CC Meets FIPS: A Hybrid Test Methodology for First Order Side Channel Analysis

Debapriya Basu Roy, Shivam Bhasin, Sylvain Guilley, Annelie Heuser, Sikhar Patranabis, Debdeep Mukhopadhyay

CC meets FIPS: A Hybrid Test Methodology for First Order Side Channel Analysis

Debapriya Basu Roy, Shivam Bhasin, Sylvain Guilley, Annelie Heuser, Sikhar Patranabis, Debdeep Mukhopadhyay

Abstract—Common Criteria (CC) and FIPS 140-3 are two popular side channel testing methodologies. Test Vector Leakage Assessment Methodology (*TVLA*), a potential candidate for FIPS, can detect the presence of side-channel information in leakage measurements. However, *TVLA* results cannot be used to quantify side-channel vulnerability and it is an open problem to derive its relationship with side channel attack success rate (*SR*), i.e. a common metric for CC. In this paper, we extend the *TVLA* testing beyond its current scope. Precisely, we derive a concrete relationship between *TVLA* and signal to noise ratio (*SNR*). The linking of the two metrics allows direct computation of success rate (*SR*) from *TVLA* for given choice of intermediate variable and leakage model and thus unify these popular side channel detection and evaluation metrics. An end-to-end methodology is proposed, which can be easily automated, to derive attack *SR* starting from *TVLA* testing. The methodology works under both univariate and multivariate setting and is capable of quantifying any first order leakage. Detailed experiments have been provided using both simulated traces and real traces on SAKURA-GW platform. Additionally, the proposed methodology is benchmarked against previously published attacks on *DPA contest v4.0* traces, followed by extension to jitter based countermeasure. The result shows that the proposed methodology provides a quick estimate of *SR* without performing actual attacks, thus bridging the gap between CC and FIPS.

Index Terms—Side Channel, Evaluation Based Testing, Validation Based Testing, *TVLA*, *NICV*

1 Introduction

Since the seminal work by Kocher et al. [1], side channels have emerged as a serious threat to implementations of cryptographic algorithms in the past two decades, with the ability to render even mathematically robust cryptographic algorithms vulnerable. A side-channel adversary observes the physical properties of a cryptographic implementation, such as timing, power or electromagnetic emanations, and tries to infer the secret key by modeling a sensitive intermediate state of the design which depends on these physical properties. Cryptographic designs must, therefore, provide security guarantees against such threats. In this context, efficient validation and evaluation methodology for testing side channel vulnerability has gathered significant interest in the research community. In particular, there exist today, two popular security certification programs - Common Criteria (CC) [2] and FIPS [3] that recommend crypto-implementations to be secure against side channel attacks. Each of these programs follows two distinct testing methodologies, namely *evaluation-style testing*, and *conformance-style testing*.

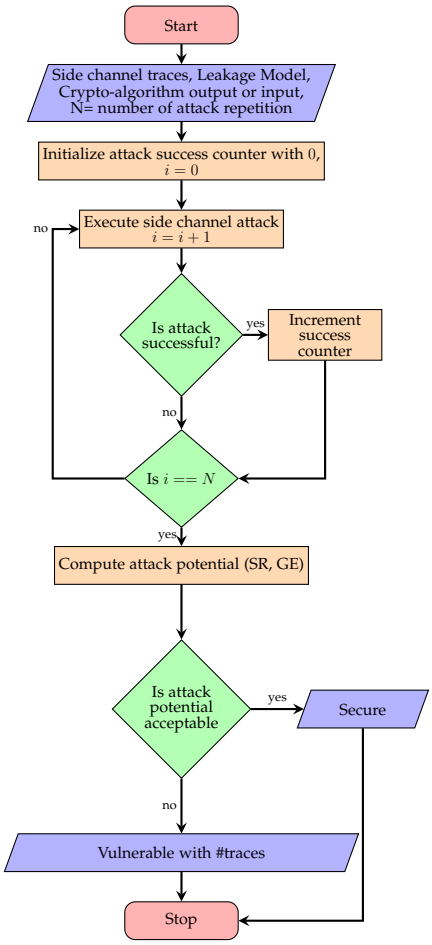
1.1 Evaluation-style Testing.

The Common Criteria (CC) certification is a prime example of evaluation-style testing. CC is essentially a set of security guidelines (ISO-15408) that define a common framework for evaluating cryptographic implementations using a standard set of pre-defined evaluation assurance levels. A typical evaluation based testing mechanism is shown in Fig. 1(a). From the point of view of detecting side channel vulnerabilities, it recommends evaluating the system against all state-of-the-art attack strategies, with the knowledge of the threat model. The evaluator needs to perform different side channel attacks starting from simple power attacks to higher order differential power attacks with different leakage models. Additionally, each of these attacks is repeated multiple times to compute metrics like success rate (*SR*). An ever-increasing list of attack strategies, together with a large number of models characterizing different leakage profiles of the device, often renders such a testing methodology cumbersome, costly and limited by the testing expertise available at hand. Additionally, the success of evaluation-style testing methodologies depends strongly on appropriate choices of the leakage models, and an error of judgement in this regard could cause a potentially vulnerable crypto-implementation to pass the test. This makes evaluation style testing mechanisms costly and dependent on lab expertise.

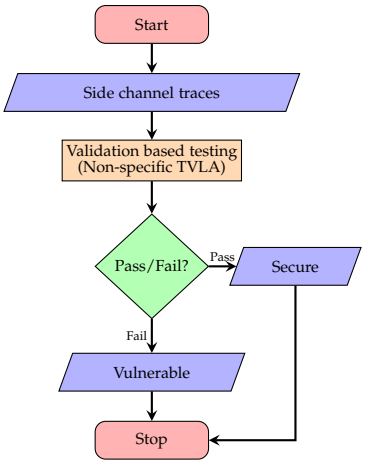
1.2 Conformance-Style Testing.

Unlike CC, FIPS [3] certification is an example of conformance-style testing that uses a cryptographic module validation program (CMVP) to validate target's compliance with necessary security levels rather than an exact evaluation of its vulnerability. With respect to side channels, it employs a simplified approach for merely detecting the presence of *any* leakage, independent of attack methodologies and leakage models. This makes it possible to have structured conformance-style testing methodologies that are

- *Debapriya Basu Roy, Sikhar Patranabish and Debdeep Mukhopadhyay are with Secured Embedded Architecture Laboratory (SEAL), IIT Kharagpur.*
- *Shivam Bhasin is with Temasek Laboratories, NTU.*
- *Annelie Heuser is with IRISA/CNRS, Rennes, France.*
- *Sylvain Guilley is with TELECOM-ParisTech, France and Secure-IC S.A.S., France.*



(a) Evaluation-style Testing



(b) Conformance-style Testing

Fig. 1: Existing Side Channel Testing Methodologies

cost-effective and consistent across different testing labs with varied testing expertise. Fortifications with precise security specifications and test plan coverage have the potential to make this style of testing against side-channel vulnerabilities highly efficient and suitable for wide-scale use. A typical conformance-style testing mechanism is shown in Fig. 1(b).

However, the conformance-style testing mechanism can only detect the presence of side channel vulnerability, it is not capable of quantifying the side channel vulnerability. On the other hand, though evaluation-style testing mechanism has many disadvantages, it can quantify side channel susceptibility, while also finding application in comparing vulnerability of two designs.

Test Vector Leakage Assessment (*TVLA*) [4] which was proposed at NIST sponsored NIAT workshop in 2011, is one of the well-known conformance style testing mechanism which has gained popularity among the researchers and especially the practitioners due to its robustness, applicability to different crypto-implementations and easy integrability with the existing testing methodologies. Multiple research papers (e.g. [5]) on side channel attacks have used this tool to show the effectiveness of their proposed attacks and countermeasures. *TVLA* uses the well known *Welch's t-test*. It was proposed as a *PASS/FAIL* test, which checks if t-value crosses the pre-defined threshold (proposed as ± 4.5 [4]). If the t-value crosses the threshold, the measurement is considered to carry data dependent information, which could be potentially exploited.

TVLA can be classified into: *non-specific* and *specific* [4]. Non-specific *TVLA* partitions traces on basis of public inputs (usually plaintext). Specific *TVLA* partitions based on intermediate key-dependent variables and thus can provide intuitions on the source of leakage. It has been shown in [6] that non-specific *TVLA* outperforms specific *TVLA* as the number of false positives will be less in case of non-specific *TVLA*. Both methods are discussed in details in section 2.

Being a conformance style testing, one demerit of *TVLA* methodology is that, a failed t-test may or may not lead to successful key extraction. In other words, we can not quantify the side channel vulnerability of a crypto-systems using the results of this t-test. Quantifying side channel vulnerability requires the knowledge of leakage models and adversary capability. As discussed earlier, evaluation style testing can achieve this objective, albeit with very high cost. In current form, the result of t-test can not be used for such side channel vulnerability quantification.

In this paper, we propose a hybrid testing methodology which has the simplicity of conformance-style testing along with the capability of side channel vulnerability quantification. Our main idea is to use *specific TVLA* to extract more information regarding the side channel vulnerability of the underlying crypto-implementation. This extracted information can then be expressed in terms of evaluation metrics like signal-to-noise ratio (*SNR*) and attack success rate (*SR*). Thus, we first derive the formal relationship between *TVLA* and *SNR*. Based on the derived formulation, the hybrid methodology is developed which allows an evaluator to quantify side-channel vulnerability through *SR*, starting from a basic *TVLA* analysis. In a nutshell, the proposed methodology bridges the gap between conformance-style and evaluation style testing. We have provided more details on this in section 4. The detailed flow chart of the proposed testing methodology is shown in Fig. 2.

The proposed methodology is also extended to the multivariate setting. As the objective of the proposed testing methodology is to evaluate real products, multivariate analysis can be often required. For instance, often commercial smart cards have a built-in clock jitter which causes measurement misalignment. A univariate testing will lead to sub-optimal results, while a multivariate analysis could combine leakages spread over different samples (due to jitter) and evaluate in an optimal manner.

1.3 Related Work

A unified framework to evaluate side channel attack was proposed by Standaert et al. [7]. It puts forward two key metrics success rate (*SR*) and guessing entropy (*GE*) as main attack metrics. The success rate of a specific side channel attack is defined as the probability of successful secret key retrieval. In simple mathematical notation, success rate (*SR*) of a side channel attack (A) is presented as follows:

$$SR = Pr[A(E_{k_0}, L) = k_0] \tag{1}$$

where k_0 is the correct key used in the encryption process E_{k_0}, L is the leakage obtained from side channel traces. In CHES 2012, Fei et al. [8] introduced the notion of confusion coefficient which can be used to compute theoretical success rate of a mono-bit differential power analysis (i.e. difference of mean) given the *SNR*. This work was further improved and extended to correlation power analysis by Thillard et al. [9]. Fei et al. [10] also extended the initial work on success rate estimation for mono-bit DPA to CPA and beyond.

On the other hand, to simplify the evaluation process, simple and model-agnostic techniques were also developed in parallel. The main technique of this class is the previously mentioned *TVLA* [4], which was proposed as a FIPS 140-3 candidate. Another simple method to detect point of leakage in a univariate first-order setting was proposed in [11], termed as *Normalized Inter Class Variance (NICV)*. Authors show that NICV is an estimate of *SNR* and approaches (squared) Pearson's correlation coefficient in absence of noise. *NICV* is actually the output of statistical F-test (also known as ANOVA (ANalysis Of VAriance)). Owing to its relationship to *SNR*, *NICV* was also used to derive *SR* for mono-bit DPA using formulation from [8]. In this work, we work on connecting the individual techniques to develop the whole chain. The main missing link in the above techniques is the relationship between *TVLA* and *SNR*. By developing that link, we are able to develop a methodology that can be automated end to end to estimate attack *SR* right from computation of specific *TVLA*.

1.4 Contribution

The main contributions of this paper are as follows:

- *SNR* of side-channel measurement and *TVLA* (both *specific and non-specific*) are independently developed metrics. We derive the relationship between *SNR* and *TVLA*. We formally show that the two metrics are equivalent.
- Next, we devise a methodology to estimate the theoretical bounds for the success rate of an attack from the *specific TVLA* results. This, to our knowledge, is the first attempt to extend *specific TVLA* results for quantification of side channel vulnerability through *SR*. The methodology uses theoretical success rate formulation for CPA by Fei et al. [10]. In other words, the developed methodology attempts to bridge the gap between conformance and evaluation based testing by setting the following chain: $Specific\,TVLA \rightarrow SNR \rightarrow SR$.
- We also show that using *non-specific TVLA* to estimate *SNR* is impractical, thus motivating the usage of *specific TVLA*.
- The developed methodology is extended to multivariate setting under first-order leakage setting.
- The methodology is practically demonstrated on unprotected AES implementation on an 8-bit microcontroller as well as publicly available traces of protected AES implementation (with accidental leakage) of DPA Contest v4.0 [12]. We validate the proposed methodology by showing a close match between our predicted SR and the practical SR achieved from the attack published in [13]

The rest of the paper is organized as follows: Section 2 briefly describes the mathematics behind different metrics for validation and evaluation of side channel vulnerabilities. Next, section 3, derives the relationship between *Welch's t-test* based *TVLA* and *ANOVA* based *NICV* (and *SNR*). Section 4 introduces the proposed hybrid design methodology for side channel vulnerability quantification. The proposed formulation is experimentally validated in section 5 followed by application of the hybrid methodology to AES in section 6. The extension of the proposed methodology to multivariate setting is discussed in section 7 followed by final conclusions in section 8.

2 Preliminaries

In this section, we introduce the notations used throughout the paper, along with brief definitions for the following concepts: *TVLA*, *NICV*, *SNR*, and *SR*. Finally, the previously proposed relationship between *SR* and *SNR* is discussed.

2.1 Notations Used

We denote by X and k a single plaintext byte and key byte, respectively. We also denote by $L = l(X, k)$ the normalized leakage model such that $\mathsf{E}(L) = 0$ and $\mathsf{Var}(L) = \mathsf{E}(L^2) = 1$. Finally, we denote by Y the leakage measurement such that

$$Y = \epsilon L + N \tag{2}$$

where ϵ is the scaling coefficient and $N \sim \mathcal{N}(0, \sigma^2)$ is the noise component, which is independent of X. Note that the derivations in this paper are based on Eqn. (2). A commonly encountered example for $l(X, k)$ is the Hamming weight leakage model on n bits, represented as:

$$l(X, k) = \frac{2}{\sqrt{n}} \left(HW(Sbox(X \oplus k)) - \frac{n}{2} \right)$$

where $Sbox$ denotes the substitution operation. $Sbox(X \oplus K)$ is the intermediate variable whose value is mapped to side-channel leakage by the leakage model (HW, for example). With the above notations in place, we present brief definitions for the different side channel metrics used in this paper.

2.2 Signal-to-Noise Ratio

Definition 1. SNR [14, § 4.3.2, page 73] The Signal-to-Noise Ratio (SNR) is defined as:

$$\mathsf{SNR} = \frac{\mathsf{Var}(\mathsf{E}(Y|X))}{\mathsf{E}(\mathsf{Var}(Y|X))} \tag{3}$$

Lemma 1 (SNR in the case of leakage model (2)***).***

$$\mathsf{SNR} = \frac{\epsilon^2}{\sigma^2} \tag{4}$$

Proof 1. Let x be a plaintext, and $l = l(x, k)$. Then $\mathsf{E}(Y|X = x) = \mathsf{E}(\epsilon L + N|L = l) = \epsilon l$, by expression of the model (2) and noise independence from the L. Therefore, $\mathsf{Var}(\mathsf{E}(Y|X)) = \mathsf{Var}(\epsilon L) = \epsilon^2$. Besides, $\mathsf{E}(\mathsf{Var}(Y|X)) = \mathsf{E}(\sigma^2) = \sigma^2$. Hence, $\mathsf{SNR} = \frac{\mathsf{Var}(\mathsf{E}(Y|X))}{\mathsf{E}(\mathsf{Var}(Y|X))} = \frac{\epsilon^2}{\sigma^2}$.

2.3 Normalized Inter Class Variance

Normalized Inter-Class Variance (*NICV*) is a technique which was designed to detect relevant point(s) of interest (PoI) in an SCA trace [11]. It has application in side channel trace compression and dimensionality reduction. NICV is based on *ANOVA (ANalysis Of VAriance)* or *F-test* [15]. The main advantage of NICV is that it is leakage model agnostic, and can be applied with the knowledge of only plain-text or cipher-text and does not require knowledge of target implementation or secret key.

Definition 2 (NICV [11, Eqn. (4) of Sec. 3.1]). The Normalized Inter-Class Variance (NICV) is defined as:

$$\mathsf{NICV} = \frac{\mathsf{Var}(\mathsf{E}(Y|X))}{\mathsf{Var}(Y)}. \tag{5}$$

Lemma 2 (NICV in the case of leakage model (2)***).***

$$\mathsf{NICV} = \frac{1}{1+\frac{\sigma^2}{\epsilon^2}}. \tag{6}$$

In particular, $0 \leq \mathsf{NICV} \leq 1$.

Proof 2. The numerator has already been proven to be equal to ϵ^2. Besides, $\mathsf{Var}(Y) = \mathsf{Var}(\epsilon L) + \mathsf{Var}(N) = \epsilon^2 + \sigma^2$, by independence of X and N. Hence $\mathsf{NICV} = \frac{\mathsf{Var}(\mathsf{E}(Y|X))}{\mathsf{Var}(Y)} = \frac{\epsilon^2}{\epsilon^2+\sigma^2} = \frac{1}{1+\frac{\sigma^2}{\epsilon^2}}$.

Proposition 1 (Link between NICV and SNR [11, Eqn. (5) of Sec. 3.1]). We have:

$$\mathsf{NICV} = \frac{1}{\frac{1}{\mathsf{SNR}}+1} \quad \text{and, conversely,} \quad \mathsf{SNR} = \frac{1}{\frac{1}{\mathsf{NICV}}-1}. \tag{7}$$

Proof 3. The proof follows from a direct application of the Lemmas 1 and 2.

2.4 Test Vector Leakage Assessment (TVLA)

Test Vector Leakage Assessment (TVLA) [4] is a direct application of *Welch's t-test* on side channel leakage traces for detection of vulnerabilities. The *TVLA* methodology can be classified into two different categories: *non-specific TVLA* and *specific TVLA*. For both the cases, one must acquire two sets of traces. In case of *non-specific TVLA*, the first set corresponds to a fixed key and fixed plaintext as input to the cryptographic IP, while the second set contains traces corresponding to the same fixed key and random plaintext. Thereafter a hypothesis testing performed by assuming a null hypothesis that these two sets of traces have identical means and variance. If the null hypothesis is accepted, it signifies that the traces carry no sensitive information. On the other hand, a rejected null hypothesis indicates the presence of exploitable leakage.

More specifically, the *non-specific TVLA* may be defined mathematically as follows:

Definition 3 (TVLA [4, page 7]). The non-specific TVLA is defined for Q queries as:

$$\widehat{\mathsf{TVLA}}_x = = \frac{\left(\frac{1}{\sum_{q/x_q=x} 1}\sum_{q/x_q=x} y_q\right) - \left(\frac{1}{\sum_q 1}\sum_q y_q\right)}{\sqrt{\frac{1}{\sum_{q/x_q=x} 1}\left(\frac{1}{\sum_{q/x_q=x} 1} y_q^2 - \left(\frac{1}{\sum_{q/x_q=x} 1} y_q\right)^2\right) + \frac{1}{\sum_q 1}\left(\frac{1}{\sum_q 1} y_q^2 - \left(\frac{1}{\sum_q 1} y_q\right)^2\right)}} \tag{8}$$

where $\sum_q$ denotes $\sum_{q=1}^{Q}$ and $\sum_{q/t_q=t}$ denotes $\sum_{1\leq q\leq Q, \text{ s.t. } t_q=t}$.

We notice that this test is consistent, in that, asymptotically,

$$\widehat{\mathsf{TVLA}}_x \xrightarrow[Q\to+\infty]{} \begin{cases} +\infty & \text{if } \mathsf{E}(Y|X=x) \neq \mathsf{E}(Y), \\ 0 & \text{otherwise.} \end{cases}$$

More precisely, according to the law of large numbers (LLN), we have that:

$$\widehat{\mathsf{TVLA}}_x \underset{Q\to+\infty}{\approx} \sqrt{Q}\frac{\mathsf{E}(Y|X=x)-\mathsf{E}(Y)}{\sqrt{\mathsf{Var}(Y|X=x)+\mathsf{Var}(Y)}}.$$

We therefore define the asymptotic constant $\lim_{Q\to+\infty} \frac{1}{\sqrt{Q}}\widehat{\mathsf{TVLA}}_x = \mathsf{TVLA}_x$ as:

Definition 4. Asymptotic constant for Test Vector Leakage Assessment (TVLA) for Fixed versus Random is:

$$\mathsf{TVLA}_x = \frac{\mathsf{E}(Y|X=x)-\mathsf{E}(Y)}{\sqrt{\mathsf{Var}(Y|X)+\mathsf{Var}(Y)}},$$

where the fixed plaintext is x. In this definition, the test is *non-specific*, since one does not need to know the key.

Lemma 3 (TVLA in the case of leakage model (2)***).***

$$\mathsf{TVLA}_x = \frac{\epsilon l(x,k)}{\sqrt{\epsilon^2+2\sigma^2}}.$$

Proof 4. Indeed, we have $\mathsf{E}(Y)=0$, hence the result follows.

For *specific TVLA*, knowledge of secret key is required as in this case the traces are partitioned depending upon the value of some intermediate data of crypto-execution [4]. Depending upon the choice of intermediate data, there could be multiple ways to do this partitioning. In [6], the superiority of *non-specific TVLA* over *specific TVLA* is established. *TVLA* is compared with mutual information based analysis techniques in [16] and comparative analysis between them is presented. In [5], authors have focused on the applicability of *TVLA*. They have extended application of *TVLA* to higher order attacks. Moreover, they have presented efficient algorithms for on-line computation of *TVLA*. A modified paired t-test based *TVLA* methodology is presented in [17]. A recent work [18] shows the limitations of t-test in security evaluation of a higher-order masking scheme, however, for first order evaluation it provides a good starting point.

2.5 SNR and SR

A closed-form expression for DPA and CPA has been derived in [8], [9], [10] that depends on three factors: number of measurements Q, SNR, confusion coefficient vector $\boldsymbol{\kappa}$, and confusion matrices $\boldsymbol{K}, \boldsymbol{K}^{**}$.

Definition 5 (Confusion vector and matrices for CPA [10]). Let k_c denote the secret key and k_{g_i} with $1 \leq i \leq 2^{n-1}$ a key guess where $k_{g_i} \neq k_c$, then the confusion vector $\boldsymbol{\kappa}$ and the confusion matrices $\boldsymbol{K}$,$\boldsymbol{K}^{**}$ are defined as

$$\boldsymbol{\kappa} = (\kappa(k_c, k_{g_1}), \ldots, \kappa(k_c, k_{g_{2^n-1}})^T$$

$$\boldsymbol{K} = \begin{pmatrix} \kappa(k_c, k_{g_1}, k_{g_1}) & \kappa(k_c, k_{g_1}, k_{g_2}) & \cdots & \kappa(k_c, k_{g_1}, k_{g_{2^n-1}}) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(k_c, k_{g_{2^n-1}}, k_{g_1}) & \kappa(k_c, k_{g_{2^n-1}}, k_{g_2}) & \cdots & \kappa(k_c, k_{g_{2^n-1}}, k_{g_{2^n-1}}) \end{pmatrix}$$

$$\boldsymbol{K}^{**} = \begin{pmatrix} \kappa^{**}(k_c, k_{g_1}, k_{g_1}) & \kappa^{**}(k_c, k_{g_1}, k_{g_2}) & \cdots & \kappa^{**}(k_c, k_{g_1}, k_{g_{2^n-1}}) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa^{**}(k_c, k_{g_{2^n-1}}, k_{g_1}) & \kappa^{**}(k_c, k_{g_{2^n-1}}, k_{g_2}) & \cdots & \kappa^{**}(k_c, k_{g_{2^n-1}}, k_{g_{2^n-1}}) \end{pmatrix}$$

with

$$\begin{aligned} \kappa(k_c, k_g) &= E((l(X, k_c) - l(X, k_g))^2) \\ \kappa(k_c, k_{g_i}, k_{g_j}) &= E((l(X, k_c) - l(X, k_{g_i})(l(X, k_c) - l(X, k_{g_j})) \\ \kappa^{**}(k_c, k_{g_i}, k_{g_j}) &= 4E((l(X, k_c) - E(l(X, k_c)))^2 \\ &\quad (l(X, k_c) - l(X, k_{g_i}))(l(X, k_c) - l(X, k_{g_j}))). \end{aligned}$$

Remark 1. In case of no-weak keys $\boldsymbol{\kappa}, \boldsymbol{K}, \boldsymbol{K}^{**}$ are not key dependent and thus can be determined without knowing the correct key by setting w.l.o.g $k_c = 0$.

Now, considering a leakage model as in Eqn. (2), the theoretical success rate is given by

$$\mathsf{SR} = \Phi_{[\boldsymbol{K} + (\frac{\epsilon}{2\sigma})^2 (\boldsymbol{K}^{**} - \boldsymbol{\kappa}\boldsymbol{\kappa}^T)]}(\sqrt{Q}\frac{\epsilon}{2\sigma}\boldsymbol{\kappa}) \tag{9}$$

where $\Phi_{[C]}(\boldsymbol{\mu})$ is the cumulative distributive function of the multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance C. Now as $\mathsf{SNR} = \frac{\epsilon^2}{\sigma^2}$ a direct relation between SNR and SR is given by

$$\mathsf{SR} = \Phi_{[\boldsymbol{K} + (\frac{1}{4})\mathsf{SNR}(\boldsymbol{K}^{**} - \boldsymbol{\kappa}\boldsymbol{\kappa}^T)]}(\sqrt{Q}\frac{1}{2}\sqrt{\mathsf{SNR}}\boldsymbol{\kappa}). \tag{10}$$

Remark 2. The formula of the theoretical success rate in [9] should yield equivalent results. The main difference between [9] and [10] is the normalization of the confusion coefficient(s). Both works are extension of the mono-bit case for DPA introduced in [8]. A further extension to masked implementations has been given in [19], however, since this work targets only first order leakage, masking and higher order attacks remain out of scope.

Note that, Eqn. (9) and Eqn. (10) hold for Eqn. (2) and thus assume that $l(X, k)$ is known. However, which has not been mentioned in previous works, is that in a practical scenario one may use an approximation of $l(X, k)$ (e.g., $HW(Sbox(X \oplus k))$. This approximation may influence the goodness of the estimation of the theoretical SR in two different ways. First, it may influence the values of $\boldsymbol{\kappa}, \boldsymbol{K}, \boldsymbol{K}^{**}$ as the approximation may not have the same (less or more) "distinguishing ability" as $l(X, k)$. Second, the error made in the approximation of $l(X, k)$ introduces additional noise (epistemic noise from the leakage model) which is not captured when estimating the SNR on the traces. From the previous experiments, we observed that the second aspect is more crucial than the first one.

To take a global look at the previous work, *NICV* is shown directly related with the SNR, which in turn is a main input for computing the minimum number of side channel traces required for performing successful CPA. However, no such formulation exist in case of *specific* or *non-specific TVLA*. In the subsequent section, we will establish the relationship between *specific TVLA* and *SNR* so that we can extend the testing mechanism of conformance-style testing.

3 Link Between *NICV*, *SNR* and *TVLA*

3.1 Motivation

Conformance based testing using *TVLA* is gaining popularity due to its simplicity and ease of computation, but it fails to quantify side channel vulnerability. On the other hand, the evaluation based testing mechanism is highly expensive and lab expertise dependent, but is capable of performing such quantification. In this work, we develop a *hybrid* methodology which provides the simplicity of conformance style testing mechanism and is able to quantify side channel vulnerability as well. We use *specific TVLA* to extract more information regarding the side channel vulnerability of the underlying crypto-implementation.

As shown in section 2.5, a series of works have already established closed relation between *SNR* and *SR* [8], [9], [10]. Further, this relationship for higher-order attacks targeting protected implementations was established in [19]. In this paper, we first establish closed form relation between *specific TVLA*, *SNR* and *NICV*, enabling the computation chain $Specific\ TVLA \rightarrow SNR \rightarrow SR$. Deriving such relation helps establishing a link between the validation and evaluation style testing, which is the main contribution of this work. We further develop a hybrid testing mechanism combining features of validation and evaluation style testing.

3.2 Linking *TVLA* and *NICV*

We follow the same methodology as *TVLA* i.e. dividing data into two groups followed by application of *NICV* (and *SNR*) to it. Let us assume that an adversary has collected n number of side channel traces. The entire set of side channel traces is designated as Y and individual side channel trace is denoted as Y_i, where $i \in [1, n]$ is the index of the corresponding side channel trace. Next following the *TVLA* approach, the traces are partitioned into two groups: Y^{G1} and Y^{G2}, having cardinality n_1 and n_2 ($n = n_1 + n_2$) respectively. Mean and variance of group Y^{G1} and group Y^{G2} are denoted by μ_1, σ_1^2 and μ_2, σ_2^2 respectively. Moreover, mean and variance of the entire set Y are denoted as μ and σ^2. The objective is to derive the relationship between *TVLA* and *NICV* metric. Since we are dealing with only two groups, the corresponding two groups *NICV* is denoted as NICV_2. This NICV_2 will be generalized in the following subsection.

Theorem 1. Consider two groups of side channel traces Y^{G1} and Y^{G2} with cardinality n_1 and n_2. The computation of *TVLA* and NICV_2 on these two groups are related by the following formula

$$\mathsf{NICV}_2 = \frac{1}{\frac{n}{\mathsf{TVLA}^2} + \frac{n}{C}(\sigma_1^2 - \sigma_2^2)\left(\frac{1}{n_2} - \frac{1}{n_1}\right) + 1} \tag{11}$$

where $C = \left(\mu_1^2 - \mu_2^2\right)^2$.

Proof 5. From Eqn. (5) we can write NICV_2 as below:

$$\mathsf{NICV}_2 = \frac{\frac{1}{n}\sum_{i=1}^{2} n_i(\mu_i - \mu)^2}{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \mu)^2} \tag{12}$$

From Eqn. (8) we can write *TVLA* as follows:

$$\begin{aligned}\mathsf{TVLA} &= \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \\ \mathsf{TVLA}^2 &= \frac{(\mu_1 - \mu_2)^2}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \\ &= \frac{C}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}},\end{aligned} \tag{13}$$

where $C = (\mu_1 - \mu_2)^2$. Now we will consider only the numerator part of the NICV_2 formulation which is

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{2} n_i\,(\mu_i - \mu)^2 \\ &= \frac{1}{n}\left(n_1\left(\mu_1 - \frac{n_1\mu_1 + n_2\mu_2}{n}\right)^2 + n_2\left(\mu_2 - \frac{n_1\mu_1 + n_2\mu_2}{n}\right)^2\right) \\ &= \frac{1}{n}\left(\frac{n_1 n_2^2}{n^2}(\mu_1 - \mu_2)^2 + \frac{n_1^2 n_2}{n^2}(\mu_1 - \mu_2)^2\right) \\ &= \frac{n_1 n_2 (n_1 + n_2)}{n^3} C \\ &= \frac{n_1 n_2}{n^2} C.\end{aligned} \tag{14}$$

Next we will consider the denominator part of the NICV computation which is as follows:

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}(Y_i - \mu)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i^2 - \frac{2Y_i\,(n_1\mu_1 + n_2\mu_2)}{n} + \frac{(n_1\mu_1 + n_2\mu_2)^2}{n^2}\right) \\ &= \frac{1}{n}\sum_{Y_i \in Y^{G1}}\left(Y_i^2 - \frac{2Y_i\,(n_1\mu_1 + n_2\mu_2)}{n}\right) \\ &+ \frac{1}{n}\sum_{Y_i \in Y^{G2}}\left(Y_i^2 - \frac{2Y_i\,(n_1\mu_1 + n_2\mu_2)}{n}\right) + \frac{(n_1\mu_1 + n_2\mu_2)^2}{n^2} \\ &= \frac{1}{n}\sum_{Y_i \in Y^{G1}}\left(Y_i^2 - 2Y_i\mu_1 + \mu_1^2 + \left(\frac{2Y_i n_2\,(\mu_1 - \mu_2)}{n} - \mu_1^2\right)\right) \\ &+ \frac{1}{n}\sum_{Y_i \in Y^{G2}}\left(Y_i^2 - 2Y_i\mu_2 + \mu_2^2 + \left(\frac{2Y_i n_1\,(\mu_2 - \mu_1)}{n} - \mu_2^2\right)\right) \\ &+ \frac{(n_1\mu_1 + n_2\mu_2)^2}{n^2} \\ &= \frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 + \frac{n_1 n_2}{n^2} C.\end{aligned} \tag{15}$$

We can now combine Eqn. (12), (13), (14) and (15) to achieve the desired formulation

$$\begin{aligned}\mathsf{NICV}_2 &= \frac{\frac{n_1 n_2}{n^2} C}{\frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 + \frac{n_1 n_2}{n^2} C} \\ &= \frac{C}{n\left(\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2} + \sigma_1^2\left(\frac{1}{n_2} - \frac{1}{n_1}\right) + \sigma_2^2\left(\frac{1}{n_1} - \frac{1}{n_2}\right)\right) + C} \\ &= \frac{1}{n\frac{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}{C} + \frac{n}{C}(\sigma_1^2 - \sigma_2^2)\left(\frac{1}{n_2} - \frac{1}{n_1}\right) + 1}.\end{aligned}$$

Thus we can write NICV_2 as

$$\mathsf{NICV}_2 = \frac{1}{\frac{n}{\mathsf{TVLA}^2} + \frac{n}{C}(\sigma_1^2 - \sigma_2^2)\left(\frac{1}{n_2} - \frac{1}{n_1}\right) + 1}.$$

Corollary 1. If both the group have the same number of side channel traces ($n_1 = n_2 = \frac{n}{2}$), Eqn. (11) transforms into

$$\mathsf{NICV}_2 = \frac{1}{\frac{n}{\mathsf{TVLA}^2} + 1}. \tag{16}$$

Remark 3. It must be noticed that *TVLA* needs to be evaluated for a finite number of traces (n), otherwise it diverges to $+\infty$. However, TVLA^2/n tends to a finite value when n tends to $+\infty$, which bounds the value of $\mathsf{NICV} \in [0, 1]$.

3.3 Generalizing the *NICV* Computation

The relationship between *TVLA* and NICV_2 (2-class *NICV*) was derived previously. However, the general application of *NICV* (or *SNR*) is not restricted to two classes. In this section, the relation between *TVLA* is extended from NICV_2 to a generic k-class *NICV* (NICV_k).

Let us now assume that n number of side channel traces can be partitioned into k number of groups where i^{th} group contains n_i number of traces. A generic example in case of ciphers like AES, where byte-wise computation is performed and the desired value of k is 256. NICV_k can be directly computed from NICV_2 by following an iterative approach. For the derived k groups, k different NICV_2 is performed and the results are combined as follows:

- $\forall i \in \mathbb{Z}_k$, create two groups: the first group contains the side channel traces with particular byte of the plain-text equal to i, the other group will contain the side channel traces with that particular byte value not equal to i. The means of these two groups are denoted as μ_i and $\overline{\mu_i}$ respectively. Similarly, we denote the cardinality of these two groups as n_i and $\overline{n_i} = n - n_i$.
- Compute NICV_2 for each of these two groups. We denote this as NICV_2^i.

Theorem 2. The computations of NICV_k and $NICV_2$ are related with the following formula

$$\mathsf{NICV}_k = \sum_{i=1}^{k} \mathsf{NICV}_2^i - \frac{\sum_{i=1}^{k}\frac{n_i^2}{n\overline{n_i}}(\mu_i - \mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j - \mu)^2}. \tag{17}$$

Proof 6. From Eqn. (12), we can compute NICV_2^i as below

$$\mathsf{NICV}_2^i = \frac{\frac{1}{n}\left(n_i\,(\mu_i - \mu)^2 + (n - n_i)\,(\overline{\mu_i} - \mu)^2\right)}{\frac{1}{n}\sum_{j=1}^{n}(Y_j - \mu)^2}$$

$$= \frac{\frac{1}{n}\left(n_i\left(\mu_i-\mu\right)^2 + \frac{1}{n-n_i}\left(\frac{n_i\sum_{j=1}^{j=k} n_j\mu_j - n n_i\mu_i}{n}\right)^2\right)}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}$$

$$= \frac{\frac{n_i}{\overline{n_i}}\left(\mu_i-\mu\right)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}, where \quad \overline{n_i} = n - n_i. \tag{18}$$

Now if we add each NICV_2^i, we will get the following relationship

$$\sum_{i=1}^{k}\mathsf{NICV}_2^i = \frac{\sum_{i=1}^{k}\frac{n_i}{\overline{n_i}}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2} = \frac{\sum_{i=1}^{k}\frac{n}{\overline{n_i}}\frac{n_i}{n}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}$$

$$= \frac{\sum_{i=1}^{k}(1+\frac{n_i}{\overline{n_i}})\frac{n_i}{n}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}$$

$$= \frac{\sum_{i=1}^{k}\frac{n_i}{n}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2} + \frac{\sum_{i=1}^{k}\frac{n_i^2}{n\overline{n_i}}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}. \tag{19}$$

From Eqn. (12), we can write NICV_k as follows

$$\mathsf{NICV}_k = \frac{\sum_{i=1}^{k}\frac{n_i}{n}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}. \tag{20}$$

Combining Eqn. (19) and (20), we arrive at the following relation

$$\sum_{i=1}^{k}\mathsf{NICV}_2^i = \mathsf{NICV}_k + \frac{\sum_{i=1}^{k}\frac{n_i^2}{n\overline{n_i}}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}. \tag{21}$$

Using the assumption of uniform setting, we presume that each group has same number of side channel traces. Then, Eqn. (19) becomes

$$\sum_{i=1}^{k}\mathsf{NICV}_2^i = \frac{\frac{1}{k}\sum_{i=1}^{k}(\mu_i-\mu)^2 + \frac{1}{k(k-1)}\sum_{i=1}^{k}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2}$$

$$= \frac{\frac{k}{k-1}\frac{1}{k}\sum_{i=1}^{k}(\mu_i-\mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j-\mu)^2} = \frac{k}{k-1}\mathsf{NICV}_k. \tag{22}$$

Thus we arrive at the desired formulation

$$NICV_k = \frac{k-1}{k}\sum_{i=1}^{k}NICV_2^i.$$

It must be noted that NICV_k is actually the generalized *NICV* which was introduced in [11].

Corollary 2. If all the k groups have same number of side channel traces, then

$$\mathsf{NICV}_k = \frac{k-1}{k}\sum_{i=1}^{k}\mathsf{NICV}_2^i. \tag{23}$$

Once we have computed NICV_k, we can easily compute *SNR* using Eqn. (7).

3.4 Extension to Non-Specific TVLA

In this part, we establish the relationship between *SNR* and *non-specific TVLA*. The first hint of the link between *SNR* and *TVLA* was qualitatively discussed in [11]. The formal relationship is derived as follows.

Proposition 2 (Link between* SNR *and* TVLA*). The *SNR* is the variance of the *TVLA* values in the fixed versus random (or non-specific) setup, the variance is computed over all possible fixed values:

$$\mathsf{SNR} = \frac{2\mathsf{Var}(\mathsf{TVLA}_X)}{1-\mathsf{Var}(\mathsf{TVLA}_X)}.$$

Proof 7. As $\mathsf{TVLA}_X = \frac{\epsilon l(x,k)}{\sqrt{\epsilon^2+2\sigma^2}}$, we have: $\mathsf{Var}(\mathsf{TVLA}_X) = \frac{\epsilon^2}{\epsilon^2+2\sigma^2}\mathsf{Var}(L) = \frac{\epsilon^2/\sigma^2}{2+\epsilon^2/\sigma^2} = \frac{\mathsf{SNR}}{2+\mathsf{SNR}}$. From here we can easily derive $\mathsf{SNR} = \frac{2\mathsf{Var}(\mathsf{TVLA}_X)}{1-\mathsf{Var}(\mathsf{TVLA}_X)}$

For *non-specific TVLA*, the traces are partitioned depending upon the entire plaintext value, where one group contains traces with fixed plaintext and other contains traces with random plaintext. If we want to extend our approach to *non-specific TVLA* to compute *SNR*, we need to compute *TVLA* for each plaintext value, which is computationally infeasible. Thus, in the following, we stick to specific *TVLA* only.

4 Proposed Hybrid Side Channel Testing Methodology

4.1 Context

Countermeasures against side-channel attack are advancing every year [20]. Alongside, there are comprehensive evaluation methodologies which are also developed [21]. However, conducting a comprehensive and detailed security evaluation can be a time-taking task. Time is a limiting factor for the evaluation process and for the same reason *CC* evaluations contain time spent for the evaluation as a metric. Some work deal with further simplifying the evaluation process [22].

Most, if not all, real implementations are currently considering basic countermeasures due to the cost of security attached. Thus, evaluation laboratories are still often dealing with unprotected or low-order protected cryptographic implementations, which might also suffer from accidental first order leakage. Automotive ECUs are a current example. In such scenarios, a simple testing methodology like *TVLA* can be a good start. However, it might also be desirable/required to quantify the side channel vulnerability. The methodology proposed in the following combines the efficiency of conformance-style testing mechanism with the purpose of evaluation style mechanism. We later extend the proposed methodology to a multivariate setting as well.

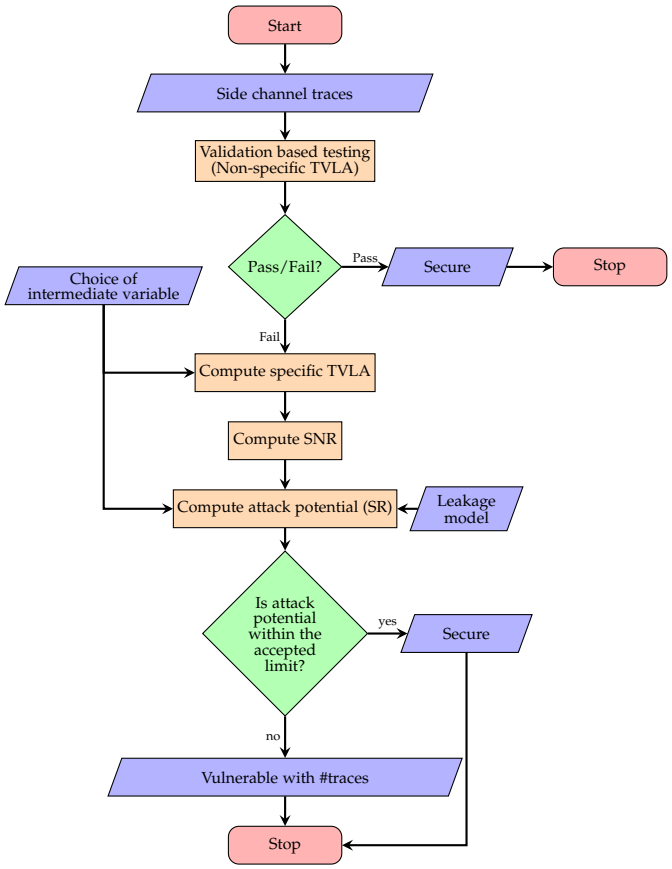


Fig. 2: Proposed Hybrid Side Channel Testing Methodology

4.2 Description of the Proposed Methodology

Side channel analysis is based on a divide and conquer approach. For instance, for an *SPN* cipher where each $b \times b$ S-box handles b bits of the entire key bits, the attack focuses on each of these b bit groups separately. In case of AES-128, $b = 8$ which means that the attack is applied on 8-bits or one byte of the secret key, also known as a sub-key. The attack can be repeated 16 times to recover all the key bytes or alternatively key enumeration methods can be applied to derive the full key [23]. The same applies to *SNR* and *NICV*. One can compute *SNR* or *NICV* byte-wise to zero down the leakage zone of each key byte and apply the attack.

In Fig. 2, we present our methodology to extend the *TVLA* computation to recover the *SNR* followed by the computation of the success rate with a given leakage model and intermediate variable. The test starts with a *non-specific TVLA* test to detect the presence of side channel leakage. If this test fails, we first perform *specific TVLA* for a chosen intermediate variable. Indeed, it is the intermediate value and leakage model that helps in binding the evaluation and conformance based testing in the proposed methodology. From *specific TVLA*, NICV_2 is computed by Eqn. (11), which further leads to NICV_k by Eqn. (17). NICV_k (or just *NICV*) can directly provide the *SNR* by Eqn. (7). Finally, *SNR* leads to *SR* for a chosen leakage model (Eqn. (10)). The computation of *SR* through the proposed methodology is presented in the Algorithm 1. As stated in section 3.4, the methodology cannot be applied to non-specific *TVLA* due to computational infeasibility.

The proposed hybrid methodology brings in several

Algorithm 1: Computing *SNR* and *SR* from *TVLA*

Input: Side channel traces and corresponding intermediate state
Output: *SR* for chosen sub-key

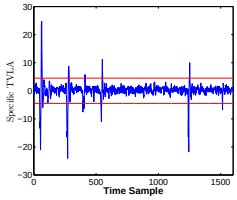
1 **for** $i = 0$ *to* k **do**
2 Partition the side channel traces into two groups: G_1 and G_2
3 G_1: Side channel traces where j^{th} byte of the intermediate data $= i$
4 G_2: Side channel traces where j^{th} byte of the intermediate data $\neq i$
5 Apply *TVLA* on groups G_1 and G_2
6 Compute NICV_2^i from the *TVLA* value by using Eqn. (11)
7 Compute NICV_k using Eqn. (17)
8 Compute $\mathsf{SNR} = \frac{1}{\frac{1}{\mathsf{NICV}_k} - 1}$
9 Compute $\mathsf{SR} = \Phi_{[\boldsymbol{K} + (\frac{1}{4})\mathsf{SNR}(\boldsymbol{K}^{**} - \boldsymbol{\kappa}\boldsymbol{\kappa}^T)]}(\sqrt{Q}\frac{1}{2}\sqrt{\mathsf{SNR}}\boldsymbol{\kappa})$
10 Return SR

TABLE 1: Comparison between existing and proposed testing methodologies

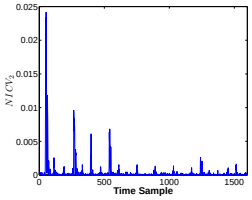
Features	Evaluation	Conformance	Proposed
Leakage model required	√	×	√
Intermediate value required	√	×	√
Vulnerability quantification	√	×	√
Analytical	×	√	√

advantages as compared to the two individual approaches (evaluation and conformance). It formally shows that the two approaches are not unrelated and propose a basis to compute one from the other. Moreover, the proposed methodology provides a computation acceleration. In comparison to Fig. 1 (a), Fig. 2 does not have any iterative loop for success rate computation. The acceleration is significant in commercial products, where even unprotected implementations might need millions of traces for an attack, repeated several times for success rate computation. The proposed methodology can compute *SR* for several leakage models in parallel, without significant additional computation, as the knowledge of leakage model is only needed in step 9 of Algo. 1. Since the leakage model projects the intermediate value to side-channel leakage, several projections can be tested in parallel, based on the attacker profile. The methodology can support a range of leakage models from a generic Hamming weight and identity, which can be erroneous, to profiled leakage model of linear and higher dimensions [24], which will be more precise. As shown later, the proposed methodology can also be applied in a multivariate setting. Nevertheless, if *TVLA* results are not required, the evaluator can directly compute *SNR* from the traces and follow the remaining methodology.

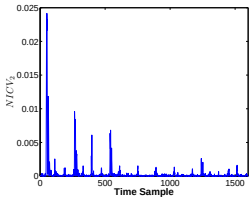
As for the disadvantages, the choice of intermediate value is required for *specific TVLA* computation. However, conformance-style testing does not require any prior knowledge of leakage model or intermediate variable. This choice of intermediate value requires expertise on part of the evaluator. From another perspective, it is the knowledge or choice of intermediate value which binds the two approaches together. The proposed methodology takes the intermediate value as an external input from the evaluator for *specific TVLA* computation and allows the user to be flexible in his choice of leakage model or test several in parallel. All these points are summarised in Tab 1.



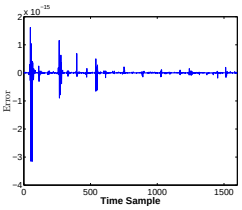
(a) Single specific *TVLA*



(b) Predicted NICV_2 from plot (a)



(c) Computed NICV_2 from traces



(d) Prediction Error in Eqn. (11) ((b) - (c))

Fig. 3: Equivalence of *TVLA* and NICV_2

5 Experimental Verification of Derived *TVLA* and *NICV* Relation

The derived relation between *specific TVLA* and *SNR* (or *NICV*) is experimentally validated in this section on an AES-128 implementation (without side-channel countermeasures) running on an ATMEGA-8515 smart-card.

5.1 Experimental Setup

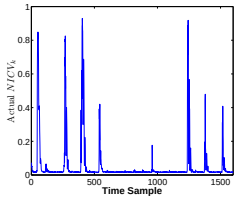
The *AES* design is implemented on a SAKURA-GW platform [25]. The SAKURA-GW platform supports communication with ATMEGA-8515 smart-card, which in our case runs an unprotected AES-128. The power measurements are taken using a Tektronix MSO4034B mixed signal oscilloscope with sampling frequency $500\ MSamples/sec$. Being an unprotected implementation, it is obvious that the AES implementation must have exploitable leakage and its *TVLA* value should be more than the threshold of 4.5.

5.2 Validation of *TVLA* and NICV_2 Relationship

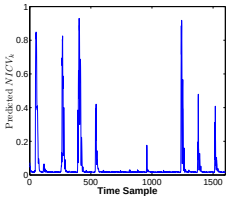
To verify the relationship between *TVLA* and NICV_2 (see Eqn. (11)) practically, we start with partitioning the traces based on the first-byte value ($k = 256$) of the output of round 9 as the intermediate state, following step 1 of Algo. 1. Next, we compute *TVLA* and NICV_2 from the partitions again following Algo. 1. The results are shown in Fig. 3. A specific *TVLA* trace is shown in Fig. 3 (a). Next, the *TVLA* trace in Fig. 3 (a) is used to compute NICV_2 using Eqn. (11) and shown in Fig. 3 (b). We also compute NICV_2 from power measurement as shown in Fig. 3 (c). The error between predicted and computed NICV_2 is in the order of 10^{-15} i.e. negligible and coming from truncation error (Fig. 3 (d)), which confirms Eqn. (11).

5.3 Validation of NICV_k and NICV_2 relationship

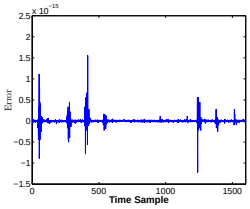
Similar validation is also done for Eqn. (17) that relates NICV_2 and NICV_k. Using the same set of traces and no. of partitions ($k = 256$), we compute NICV_k from the traces and predict it from previously computed NICV_2. The results are shown in Fig. 4. As the computed NICV_k(Fig. 4 (a)) follows closely the predicted NICV_k (Fig. 4 (b)), the prediction error (Fig. 4 (c)) also stays in the range of 10^{-15}.



(a) Computed NICV_k



(b) Predicted NICV_k



(c) Prediction Error in Eqn. (17) ((a) - (b))

Fig. 4: Prediction of NICV_k

6 Case Study: Application to AES

The equivalence of *TVLA* and *SNR* was theoretically derived and experimentally verified in the previous sections. The step by step procedure to compute *SNR* (and *SR*) from the *specific TVLA* value was presented in Algo. 1. In this section, we focus on the application of these relations towards testing AES in three different settings. First results are shown on simulated power traces, followed by application of the evaluation methodology on actual power traces acquired from unprotected AES implementation running on the same ATMEGA-8515 smart-card which was used in section 5. Finally, the methodology is tested on publicly available DPA Contest v4.0 traces corresponding to a protected AES-256 implementation with some first order leakage.

6.1 Under Simulated Setting

Simulated traces are generated for an $8-$bit microcontroller, assuming perfect Hamming weight leakage and added zero mean Gaussian noise ($\mathcal{N}(0, \sigma^2)$), where σ^2 denotes the variance of the noise distribution. The side channel trace can be represented as $Y = HW(v) + \mathcal{N}$, where v is the chosen intermediate value, which in this case is first 8-bits of round 9 output. We have generated side channel traces for different *SNR* values ranging from 0.03 to 2.

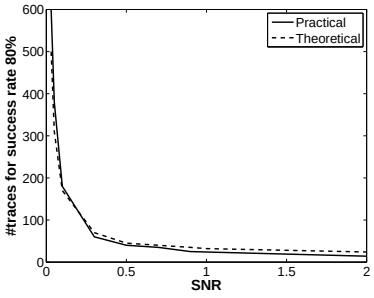
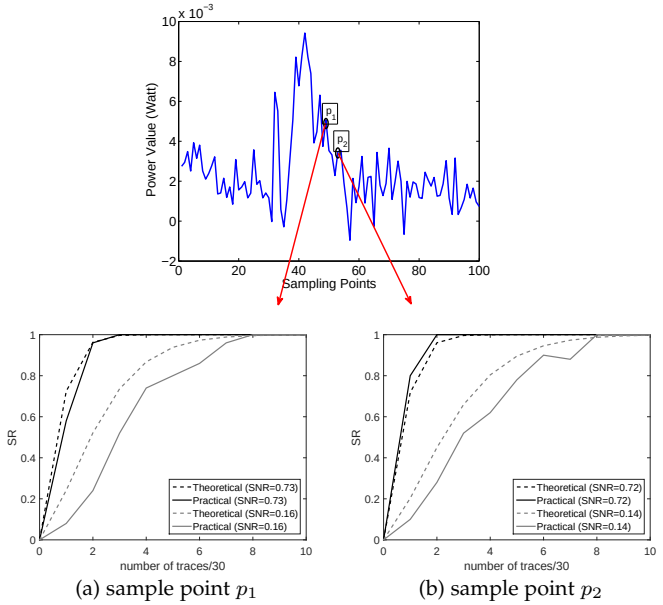


Fig. 5: Estimation of Number of Traces to Reach 80% *SR* For Theoretical *SR* and Practical *SR* for different *SNRs*

Next, we directly apply Algo. 1 to first derive *SNR* and then Eqn. (10) to estimate the number of traces required to achieve 80% *SR*. A practical CPA attack is also performed

repeatedly on the set of the simulated traces to compute the number of traces required to achieve 80% *SR*. The corresponding result is shown in Fig. 5, which shows a very close match between the theoretical and practical evaluation.

It can be observed that under perfect HW model assumption, the estimated theoretical estimation and practical computation fits quite closely. A minor overshoot for practical *SR* is seen for high *SNR* (> 0.5). This overshoot is an approximation glitch in the theoretical formulation under central limit theorem and law of large numbers, which needs few dozen traces to converge. Otherwise, the approximation overshoot remains constant even for extremely high *SNR* (tested up to *SNR=20*). The overshoot can be seen in real traces as well for high *SNR* scenarios in the next subsection.



(a) sample point p_1 (b) sample point p_2

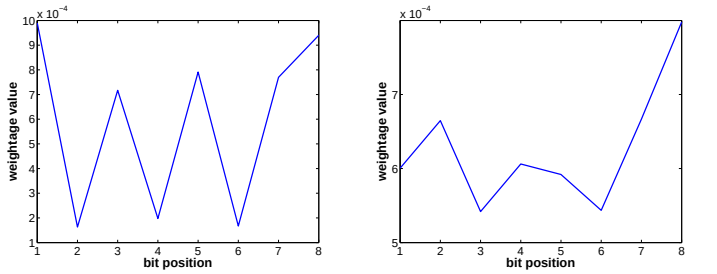
Fig. 6: Comparison Between Theoretical *SR* and Practical *SR* for different *SNRs* using Hamming weight model at different sample points

6.2 On Real Power Traces

The experimental setup for the acquisition of power traces is equivalent to the one described in section 5.1. Further white Gaussian noise is added to experiment in low-*SNR* scenarios. The experiments were performed with 20,000 traces. For practical *SR*, a CPA was mounted on a randomly chosen set of 300 traces (from those 20000), repeated 50 times. Following Algo. 1 and assuming that the ATMEGA-8515 smart-card on the SAKURA-GW board leaks in HW model, we generate plots for estimated theoretical success rate.

The results are shown in Fig. 6 for two distinct points p_1 and p_2 on the trace. We compute the practical *SR* and theoretical *SR* in the interval of 30 traces. The x-axis in the Fig. 6 denotes the number of such intervals (which is equal to number of traces/30) and the y-axis denotes the corresponding *SR* value. As we can see, in the low *SNR* scenario, there is a gap between theoretical *SR* and practical *SR* which is due to the improper leakage model.

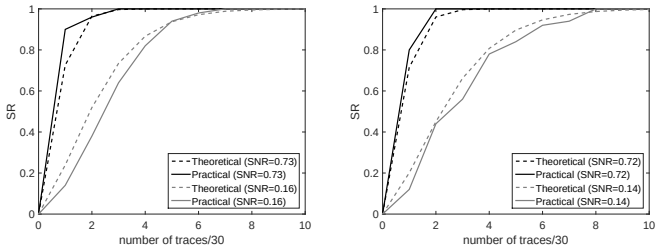
Finding a device with perfect HW leakage model is a very strong assumption. The two distinct points: p_1 and p_2 are chosen as such that one point has leakage very close to



(a) sample point p_1 with imperfect HW leakage model (b) sample point p_2 which is more close to HW leakage model

Fig. 7: Sample points with perfect and imperfect HW leakage model

HW model while the other deviates from the model. More specifically, the sample point p_2 has a leakage model closer to HW model, whereas the sample point p_1 has a leakage model which deviates significantly from the HW model. A closer estimation to the actual model is computed using profiling based on stochastic modeling [24] of leakage into 9 dimensions as $\Sigma_{i=1}^{8}\beta_i v_i$. The β weights of different points are shown in Fig. 7. Fig. 7(a) shows that in case of point p_1, the leakage model deviates from HW model, whereas Fig. 7(b) shows that leakage model of point p_2 stays close to HW model. Referring back to Fig. 6, when the *SNR* is high, the practical *SR* for both sampling point p_1 and p_2 closely matches the theoretical prediction. However, as the *SNR* reduces, the deviation between theoretical and practical *SR* increases. This deviation is even worse when the model is imperfect (see Fig. 6(a)).



(a) sample point p_1 (b) sample point p_2

Fig. 8: Comparison Between Theoretical *SR* and Practical *SR* for different *SNRs* using first order stochastic model at different sample points

We repeat the experiments by taking the actual model into the account and re-running Algo. 1[1]. Precisely it is only the last step of Algo. 1 which is affected by the leakage model as stated in Eqn. (10). The results are shown in Fig. 8. Again under high *SNR*, the practical attack results match with the theoretical estimation. However, by taking the correct leakage model into the account, the theoretical estimation of *SR* and practical *SR* also matches closely for sample point p_1 (with imperfect HW leakage model) and sample point p_2 (with leakage model close to HW leakage model). This match is due to the application of correct leakage model in Algo. 1 which confirms the importance of leakage modeling in a side channel attack. From the methodology aspect, it shows that the better profiled the model is, the more realistic prediction

1. The computation of *SR* using HW model and stochastic modeling can be executed in parallel.

of *SR* can be made from the *TVLA* results. Nevertheless, the evaluator can test several leakage models in parallel at negligible computation overhead.

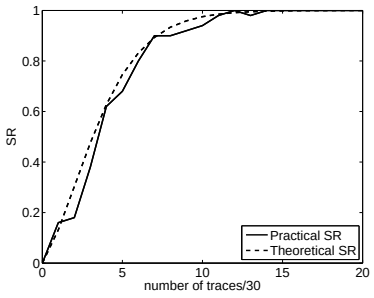


Fig. 9: Comparison Between Theoretical *SR* and Practical *SR* on DPA Contest v4.0 Traces

6.3 First Order Protected Implementation with Leakage: DPA Contest v4.0 Traces

Till now, we have discussed the application of the proposed methodology on the unprotected implementations. The proposed hybrid testing methodology can be also applied to the flawed first order protected implementation which exhibits first order leakages due to inefficient implementation or glitches inside the circuits. The side channel traces used in *DPA Contest* $v4.0$ [12] is an example of such scenarios. The AES-256 implementation used in *DPA Contest* $v4.0$ is based on rotating S-Box making scheme. However, it was shown in [13] that the implementation exhibits univariate first order leakage. Precisely, the attack in [13] exploits the accidental leakage on a single bit when the round 0 key addition result is overwritten by the round 1 Sbox output. The leakage is exploited over a single bit and denoted as $(((x \oplus k) \oplus S(x \oplus k))\&1)$ ($\&$ is logical AND function). In the following, we first compute the practical *SR* for the attack published in [13], on the available traces with this model to use it as a benchmark. Next, the proposed methodology is applied to predict the *SR* from *specific TVLA* testing using the same leakage model.

The practical and theoretical *SR* are shown in Fig. 9. As shown in Fig. 9, the values of practical and theoretical *SR* matches very closely which in turn proves the efficiency of the proposed methodology. $(((x \oplus k) \oplus S(x \oplus k))\&1)$ is used as a intermediate value and the applied leakage model is identity, i.e. *specific TVLA* and thus *SNR* are computed for a single bit of model.

7 Multivariate Analysis

Traditionally, multivariate side channel analysis is applied for higher order attacks where leakages from multiple points are combined. Multivariate analysis can be useful even in a first order leakage context because an adversary can retrieve the key much earlier if he combines multiple leakage points in an optimal manner. A relevant scenario where such analysis can be useful is a real industrial product with clock jitter that leads to side-channel measurement misalignment. The leakage is thus spread over multiple time samples due to the jitter. While a univariate analysis in such scenario might be sub-optimal, a multivariate approach can lead to fair evaluation.

In its current form, *TVLA* metric can not be applied in multivariate analysis without modifying its formulation. Recently in [18], the limitations of *TVLA* in detection of multivariate side channel vulnerabilities were addressed in details for higher order analysis. In [5], the authors have focussed on extending *TVLA* methodology to higher order leakage detection. Consequently, a strategy for applying *d-th order d-variate TVLA test* is given. A typical application for such analysis can be a software implementation of d^{th} order masking, where shares are executed sequentially.

Our approach in this section is different from them as we focus on *1st order d-variate TVLA test* where d denotes the dimension of a single side channel trace. We investigate the extension of proposed methodology for unprotected implementation in the multivariate setting for side-channel vulnerability quantification. Therefore, the weaknesses pointed out in [18], do not apply to our setting. Moreover, in this section, we try to extend the applicability of *TVLA* from univariate to multivariate settings to address one of the shortcomings of traditional *TVLA* [18].

7.1 Proposed Formulation

To obtain *SR* for multivariate side channel analysis, we can follow two different approaches. We can either compute *TVLA* on each sample and then combine those values to get the corresponding *SR* in multivariate settings or combine the different sample points using an optimal dimensionality reduction formulation to convert the multivariate side channel traces into a single point. For latter, we use the framework of [26]. In particular, the traces Y arise from a single leakage model L, which depends on the correct key $k = k^*$, and which is taken standard (i.e., $\mathsf{E}(L) = 0$, $\mathsf{Var}(L) = 1$), through the relationship:

$$Y_d = \alpha_d L(k^*) + N_d,$$

where d is the dimensionality ($1 \leq d \leq D$).

Remark 4. This equation implies $\mathsf{E}(Y) = 0$. When computing a t-test, using non-specific or specific, the evaluator also has to evaluate $\mathsf{E}(Y|X = x_0)$ for a given plaintext (or a given byte value of the plaintext) x_0. Let's assume that $\mathsf{E}(Y|X = x_0) = c \neq 0$. The condition $\neq 0$ is here to avoid having $\mathsf{E}(Y) = \mathsf{E}(Y|X = x_0)$, in which case the attacker would conclude the device is secure whereas in practice it is not (e.g. for a different value of x_0', we would have $\mathsf{E}(Y) \neq \mathsf{E}(Y|X = x_0')$).

In matrix form, for Q number of side channel traces, we can write the above equation as below:

$$Y^{D,Q} = \alpha^D L^Q(k^*) + N^D,$$

Here α_D is a non-zero vector of length D, and can be calculated as follows [26]:

$$\alpha^D = \frac{Y^D (L^Q(k^*))^T}{L^Q(k^*) L^Q(k^*)^T}. \tag{24}$$

We assume that the noise N^D is multivariate normal, and we denote by $\mathbf{\Sigma}$ its $D \times D$ covariance matrix. The value of $\mathbf{\Sigma}$ can be computed as below [26]:

$$\Sigma = \frac{1}{Q-1}(Y^{D,Q} - \alpha^D L^Q(k^*))(Y^{D,Q} - \alpha^D L^Q(k^*))^T. \tag{25}$$

With the knowledge of α^D and Σ, we can now calculate the optimal dimensionality reduction formulation which is $\frac{(\boldsymbol{\alpha}^D)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}^{D,Q}}{(\boldsymbol{\alpha}^D)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}^D}$ [26].

7.1.1 SNR *and* TVLA *in multivariate settings*

To compute the *SNR* and *TVLA* in multivariate settings, we propose following pre-processing steps. Hereby $\boldsymbol{boldface}$ we denote multivariate trace of dimension D.

- Step 1: Compute $\boldsymbol{\Sigma}$,
- Step 2: Standardize the measurements, that is: $\boldsymbol{Y}$ becomes $\boldsymbol{Y}' = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}$.

Notice that $\boldsymbol{Y}' = (\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\alpha})L + \boldsymbol{N}'$, where $\boldsymbol{N}'$ is now an isotropic standard noise (all D samples of noise are i.i.d., of mean 0 and variance 1). Indeed,

$$\begin{aligned}\mathsf{E}(\boldsymbol{N}'(\boldsymbol{N}')^\mathsf{T}) &= \mathsf{E}(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{N}\boldsymbol{N}^\mathsf{T}\boldsymbol{\Sigma}^{-1/2}) \\ &= \boldsymbol{\Sigma}^{-1/2}\mathsf{E}(\boldsymbol{N}\boldsymbol{N}^\mathsf{T})\boldsymbol{\Sigma}^{-1/2} = \boldsymbol{I}, \end{aligned} \tag{26}$$

where $\boldsymbol{I}$ is the $D \times D$ identity matrix.

On step 2, we can now re-estimate $\boldsymbol{\mu}'_1$, as $\mathsf{E}(\boldsymbol{Y}')$. For the sake of clarity, we drop index 1 and 2 in $\boldsymbol{\mu}$ (when it is clear given the context). We see that the optimal dimensionality reduction is (theorem 1 of [26])

$$\frac{(\boldsymbol{\mu}')^\mathsf{T}\boldsymbol{Y}'}{(\boldsymbol{\mu}')^\mathsf{T}\boldsymbol{\mu}'} = \|\boldsymbol{\mu}'\|^{-2}\ (\boldsymbol{\mu}')^\mathsf{T}\boldsymbol{Y}'. \tag{27}$$

Consequently, we can define multivariate *SNR* and multivariate *TVLA* as follow:

$$\mathsf{SNR} = (\boldsymbol{\mu}')^\mathsf{T}\boldsymbol{\mu}' = \sum_{d=1}^{D}(\mu'_d)^2. \tag{28}$$

$$\mathsf{TVLA}^2 = \sum_{d=1}^{D}\frac{(\mu'_{1,d} - \mu'_{2,d})^2}{\frac{1}{n_1} + \frac{1}{n_2}} \tag{29}$$

because $\sigma'_{1,d} = \sigma'_{2,d} = 1$ (by Eqn. (26)).

Remark 5. This is equal to (up to an irrelevant $\frac{1}{4}$ proportionality factor) the Hotelling's T-Square [27]). Indeed, let us consider that $n_1 = n_2 = n/2$. We have:

$$\begin{aligned}\mathsf{TVLA}^2 &= \sum_{d=1}^{D}\frac{(\mu'_{1,d} - \mu'_{2,d})^2}{\frac{1}{n_1} + \frac{1}{n_2}} \\ &= \frac{1}{4}n(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\mathsf{T}\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \end{aligned} \tag{30}$$

The definition of multivariate *SNR* (Eqn. (28)) and multivariate *TVLA* (Eqn. (29)) remains consistent with the dimensionality reduction (Eqn. (27)). Namely, we have:

Proposition 3. The application of univariate *SNR* (resp *TVLA*) of reduced trace (Eqn. (27)) yields multivariate *SNR* (Eqn. (28)) (resp. multivariate *TVLA* (Eqn. (29))).

Proof 8. After dimensionality reduction, we get:

$$Y'' = L + \frac{1}{\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}\boldsymbol{\mu}'^T\boldsymbol{N}'.$$

For the *SNR*, we thus have:

- **signal**: $\mathsf{Var}(L) = 1$;
- **noise**:

$$\frac{1}{(\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})^2}\mathsf{Var}(\boldsymbol{\mu}'^T\boldsymbol{N}') = \frac{1}{\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}. \tag{31}$$

Hence *SNR* is $\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$, which is equal to Eqn. (28).

Regarding *TVLA*, we will assume that $\mathsf{E}(Y) = \boldsymbol{\mu_1} = 0$, and $\mathsf{E}(Y|X = x_0) = \boldsymbol{\mu_2} = c\boldsymbol{\mu}$. Hence, after dimensionality reduction (Eqn. (27)), one gets

- reduced average for random plaintext: 0,
- reduced average for fixed plaintext = x_0: c,
- reduced noise has variance (Eqn. (31)).

Hence the univariate (squared) *TVLA* on reduced traces is

$$c^2(\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}).$$

Now, the multivariate (squared) *TVLA* (Eqn. (29)) is (using Hotteling formula (Eqn. (30))):

$$\frac{1}{4}n(\mathbf{0} - c\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{0} - c\boldsymbol{\mu}),$$

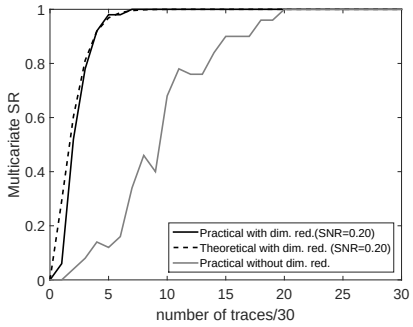
which also match with the *TVLA* expression obtained after dimensionality reduction. It must be noted that this formulation is applicable to both specific and non-specific *TVLA* test.

7.2 Experimental Results

The multivariate setting of the proposed methodology is now experimentally validated on real power traces of an unprotected AES-128 (same as section 6.2). We first apply *optimal dimension reduction* on the acquired traces to project the multivariate leakage to a single point. As shown in Prop. 3 multivariate *SNR* computed on the multivariate traces is equivalent to the univariate *SNR* computed on the dimension reduced traces. Hence, we can use our proposed methodology for univariate traces on the dimension reduced traces and can compute the theoretical *SR* and practical *SR* (see Fig. 10). Firstly, the practical *SR* on dimension reduced traces (multivariate) is much better than traces without dimension reduction (univariate). This shows that if an adversary applies multivariate analysis for first order side channel attack, he can obtain the correct key within very few traces compared to univariate analysis. Even on an unprotected implementation, the leakage is spread over samples and cannot be optimally exploited in a univariate setting. This observation validates the motivation behind developing our *1st order d-variate* side channel vulnerability quantification methodology. Figure 10 also shows that the proposed formulation for computation of the theoretical *SR* follows the practical *SR* which successfully validates our proposed methodology for computation of *SR* in first order multivariate settings. It must be noted that the *SNR* shown in Fig. 10 is computed after applying dimension reduction.

7.3 Application to Jitter-based Countermeasures

As stated before, the proposed hybrid evaluation methodology can be applied to any first order side-channel leakage. The analysis was extended from univariate to multivariate setting in the previous subsection. The extension to multivariate setting brings several countermeasures under the scope of this scheme. We next apply the proposed methodology to a jitter based countermeasure.



(a)

Fig. 10: Comparison Between Theoretical *SR* and Practical *SR* in multivariate settings

Insertion of jitter during computation of cryptographic operations, results in misalignment of traces. The misalignment causes reduction of SNR. Such countermeasure are often deployed in commercial products and also used to strengthen other countermeasures like masking. To perform a successful attack the attacker has to increase the number of traces or apply realignment methods or multivariate attacks or a combination of these methods. Fr our experiments, ee introduce jitter on the acquired traces using the same methodology as [28] and the ASCAD database [29]. A jitter in the power trace was introduced by shifting each power trace by a random number ($\in [0, 75]$) of sample points. An instance of such jittery power trace is shown in Fig.11.

As expected, the application of univariate attack on 300 unprotected AES-128 traces (same as section 6.2) failed. Next, we apply the previously proposed hybrid methodology in multivariate setting.

In Fig. 12, we show the practical and theoretical success rate of the multivariate analysis on the jittery power traces. The theoritical prediction stays close to practical attacks, even in presencce of jitter-based countermeasure, expanding the applicability of the proposed hybrid evaluation methodology.

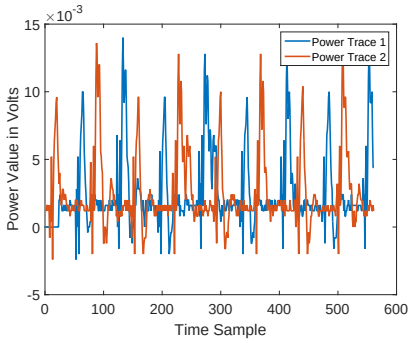


Fig. 11: Sample power traces after introduced jitter

8 Conclusion

Though conformance-style testing methodology is becoming popular due to its simplicity and integrability with standard testing mechanism, it does not give much information about the side-channel resistance of the target. In this paper, we make a first attempt to extend the *TVLA* based conformance-style testing methodology beyond its current scope. The analytic relationship between *specific TVLA* and *SNR* is derived, which allows to directly compute *SR* from

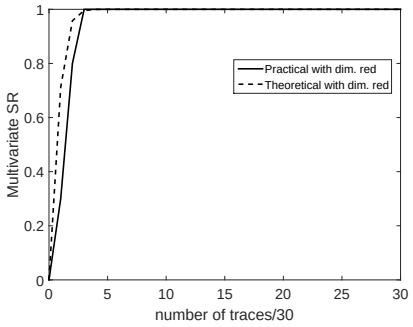


Fig. 12: Comparison between theoretical SR and practical SR on jitter-based countermeasure in multivariate setting

specific TVLA test with the knowledge of leakage model and intermediate variable. We have also shown that *non-specific TVLA* can not be used in this context due to computational infeasibility. By connecting *specific TVLA* with *SR*, an attempt is made to bridge the gap between conformance based testing and evaluation based testing, addressing both side channel leakage detection and side channel leakage quantification. The methodology is successfully verified on an unprotected AES smart-card implementation in a simulated setting as well as practical measurements. The proposed methodology is further extended to address multivariate leakage. As the proposed methodology addresses only first-order side-channel leakage, it can be applied to test several countermeasures. We verified this methodology on two specific countermeasures: a masking countermeasure with accidental first-order leakage (in publicly available *DPA Contest v*4.0 traces) and jitter based countermeasures. The theoretical and practical results are shown to match, especially under a well profiled model. Further extension of this approach to protected implementation, especially using the formulation of [5], [19] would be an interesting direction.

References

[1] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual International Cryptology Conference*, pages 388–397. Springer, 1999.

[2] The Common Criteria. https://www.commoncriteriaportal.org/. Accessed: 2016-09-25.

[3] FIPS 1403 DRAFT Security Requirements for Cryptographic Modules (Revised Draft). http://csrc.nist.gov/publications/drafts/fips1403/reviseddraftfips1403_PDFzip_documentannexAtoannexG.zip.

[4] Jaffe J. Goodwill G., Jun B. and Rohatgi P. A testing methodology for side-channel resistance validation. http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf, 2011.

[5] Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99, 2016.

[6] G. Becker, J. Cooper, E. DeMulder, G. Goodwill, J. Jaffe, G. Kenworthy, T. Kouzminov, A. Leiserson, M.Marson, P. Rohatgi, and S. Saab. Test Vector Leakage Assessment (TVLA) methodology in practice. http://icmc-2013.org/wp/wp-content/uploads/2013/09/Rohatgi_Test-Vector-Leakage-Assessment.pdf, 2013.

[7] François-Xavier Standaert, Tal G Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 443–461. Springer, 2009.

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

[9] Adrian Thillard, Emmanuel Prouff, and Thomas Roche. Success through confidence: Evaluating the effectiveness of a side-channel attack. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013*, pages 21–36, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[10] Yunsi Fei, A. Adam Ding, Jian Lao, and Liwei Zhang. A statistics-based success rate model for DPA and CPA. *J. Cryptographic Engineering*, 5(4):227–243, 2015.

[11] Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. Side-channel Leakage and Trace Compression Using Normalized Inter-class Variance. In *Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '14, pages 7:1–7:9, New York, NY, USA, 2014. ACM.

[12] AES-256 RSM Traces. http://www.dpacontest.org/v4/rsm_traces.php.

[13] Amir Moradi, Sylvain Guilley, and Annelie Heuser. Detecting hidden leakages. In *Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings*, pages 324–342, 2014.

[14] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, http://www.dpabook.org/.

[15] Sébastien Tiran, Guillaume Reymond, Jean-Baptiste Rigaud, Driss Aboulkassimi, Benedikt Gierlichs, Mathieu Carbone, Gilles R. Ducharme, and Philippe Maurine. Analysis of variance and CPA in SCA. *IACR Cryptology ePrint Archive*, 2014:707, 2014.

[16] Luke Mather, Elisabeth Oswald, Joe Bandenburg, and Marcin Wójcik. Does my device leak information? an a priori statistical power analysis of leakage detection tests. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 486–505, 2013.

[17] A. Adam Ding, Cong Chen, and Thomas Eisenbarth. Simpler, faster, and more robust t-test based leakage detection. Cryptology ePrint Archive, Report 2015/1215, 2015. http://eprint.iacr.org/2015/1215.

[18] Franois-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. Cryptology ePrint Archive, Report 2017/138, 2017. http://eprint.iacr.org/2017/138.

[19] Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2014.

[20] Jean-Sébastien Coron, Aurélien Greuet, Emmanuel Prouff, and Rina Zeitoun. Faster evaluation of sboxes via common shares. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 498–514. Springer, 2016.

[21] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 459–476. Springer, 2014.

[22] François Durvaux, François-Xavier Standaert, and Santos Merino Del Pozo. Towards easy leakage certification. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 40–60. Springer, 2016.

[23] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In *International Conference on Selected Areas in Cryptography*, pages 390–406. Springer, 2012.

[24] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 30–46. Springer, 2005.

[25] SAKURA-GW. http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-W.html.

[26] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is More - Dimensionality Reduction from a Theoretical Perspective. In *Cryptographic Hardware and Embedded*

[28] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 45–68. Springer, 2017.

Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, pages 22–41, 2015.

[27] Harold Hotelling. The generalization of student's ratio. *Ann. Math. Statist.*, 2(3):360–378, 08 1931.

[29] Emmanuel Prouff, Remi Strullu, Ryad Benadjila, Eleonora Cagli, and Cecile Dumas. Study of deep learning techniques for side-channel analysis and introduction to ascad database. Cryptology ePrint Archive, Report 2018/053, 2018.

Debapriya Basu Roy is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur, India. His current research interests include design and analysis of side channel secure Elliptic Curve Cryptography, hardware security and FPGA based system design.

Shivam Bhasin is currently a Senior Research Scientist and Principal Investigator at PACE Lab, Nanyang Technical University, Singapore since 2015. His research interests include embedded security, trusted computing and secure designs.

Sylvain Guilley is a Co-founder and CTOSecure-IC, France and professor at TELECOM-ParisTech, France. His research interest is ESS (Embedded Systems Security). This field includes trusted computing, cyber-physical security, secure prototyping in FPGA and ASIC, and formal / mathematical methods. Sylvain has co-authored 100+ research papers and filed 20+ patents.

Annelie Heuser is a researcher of the French National Center for Scientific Research (CNRS) at IRISA, Rennes, France. Her main research interests lie in the area of side-channel analysis, machine learning, hardware security, and malware detection/ classification.

Sikhar Patranabis has been pursuing Ph.D. in Dept. of Computer Science and Engineering, IIT Kharagpur since 2015. His research interests include public key cryptography, lightweight cryptography and hardware security.

Debdeep Mukhopadhyay received his PhD from Dept. of Computer Science and Engineering, IIT Kharagpur in 2007, where he is presently an Associate Professor. His research interests include cryptography, VLSI of cryptographic algorithms, hardware security and side channel analysis.