NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage*

Shivam Bhasin[1] Jean-Luc Danger[1,2] Sylvain Guilley[1,2] Zakaria Najm[1]

[1] Institut MINES-TELECOM, TELECOM ParisTech,
Department COMELEC, 46 rue Barrault,
75 634 PARIS Cedex 13, FRANCE.
[2] Secure-IC S.A.S., 80 avenue des Buttes de Coësmes,
35 700 Rennes, FRANCE.

{bhasin,danger,guilley,znajm}@telecom-paristech.fr

Abstract. Side-Channel Attacks (SCA) are considered a serious threat against embedded cryptography. Therefore security-critical chips must be tested for SCA resistance before deployment or certification. SCA are powerful but can need a lot of computation power, especially in the presence of countermeasures. The computation complexity of these attacks can be reduced by selecting a small subset of points where leakage prevails. In this paper, we propose a method to detect relevant leakage points in side-channel traces. The method is based on Normalized Inter-Class Variance (NICV). A key advantage of NICV over state-of-the-art is that NICV does neither need a clone device nor the knowledge of secret parameters of the crypto-system. NICV has a low computation requirement and it detects leakage using public information like input plaintexts or output ciphertexts only. It can also be used to test the efficiency of leakage models, the quality of traces and robustness of countermeasures. A theoretical rationale of NICV with practical application on real crypto-systems are provided to support our claims.

Keywords: Cryptography, side-channel analysis, leakage detection, ANOVA, NICV, AES, RSA.

1 Introduction

Security-critical devices must undergo a certification process before being launched into the public market. One of the many security threats tested in the certification process is Side-Channel Attacks (SCA [1,2]). SCA pose a serious practical threat to physical implementation of secure devices by exploiting unintentional leakage from a device like the power consumption, electromagnetic emanation or timing. Several certification/evaluation labs are running SCA daily on devices under test to verify their robustness.

* Some ideas contained in this paper have also been presented orally at the first International Cryptographic Module Conference (ICMC 2013), Sept. 24–26, 2013, in Gaithersburg area (MD, USA).

The certification process is expensive and very time-consuming which also increases the overall time-to-market for the device under test. It worsens when the desired security level increases. For instance, it is usually considered that a Common Criteria (CC [3]) evaluation at highest assurance level for penetration attacks (AVA.VLAN.5) requires the device to resist attacks with 1 million traces. Similarly, the draft ISO standard 17,825 [4] (extension of FIPS 140-2) demands resistance against side-channel analysis with 10,000 traces (level 3) and with 100,000 traces (level 4). The traces can have millions of points and thus running SCA on these traces can be really time consuming. Also several attacks must be tested on the same set of traces before certifying a device. To accelerate the evaluation process, a methodology should be deployed which compress the enormous traces to a small set of relevant points.

Related Works The compression of SCA traces which results in reduced time complexity of the attacks, can be achieved by selecting a small subset of points where leakage prevails. This issue of selecting relevant time samples have been dealt previously by some researchers. Chari et al. [2] use templates to spot interesting time samples. The method involves building templates T on n different values of the subkey. Interesting time samples can then be found as points which maximizes $\sum_{i,j=1}^{n}(T_i - T_j)$. In this equation, T_i is the average of the traces when the sensitive variable belongs to the class i. Two further improvements were then proposed by Gierlichs et al. [5]. The first improvement, also called as Sum Of Squared pairwise Differences (SOSD), simply computes $\sum_{i,j=1}^{n}(T_i - T_j)^2$ for $i \geq j$. SOSD avoids cancellation of positive and negative differences. SOSD can be further improved by normalizing it by some variance. This normalized SOSD is called SOST (Sum Of Squared pairwise T-differences [5], where a *T-difference* means a *Student T-test*) and computed as

$$\sum_{i,j=1}^{n} \left(\frac{T_i - T_j}{\sqrt{\frac{{\sigma_i}^2}{m_i} + \frac{{\sigma_j}^2}{m_j}}} \right)^2 ,$$

where σ_i is the variance of T in class i, and m_i is the number of samples in class i. If m is the total number of traces, we have $\sum_{i=1}^{n} m_i = m$, and m_i is also m times the estimated probability for the traces to belong to class i. When the classes are equally populated (i.e., $\forall i,\ m_i = m/n$), the SOST rewrites as:

$$\frac{m}{n} \sum_{i,j=1}^{n} \frac{(T_i - T_j)^2}{{\sigma_i}^2 + {\sigma_j}^2} ,$$

A practical problem with template-based detection techniques comes from the computation of templates. First of all, templates require an access to a clone device. Secondly, templates need two sets of traces: one for profiling with random keys and another for attacking with an unknown but fixed key. An alternative to the latter

limitation is model-based templates which can exploit the same set of traces as proposed in [6]. Although model-based templates can be really efficient, they are relevant for the chosen power model only.

Another method proposed in this context is the Principal Components Analysis (PCA). PCA is used for dimensionality reduction. It yields a new basis of the time samples in which the inter-class variance is greater. This basis takes into account the covariance of the samples. In side-channel analysis, the goal of PCA is to gather all the information in a single (or few) component(s) [7]. Eventually, other empirical methods use chosen plaintext attacks, such as the differences between plaintexts `0x00...0000` and `0x00...00ff`. This technique requires many requests to check for all the bytes, not only the least significant byte. Furthermore, it is not always possible to choose the plaintext messages (*e.g.* when modes of operations with initial vectors are used).

In this paper, we propose a new method relying on a metric called "Normalized Inter-Class Variance" (NICV). This NICV method allows to detect interesting time samples, without the need of a profiling stage on a clone device. Hence the SCA traces can be compressed and the analysis could be greatly accelerated. The main characteristics of the proposed method are:

- NICV operates without the need of a clone device, *i.e.* it requires no profiling stage and use the same set of traces which are to be analyzed,
- it uses only public information like plaintext or ciphertext,
- the method is leakage model agnostic, it is not an analysis tool but a helper to speed up the analysis, but
- it can serve to evaluate the accuracy of various leakage models and choose which is the best applicable.

Compared to PCA, the purpose of NICV is to return the total variation of the traces at each time sample, so as to test which leakage model causes inter-class variation.

The rest of the paper is organized as follows. General background to SCA is recalled in Sec. 2. The rationale of NICV to select SCA relevant time samples is detailed in Sec. 3. We also derive in this section a lower bound on the number of traces to recover the key. This is followed by some practical use cases applied on real devices like FPGA and smartcards in Sec. 4. Finally, Sec. 5 draws general conclusions.

2 General Background

Side-channel analysis consists in exploiting dependencies between the manipulated data and the analog quantities (power consumption, electromagnetic radiation, ...) leaked from a CMOS circuit. Suppose that several power consumption traces, denoted Y, are recorded while a cryptographic device is performing an encryption or decryption operation. An attacker predicts the intermediate leakage $L(X)$, for

a known part of the ciphertext (or plaintext) X and key hypothesis K. Next, the attacker uses a distinguisher like Correlation Power Analysis (CPA [1]), to distinguish the correct key $k^\star$ from other false key hypotheses. CPA is a computation of the *Pearson Correlation Coefficient* ρ between the predicted leakage $L(X)$ and the measured leakage Y, which is defined as:

$$\text{CPA} : \rho\left[L(X); Y\right] = \frac{\mathbb{E}\left[\left(L(X) - \mathbb{E}\left[L(X)\right]\right) \cdot \left(Y - \mathbb{E}\left[Y\right]\right)\right]}{\sqrt{\mathsf{Var}\left[L(X)\right] \cdot \mathsf{Var}\left[Y\right]}} \in [-1; +1] \ ,$$

where $\mathbb{E}$ and Var denote the mean and the variance respectively.

Various distinguishers have been proposed in literature. In [8], authors show that all statistical distinguishers eventually turn out to be equivalent when the signal-to-noise ratio gets high. The differences observed by an attacker are due to statistical artifact which arises from imprecise estimations due to limited numbers of observations. In the rest of the paper without loss of generality, we use CPA as a distinguisher.

Authors of [8] also show that a proper estimation of leakage model $L(X)$ can define the efficiency of the attack. Therefore a detection technique is needed which can detect the relevant leakage points and the most efficient leakage model. In the following, we introduce NICV as a leakage detection technique and its power to evaluate estimated leakage models. As shown later, NICV is not a SCA channel distinguisher itself. NICV works in co-ordination with any SCA distinguishers like CPA to enhance their performance. Even variance-based distinguishers as introduced in [9,10,11] can be made efficient using NICV. The metrics and distinguishers based on *Linear Discriminant Analysis* (LDA) [12] or on *Principal Component Analysis* (PCA) [13,12,14] are also using some kind of L^2 distance between classes (as does the NICV, see next section).

3 Leakage Detection using NICV

In this section, we first describe our *normalized inter-class variance* (NICV) detection technique. We provide the mathematical background of NICV and then discuss its behavior in a side-channel context.

3.1 Rationale of the NICV Detection Technique

Let us call X one byte of the plaintext or of the ciphertext (that is, the domain of X is $\mathcal{X} = \mathbb{F}_2^8$), and $Y \in \mathbb{R}$ the leakage measured by the attacker[1]. Both random variables are public knowledge. Then, for all leakage prediction function L of the leakage knowing the value of x taken by X (as per Proposition 5 in [15]), we have:

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

[1] In general, Y can be continuous, but X must be discrete (and $\mathcal{X}$ must be of finite cardinality).

Again in Corollary 8 of [15], the authors derive:

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

which we refer to as the *normalized inter-class variance* (NICV). It is an ANOVA (ANalysis Of VAriance) F-test, as a ratio between the explained variance and the total variance (see also the recent article [16]).

Once combined, equations (1) and (2) yield that for all prediction function $L : \mathbb{F}_2^8 \to \mathbb{R}$, we have:

$$0 \leq \rho^2 \left[L(X); Y \right] \leq \frac{\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]}{\mathsf{Var}\left[Y\right]} = \text{NICV} \leq 1 \ . \tag{3}$$

Therefore, the NICV is the *envelop* or maximum of all possible correlations computable from X with Y. There is an equality in (3) if and only if $L(x) = \mathbb{E}\left[Y|X = x\right]$, which is the optimal prediction function[2].

In practice, the (square) CPA value does not attain the NICV value, owing to noise and other imperfections. This is illustrated in Fig. 1. The difference can come from various reasons like:

- The attacker knows the exact prediction function, but as usual not the actual key. For instance, let us assume the traces can be written as $Y = w_H(S(X \oplus k^\star)) + N$, where $k^\star \in \mathbb{F}_2^8$ is the correct key, $S : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ is a substitution box, w_H is the Hamming weight function, and N is some measurement noise, that typically follows a centered normal distribution $N \sim \mathcal{N}(0, \sigma^2)$. In this case, the optimal prediction function $L(x) = \mathbb{E}\left[Y|X = x\right]$ is equal to: $L(x) = w_H(S(X \oplus k^\star))$ (the only hypothesis on the noise is that it is *centered* and *mixed additively* with the sensitive variable). This argument is at the base of the soundness of CPA: $\forall k \neq k^\star, \rho\left[w_H(S(X \oplus k)); Y\right] \leq \rho\left[w_H(S(X \oplus k^\star)); Y\right] \leq \sqrt{\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right] / \mathsf{Var}\left[Y\right]}$.
- The CPA is smaller than NICV when the attacker assumes a wrong model, for instance $L(x) = w_H(x \oplus k^\star)$, when $Y = w_H(S(x \oplus k^\star)) + N$.
- Eventually, the attacker can have an approximation of the leakage model, for instance $L(x) = w_H(S(x \oplus k^\star))$, whereas actually $Y = \sum_{i=1}^{8} \beta_i \cdot S_i(x \oplus k^\star) + N$, where $\beta_i \approx 1$, but slightly deviate from one.

The distance between CPA and NICV is, in non-information theoretic attacks (*i.e.* attacks in the proportional / ordinal scale, as opposed to the nominal scale [17]) is similar to the distance between perceived information (PI) and mutual information (MI) [18].

[2] Rigorously: if and only if $L(x)$ is an affine function of $\mathbb{E}\left[Y|X = x\right]$.

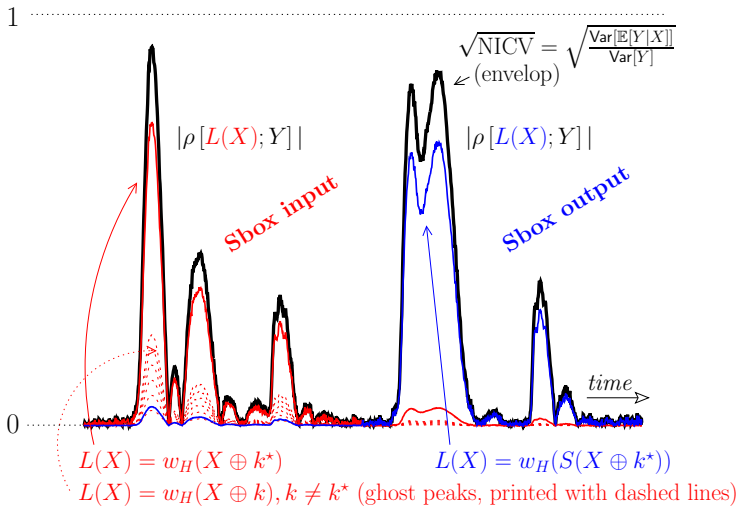


Fig. 1. NICV metric when $Y = \sum_{i=1}^{8} \beta_i \cdot S_i(x \oplus k^\star) + N$, and attack results for some prediction functions

Now, if X is uniformly distributed, the NICV in itself *is not* a distinguisher. Indeed, if we assume that $Y = w_H(S(X \oplus k^\star)) + N$, then:

$$\begin{aligned}
\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right] &= \sum_{x \in \mathcal{X}} \mathsf{P}[X = x]\mathbb{E}\left[Y|X = x\right]^2 - \mathbb{E}\left[Y\right]^2 \\
&= \frac{1}{2^8} \sum_{x \in \mathcal{X}} \mathbb{E}\left[w_H(S(x \oplus k^\star)) + N\right]^2 - \left(\sum_{x \in \mathcal{X}} \mathbb{E}\left[w_H(S(x \oplus k^\star)) + N\right]\right)^2 \\
&= \frac{1}{2^8} \sum_{x' = x \oplus k^\star \in \mathcal{X}} \mathbb{E}\left[w_H(S(x'))\right]^2 - \left(\sum_{x' = x \oplus k^\star \in \mathcal{X}} \mathbb{E}\left[w_H(S(x'))\right]\right)^2 \\
&= \mathsf{Var}\left[w_H(S(X))\right] \quad \text{on the one hand, and} \\
\mathsf{Var}\left[Y\right] &= \mathsf{Var}\left[w_H(S(X))\right] + \mathsf{Var}\left[N\right] \quad \text{on the other hand, because } X \perp\!\!\!\perp N.
\end{aligned}$$

All in one:

$$\text{NICV} = \frac{\mathsf{Var}\left[\mathbb{E}\left[Y|X\right]\right]}{\mathsf{Var}\left[Y\right]} = \frac{1}{1 + \frac{1}{\text{SNR}}} \ , \tag{4}$$

where the signal-to-noise ratio SNR is the ratio between:

- the signal, *i.e.* the variance of the informative part, namely $\mathsf{Var}\left[w_H(S(X \oplus k^\star))\right]$, and
- the noise, considered as the variance $\mathsf{Var}\left[N\right]$.

Clearly, Eqn. (4) does not depend on the secret key $k^\star$ as both Y and X are public parameters known to the attacker.

Remark 1. In the binary case ($n = 2$) when both classes are equally probable, the expression of NICV simplifies to: $\mathrm{NICV} = \frac{\mathsf{Var}[\mathbb{E}[Y|X]]}{\mathsf{Var}[Y]} = \frac{\left(\frac{\mathbb{E}[Y|X=0]-\mathbb{E}[Y|X=1]}{2}\right)^2}{\mathsf{Var}[Y]}$.

It is similar to Cohen's d metric.

Now comparing NICV with other leakage detection techniques like SOST and SOSD we can give the following remarks.

Remark 2. SOSD is actually proportional to the inter-class variance (this point *was not* made by the authors of [5]). Indeed, with our notations, $T_i \doteq \mathbb{E}[Y|X = i]$. And thus:

$$\sum_{i,j}(T_i - T_j)^2 = 2 \times 2^n \sum_i T_i^2 - 2\left(\sum_i T_i\right)^2 =$$
$$2 \times 2^{2n} \sum_x \mathbb{E}\left[Y^2|X = x\right] \mathsf{P}[x] - 2\left(2^8 \mathbb{E}[Y]\right)^2 = 2^{2n+1}\mathsf{Var}\left[\mathbb{E}[Y|X]\right] \ .$$

But this inter-class variance *is not* normalized. Therefore, the SOSD can be large at samples where $\mathsf{Var}[N]$ is large, although not containing (much) information.

Remark 3. SOST which was proposed as an improvement over SOSD is normalized. Using our notations, SOST is equal to

$$m \times \sum_{(x,x')\in\mathcal{X}^2} \frac{\left(\mathbb{E}[Y|X = x] - \mathbb{E}[Y|X = x']\right)^2}{\sqrt{\mathsf{Var}\left[\mathbb{E}[Y|X = x]\right]/\mathsf{P}[X = x] + \mathsf{Var}\left[\mathbb{E}[Y|X = x']/\mathsf{P}[X = x']\right]}} \ ,$$

where m is the number of traces. This certainly is an expression that is not usual in statistics, and *a priori* cannot be simplified.

3.2 Lower Bound on the Number of Traces to Break the Key

The success rate ($0 \leq \mathsf{SR} \leq 1$) of a CPA has been computed theoretically by Thillard, Prouff and Roche [19] (a result that extends the previous analytical formula obtained for the "difference-of-means" test by Fei, Luo and Ding [20]). It is related to the various factors, namely:

- the signal-to-noise ratio (which is also directly related to the NICV metric),
- the sensitive variable expression, which discriminates more or less easily the correct key (which relates to an algorithmic parameter known as the confusion coefficient),
- the number of traces m.

The expression takes the following form:

$$\mathsf{SR} = \Phi(\sqrt{m}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}) \ .$$

In this expression, Φ is the cumulative distribution function of the multivariate Gaussian. Let N_k be the number of key hypotheses. Then $\boldsymbol{\Sigma}$ is a $(N_k - 1) \times (N_k - 1)$ matrix, and $\boldsymbol{\mu}$ is column of length $(N_k - 1)$. In the worst case, the leakage model is very discriminating. This means that the wrong key guesses are all equivalent and yield a null value for the distinguisher. It is a strong assumption, but for good ciphers, such as the AES, this pessimistic approximation is not too far from the reality. In this case, $\boldsymbol{\Sigma}$ degenerates to a 1×1 matrix (a scalar), equal to: $\boldsymbol{\Sigma} = 2\kappa_0/\sigma^2$. In this equation, κ_0 is "generalized" confusion coefficient, equal to $\frac{1}{2^n}(L \otimes L)(0)$, where $\otimes$ is the convolution function. Also, $\boldsymbol{\mu}$ is a scalar, equal to κ_0. So, we have:

$$\mathsf{SR} = \Phi\left(\sqrt{m \times \frac{1}{2}\frac{\kappa_0}{\sigma^2}}\right) \quad . \tag{5}$$

The ratio $\sqrt{\frac{\kappa_0}{\sigma^2}}$ can be seen as the square of a signal-to-noise ratio.

So, for a given confidence level in the result of the attack, expressed as targetted success rate SR, we have that the lower bound on the number of traces to break the key is m_{SR}, equal to:

$$m_{\mathsf{SR}} = 2\frac{\sigma^2}{\kappa_0} \times \left(\Phi^{-1}\left(\mathsf{SR}\right)\right)^2 \quad , \tag{6}$$

where Φ^{-1} is the reciprocal function of Φ (recall that Φ is monotonic decreasing).

So, any CPA will require more traces to recover the key than m_{SR}.

3.3 Discussion

The mathematical background of NICV as a leakage detection technique was previously discussed. We learned that NICV has evident advantages over other methods because all its input parameters are public like side-channel traces and associated plaintexts/ciphertexts. Since public parameters are used for computation of NICV, there is no need for access to clone device which is a limiting requirement in template-based detection techniques. Another interesting observation is that the expression of NICV (Eq. (1)) does not contain $L(x)$. In other words, NICV is leakage model agnostic. Moreover from Eq. (3), we learn that NICV forms the envelope of all correlation coefficients for all leakage models. NICV provides the worst case leakage of a devise and therefore estimates the accuracy of leakage model used as illustrated in Fig. 1. Thus NICV has a clear application in comparing various leakage models.

4 Use Cases

We detailed the theoretical soundness and advantages of NICV as a leakage detection technique in Sec. 3. In this section, we apply NICV in practical side-channel evaluation scenarios. Several use cases of NICV are discussed in the following.

4.1 Accelerating Side-Channel Attacks

The main application of NICV is to find the interesting time samples for accelerating SCA. A simple trace of an AES execution can have millions of points. Therefore it is of interest for the evaluator to know few interesting points rather than attacking the whole trace. We first apply the metric on traces of an AES-128 implementation running on an FPGA which performs one round per clock cycle. These traces are small and contain only $1,000$ points. The comparison of our metric with a correlation coefficient computed with the good key is shown in Fig. 2. The correlation is based on Hamming distance model of the state register of the AES core. The model can be expressed as $w_H(val_i \oplus val_f)$ where val_i and val_f are initial and final value of the register. This leakage model is shown to be very efficient in CMOS technology. A relevant peak of NICV is seen at the same moment as in correlation peak. Thus NICV is able to detect the point of leakage in the trace. It can be noticed that NICV curve shows several other peaks apart from the correlation peak. As shown later in Sec. 4.2, other peaks in the NICV curve comes either from other leakage models or post-processing of cipher in the circuit.

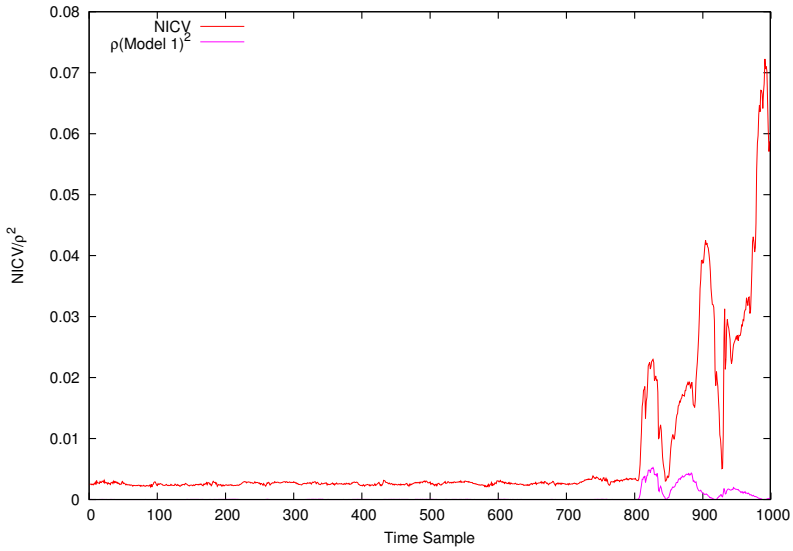


Fig. 2. NICV vs Correlation for a AES-128 hardware implementation

Next we apply our metric on a software implementation of AES-256 running on an ATMEL AVR microcontroller. It is here that we can see the advantage of NICV. A single trace of this implementation contains 7 million points and needs roughly 5.3 Mbytes of disk space when stored in the most compressed format. These details are in respect to a LeCroy wavescanner 6100A oscilloscope with a bandwidth of

1 GHz. We applied NICV on these traces to find the leakage points related to each of the 16 bytes of the AES. Fig. 3 shows the computation of NICV on the first round only (for better resolution of results). The computations of Sbox0 for round 1 takes only ≈ 1000 time samples. Once the interesting time samples corresponding to each executed operation is known, the trace size is compressed from 7000000 to 1000, i.e. a gain of roughly $7000\times$.

One very interesting application of NICV that we found during our experiments is to reverse engineering. We computed NICV for all the 16 bytes of the plaintext and plotted the 16 NICV curves in Fig. 3 (depicted in different colors). By closely observing Fig. 3, we can distinguish individual operations from the sequence of byte execution. Each NICV curve (each color) shows all sensitive leakages related to that particular byte. Moreover, with a little knowledge of the algorithm, one can easily follow the execution of the algorithm. For example, the execution of all the bytes in a particular sequence indicates the SubBytes or AddRoundKey operation. Manipulation of bytes in sequence $\{1, 5, 9, 13\}$, $\{2, 6, 10, 14\}$ and $\{3, 7, 11, 15\}$ indicates the ShiftRows operations. The ShiftRows operation of AES shifts circularly 3 out of 4 rows with different constant. This can be clearly seen is Fig. 3: only three rows are manipulated and the bytes in the first row i.e. $\{0, 4, 8, 12\}$ are not used during this time. Similarly MixColumns can also be identified by just looking the bytes manipulated together. Moreover, detecting precise leakage points of each operation can help an attacker run collision attacks.

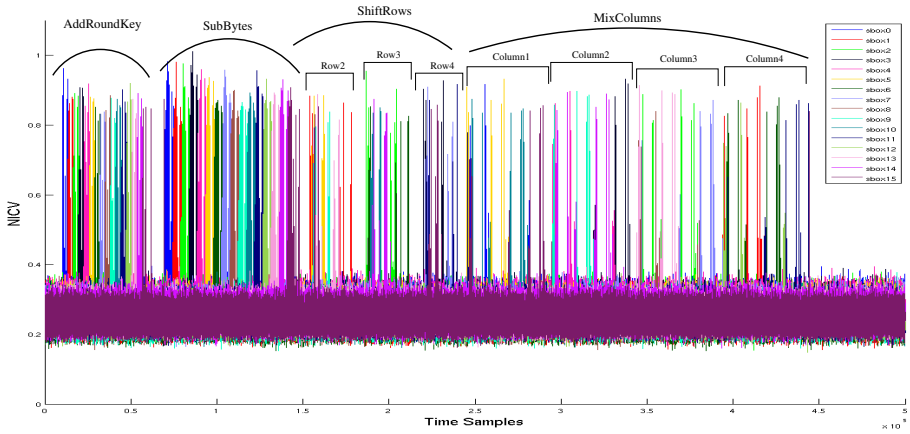


Fig. 3. NICV computed for a AES-128 software implementation to detect each round operation.

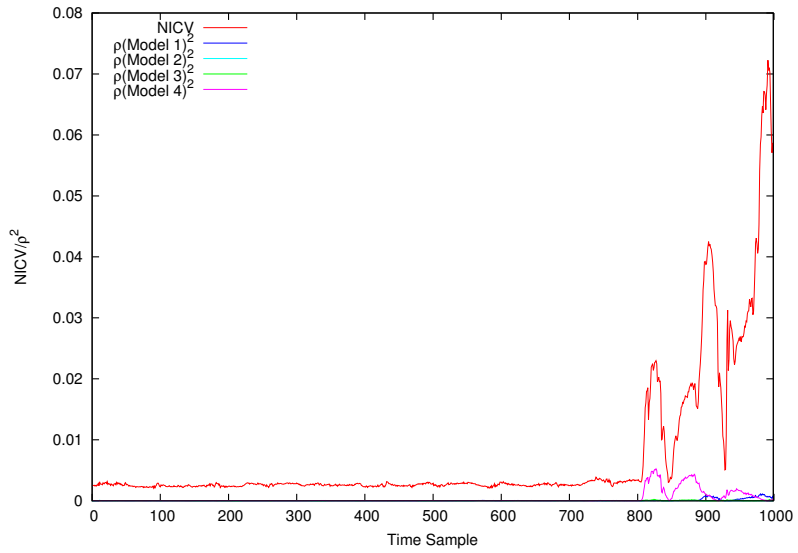
4.2 Testing Leakage Models

A common problem in SCA is the choice of leakage model which directly affects the efficiency of the attack. As shown in Sec. 3.1, the square of the correlation between modeled leakage ($L(X, K)$) and traces ($Y = L(X, K^\star) + N$) is smaller or equal to NICV, where N represents a noise. The equality exists only if the modeled leakage is the same as the traces. We tested two different leakage models for the state register resent before the Sbox operation of AES *i.e.* $w_H(val_i \oplus val_f) \in [\![0, 8]\!]$ (Model 1) and $val_i \oplus val_f \in [\![0, 255]\!]$ (Model 2). w_H is the Hamming weight function. Similar models are built for another register which is intentionally introduced at the output of the Sbox *i.e.* $w_H(S(val_i) \oplus S(val_f)) \in [\![0, 8]\!]$ (Model 3) and $S(val_i) \oplus S(val_f) \in [\![0, 255]\!]$ (Model 4). We implemented the AES on an FPGA and acquired SCA traces to compare the leakage models. Fig. 4 shows the square of correlation of four different leakage models with the traces against the NICV curve. It can be simply inferred from Fig. 4(b) that Model 4 performs the best while Model 2 is the worst. The gap between NICV and $\rho(\text{Model 4})^2$ is quite large due to reasons mentioned in Sec 3.1. This means that there exist other leakage models which could perform better than Model 4. However, finding these models might not be easy because of limited knowledge of design and device characteristics available. Methods based on linear regression [21] can be leveraged to determine the most relevant leakage model.

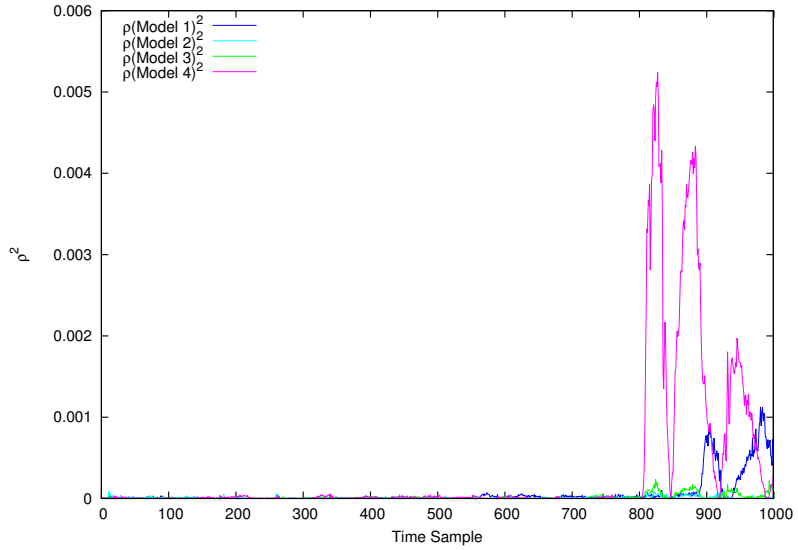
4.3 Testing Countermeasure Implementation

An application where NICV can also come handy to a designer is in evaluation of SCA countermeasures. Several countermeasures can be deployed to make SCA harder or impossible. However, if the countermeasure is badly implemented or not balanced, simple SCA like CPA can extract the key. One can also use NICV in this context to make the analysis faster. Unlike CPA, NICV does not need computation of intermediate values or repeating the attack for all key hypothesis. Application of NICV on a protected implementation will detect any linear leakage if the countermeasure is badly implemented. We test NICV on a Dual-rail Precharged Logic (DPL [22]) countermeasure applied on AES-128 hardware implementation. The security of DPL largely depends on the amount of imbalance in routing of individual wires. The curves in Fig. 5 represent two different bytes of the AES, one which is properly routed and the other badly. NICV clearly distinguishes the badly routed byte of the AES, giving a feedback to the designer about the point of vulnerability. Fig. 5(b) has two NICV peaks, one w.r.t correlation and the other due to post-processing of cipher. On the other hand, Fig. 5(a) contains a unique NICV peak due to post-processing of cipher. We know that the second peak is not related to the secret key because of the extremely low correlation value at time samples (800—1, 000).

Even from an evaluator's viewpoint, NICV can also help analyzing protected implementations. For example, in the case of masked implementation, an evaluator has knowledge of ciphertext Z and mask M. This allows the evaluator to compute

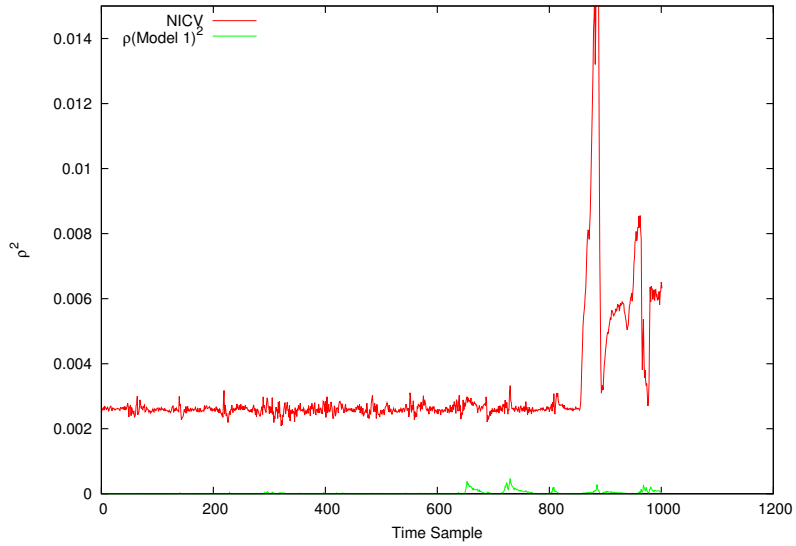


(a)

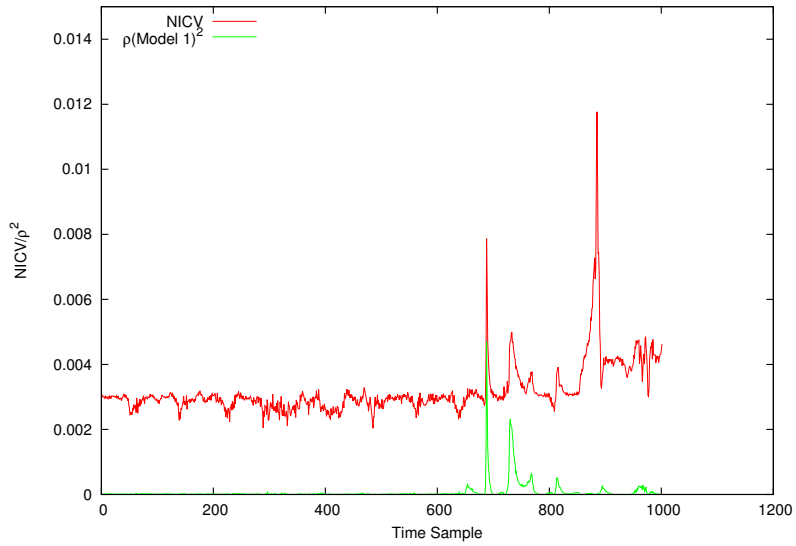


(b)

Fig. 4. (a) NICV vs ρ^2 of four different models, (b) and its zoom



(a)



(b)

Fig. 5. NICV vs Correlation for (a) well, (b) badly protected bytes of a DPL implementation

NICV as $\frac{\mathsf{Var}[\mathbb{E}[Y|(Z \oplus M)]]}{\mathsf{Var}[\mathbb{E}[Y]]}$ and detect points where masked data is computed. A direct application NICV with mask is collision attacks.

4.4 Comparing Quality of Measurements

SNR is often used to estimate the quality of a measurement setup/traces to compare different measurement setups. The problem with SNR is that it is computed using a specific leakage model. NICV is a good candidate for quality comparison owing to the independence from choice of leakage model.

4.5 Accelerating SCA on Asymmetric Key Cryptography

Asymmetric key cryptography consists in computing exponentiations. For example, in RSA [23], the computation consists in X^d (modulo N) from X. For the sake of simplicity, let us consider a right-to-left exponentiation. Such exponentiation is illustrated in Alg. 1, where N is the modulus (e.g. that fits on 1024 bits), and $R[1]$ and $R[2]$ are two 1024 bit temporary registers. Let us call d_i the 1024 bits of d. We assume $d_0 = 1$.

Algorithm 1: Unprotected right-to-left 1024 bit RSA implementation

Input : $X \in \mathbb{Z}_N, d = (d_{1023}, \cdots, d_0)_2$
Output: $X^d \in \mathbb{Z}_N$

1 $R[1] \leftarrow 1$
2 $R[2] \leftarrow X$
3 **for** $i \in [\![0, 1023]\!]$ **do**
4 **if** $d_i = 1$ **then**
5 $R[1] \leftarrow R[2] \cdot R[1]$ /* Multiply */
6 **end**
7 $R[2] \leftarrow R[2] \cdot R[2]$ /* Square */
8 **end**
9 **return** $R[1]$

Hence the number X^3 will be computed (in $R[1]$; refer to line 5) if and only if $d_1 = 1$. This conditional operation is at the basis of the SCA on RSA [24]: if a correlation between the traces Y and the prediction $L(X) = X^3$ exists, then $d_1 = 1$; otherwise, $d_1 = 0$. For this alternative to be tested with NICV, one should compute $Y|X^3$, where X^3 (modulo N) is a large number (*e.g.* 1,024 bits). To be tractable, small parts of X^3 like the least significant byte (LSB) shall be used instead of X^3. In this case, a leakage can be detected by computing $\mathsf{Var}\left[\mathbb{E}\left[Y|\mathrm{LSB}(X^3)\right]\right] / \mathsf{Var}\left[Y\right]$. The corresponding attack would use the prediction function $L(X) = \mathrm{LSB}(X^3)$.

For sure, the test is relevant only if the bit d_1 is set in the private key d. But if it is not, then maybe d_2 is set. In this case, a leakage can be detected by computing

$\mathsf{Var}\left[\mathbb{E}\left[Y|\mathrm{LSB}(X^5)\right]\right]/\mathsf{Var}\left[Y\right]$. Similarly, if $d_1 = d_2 = 0$, it is plausible that $d_3 = 1$, and thus X^9 is computed. Thus, it is sufficient, in order to detect a leakage to compute $\mathsf{Var}\left[\mathbb{E}\left[Y|\mathrm{LSB}(X^{2^i+1})\right]\right]/\mathsf{Var}\left[Y\right]$ for a couple of small $i > 0$. Any significant peak indicates a potential vulnerability.

If, for example, the 10 NICV quantities $\mathsf{Var}\left[\mathbb{E}\left[Y|\mathrm{LSB}(X^{2^i+1})\right]\right]/\mathsf{Var}\left[Y\right]$, for $1 \leq i \leq 10$, are computed (without knowing the key d), then a vulnerability is detected with probability $1 - 2^{-10}$ (indeed, 2^{-10} is the probability of having $d_1 = \ldots = d_{10} = 0$). This methodology is illustrated in Fig. 6.

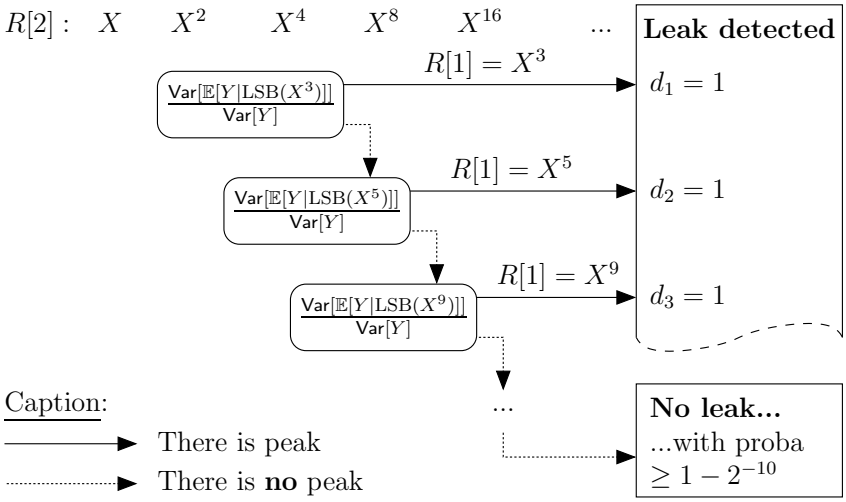


Fig. 6. Illustration of the application of NICV to RSA

5 Conclusions and Perspectives

We presented NICV as a leakage detection technique for side-channel leakage. NICV uses public information like plaintext or ciphertext for detecting leakage and therefore has a low computation footprint. It can be seen as the worst case leakage analysis which *envelops* correlation coefficient of all possible leakage models. However NICV cannot be used directly as a distinguisher for an attack. Unlike templates, NICV can operate on the same set of traces which are used for attack. We demonstrated the power of NICV in several use cases related to SCA like detecting relevant time samples, comparing leakage models, testing countermeasures etc. NICV can also be used in context of accelerating SCA on asymmetric cryptography.

Future works can focus on extending the power of NICV in detecting higher-order leakage and extensive application to asymmetric key cryptography.

References

1. Brier, É., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: CHES. Volume 3156 of LNCS., Springer (2004) 16–29 Cambridge, MA, USA.
2. Chari, S., Rao, J.R., Rohatgi, P.: Template Attacks. In: CHES. Volume 2523 of LNCS., Springer (2002) 13–28 San Francisco Bay (Redwood City), USA.
3. Consortium, C.C.: Common Criteria (*aka* CC) for Information Technology Security Evaluation (ISO/IEC 15408) (2013)
   Website: http://www.commoncriteriaportal.org/.
4. Easter, R.J.: Text for ISO/IEC 1st WD 17825 – Information technology – Security techniques – Non-invasive attack mitigation test metrics for cryptographic modules (2012) Prepared within ISO/IEC JTC 1/SC 27/WG 3. (Online).
5. Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. Stochastic Methods. In: CHES. Volume 4249 of LNCS., Springer (2006) 15–29 Yokohama, Japan.
6. Aabid, M.A.E., Guilley, S., Hoogvorst, P.: Template Attacks with a Power Model. Cryptology ePrint Archive, Report 2007/443 (2007) http://eprint.iacr.org/2007/443/.
7. Archambeau, C., Peeters, É., Standaert, F.X., Quisquater, J.J.: Template Attacks in Principal Subspaces. In: CHES. Volume 4249 of LNCS., Springer (2006) 1–14 Yokohama, Japan.
8. Mangard, S., Oswald, E., Standaert, F.X.: One for All - All for One: Unifying Standard DPA Attacks. Information Security, IET **5** (2011) 100–111 ISSN: 1751-8709 ; Digital Object Identifier: 10.1049/iet-ifs.2010.0096.
9. Standaert, F.X., Gierlichs, B., Verbauwhede, I.: Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In: ICISC. Volume 5461 of LNCS., Springer (2008) 253–267 Seoul, Korea.
10. Batina, L., Gierlichs, B., Lemke-Rust, K.: Differential Cluster Analysis. In Clavier, C., Gaj, K., eds.: Cryptographic Hardware and Embedded Systems – CHES 2009. Volume 5747 of Lecture Notes in Computer Science., Lausanne, Switzerland, Springer-Verlag (2009) 112–127
11. Moradi, A., Mischke, O., Eisenbarth, T.: Correlation-Enhanced Power Analysis Collision Attack. In: CHES. Volume 6225 of Lecture Notes in Computer Science., Springer (2010) 125–139 Santa Barbara, CA, USA.
12. Karsmakers, P., Gierlichs, B., Pelckmans, K., Cock, K.D., Suykens, J., Preneel, B., Moor, B.D.: Side channel attacks on cryptographic devices as a classification problem. COSIC technical report (2009)
13. Guilley, S., Chaudhuri, S., Sauvage, L., Hoogvorst, P., Pacalet, R., Bertoni, G.M.: Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks. IEEE Transactions on Computers **57** (2008) 1482–1497
14. Souissi, Y., Nassar, M., Guilley, S., Danger, J.L., Flament, F.: First Principal Components Analysis: A New Side Channel Distinguisher. In Rhee, K.H., Nyang, D., eds.: ICISC. Volume 6829 of Lecture Notes in Computer Science., Springer (2010) 407–419
15. Prouff, E., Rivain, M., Bevan, R.: Statistical Analysis of Second Order Differential Power Analysis. IEEE Trans. Computers **58** (2009) 799–811
16. Choudary, O., Kuhn, M.G.: Efficient Template Attacks. Cryptology ePrint Archive, Report 2013/770 (2013) http://eprint.iacr.org/2013/770.
17. Whitnall, C., Oswald, E., Standaert, F.X.: The myth of generic DPA...and the magic of learning. Cryptology ePrint Archive, Report 2012/256 (2012) http://eprint.iacr.org/2012/256.
18. Renauld, M., Standaert, F.X., Veyrat-Charvillon, N., Kamel, D., Flandre, D.: A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In: EUROCRYPT. Volume 6632 of LNCS., Springer (2011) 109–128 Tallinn, Estonia.
19. Thillard, A., Prouff, E., Roche, T.: Success through confidence: Evaluating the effectiveness of a side-channel attack. In Bertoni, G., Coron, J.S., eds.: CHES. Volume 8086 of Lecture Notes in Computer Science., Springer (2013) 21–36

20. Fei, Y., Luo, Q., Ding, A.A.: A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In Prouff, E., Schaumont, P., eds.: CHES. Volume 7428 of LNCS., Springer (2012) 233–250
21. Doget, J., Prouff, E., Rivain, M., Standaert, F.X.: Univariate side channel attacks and leakage modeling. J. Cryptographic Engineering **1** (2011) 123–144
22. Bhasin, S., Guilley, S., Souissi, Y., Graba, T., Danger, J.L.: Efficient Dual-Rail Implementations in FPGA using Block RAMs. In: ReConFig, IEEE Computer Society (2011) 261–267 Cancún, Quintana Roo, México. DOI: 10.1109/ReConFig.2011.32.
23. Rivest, R.L., Shamir, A., Adleman, L.M.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM **21** (1978) 120–126
24. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Power Analysis Attacks of Modular Exponentiation in Smartcards. In Koç, Ç.K., Paar, C., eds.: CHES. Volume 1717 of LNCS., Springer (1999) 144–157