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

Shivam Bhasin*, Jean-Luc Danger*†, Sylvain Guilley*† and Zakaria Najm*
*TELECOM-ParisTech 46 rue Barrault, 75 634 Paris Cedex 13, FRANCE.
†Secure-IC S.A.S. 80 avenue des Buttes de Coësmes, 35 700 Rennes, FRANCE.
Email: firstname.lastname@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. With the advancement of electromagnetic (EM) measurement techniques in side-channel, high quality traces with enhanced resolution (or more points) are used in order to carry out SCA evaluations. 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. It is shown that NICV can be related to Pearson correlation and signal to noise ratio (SNR) which are standard metric in side-channel and EMC community. 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.

I. 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 (EM) emanation or timing. Recently, measurements using EM probes are gaining popularity over power consumption measurement. This is because EM probe can be designed to provide more localized information of the device under test, resulting in a higher SNR.

Several certification/evaluation labs are running SCA daily on devices under test to verify their robustness. The certification process is expensive and very time-consuming which also increases the overall time-to-market for the device under test. A good example can be of hardware targets tested under Common Criteria evaluations. Such devices should undergo all penetration tests (invasive and non-invasive) over a maximum period of 3 months, which leaves very little time for SCA evaluations. 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. When EM probes are used to acquire side channel traces, often high resolution is required to test short-lived unintentional leakages through electromagnetic emanations. This further explodes the size of SCA traces.

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. Interested readers can refer to [5] for an overview on state-of-the-art of leakage detection techniques.

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 is greatly accelerated. The three main properties of NICV are that it uses only public information like plaintext or ciphertext, it is leakage model agnostic and it can operate without an access to a clone device. NICV can also be used to evaluate the accuracy of leakage models and choose the best applicable model.

The rest of the paper is organized as follows. General background to SCA is recalled in Sec. II. The rationale of NICV to select SCA relevant time samples is detailed in Sec. III. This is followed by some practical use cases applied on real devices like FPGA and smartcards spied by EM probes in Sec. IV. Finally, Sec. V draws general conclusions.

II. 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:

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

where $\mathbb{E}$ and Var denote the mean and the variance respectively. ρ is a normalized coefficient whose value always stays in the range $[-1;+1]$.

Various distinguishers have been proposed in literature. In [6], 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 [6] 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 [2], [7] can be made efficient using NICV.

III. Leakage Detection using NICV

A. 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 parameters known to the attacker. Then, for all leakage prediction function L of the leakage knowing the value of x taken by X (as per Proposition 5 in [8]), we have:

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

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

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

which we refer to as the *normalized inter-class variance* (NICV). It is an ANOVA test (ANalysis Of VAriance). Once combined, Eqn. (1) and (2) yield that for all prediction function $L : \mathbb{F}_2^8 \to \mathbb{R}$, we have:

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

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]. NICV can also be directly related to signal-to-noise ratio (SNR [5]) which is a standard metric in EMC community.

$$\text{NICV} = \frac{1}{1+\frac{1}{\text{SNR}}} \ , \quad (4)$$

Thus NICV has evident advantages over other methods because all its input parameters are public and no clone device is needed for learning. NICV provides the worst case leakage of a device and therefore estimates the accuracy of leakage model used as illustrated in Fig. 1. Further properties of NICV and its comparison with other detection techniques are in [5].

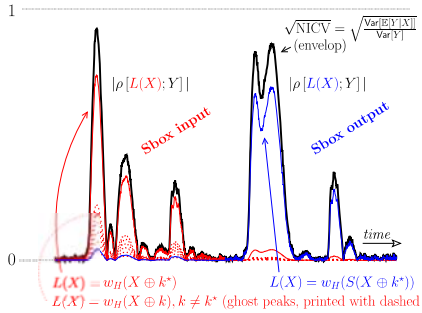


Fig. 1. NICV metric when $Y = \sum_{i=1}^{8} \beta_i \cdot S_i(x \oplus k^\star) + N$, and attack results for some prediction functions

IV. Use Cases

We discussed the theoretical background of NICV as a leakage detection technique in Sec. III. In this section, we apply NICV in practical side-channel evaluation scenarios. All the measurements have been done using a near field EM RFU5-2 probe from Langer EMV. The EM traces are sampled using a LeCroy WaveRunner 6100A oscilloscope for smartcard implementations and 54855 Infiniium Agilent oscilloscope for hardware targets. Several use cases of NICV are discussed in the following.

A. 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 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

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

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

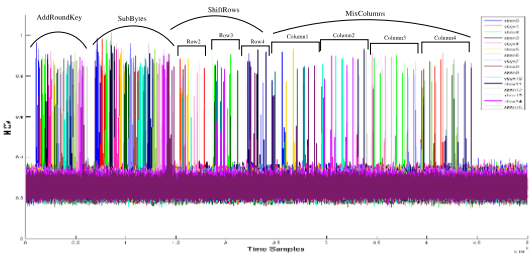


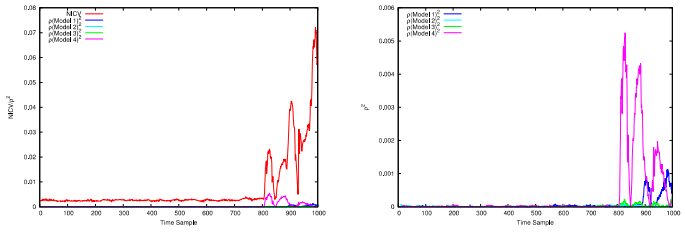
Fig. 2. NICV computed for a AES-128 software implementation to detect each round operation.

when stored in the most compressed format. We applied NICV on these traces to find the leakage points related to each of the 16 bytes of the AES. Fig. 2 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. 2 (depicted in different colors). By closely observing Fig. 2, 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. It can be clearly seen is Fig. 2 that 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.

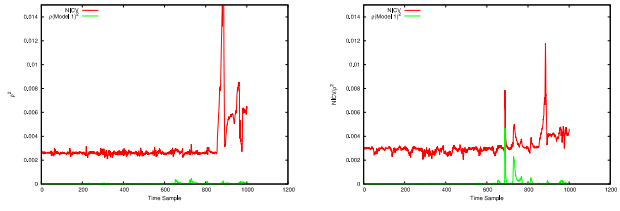
B. Testing Leakage Models

A common problem in SCA is the choice of leakage model which directly affects the efficiency of the attack. As shown in Sec. III-A, 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



(a) (b)

Fig. 3. (a) NICV vs ρ^2 of four different models, (b) and its zoom



(a) (b)

Fig. 4. NICV vs Correlation for (a) well, (b) badly protected bytes of a DPL implementation

the leakage models. Fig. 3 shows the square of correlation of four different leakage models with the traces against the NICV curve. It can be simply inferred from Fig. 3(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 III-A. 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 [9] can be leveraged to determine the most relevant leakage model.

C. Testing Countermeasure Implementation

NICV can be used by designers to test the effectiveness of implemented SCA countermeasures. Application of NICV on a protected implementation will detect any linear (uni-variate) leakage if the countermeasure is badly implemented. We test NICV on a Dual-rail Precharged Logic (DPL [10]) countermeasure applied on AES-128 hardware implementation. The security of DPL largely depends on the amount of imbalance in routing of individual wires. Therefore security offered by two instances of same circuit can offer varied SCA resistance. The curves in Fig. 4 represent two different bytes of the AES with the same HDL code, however one 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. 4(b) has two NICV peaks, one w.r.t correlation and the other due to post-processing of cipher. On the other hand, Fig. 4(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 $(800—1,000)$.

D. 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.

E. Accelerating SCA on Asymmetric Key Cryptography

Asymmetric key cryptography consists in computing exponentiations. For example, in RSA [11], 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 the basis of the SCA on RSA [12]: 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. This methodology is illustrated in Fig. 5.

V. Conclusions and Perspectives

Advanced EM measurement techniques are widely deployed to acquire high quality side-channel traces with enhanced resolution. With the increase in resolution, there is a need for methodology to compress the enormous traces to a small set of relevant points. We presented NICV as a leakage detection technique for side-channel leakage. It can be related to Pearson correlation and SNR. NICV uses public information like plaintext or ciphertext for detecting leakaging points in an EM trace

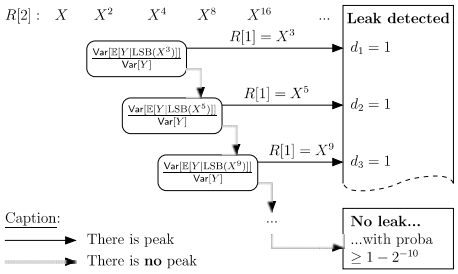


Fig. 5. Illustration of the application of NICV to RSA

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

References

[1] É. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *CHES*, ser. LNCS, vol. 3156. Springer, August 11–13 2004, pp. 16–29, Cambridge, MA, USA.

[2] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in *CHES*, ser. LNCS, vol. 2523. Springer, August 2002, pp. 13–28, San Francisco Bay (Redwood City), USA.

[3] C. C. Consortium, "Common Criteria (*aka* CC) for Information Technology Security Evaluation (ISO/IEC 15408)," 2013, Website: http://www.commoncriteriaportal.org/.

[4] R. J. Easter, "Text for ISO/IEC 1st WD 17825 – Information technology – Security techniques – Non-invasive attack mitigation test metrics for cryptographic modules," January 19 2012, Prepared within ISO/IEC JTC 1/SC 27/WG 3. (Online).

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

[6] S. Mangard, E. Oswald, and F.-X. Standaert, "One for All - All for One: Unifying Standard DPA Attacks," *Information Security, IET*, vol. 5, no. 2, pp. 100–111, 2011, ISSN: 1751-8709 ; Digital Object Identifier: 10.1049/iet-ifs.2010.0096.

[7] A. Moradi, O. Mischke, and T. Eisenbarth, "Correlation-Enhanced Power Analysis Collision Attack," in *CHES*, ser. Lecture Notes in Computer Science, vol. 6225. Springer, August 17-20 2010, pp. 125–139, Santa Barbara, CA, USA.

[8] E. Prouff, M. Rivain, and R. Bevan, "Statistical Analysis of Second Order Differential Power Analysis," *IEEE Trans. Computers*, vol. 58, no. 6, pp. 799–811, 2009.

[9] J. Doget, E. Prouff, M. Rivain, and F.-X. Standaert, "Univariate side channel attacks and leakage modeling," *J. Cryptographic Engineering*, vol. 1, no. 2, pp. 123–144, 2011.

[10] S. Bhasin, S. Guilley, Y. Souissi, T. Graba, and J.-L. Danger, "Efficient Dual-Rail Implementations in FPGA using Block RAMs," in *ReConFig*. IEEE Computer Society, November 30 – December 2 2011, pp. 261–267, Cancún, Quintana Roo, México. DOI: 10.1109/ReConFig.2011.32.

[11] R. L. Rivest, A. Shamir, and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[12] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power Analysis Attacks of Modular Exponentiation in Smartcards," in *CHES*, ser. LNCS, Ç. K. Koç and C. Paar, Eds., vol. 1717. Springer, 1999, pp. 144–157.