What Lies Ahead: Extending TVLA Testing Methodology Towards Success Rate

Debapriya Basu Roy[1], Shivam Bhasin[2], Sikhar Patranabis[1], and Debdeep Mukhopadhyay[1]

[1] Secured Embedded Architecture Laboratory (SEAL)
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
dbroy24@gmail.com, sikharpatranabis@gmail.com,
debdeep.mukhopadhyay@gmail.com

[2] Temasek Laboratories, NTU, Singapore.
sbhasin@ntu.edu.sg

Abstract. Evaluation of side channel vulnerability of a cryptosystem has seen significant advancement in recent years. Researchers have proposed several metrics like *Test Vector Leakage Assessment Methodology (TVLA)*, *Normalized Inter Class Variance (NICV)*, *Signal to Noise Ratio (SNR)*, *Guessing Entropy* to determine side channel security of crypto-implementations. Among these, *TVLA* has emerged as the front-runner as it can determine side channel vulnerability of a crypto-system irrespective of the underlying leakage model and hence can be integrated into the testing mechanism very easily. TVLA which is actually similar to *statistical t-test* acts as a powerful tool which provides a *pass-fail* testing mechanism of crypto-implementations. More precisely it can determine whether the system is secure or not, it does not quantify the security of the crypto-implementations in terms of number of side channel traces required or signal-to-noise ratio (SNR) of the crypro-implementations. *Statistical F test*, on the other hand, can easily compute the SNR, which in turn can quantify the side channel vulnerability in terms of number of side channel traces required. In this work, we aim to connect the TVLA metric to the computation of SNR, leading to establishing lower bound for the number of traces for a successful attack. This work will also show the equivalence of the required existing side channel evaluation metrics.

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 is then correlated with 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.*

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 crypto-implementations using a standard set of pre-defined evaluation assurance levels. 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. 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 less favourable for testing crypto-implementations against side channel vulnerability.

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 a design in terms of whether it meets the necessary security levels or not, rather than an exact evaluation of its vulnerability. With respect to side channels, it employs a simplified approach of 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 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.

Test Vector Leakage Assessment (TVLA) [4] which was proposed at NIST sponsored NIAT workshop 2011, is one of such conformance style testing mechanism which has gained huge popularity among the researchers and specially the practitioners due to its robustness, applicability to different crypto-implementations and easy integrability with the exiting testing methodologies. Multiple research papers on side channel attacks have used this tool to show the effectiveness of their proposed attacks and countermeasures. TVLA exploits well known *Welch's t-test* which is actually statistical hypothesis testing mechanism. It can be classified into two different categories: *non-specific* and specific [4]. In case of non-specific TVLA, the validator does not need to have the knowledge of the secret key. He just needs to collect the side channel traces of the crypto-implementations and classifies them into two different classes: one with fixed test-vector (plain-text) as input and another with random test-vectors as input. He then requires to perform *Welch's t-test* (also known as Student's t-test) on these two classes and compute the *t-value*. If the t-value crosses the pre-defined threshold (which for TVLA is ±4.5 [4]), the crypto-implementation is considered to be vulnerable to side channel attacks. In case of specific TVLA, the validator needs to know the secret key of the crypto-implementation as the classification of the side channel traces is done depending upon the intermediate value of the crypto-designs. There are various types of specific TVLA test which we will discuss in details in section 2. It has been shown in [5] that non-specific TVLA outperforms specific TVLA as the number of false positives will be less in case of non-specific TVLA. It must be noted that in case of TVLA the focus is on identifying statistically significant information leakage and not on key extraction. Hence, observation of TVLA leakage may or may not lead to successful key extraction. Key extraction procedure depends upon the complexity of the attacks and correctness of hypothetical leakage model which again varies from device to device. Hence, it may happen that key extraction procedure fails due to wrong assumption of or high complexity of the hypothetical power model in-spite of having high TVLA leakage [6].

More elaborately, TVLA does not lead to key extraction neither it quantifies the side channel vulnerability. It is a *Pass-Fail* test which determines whether the crypto-implementation is safe or not. However, in some cases, it would be useful to know how unsafe the design is, which demands the need of quantification of side channel vulnerability. For example, a correct feedback on potential vulnerability to designers of crypto-systems, can lead to better implementations. However, in current form, TVLA fails to report side-channel vulnerabilities and evaluation based testing are too costly and expertise dependent to be deployed for this objective.

Related Work: Prior to TVLA, few information theoretic tests [7, 8] tests were proposed to analyse side channel vulnerability of crypto-systems. These tests are based on mutual information and can not be scaled to higher order attacks [9]. Additionally, these tests are complex and require computation of probability distribution of leakage and hence suffers from similar disadvantages of evaluation based testing. TVLA was first proposed in [4] where authors analysed validity of TVLA on AES. Subsequently, TVLA was applied to RSA [10] to show its effectiveness on public key cryptography. In this context, recently in [11] the authors have shown how to apply TVLA to asses horizontal attack vulnerabilities. In [5], results of [4] and [10] are brought together and superiority of *non-specific TVLA* over *specific TVLA* is established. TVLA is compared with mutual information based analysis techniques in [12] and comparative analysis between them is presented. In [9], authors have focussed on 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. An improved version of TVLA, based on *matched pair t-test* is presented in [13]. The advantage of TVLA is that it can detect leakage of any order and is independent of underlying architecture and hypothetical power model. It does not give any information regarding the ease of actual attack or exploitable leakage model. Hence, TVLA can not be extended to evaluation based testing which is a requirement for quantification of side channel vulnerability of crypto-designs.

Evaluation based testing requires the evaluator to check whether he can retrieve the secret key or not. Success of such evaluation based testing can be measured by two metrics: *Success Rate* [14] and *Guessing Entropy* [15]. 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, denoted as F_{k_0}, L is the leakage obtained from side channel traces. Lower the SR, higher is the resistance of crypto-implementation against the side channel attack A. It must be noted that SR indicates efficiency of a particular side channel attack and not the security of the design. In literature, multiple statistical distinguishers have been proposed to differentiate the correct key from the wrong key guesses. Most notable among them are *Difference of Mean (DoM)* and *Pearson's correlation coefficient* [16]. There have been multiple works which have analysed SR from the point of view of statistical distinguisher. In [14], the authors have defined SR for difference of mean (DoM) attack as follows:

$$SR = Pr[\delta_{k_0} > \delta_{<\overline{k_0}>}] \tag{2}$$

where k_0 is the correct key, $< \overline{k_0} >$ is the set of wrong key guesses and δ_{k_i} indicates DoM value of each key guess k_i. This definition was extended to address correlation power attack (CPA) in [17]. Additionally, authors in [14] have introduced a new parameter *confusion coefficient* which is used to estimate SR in terms side channel traces required to learn the secret key of the crypto-system for either DoM and CPA. SR of a side channel attack is often characterized by the *order* of the SR. For side channel attacks using either DoM or CPA, we rank all the possible candidate keys according to their DoM or correlation value where the key with highest DoM or correlation value is ranked 1. SR of *order* o indicates that rank of correct key is not more than o. Guessing entropy on the other hand is the measure of the post attack workload. It indicates the number of key hypothesises required to be tested after the side channel attack. Lower the guessing entropy, higher the success rate of the attack.

Generally, all sampling points on a side channel trace do not have equivalent leakage. There are some sampling points which provide more information leakage compared to others. Quality of sampling points from information leakage point of view is measured by a parameter known as Signal to Noise ratio (SNR). As an adversary, it is beneficial to focus only on the sampling points with high SNR as it increases the efficiency of the attacks by reducing the number of ghost peaks (wrong key guesses getting lower rank compared to correct key guess) [18, 19]. Additionally, for *Template Attack* [20], the complexity of attack increases significantly for side channel traces with large sampling points. In this context, it is extremely important to reduce the length of the side channel traces by focussing only on the high SNR sampling point of side channel trace. Various statistical and machine learning based techniques have been produced for such purpose. In [20], authors have used a template based approach which involves building templates for n different value of sub-key. High SNR points are then obtained by taking pairwise difference between these templates where high difference indicates high information leakage. This approach was improved in [21] where the authors have deployed *sum of squared difference (SOSD)* instead of pairwise difference of built templates. They have further modified their approach by executing *Student's t-test* on the templates to find out high SNR points.

From perspective of machine learning, compression of side channel traces to find out high SNR (or leakage) points leads to the problem of dimensionality reduction. In this context, authors in [22] proposed usage of *Principal Component Analysis (PCA)* whose goal is to gather all the information from high leakage points and reflect them on a new time basis with few points. This actually reduces the length of the side channel trace significantly which helps in efficient computation of the covariance noise matrix. Further, in [23], authors have used *Linear Discriminant Analysis (LDA)* with the objective of reaching optimal limits of a non-profiled *CPA*. In [24], the authors also support that *LDA* indeed leads to optimal dimensionality reduction.

On the other hand, template attacks need to have access to the cloned device, where adversary can build profiles of templates for different value of the sub-key. This may be a strong assumption in certain scenarios where such profiling is not feasible. Hence it is imperative to have some methodology which will bring out the high SNR leakage points without an explicit profiling step, thus not requiring access to a clone of the device. Such a strategy was proposed in [25] where authors introduced a new parameter *Normalized Inter Class Variance (NICV)* which can be used to estimate SNR of the sample points of side channel traces without any access to a cloned device. *NICV* is actually output of statistical F-test (also known as ANOVA (ANalysis Of VAriance)). It was shown in [25] that NICV approaches (squared) Pearson's correlation coefficient in absence of noise. Additionally, we can compute *SR* from *NICV* value which relates *SNR* with the *success rate (SR)*.

From the above discussion, it is clear that till now the research for validation and evaluation of side channel vulnerabilities of a crypto-implementation has followed independent paths. Testing for validation for side channel vulnerability can not quantify the side channel security whereas evaluation based testing is costly and expertise dependent. For quantification of side channel security, various metrics like *SR*, *Guessing Entropy* and *SNR* have been already proposed in the literature. On the other hand, recently proposed metric *TVLA* which is used for validation of side channel vulnerability has gathered significant interest among the researchers as it is independent of attack methodology and hypothetical power model. Nonetheless, till now any relationship between $TVLA$ and evaluation style based testing metrics (*SR*,*GE* and *SNR*) are not explored in the literature. Such relationship is actually of great importance as this will help quantify side channel security from $TVLA$ value and as a result extend its scope. In this paper, we try to formulate the relationship between $TVLA$ and *SNR* and there after estimate the lower bounds of the side channel traces required to break a given crypto-implementation.

Our Contribution: The main contribution of this paper are as follows:

- In this paper, we show how to formulate *SNR* of a crypto-implementation from the $TVLA$ metric. This allows us to estimate the *SR* from the $TVLA$ value, which in turn let us quantify side channel vulnerability of vulnerable designs.
- We will show that *non-specific* $TVLA$ actually captures only a fraction of the total *SNR*. On the other hand, from *specific* $TVLA$, we can compute the total *SNR* from $TVLA$.
- With the above results, we will extend the $TVLA$ based testing mechanism, to also quantify the side channel vulnerability in terms of number of side channel traces to attack. Our results also unify side channel metrics for both validation and evaluation and shows that all these metrics are actually equivalent.

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*). The derived relationship is experimentally validated in section 4 followed by application to AES in section 5. Finally in section 6 the conclusions are drawn.

2 Preliminaries

In this section we will provide a brief description of statistical hypothesis testing. As we have mentioned in the previous section, both $TVLA$, which is validation based testing mechanism and *NICV* which is an evaluation based testing mechanism are actually built on *Welch's t-test* and $ANOVA$ respectively. We will follow this discussion with a short note on *SR* and *SNR*.

2.1 Statistical Hypothesis Testing

Statistical tests often require to make decisions about a statistical population on the basis of sample observations. For example, given a random sample, it may be required to decide whether the population from which the sample has been obtained, is a normal distribution with a specific mean and standard deviation. Any statement or assertion about a statistical population or its parameters is called a Statistical Hypothesis. The procedure which enables us to decide whether a certain hypothesis is true or not is called Test of Significance or Statistical Hypothesis Testing.

A statistical hypothesis which is set up (i.e. assumed) and whose validity is tested for possible rejection on the basis of sample observations is called Null Hypothesis. It is denoted as H_0 and tested for acceptance or rejection. On the other-hand, an Alternative Hypothesis is a statistical hypothesis which differs from the null hypothesis, and is denoted as H_1. This hypothesis is not tested, its acceptance (or rejection) depends on the rejection (or acceptance) of that of the null hypothesis. The sample is then analysed to decide whether to reject or accept the null hypothesis. For this purpose, a suitable statistic, called Test Statistic is chosen. Its sampling distribution is determined, assuming that the null hypothesis is true. The observed value of the statistic would be in general different from the expected value because of sampling fluctuations. However if the difference is very large then the null hypothesis is rejected, Whereas, if the differences is less than a tolerable limit then H_0 is not rejected. Thus it is necessary to formally determine these limits.

Assuming the null hypothesis to be true, the probability of obtaining a difference equal to or greater than the observed difference is computed. If this probability is found to be small, say less than 0.05, the conclusion is that the observed value of the statistic is rather unusual, and has arisen because the underlying assumption, i.e. the null hypothesis is not true. We say that the observed difference is significant at 5 per cent level of significance, and hence the null hypothesis is rejected at 5 per cent level of significance. The level of significance, say α also corresponds to a $(1-\alpha)$ level of confidence. If however this probability is not very small, say more than 0.05, the observed difference cannot be considered unusual and is attributed to sampling fluctuations only. The difference, now is not significant at 5 per cent level of significance. The region in which null hypothesis is rejected is known as the *critical region.*

To formulate the above discussion mathematically, we need to introduce a term *Standard Error (SE). SE* is defined as the standard deviation of sampling distribution. We state formally a subsequent result on standard errors which will be useful to understand the subsequent discussion on the detection test.

Theorem 1. *Consider two independent simple samples of sizes n_1 and n_2, with means μ_1 and μ_2, and standard deviations σ_1 and σ_2 respectively, then:*

$$SE(\mu_1 - \mu_2) = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \tag{3}$$

Once we have defined the *critical region* and *level of significance (α)*, we compute the following parameter

$$z = \frac{\text{(Observed Value)} - \text{(Expected Value)}}{\text{Standard Error (SE)}} \tag{4}$$

We will assume the null hypothesis H_0 to be true if it gets rejected with α percent of level of significance. If the percentage of rejection goes beyond α, we assume the null hypothesis H_0 is false.

There are different methods for computation of parameter z. In the next subsection, we will focus on two such test: *Welch's t-test* and *ANOVA*.

2.2 Welch's t-test

Welch's t-test is essentially a test of equality of two moments drawn independently and randomly from two populations. The starting point is the first moment, where equality of two means from the two samples are tested for equality. In this case, the null hypothesis is $H_0(\mu_1 = \mu_2)$, where μ_1 and μ_2 are the two means for the two independent samples. As discussed, the standard error of the difference of means $\mu_1 - \mu_2$ is $SE(\mu_1 - \mu_2) = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$. We denote the output of *Welch's t-test* as t and it is computed as follows:

$$t \frac{= \mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{5}$$

where σ_1 and σ_2 are the standard deviations of two independent random samples. For a large distribution, the test statistic t follows standard normal distribution. However, for tests with any sample sizes, a more exact sampling distribution for t is the t-distribution, and this gives rise to the *Welch's t-test.* The statistic t then follows the t-distribution with degrees of freedom calculated according to Welch-Satterthwaite, as $v = \frac{SE(\mu_1 - \mu_2)}{\frac{(\sigma_1^2/n_1)}{n_1 - 1} + \frac{(\sigma_2^2/n_2)}{n_2 - 1}}$.

2.3 ANOVA

In the previous section, we have introduced *Welch's t-test* for statistical hypothesis testing. *Welch's t-test* is applicable when the number of independent samples classes are two. However, for many real life scenarios, the number of independent sample classes could be more than two. In such cases, to test null hypothesis , we need to apply *t-test* multiple times. Alternatively, we can execute a single *F-test* to check the null hypothesis. In statistical terms, *F-test* is defined as follows

$$F = \frac{Explained\ Variance}{Unexplained\ Variance} = \frac{Inter - Class\ Variance}{Intra - Class\ Variance} \tag{6}$$

This *F-Test* is also known as *ANOVA (ANalysis Of VAriance)*. Computation of *F-test* or *ANOVA* involves computation of two terms: *error sum of squares (SS_{err})* and *treatment sum of squares (SS_{treat})* [26]. Before we define these parameters, we need to define the statistical experiment on which we will enact *F-test.* In this study, we want to analyse behaviour of a population $\mathcal{Y}$, whose variation depends upon a random variable X. The domain of this random variable is denotes as $\mathcal{X}$. From side channel perspective, $\mathcal{Y}$ can be considered as side channel information leakage whose variation depends upon leakage model (e.g. Hamming weight or Hamming distance) which can be denoted as X. We also assume that cardinality of $\mathcal{X}$ is Q. The first step is to sample the population of $\mathcal{Y}$

and partition those samples into Q number of groups. We assume that i^{th} group has n_i number of elements where $i \in \mathbb{N}_{\mathbb{Q}}$. We denote each element of these groups as $Y_{i,j}$ where i indicates the group and j indicates the element inside the group i. We also create another group by accumulating all elements together denoted as $\mathbf{Y} = \{Y_{1,1}, \ldots, Y_{1,n_1}, \ldots, Y_{Q,1}, \ldots, Y_{Q,n_Q}\}$. The mean of $\mathbf{Y}$ is denoted as μ, whereas mean of individual groups are denoted as $\mu_i, i \in \{1, 2, \ldots, Q\}$. With this definitions in mind, we now define parameters SS_{err} and SS_{treat} as follows:

$$SS_{err} = \sum_{x \in \mathcal{X}} \sum_{i=1}^{n_x} (Y_{x,i} - \mu_x)^2 \tag{7}$$

$$SS_{treat} = \sum_{x \in \mathcal{X}} n_x (\mu_x - \mu)^2 \tag{8}$$

The value of F-test is computed as follows

$$F = \frac{SS_{treat} \times (N - Q)}{(Q - 1) \times SS_{err}}, N = \sum_{x \in \mathcal{X}} n_x \tag{9}$$

Like *Welch's t-test*, the objective of *F-test* also is to check the validity of null hypothesis H_0, which in this case is defined as follows

$$H_0 : \mu_1 = \mu_2 = \ldots = \mu_k \tag{10}$$

Given a level of significance α, we determine the region of rejection from the *F-distribution* table. If the result of *F-test* belongs to the region of rejection we reject the null hypothesis H_0, otherwise we accept it.

For side channel vulnerability measurement, we are not interested in the exact value of F-test. However, in subsection 2.5 we will show that the concept of *ANOVA* helps us to build a very useful metric *Normalized Intra Class Variance* which we can directly relate to *SNR*. But before that in the next subsection we will introduce *TVLA*.

2.4 Test Vector Leakage Assessment (TVLA)

In section 2.2, we have introduced *Welch's t-test* as statistical hypothesis testing mechanism. As we have mentioned in section 1, *Test Vector Leakage Assessment (TVLA)* is direct application of *Welch's t-test* on side channel traces for validation of side channel vulnerabilities.

TVLA methodology can be classified in to 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*, one set corresponds to a fixed key and fixed plain-text as input to the cryptographic IP, the second set collects traces corresponding to same fixed key and random plain-text. We consider the side channel information leakage as a random variable $\mathcal{Y}$ and the set of side channel traces captured is denoted by Y. The captured side channel traces are then partitioned into two different sets: Y^f (fixed plain-text as input) and Y^r (random plain-text as input). Thereafter a hypothesis testing performed by assuming a null hypothesis that the 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 presence of exploitable leakage. This can be expressed as:

$$TVLA = \frac{\mu_r - \mu_f}{\sqrt{\frac{\sigma_r^2}{n_r} + \frac{\sigma_f^2}{n_f}}} \ , \tag{11}$$

where n_r, n_f signifies the number of traces in set Y^r, Y^f respectively. The mean and standard deviation of set Y_r is denoted by μ_r and σ_r. Similarly, μ_f and σ_f refer to mean and standard deviation of Y^f. The null hypothesis of two equal means is rejected when the $TVLA$ exceeds a threshold of ±4.5, which ensures with degrees of freedom > 100, $P[|TVLA| > 4.5] < 0.00001$, this threshold leads to a confidence of 0.99999. Thus, if the $TVLA$ value is within ±4.5, we can claim that the crypto-implementation is secure with high confidence. Otherwise, we reject the null hypothesis and declare the crypto-implementation to leak exploitable side-channel information.

In case of *non-specific TVLA*, we partition the side channel traces according to the plain-text. Hence, knowledge of secret key is not required for performing *non-specific TVLA*. However, 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 way to do this partitioning.

- In the first case, a particular round is selected and the intermediate data is computed by xoring the input and output of that round. Then, for each bit of the computed intermediate data, we partition the side channel traces depending upon whether that particular bit is zero or one. $TVLA$ is computed for each bit of the intermediate data and its value should be within ±4.5 for all of them. Similar analysis can be carried out by considering *S-Box* output or a particular round output as the intermediate data.
- In the second case, we consider the first byte of a particular round output as the intermediate data. $\forall i \in \mathbb{Z}_{256}$, we partition the traces into two groups depending upon whether the value of intermediate data is equal to i or not. Once the partitioning is done, we compute $TVLA$ for each value of i.

2.5 Normalized Inter Class Variance

Normalized Inter-Class Variance (NICV) is a technique which was designed to detect relevant point of interest (PoI) in an SCA trace [25]. This is an extremely useful tool for side channel trace compression and dimensionality reduction. *NICV* is based on *ANOVA*, introduced in section 2.3. The advantage of *NICV* is that, like *non-specific TVLA*, *NICV* can be applied with the knowledge of only plain-text and cipher-text and does not require knowledge of target implementation or secret key. A side-channel adversary acquired leakage measurement $Y \in \mathbb{R}$ corresponding to a public parameter X (lets say a byte of plaintext or ciphertext i.e $\mathcal{X} = \mathbb{F}_2^8$). For this paper we consider the public parameter X is a k bit parameter, having 2^k possible values. The leakage prediction function is denoted as L which takes public parameter X as input. As shown in [25, 27],we can define the following relation

$$\rho^2 \left[L(X); Y\right] = \underbrace{\rho^2 \left[L(X); \mathbb{E}\left[Y|X\right]\right]}_{0 \leq \ \cdot \ \leq 1} \times \rho^2 \left[\mathbb{E}\left[Y|X\right]; Y\right] \quad . \tag{12}$$

Here, $\mathbb{E}$ and Var denotes the expectation and the variance respectively, whereas ρ represents correlation. Eq. (12) was further simplified in [25, 27] to derive:

$$\rho^2 \left[\mathbb{E}\left[Y|X\right]; Y\right] = \frac{\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]}{\mathsf{Var}\left[Y\right]} \quad , \tag{13}$$

The term in Eq. (13) is further called as the *normalized inter-class variance (NICV)*. *NICV* can also be expressed in terms of the result of *F-test* or *ANOVA*, introduced in section 2.3, as it is a ratio between the explained variance and the total variance. *F-test* depends upon two parameter: SS_{treat} and SS_{error}. For the computation of *NICV*, we define another term SS_{total} below

$$\begin{aligned} SS_{total} &= SS_{treat} + SS_{err} \\ &= \sum_{x \in \mathcal{X}} n_x (\mu_x - \mu)^2 + \sum_{x \in \mathcal{X}} \sum_{i=1}^{n_x} (Y_{x,i} - \mu_x)^2 \\ &= \sum_{x \in \mathcal{X}} \sum_{i=1}^{n_x} (Y_{x,i} - \mu)^2 \\ &= N \times \mathsf{Var}\left[Y\right] \end{aligned} \tag{14}$$

$$\begin{aligned} SS_{treat} &= \frac{\sum_{x \in \mathcal{X}} n_x (\mu_x - \mu)^2}{\sum_{x \in \mathcal{X}} n_x} \times \sum_{x \in \mathcal{X}} n_x \\ &= N \times \mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right] \end{aligned} \tag{15}$$

The different symbols used in the above equations are defined in section 2.3. From equation (14) and equation (15), we can define *NICV* as below:

$$\boxed{NICV = \frac{\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]}{\mathsf{Var}\left[Y\right]} = \frac{SS_{treat}}{SS_{total}}} \tag{16}$$

Combining equation (9) and equation (16), we can derive the relation between the *F-test* and *NICV* which is given in equation (17).

$$\boxed{F = \frac{NICV \times (N - Q)}{(1 - NICV) \times (Q - 1)}} \tag{17}$$

In [25], the authors have shown that $NICV$ is the maximum of all possible correlation from X with Y. Also in the same paper, the authors have given the relationship between $NICV$ and SNR which is shown in equation (18).

$$\boxed{\mathrm{NICV} = \frac{\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]}{\mathsf{Var}\left[Y\right]} = \frac{1}{1+\frac{1}{\mathrm{SNR}}}}\ , \tag{18}$$

For details of this derivation, the readers may refer to [25]. The value $\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]$ constitutes *signal* of the *SNR*. On the other hand, $\mathsf{Var}\left[Y\right] - \mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]$ denotes the noise part. Equation (18) is actually a very useful expression as it relates *NICV* with *SNR*, which itself is related with *SR*. In the next subsection, we focus on this relationship

2.6 SNR and SR

We have already presented the relationship between *SNR* and *NICV* in the previous subsection. In [14, 17], the authors have proposed following formulation for computation of *SR*

$$SR = \phi(\sqrt{m}\Sigma^{-1/2}\boldsymbol{\mu}) \tag{19}$$

where ϕ is a multivariate Gaussian cumiliative distributive function, m denotes number of side channel traces captured. Additionally, assuming N_k is the total number of candidate keys, Σ is a $(N_k - 1) \times (N_k - 1)$ matrix. $\boldsymbol{\mu}$ is a column vector of cardinality $N_k - 1$, whose elements are function of *confusion coefficient* [14, 17]. Σ is actually covariance matrix whose value depends upon the difference of correlation value between correct key and wrong keys. In [25], the authors have further simplified equation (19) which gives us the following relationship

$$SR = \phi\left(\sqrt{m \times \frac{\kappa_0 - \kappa_1}{2\sigma^2}}\right) \tag{20}$$

Here κ_0 is the *generalized confusion coefficient* and σ^2 is the variance of the noise. The term $\frac{\kappa_0 - \kappa_1}{\sigma^2}$ is actually the *SNR* of the side channel leakage [25]. Hence we can rewrite equation (20) as below

$$SR = \phi\left(\sqrt{m \times 2 \times SNR}\right) \tag{21}$$

$$m = \frac{2}{SNR} \times (\phi^{-1}(SR))^2 \tag{22}$$

This value of m denotes the minimum number of traces that a side channel adversary must capture to get access to the corresponding key byte. It must be noted that in [25], the authors have formulated equation (20) with the assumption that the correct key will have a significantly larger correlation value compared to the wrong key guesses. In actual attack, due to the occurrence of ghost peaks for wrong key guesses, the number of traces required to do *CPA* would be larger compared to m. Hence the value of m is the lower bound to the number of side channel traces required. For 80% SR, $\phi^{-1}(SR = 80\%)$ can be computed as .9056 from error function table. Thus Eq. (21) simplifies to $m_{SR(80\%)} = \frac{1.64}{SNR}$.

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

3 Equivalence of TVLA and NICV

The objective of this section is to establish relationship between *TVLA* and *NICV*, which will be the first step in connecting *TVLA* with *SNR*. 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 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 in this case, the corresponding two group *NICV* is denoted as $NICV_2$. This $NICV_2$ will be generalized in the following subsection.

Theorem 2. *Consider two group of side channel traces Y_1 and Y_2 with cardinality n_1 and n_2. The computation of* TVLA *and $NICV_2$ on these two groups are related by the following formula*

$$\boxed{\boldsymbol{NICV_2 = \frac{1}{\frac{n}{TVLA^2} + \frac{n}{C}\left(\sigma_1^2 - \sigma_2^2\right)\left(\frac{1}{n_2} - \frac{1}{n_1}\right) + 1}}} \tag{23}$$

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

Proof. From equation (16) we can write $NICV_2$ as below:

$$NICV_2 = \frac{\frac{1}{n}\sum\limits_{i=1}^{2} n_i(\mu_i - \mu)^2}{\frac{1}{n}\sum\limits_{i=1}^{2}\sum_{j=1}^{n_i}(Y_{i,j} - \mu)^2}$$

$$= \frac{\frac{1}{n}\sum_{i=1}^{2} n_i(\mu_i - \mu)^2}{\frac{1}{n}\sum_{j=1}^{n}(Y_j - \mu)^2} \tag{24}$$

From equation (11) we can write $TVLA$ as follows:

$$\begin{aligned} TVLA &= \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \\ TVLA^2 &= \frac{(\mu_1 - \mu_2)^2}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \\ &= \frac{C}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \end{aligned} \tag{25}$$

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

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{2} n_i\left(\mu_i - \mu\right)^2 \\ &= \frac{1}{n}\left(n_1\left(\mu_1 - \mu\right)^2 + n_2\left(\mu_2 - \mu\right)^2\right) \\ &= \frac{1}{n}\left(n_1\left(\mu_1 - \frac{n_1\mu_1 + n_2\mu_2}{n}\right)^2 + n_2\left(\mu_2 - \frac{n_1\mu_1 + n_2\mu_2}{n}\right)^2\right) \\ &= \frac{1}{n}\left(n_1\left(\frac{n_1\mu_1 + n_2\mu_1 - n_1\mu_1 - n_2\mu_2}{n}\right)^2 + n_2\left(\frac{n_1\mu_2 + n_2\mu_2 - n_1\mu_1 - n_2\mu_2}{n}\right)^2\right) \\ &= \frac{1}{n}\left(\frac{n_1 n_2^2}{n^2}\left(\mu_1 - \mu_2\right)^2 + \frac{n_1^2 n_2}{n^2}\left(\mu_1 - \mu_2\right)^2\right) \\ &= \frac{n_1 n_2}{n^3}\left(n_2(\mu_1 - \mu_2)^2 + n_1(\mu_1 - \mu_2)^2\right) \\ &= \frac{n_1 n_2(n_1 + n_2)}{n^3}C \\ &= \frac{n_1 n_2}{n^2}C \end{aligned} \tag{26}$$

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

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \mu\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \frac{n_1\mu_1 + n_2\mu_2}{n}\right)^2 \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i^2 - \frac{2Y_i\left(n_1\mu_1+n_2\mu_2\right)}{n} + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\right)\\
&= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i^2 - \frac{2Y_i\left(n_1\mu_1+n_2\mu_2\right)}{n}\right) + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\\
&= \frac{1}{n}\sum_{Y_i\in Y^{G1}}\left(Y_i^2 - \frac{2Y_i\left(n_1\mu_1+n_2\mu_2\right)}{n}\right) + \frac{1}{n}\sum_{Y_i\in Y^{G2}}\left(Y_i^2 - \frac{2Y_i\left(n_1\mu_1+n_2\mu_2\right)}{n}\right) + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\\
&= \frac{1}{n}\sum_{Y_i\in Y^{G1}}\left(Y_i^2 - \frac{2Y_i\left(n-n_2\right)\mu_1+2Y_in_2\mu_2}{n}\right) + \frac{1}{n}\sum_{Y_i\in Y^{G2}}\left(Y_i^2 - \frac{2Y_in_1\mu_1+2Y_i\left(n-n_1\right)\mu_2}{n}\right)\\
&+ \frac{\left(n_1\mu_1+n_2\mu_2\right)^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_in_2\left(\mu_1-\mu_2\right)}{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_in_1\left(\mu_2-\mu_1\right)}{n}-\mu_1^2\right)\right) + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\\
&= \frac{1}{n}\sum_{Y_i\in Y^{G1}}\left(Y_i-\mu_1\right)^2 + \frac{1}{n}\sum_{Y_i\in Y^{G2}}\left(Y_i-\mu_2\right)^2 - \frac{n_1}{n}\mu_1^2 - \frac{n_2}{n}\mu_2^2\\
&+ \frac{2n_2\left(\mu_1-\mu_2\right)}{n^2}\sum_{Y_i\in Y^{G1}}Y_i + \frac{2n_1\left(\mu_2-\mu_1\right)}{n^2}\sum_{Y_i\in Y^{G2}}Y_i + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\\
&= \frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 - \frac{n_1}{n}\mu_1^2 - \frac{n_2}{n}\mu_2^2 + \frac{2n_1n_2\mu_1\left(\mu_1-\mu_2\right)}{n^2} + \frac{2n_1n_2\mu_2\left(\mu_2-\mu_1\right)}{n^2} + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\\
&= \frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 - \frac{n_1}{n}\mu_1^2 - \frac{n_2}{n}\mu_2^2 + \frac{2n_1n_2}{n^2}\left(\mu_1-\mu_2\right)^2 + \frac{\left(n_1\mu_1+n_2\mu_2\right)^2}{n^2}\\
&= \frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 + \frac{-\mu_1^2n_1(n_1+n_2)-\mu_2^2n_2(n_1+n_2)+2n_1n_2(\mu_1-\mu_2)^2+(n_1^2\mu_1^2+n_2^2\mu_2^2+2n_1n_2\mu_1\mu_2)}{n^2}\\
&= \frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 + \frac{n_1n_2}{n^2}(\mu_1-\mu_2)^2\\
&= \frac{n_1}{n}\sigma_1^2 + \frac{n_2}{n}\sigma_2^2 + \frac{n_1n_2}{n}C \qquad (27)
\end{aligned}
$$

We can now combine equation (15), (25), (26) and (27) to achieve the desired formulation

$$
\begin{aligned}
NICV_2 &= \frac{\frac{n_1n_2}{n^2}C}{\frac{n_1}{n}\sigma_1^2+\frac{n_2}{n}\sigma_2^2+\frac{n_1n_2}{n^2}C}\\
&= \frac{C}{\frac{n}{n_2}\sigma_1^2+\frac{n}{n_1}\sigma_2^2+\frac{n_1n_2}{n^2}C}\\
&= \frac{C}{n\left(\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}+\sigma_1^2\left(\frac{1}{n_2}-\frac{1}{n_1}\right)+\sigma_2^2\left(\frac{1}{n_1}-\frac{1}{n_2}\right)\right)+C}
\end{aligned}
$$

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

Thus we can write $NICV_2$ as

$$\boxed{\boldsymbol{NICV_2 = \frac{1}{\frac{n}{TVLA^2} + \frac{n}{C}(\sigma_1^2 - \sigma_2^2)\left(\frac{1}{n_2} - \frac{1}{n_1}\right) + 1}}}$$

Corollary 1. *If both the group have same number of side channel traces ($n_1 = n_1 = \frac{n}{2}$), equation* (23) *transforms into*

$$\boxed{\boldsymbol{NICV_2 = \frac{1}{\frac{n}{TVLA^2} + 1}}} \tag{28}$$

3.1 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 extented 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 k is 256. $NICV_k$ can be directly computed from $NICV_2$ by following an iterative approach. For the derived k groups, pairwise computation of (k-1) 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 mean of these two groups are denoted as μ_i and $\mu_{\bar{i}}$ respectively.
- Compute $NICV_2$ for each of these two groups. We denote this as $NICV_2^i$.

Theorem 3. *The computation of $NICV_k$ and $NICV_2^i$ are related by the following formula if all k groups have same number of side channel traces*

$$\boxed{\boldsymbol{NICV_k = \frac{k-1}{k}\sum_{i=1}^{k} NICV_2^i}} \tag{29}$$

Proof. From equation (15), we can compute $NICV_2^i$ as below

$$\begin{aligned}
NICV_2^i &= \frac{\frac{1}{n}\left(n_i\left(\mu_i-\mu\right)^2+\left(n-n_i\right)\left(\mu_{\bar{i}}-\mu\right)^2\right)}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \\
&= \frac{\frac{1}{n}\left(n_i\left(\mu_i-\mu\right)^2+\left(n-n_i\right)\left(\frac{n\sum\limits_{j=1,j\neq i}^{k} n_j\mu_j-(n-n_i)\sum\limits_{j=1}^{j=k} n_j\mu_j}{n(n-n_i)}\right)^2\right)}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \\
&= \frac{\frac{1}{n}\left(n_i\left(\mu_i-\mu\right)^2+\frac{1}{n-n_i}\left(\frac{n_i\sum\limits_{j=1}^{j=k} n_j\mu_j-nn_i\mu_i}{n}\right)^2\right)}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \\
&= \frac{\frac{1}{n}\left(n_i\left(\mu_i-\mu\right)^2+\frac{n_i^2}{n-n_i}\left(\mu_i-\mu\right)^2\right)}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \\
&= \frac{\frac{n_i}{n-n_i}\left(\mu_i-\mu\right)^2}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \qquad (30)
\end{aligned}$$

Let us further assume that each group has same number of side channel traces. equation (30) becomes

$$NICV_2^i = \frac{\frac{1}{k-1}\left(\mu_i-\mu\right)^2}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \qquad (31)$$

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

$$\begin{aligned}
\sum_{i=1}^{k} NICV_2^i &= \frac{\frac{1}{k-1}\sum\limits_{i=1}^{k}(\mu_i-\mu)^2}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \\
&= \frac{\frac{k}{k-1}\frac{1}{k}\sum\limits_{i=1}^{k}(\mu_i-\mu)^2}{\frac{1}{n}\sum\limits_{j=1}^{n}(Y_j-\mu)^2} \\
&= \frac{k}{k-1}NICV_k \qquad (32)
\end{aligned}$$

Algorithm 1: Computing *SNR* and $m_{SR(90\%)}$ from $TVLA$

```
   Input: Side channel traces and corresponding intermediate state
   Output: SNR, $m_{SR(80\%)}$ 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 equation (23)
7  Compute $NICV_k = \frac{k-1}{k} \sum_{i=1}^{k} NICV_2^i$
8  Compute $SNR = \frac{1}{\frac{1}{NICV_k} - 1}$
9  $m_{SR(80\%)} = \frac{1.64}{SNR}$
10 Return $SNR$, $m_{SR(80\%)}$
```

From Eq. (32) this we can simply derive Eq. (29). It must be noted that $NICV_k$ is actually the generalized *NICV* which was introduced in [25].

3.2 Extending *TVLA* flow to Side-Channel Analysis

Side channel analysis works using divide and conquer approach. For instance, *SPN* cipher where each $b \times b$ S-box handle 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 sub-key. The attack is repeated 16 times to recover all the key bytes in AES-128. This reduces the complexity of the attack significantly. The same applies to 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. Thus in this case, the value of k reduces to

Now we present the methodology to extend the $TVLA$ computation to recover *SNR*. As *SNR* is a main component in Eq. (21), one can directly use $TVLA$ results to derive the lower bound on minimum number of traces required for a successful side-channel attack. The methodology is presented in Algorithm 1. The algorithm is repeated for each sub-key to recover the whole secret key.

It must be noted that partitioning the side channel traces, depending upon a particular byte value of the intermediate state was deployed for *specific* $TVLA$ also. Steps 1 and 2 of algorithm 1 are actually application of *specific* $TVLA$. Thus using the formalization approach presented in this and previous sections, we can compute *SNR* of the crypto-system from *specific* $TVLA$ computation. For *non-specific* $TVLA$, the traces are partitioned depending upon the entire plain-text value, where one group contains traces with fixed plain-text and other contains traces with random plain-text. Thus, if we want to extend our approach to *non-specific* $TVLA$, we need to follow the following steps.

- Choose a plain-text value
- Collect side channel traces and partition them into two groups, one group contains traces with the chosen plain-text and the other group contains traces with random plain-text
- Apply $TVLA$ on these two groups
- Repeat this procedure for all possible values of plain-text

The last step is practically infeasible. It would need combinination of all possible $NICV_2$ value for computation of *generalized* $NICV$ which is equivalent to brute force. Hence the i^{th} instance of *non-specific* $TVLA$ captures only $NICV_2^i$ which leads to only a fraction of *SNR*, whereas using *specific* $TVLA$ we can compute the *SNR* for each sub-key. Finally, *SNR* leads to (lower-bound) success-rate of by side-channel attack following Eq. (21).

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

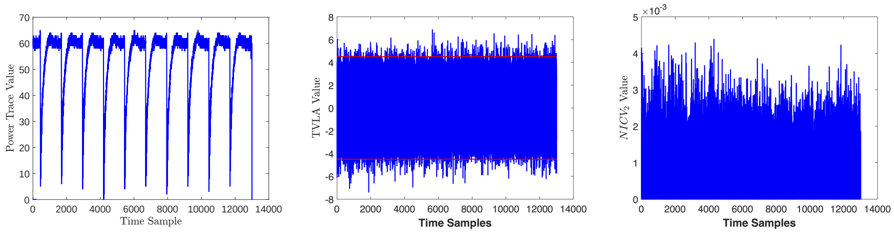
The derived relation between *specific* $TVLA$ and *SNR* (or $NICV$) will be experimentally validated in this section on an AES-128 implementation (without side-channel countermeasures) running on an FPGA.

4.1 Experimental Setup

The *AES* design is implemented on a SASEBO-GII platform [28]. SASEBO-GII has two FPGAs, one for controlling communication with the board (*SPARTAN-3A (XC3S400A)*) and another for execution of cryptographic operations (*VIRTEX-5 (XC5VLX50)*). Thus the AES is implemented on Virtex-5. The power measurements are taken using a Tektronix MSO4034B mixed signal oscilloscope with sampling frequency 2.5 GHz. 10000 traces corresponding to randomly generated plain-text are measured and used for the following computation. A sample trace is shown in Fig. 1 (a), while its worst case $TVLA$ and $NICV$ plots are shown in Fig. 1 (b) and (c). It is obvious that the AES has exploitable leakage as the $TVLA$ value is more than the threshold of 4.5.

4.2 Validation of *TVLA* and $NICV_2$ Relationship

$TVLA$ and $NICV_2$ are related by Eq. (23). It is verified on the previously collected power measurement for AES on FPGA. We start with partitioning the traces based on the first byte value ($k = 256$) of the intermediate state (round output), 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. 2. An example specific $TVLA$ trace is shown in Fig. 2 (a).Next the $TVLA$ trace in Fig. 2 (a) is used to compute $NICV_2$ using Eq. (23) and shown in Fig. 2 (b). We also compute $NICV_2$ from power measurement as shown in Fig. 2 (c). The error between predicted and computed $NICV_2$ is in the order of 10^{-16} i.e. negligible (Fig. 2 (d)). Thus, the correctness of the Eq. (23) is verified.



(a) Single AES Trace

(b) Maximum of all 256 possible TVLA plots

(c) $NICV_2$ Plot for all possible values of a particular byte

Fig. 1: $TVLA$ and $NICV_2$ on AES

4.3 Validation of $NICV_k$ and $NICV_2$ relationship

Similar validation is also done for Eq. (29) 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. 3. As the computed $NICV_k$(Fig. 3 (a)) follows closely the predicted $NICV_k$ (Fig. 3 (b)), the prediction error (Fig. 3 (c)) also stays in the range of 10^{-15}.

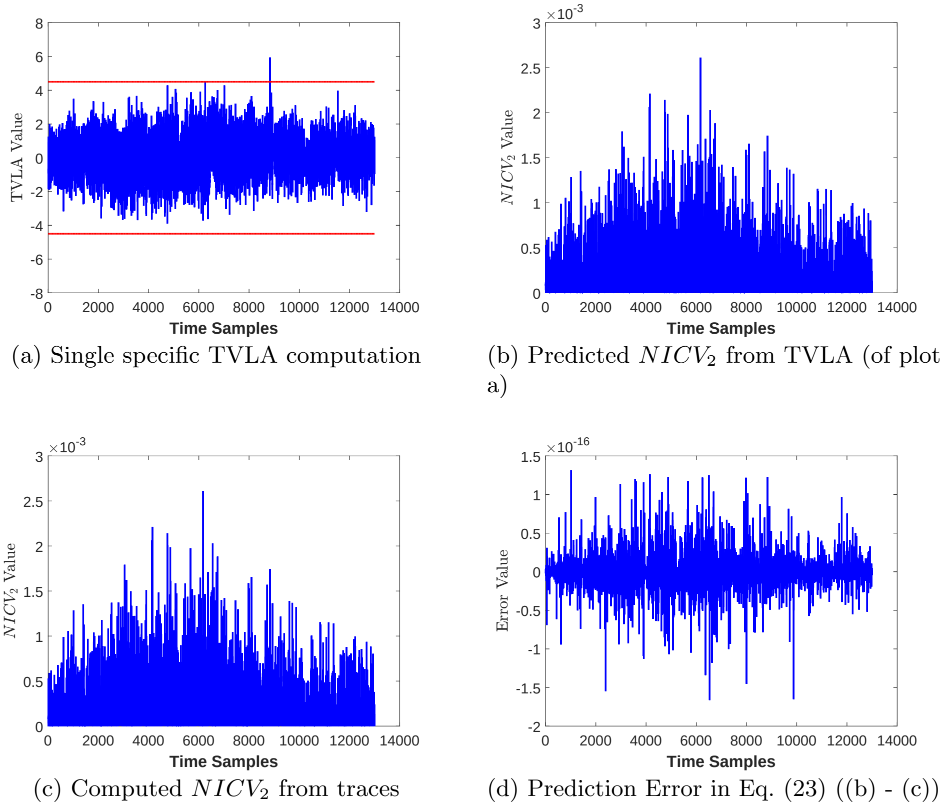
5 Case Study: Application to AES

The equivalence of $TVLA$ and SNR theoretically derived and experimentally verified in the previous sections. The step by step procedure to compute SNR from the *specific* $TVLA$ value was also presented in Algo. 1. In this section, we will focus on the application of these relations towards testing an unprotected AES-128 design.

5.1 Under Simulated Setting

The first result that we will present in this section is built on simulated AES side channel traces, with the assumption of 32 bit micro-controller as implementation platform. The side channel traces are built using *Hamming weight* leakage model, where the information leakage is proportional to the sum of bits set to $'1'$. We also assume that the side channel traces are contaminated with a zero mean Gaussian noise ($\mathcal{N}(0, \sigma)$), where σ denotes the standard deviation of the noise distribution. Thus the side channel trace can be represented as $Y = HW(x) + \mathcal{N}$, where x is the chosen intermediate value, which in our case is first 32-bits of round 9 output and $\sigma \in [0.0, 0.4, 0.8, 1.2, 1.6, 2.0]$.

Next, we directly apply Algo. 1 to first derive SNR followed by $m_{SR(80\%)}$. A practical CPA attack is also performed on the set of the traces to compare actual number of traces against predicted lower bound for $SR = 80\%$. The procedure to compute the number of side channel traces required for a given success rate SR



(a) Single specific TVLA computation

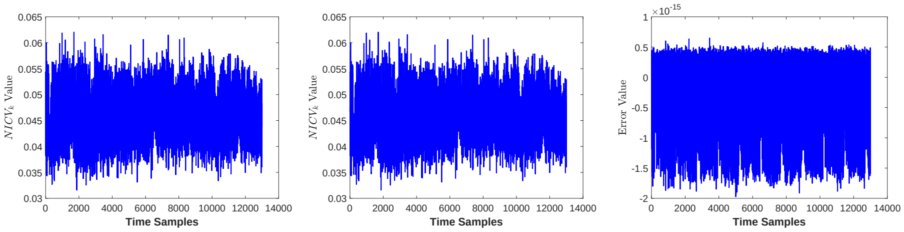
(b) Predicted $NICV_2$ from TVLA (of plot a)

(c) Computed $NICV_2$ from traces

(d) Prediction Error in Eq. (23) ((b) - (c))

Fig. 2: Equivalence of $TVLA$ and $NICV_2$

for a given noise variance σ is given in Appendix A. In our experimentation, we apply this algorithm to compute the number of side channel traces required for $SR = 80\%$. The corresponding result is shown in Fig 4 and Fig. 5. As expected the SNR reduces at higher noise σ (refer Fig. 4). Fig. 5 plots the predicted value of $m^{predicted}_{SR(80\%)}$ as derived from Algo. 1 as a function of σ. The traces were also attacked using CPA to find $m^{actual}_{SR(80\%)}$ i.e. the actual number of traces to achieve 80% SR are also plotted in Fig. 5. It can be clearly observed that the actual and predicted number of traces differ by a significant margin which deepens as noise increases. The prediction error comes from several sources. One major source of error is the assumption that wrong key have near zero correlation, which is not true in actual attack. Moreover, the confusion coefficient in the formula of m_{SR} is dependant on leakage model. Thus inaccurate estimation of leakage model would further increase the error. This phenomena will be stronger in real measurements as the leakage model will have some estimation error due to underlying non-linearities.



(a) Computed $NICV_k$ (b) Predicted $NICV_k$ (c) Error in Eq. (29) ((a) - (b))

Fig. 3: Prediction of $NICV_k$

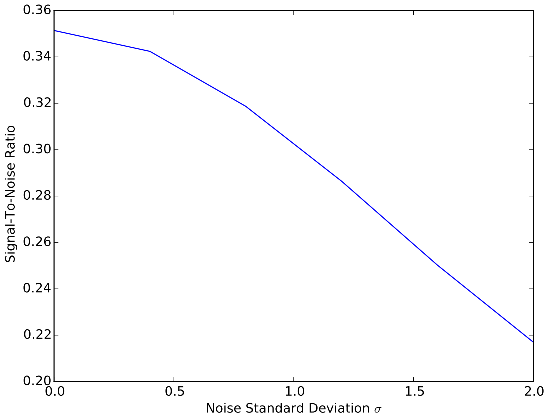


Fig. 4: Computed SNR in AES traces as a function of σ

5.2 On Real FPGA Target

The computation of SNR and SR from $TVLA$ is performed on real power measurements. Power measurements are acquired from a SASEBO-GII board running AES-128 on Virtex-5 FPGA using Tektronix MSO4034B mixed signal oscilloscope with sampling frequency 2.5 GHz. Two versions of AES are tested: unprotected AES, AES with a Linear Feedback Shift Register (LFSR) based noise generator. Only last round of AES is recorded to reduce the attack complexity. LFSR based noise generator aims at reducing the SNR, thus increasing the number of traces to perform the attack. However, introducing $LFSR$ to increase the noise in the circuit actually is not a very sound countermeasure and can be broken by a doing the attack on few additional traces. Hence, both unprotected AES and $LFSR$ based AES will fail the $TVLA$ test as shown in Fig 6(a) and Fig 6(b). The corresponding SNR, which we have computed from the $TVLA$ value, are shown in Fig. 6(c) and Fig. 6(d). The maximum SNR value for normal AES execution is 0.0660 whereas for *noise injected AES*, SNR reduces to 0.0650,

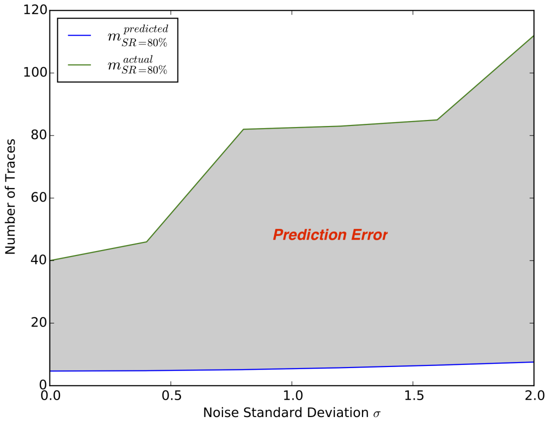


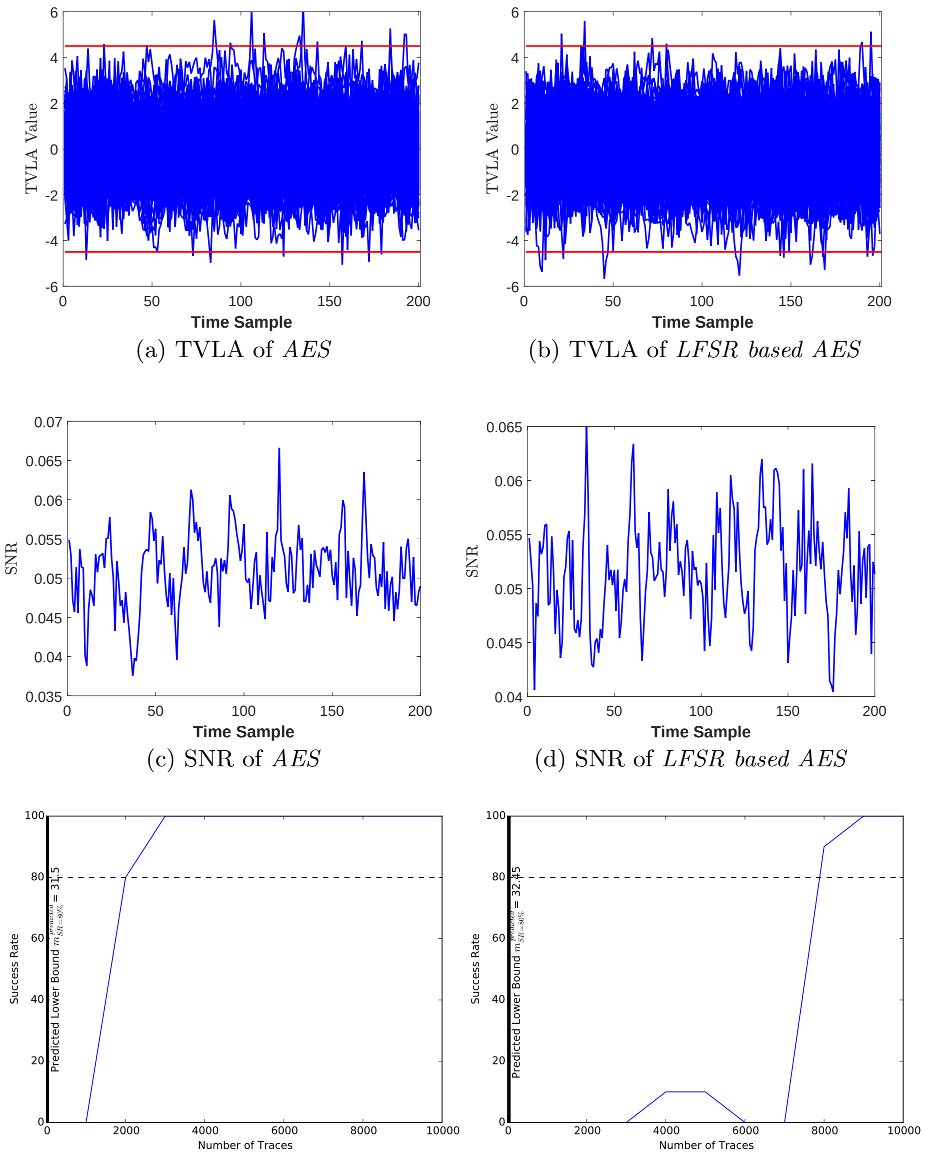
Fig. 5: Comparison Between Predicted Lower Bound and Actual Trace Required

indicating higher side channel resistance. The lowest bound of side channel traces required for 80% success rate is found to be 32.45 for *noise injected* AES, whereas for normal AES, this value is 31.635. According to our proposed methodology, this indicates that noise injected AES is more resistant against CPA compared to normal AES, however the difference of lower bound is too low to be considered as a viable countermeasure. This claim is supported by Fig. 6(e) and 6(f) where we have deployed CPA on the acquired traces to compute the number of traces required for 80% SR. Normal AES provides 80% SR within 1000 traces. However, *noise injected* AES provides 80% SR after 7000 traces.

Thus we verified the extended $TVLA$ test on AES in simulated as well as practical settings to recover SNR values and lower bound on the side channel traces required for a given success rate.

5.3 Further Discussions

Although $m^{predicted}_{SR(80\%)}$ and $m^{actual}_{SR(80\%)}$ were following an expected trend, there was a huge prediction error as shown in Fig. 5, Fig 6(e) and Fig 6(f). Under a real evaluation scenario, this leads to under-estimation of security (false positives) which is not desirable from a customer/designer view-point. This error can be owing to certain underlying strong assumption, as a result of which the lower bound of side channel traces is much below compared to actual traces required for achieving the given SR. First factor, as previously mentioned, is the assumption in Eq. (21). This equation assumes that the wrong key will have null correlation, which is not true in a real attack. Thus further improvements of Eq. (21) towards reduction in prediction error will enhance the applicability of proposed methodology in an evaluation laboratory. The other and a dominant source of error is the leakage model. When the evaluation was carried out under simulated setting with perfect leakage model, the prediction error was low.



(a) TVLA of *AES*

(b) TVLA of *LFSR based AES*

(c) SNR of *AES*

(d) SNR of *LFSR based AES*

(e) SR of *AES*

(f) SR of *LFSR based AES*

Fig. 6: Comparing Side Channel Vulnerabilities of unprotected AES and LFSR-based AES

However, in FPGA where the leakage model is non-optimal, the prediction error increases drastically. This is owing to the confusion coefficient term in Eq. (21), which is model dependant, thus leading to high error in prediction. Thus further work on better modelling capabilities will further reduce the prediction error.

6 Conclusion

TVLA based testing methodology is gaining popularity in recent years. Designed as a *PASS/ FAIL* test, 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 testing methodology beyond its current scope. Analytic relationship between *TVLA* and *SNR* is derived for this purpose. The computed *SNR* is used in determining the lower bound of side-channel traces in order to mount an attack at a desired success rate. The methodology is applied on AES in a simulated and practical setting. Predicted number of side-channel traces were compared against actual attack results, reporting some prediction error. Further relaxation of assumptions on the derived formulae is desired to reduce the prediction error. Bridging this gap between the lower bound of trace and actual trace requirement is an interesting research problem.

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. E. DeMulder G. Goodwill J. Jaffe G. Kenworthy T. Kouzminov A. Leiserson M.Marson P. Rohatgi G. Becker, J. Cooper 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.
6. François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. *IACR Cryptology ePrint Archive*, 2015:536, 2015.
7. Konstantinos Chatzikokolakis, Tom Chothia, and Apratim Guha. Statistical measurement of information leakage. In *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, pages 390–404, 2010.
8. T. Chothia and A. Guha. A statistical test for information leaks using continuous mutual information. In *2011 IEEE 24th Computer Security Foundations Symposium*, pages 177–190, June 2011.
9. Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99, 2016.

10. Rohatgi P. Jaffe J. and Witteman M. Efficient side-channel testing for public key algorithms:RSA case study. `http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/09_Jaffe.pdf`, 2011.
11. Michael Tunstall and Gilbert Goodwill. Applying TVLA to public key cryptographic algorithms. *IACR Cryptology ePrint Archive*, 2016:513, 2016.
12. 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.
13. A. Adam Ding, Cong Chen, and Thomas Eisenbarth. Simpler, faster, and more robust t-test based leakage detection. In *Constructive Side-Channel Analysis and Secure Design - 7th International Workshop, COSADE 2016, Graz, Austria, April 14-15, 2016, Revised Selected Papers*, pages 163–183, 2016.
14. 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.
15. François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected CMOS devices. In *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, pages 253–267, 2008.
16. Oswald Elisabeth Popp Thomas Mangard, Stefan. *POWER ANALYSIS ATTACKS*. Springer, 2007.
17. Adrian Thillard, Emmanuel Prouff, and Thomas Roche. Success through confidence: Evaluating the effectiveness of a side-channel attack. *IACR Cryptology ePrint Archive*, 2015:402, 2015.
18. Suvadeep Hajra and Debdeep Mukhopadhyay. Reaching the limit of nonprofiling DPA. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(6):915–927, 2015.
19. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks will fall off - higher-order optimal distinguishers. *IACR Cryptology ePrint Archive*, 2015:452, 2015.
20. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 13–28, 2002.
21. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, pages 15–29, 2006.
22. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, pages 1–14, 2006.
23. Suvadeep Hajra and Debdeep Mukhopadhyay. Reaching the limit of nonprofiling DPA. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(6):915–927, 2015.
24. 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 Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, pages 22–41, 2015.

25. Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. Side-channel leakage and trace compression using normalized inter-class variance. *IACR Cryptology ePrint Archive*, 2014:1020, 2014.
26. 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.
27. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
28. SASEBO-GII. `satoh.cs.uec.ac.jp/SAKURA/hardware/SASEBO-GII.html`. Accessed: 2016-09-25.

Algorithm 2: Computing Number of Side Channel Traces for a Given SR and σ

```
Input: Side channel traces and corresponding cipher text, $SR$, $\sigma$
Output: $m_{SR}^{actual}$
1. Initialize SR=zeros(Number of Traces,1);
for $iteration = 0$ to $I$ do
    2. Generate Gaussian Noise distribution ($\mathcal{N}(0, \sigma)$)
    3. Generate simulated trace $Y = HW(x) + \mathcal{N}$ by adding the Gaussian noise $\mathcal{N}$
    for $run = 1$ to Number of traces do
        4. $Key_{CPA}$=CPA(T(1:run),cipher-text)
        5. if ($Key_{CPA}$==Correct Key) then
            SR(run)+=1
    for i=1 to Number of Traces do
        if ($SR(i) \geq I\times SR$) then
            $m_{SR}^{actual} = i$
            break
    Return $m_{SR}^{actual}$
```

A Computation of Success Rate from CPA