guilley, J.-L. Danger, and F. Flament. First Principal Components Analysis: A New Side Channel Distinguisher. In K. H. Rhee and D. Nyang, editors, *ICISC*, volume 6829 of *Lecture Notes in Computer Science*, pages 407–419. Springer, 2010.

[32] F.-X. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[33] O.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater. An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays. *Proceedings of the IEEE*, 94(2):383–394, Feb 2006.

[34] A. Thillard, E. Prouff, and T. Roche. Success through Confidence: Evaluating the Effectiveness of a Side-Channel Attack. In G. Bertoni and J.-S. Coron, editors, *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2013.

[35] N. Veyrat-Charvillon and F.-X. Standaert. Adaptive Chosen-Message Side-Channel Attacks. In J. Zhou and M. Yung, editors, *ACNS*, volume 6123 of *Lecture Notes in Computer Science*, pages 186–199, 2010.

[36] Wikipedia. Matched filter — Wikipedia, The Free Encyclopedia, 2013. [Online; accessed 20-December-2013].

Appendix A
Experimental Setup and Pre-processing

For all the experiments, we have used standard side-channel evaluation board SASEBO-GII [16]. It consists of two FPGA device: Spartan-3A XC3S400A and Virtex-5 XC5VLX50. Spartan-3A acts as the control FPGA where as Virtex-5 contains the target cryptographic implementation. The cryptographic FPGA is driven by a clock frequency of 2 MHz. During the encryption process, voltage drops across VCC and GND of Virtex-5 are captured by Tektronix MSO 4034B Oscilloscope at the rate of 2.5 GS/s i.e. $1,250$ samples per clock period.

The traces acquired using the above setup are already horizontally aligned. However, they are not vertically aligned. The vertical alignment of the traces are performed by subtracting the DC bias from each sample point of the trace. The DC bias of each trace is computed by averaging the leakages of a window taken from a region when no computation is going on. This step is also necessary since the derived impulse response of the filters is sensitive to the absolute value of mean leakages.

For mounting the attacks, we selected a window of 300 sample points around the last round register update. After transforming into a different domain, variance of some of the sample points may become very close to zero in the new domain. As a result, while applying AOF in this new domain, the weights (which are mean/variance of the sample points) of those sample points may become very high even if their mean values are very less. In other words, due to very low variance, some low SNR sample points may get very high weight. We solved this problem by setting the weight of a sample point having variance less than a fraction of $1/500$ of the maximum variance to zero.

Appendix B
Degenerate Normal Distribution

To have a density, the covariance matrix $\mathbf{\Sigma}$ of the multivariate normal distribution $\mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$ must be a positive definite matrix [26](i.e. all of its eigenvalues must be positive). When the covariance matrix is positive semi-definite, the distribution is said to be degenerate normal distribution and does not have a density. In that case, we converted it into a positive definite matrix by the following way. We first performed eigendecomposition of $\mathbf{\Sigma}$ by factoring it into $\mathbf{Q}\Lambda\mathbf{Q}'$ where each column of $\mathbf{Q}$ represents a eigenvector of $\mathbf{\Sigma}$ and Λ is a diagonal matrix whose j^{th} diagonal element is the eigenvalue of the j^{th} eigenvector. Then, we computed a new positive definite matrix $\mathbf{\Sigma}_1 = \mathbf{Q}\Lambda_1\mathbf{Q}'$ where Λ_1 is obtained by replacing the diagonal elements having zero (in practice, elements having values below a cut-off value) by a very small positive value. The new matrix $\mathbf{\Sigma}_1$ is then used in the multivariate density.

Appendix C
Proof of Theorem 1

A formal proof of the theorem can be found in [30]. However, for the shake of completeness, we provide a proof following the proof in [36]. In Eq. (12), SNR is given as,

$$\mathrm{SNR}_{L_o} = \frac{|\mathbf{h}'\mathbf{a}|^2\sigma^2_{k^*}}{\mathbf{h}'\Sigma_{\tilde{\mathbf{N}}}\mathbf{h}}$$

The term $\sigma^2_{k^*}$ in the RHS of the above expression does not have any influence when we maximize the SNR. Thus by neglecting it, we re-write the above expression as

$$\mathrm{SNR}_{L_o} = \frac{|\mathbf{h}'\mathbf{a}|^2}{\mathbf{h}'\Sigma_{\tilde{\mathbf{N}}}\mathbf{h}}$$

Now, if $\Sigma_{\tilde{\mathbf{N}}}$ is not invertible, a subset of the τ time instants of size $rank(\Sigma_{\tilde{\mathbf{N}}})$ can be chosen such that the covariance matrix of the chosen sample points is invertible and all the computations can be carried out in this lower dimension. Thus without loss of generality, we assume $\Sigma_{\tilde{\mathbf{N}}}$ is positive-definite. Thus, SNR can be written as

$$\mathrm{SNR}_{L_o} = \frac{|(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})'(\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a})|^2}{(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})'(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})}$$

Using the Cauchy-Schwarz inequality, the numerator of the RHS of the above equation can be bounded by

$$|(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})'(\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a})|^2 \leq [(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})'(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})][(\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a})'(\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a})]$$

Thus,

$$\begin{aligned}\mathrm{SNR}_{L_o} &\leq \frac{[(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})'(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})][(\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a})'(\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a})]}{(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})'(\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h})} \\ &= \mathbf{a}'\Sigma^{-1}_{\tilde{\mathbf{N}}}\mathbf{a}\end{aligned}$$

Moreover, this upper bound is achieved when $\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h}$ and $\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a}$ are linearly dependent or $\Sigma^{1/2}_{\tilde{\mathbf{N}}}\mathbf{h} = \alpha\Sigma^{-1/2}_{\tilde{\mathbf{N}}}\mathbf{a}$, which simplifies to $\mathbf{h} = \alpha \Sigma_{\tilde{\mathbf{N}}}^{-1} \mathbf{a}$ for some normalization factor α. Setting the value of α to one, we complete the proof.