On the Use of Independent Component Analysis to Denoise Side-Channel Measurements

Houssem Maghrebi, Emmanuel Prouff

On the Use of Independent Component Analysis to Denoise Side-Channel Measurements

Houssem Maghrebi[1] and Emmanuel Prouff[2,3]

[1] Underwriters Laboratories***, France
houssem.maghrebi@ul.com
[2] ANSSI†, France
emmanuel.prouff@ssi.gouv.fr
[3] Sorbonne Universités, UPMC Univ Paris 06, CNRS, INRIA, Laboratoire d'Informatique de Paris 6 (LIP6), Équipe PolSys, 4 place Jussieu, 75252 Paris, France

Abstract. Independent Component Analysis (ICA) is a powerful technique for blind source separation. It has been successfully applied to signal processing problems, such as feature extraction and noise reduction, in many different areas including medical signal processing and telecommunication. In this work, we propose a framework to apply ICA to denoise side-channel measurements and hence to reduce the complexity of key recovery attacks. Based on several case studies, we afterwards demonstrate the overwhelming advantages of ICA with respect to the commonly used preprocessing techniques such as the singular spectrum analysis. Mainly, we target a software masked implementation of an AES and a hardware unprotected one. Our results show a significant Signal-to-Noise Ratio (SNR) gain which translates into a gain in the number of traces needed for a successful side-channel attack. This states the ICA as an important new tool for the security assessment of cryptographic implementations.

Keywords: independent component analysis, side-channel analysis, preprocessing, noise filtering, correlation power analysis, Boolean masking scheme.

1 Introduction

Side-Channel Attacks. Side-Channel Attacks (SCA) are nowadays well known and most designers of secure embedded systems are aware of them. Since the first public reporting of these threats [41] in 1996, a lot of effort has been devoted towards the research about side-channel attacks and the development of corresponding countermeasures. SCA take advantage of the fact that the power consumption (or the electromagnetic radiation) of a cryptographic device depends on the internally used secret key. Since this property can be exploited

*** This work has been done when the author was working at Safran Identity and Security.

† This work has been done when the author was working at Safran Identity and Security.

with relatively cheap equipment, these attacks pose a serious practical threat to cryptographic embedded systems. To perform a successful side-channel attack against embedded cryptographic implementations, several steps should be carefully followed [31]. First, the physical leakage (*e.g.* the power consumption or the electromagnetic radiation) of the target device must be measured during the processing of cryptographic algorithms. Second, it is common to preprocess the collected measurements by applying for instance: traces alignment, noise filtering, Points-Of-Interest (POI) selection and dimensionality reduction [45]. Finally, statistical distinguishers are applied on the (preprocessed) traces to discriminate key hypotheses.

Preprocessing Tools in SCA Context. When looking at the broad literature of side-channel attacks, several techniques and tools have been proposed to preprocess the measurements. The goal behind is to reduce the attack complexity in terms of computational time and number of traces needed for a successful attack. From the side-channel evaluation perspective, the preprocessing step is of great importance especially when targeting modern embedded systems (*e.g.* mobile phone) [27] and System-on-Chip (SoC) devices with high clock frequencies [3,43].

We provide hereafter a brief overview of the most commonly used preprocessing techniques in side-channel context:

- **Traces synchronization**: to conceal the traces misalignment typically caused by inaccuracies in triggering the power measurements or by some activated countermeasures (*e.g.* clock jitter), several works have proposed to apply synchronization techniques like the alignment [45] (*i.e.* performing a cross-correlation with sliding widow to search a pattern) or elastic alignment [57] based on the Dynamic Time Warping (DTW) algorithm.
- **Noise filtering**: several techniques have been applied to deal with traces denoising. These techniques range from simple ones like averaging to sophisticated ones like the use of the fourth-order cumulant [42] or the application of some linear filters (*e.g.* Wiener filter or Kalman filter [53]). Recently at CHES 2014, Del Pozo *et al.* have suggested using the Singular Spectrum Analysis (SSA) as a filtering technique to improve the efficiency of side-channel attacks [46]. The results obtained on various scenario (*e.g.* unprotected and masked software implementations of an AES and a Hardware implementation of PRESENT) have shown the overwhelming advantages of using this technique. However, some (hyper) parameters (*i.e.* the choice of the window length for constructing the *trajectory matrix* and the principal components selection for the reconstruction [46]) are ad-hoc and thus, if not properly executed could diminish the associated gains of SSA.
- **POI selection**: the computation complexity of side-channel attacks can be reduced by selecting a small subset of time samples where leakage prevails. To achieve this goal, several works have proposed some preprocessing techniques amongst which we identify the Sum Of Squared pairwise Differences (SOSD) and the Sum Of Squared pairwise T-differences (SOST) [30] based on the

T-Test algorithm [22,52] to choose the most relevant time samples. Other techniques exist and are rather based on SNR computation [45], variance tests [8], correlation and mutual information [23,29].

– **Dimensionality reduction**: the most commonly used methods for dimensionality reduction in side-channel context are the Principal Component Analysis (PCA) and the Linear Discriminant Analysis (LDA) [2,5,15,54] or the Kernel Discriminant Analysis (KDA) [11]. While the first provides a set of vectors (*aka* the principal components) onto which the data are first projected and then only few projections (these that maximize the variance between the mean leakage traces) are kept, the second one projects the data on the directions that maximize the ratio between inter-class and intra-class variances. So, reducing the data complexity aims at decreasing the computation time during the key recovery phase.

Our Contribution. By contrast, the denoising techniques are in general discussed less, despite their importance in reducing the complexity of side-channel attacks especially for Common Criteria evaluation [18]. In this paper, we propose the use of the Independent Component Analysis (ICA) [16,17,40] to denoise side-channel measurements. This technique is widely applied for Blind Source Separation (BSS) (see *e.g.* [36] for an application of the ICA in reducing the noise in natural images) and aims at finding a linear representation of the processed multivariate data so that the resulting components are statistically independent. To the best of our knowledge this is the first complete attempt to apply ICA as a preprocessing technique in side-channel context. Actually, in [26] Gao *et al.* have proposed a new profiled attack based on the ICA and they claimed that it could be used to improve the signal-to-noise (SNR) ratio, but they left this for further research. In another paper [9], Bohy *et al.* have also suggested a similar application but they didn't provide a practical framework on how to efficiently apply it.

Throughout several practical experiments (see Sec. 4), we argue that ICA outperforms the commonly used denoising methods in side-channel context and leads to a significant SNR gain which translates into a significant advantage in terms of number of traces needed to succeed an attack. For instance, we represent in Fig. 1 the results of a first-order Correlation Power Analysis (CPA) attack [10] when targeting an unprotected software implementation of the AES running on an ATMega163 micro-controller. Several denoising techniques have been applied for the sake of comparison.

From Fig. 1, one can conclude that the gain in terms of number of traces needed to succeed the CPA attack, with respect to our specific experiments, is about 120% compared to the SCA state-of-the-art filtering techniques.

Moreover, we compare ICA to the well-known preprocessing techniques used in SCA context to ensure dimensionality reduction and POI selection (*i.e.* PCA [54], LDA [15] and the Projection Pursuit (PP) [25]). Despite the fact that these methods are applied for different purposes than measurements denoising, we pinpoint several similarities with ICA that we discuss in Sec. 2.4.

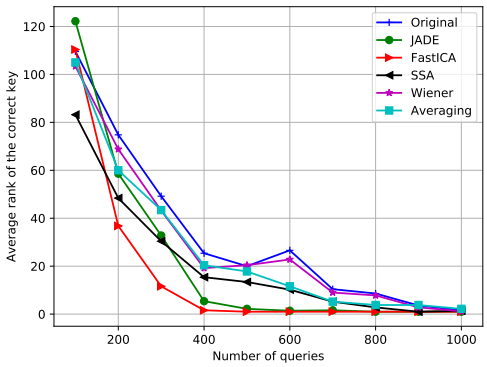


Fig. 1: Evolution of the correct key rank (y-axis) according to an increasing number of traces (x-axis) for several filtering techniques when targeting an unprotected software implementation of the AES.

2 Independent Component Analysis

2.1 Notations

In the rest of the paper, bold block capitals $\mathbf{X}$ denote matrices and bold lower cases $\mathbf{x}$ denote real row vectors. The identity matrix of dimension n is denoted by $\mathbf{I}_n$. The i^{th} row vector of a matrix $\mathbf{X}$ is denoted by $\mathbf{x}_i$, while its i^{th} coordinate is denoted by $\mathbf{x}[i]$. The transpose and the inverse of a matrix $\mathbf{X}$ are respectively denoted by $\mathbf{X}^T$ and $\mathbf{X}^{-1}$. The capital letters X are used for random variables while the lower-case letter x for their realizations. The mean, the variance and the entropy of a random variable X are respectively denoted by $\mathbb{E}[X]$, $\mathbb{V}[X]$ and $\mathbb{H}[X]$. We any (n, m)-matrix $\mathbf{M}$, we shall denote by $\mathbb{E}[\mathbf{M}]$ the mean of the matrix when drawn uniformly at random in its definition set. The dot/inner product and the matrix product shall be denoted by $\cdot$, while the product over $\mathbb{R}$ and the product between a scalar and a vector shall be denoted by $\times$.

2.2 Overview of ICA

ICA [16,17,40] is one of the most widely used techniques for blind source separation [48]. It assumes that the observed data are drawn from multiple source signals and aims at recovering these individual signals. A typical example is the so-called *cocktail party problem*: in a room, multiple people are speaking simultaneously while there are some recorders in different places of the room capturing the superimposition of their voices. The objective is to recover the speech of each individual speaker from the recorded voices.

We present hereafter a mathematical model of this problem. Let $\mathbf{x}_i$ and $\mathbf{s}_i$ respectively denote an observation and a source p-dimensional vector over

$\mathbb{R}$. We define $\mathbf{X} = (\mathbf{x}_i)_{1 \leq i \leq n}$ and $\mathbf{S} = (\mathbf{s}_i)_{1 \leq i \leq n}$ as respectively the so-called *observations* (n, p)*-matrix* and the so-called *sources* (n, p)*-matrix* both defined over $\mathbb{R}^{n \times p}$ such that:

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S} \ , \tag{1}$$

where $\mathbf{A} = (a_{i,j})_{1 \leq i,j \leq n}$ is the so-called *mixing matrix* defined over $\mathbb{R}^{n \times n}$. Hence, (1) implies that the observations (*e.g.* the recorded speech) are considered as a linear combination of the sources (*e.g.* the individual voice of each speaker). For instance, the i^{th} p-dimensional row vector of $\mathbf{X}$ rewrites:

$$\mathbf{x}_i = \sum_{j=1}^{n} a_{i,j} \times \mathbf{s}_j \ . \tag{2}$$

Remark 1. In the rest of this paper, we will often consider the $\mathbf{x}_i$ and $\boldsymbol{s}_j$ as random variables drawn uniformly from their respective definition set.

The goal of ICA is to solve the following problem:

Problem 1. Approximate the unknown matrix $\mathbf{A}$ in order to recover the latent signals $\boldsymbol{s}_j$ from the observable data $\mathbf{X}$.

To address Problem 1, the ICA first looks for an estimation $\hat{\mathbf{W}}$ of the so-called *unmixing matrix* $\mathbf{W}$ defined over $\mathbb{R}^{n \times n}$ such that $\mathbf{W} = \mathbf{A}^{-1}$ and secondly recovers an approximation of the sources matrix by computing:

$$\hat{\mathbf{S}} = \hat{\mathbf{W}} \cdot \mathbf{X} \ .$$

The ICA asymptotically succeeds in solving Problem 1 (*i.e.* $\hat{\mathbf{S}} = \mathbf{S}$) if the two following assumptions are satisfied [38].

Assumption 1 (Statistical independence) *The source signals* $\mathbf{s}_j$ *are mutually independent.*

Assumption 2 (Non-Gaussian distribution) *The source signals* $\mathbf{s}_j$ *have non-Gaussian distributions.*

Remark 2. Remarkably, Assumption 2 can be relaxed by allowing at most one source signal to have a Gaussian distribution [38]. This is an important remark in our case study since, as we will see in the following, one of the source signals corresponds to a noise observation (often assumed to have a Gaussian distribution).

From the ICA model in (1), the following ambiguities may already be discussed:

– **whitening**: it is impossible to estimate the original variance and sign of the source signals. Indeed, since both $\mathbf{S}$ and $\mathbf{A}$ are unknown, then any scalar multiple of one of the sources $\boldsymbol{s}_j$ can always be cancelled by dividing the corresponding column vector $\mathbf{a}_j$ of $\mathbf{A}$ by the same scalar; namely, for every $\alpha \in \mathbb{R}$ the relation $\mathbf{X} = (\mathbf{A} \cdot \alpha^{-1}) \cdot (\alpha \cdot \mathbf{S})$ holds.

As a consequence, and unlike PCA which focus on the variance maximization problem to find the optimal projections of the data [39], ICA exploits higher-order statistical moments to recover the sources $\boldsymbol{s}_j$ [38]. Indeed, before performing ICA, the observations $\mathbf{X}$ are centered and *whitened*, that is, modified to have identity covariance matrix [38]. This is typically done by applying first the Eigen-Value Decomposition (EVD) of the covariance matrix $\mathbb{E}[\mathbf{X}\cdot\mathbf{X}^T]$ defined by $\mathbb{E}[\mathbf{X}\cdot\mathbf{X}^T] = \mathbf{E}\cdot\mathbf{D}\cdot\mathbf{E}^T$ where $\mathbf{E}$ and $\mathbf{D}$ denote respectively the *orthogonal matrix of eigenvectors* and the *diagonal matrix of eigenvalues*. Then, the *whitened observations* are defined as $\tilde{\mathbf{X}} = \mathbf{E}\cdot\mathbf{D}^{1/2}\cdot\mathbf{E}^T\cdot\mathbf{X}$ and one can easily check that the covariance matrix $\mathbb{E}[\tilde{\mathbf{X}}\cdot\tilde{\mathbf{X}}^T]$ is the identity matrix $\mathbf{I}_n$.

– **Order invariant**: it is impossible to determine the order of the source signals. In fact, since both $\mathbf{S}$ and $\mathbf{A}$ are unknown, then any permutation of the source signals could always be canceled by applying the inverse permutation on the mixing matrix $\mathbf{A}$. Let $\mathbf{P}$ be a permutation matrix defined over $\mathbb{R}^{n\times n}$, then from (1) we have $\mathbf{X} = (\mathbf{A}\cdot\mathbf{P}^{-1})\cdot(\mathbf{P}\cdot\mathbf{S})$. Consequently, we shall say that $\hat{\mathbf{W}}$ is a good estimation of $\mathbf{W}$ if there exists a matrix $\mathbf{Q}$ permutation of the identity matrix $\mathbf{I}_n$ such that $\hat{\mathbf{W}} = \mathbf{Q}\cdot\mathbf{W}$.

2.3 ICA Estimation

Let us denote by $\mathbf{y}_j$ the j^{th} p-dimensional row vector $\hat{\mathbf{w}}_j\cdot\mathbf{X}$ where $\hat{\mathbf{w}}_j$ is the j^{th} row of $\hat{\mathbf{W}}$. Then, as a direct consequence of (1) and after denoting $\mathbf{z}_j = \hat{\mathbf{w}}_j\cdot\mathbf{A}$, for $j\in[1;n]$ we have:

$$\mathbf{y}_j = \sum_{i=1}^{n} \mathbf{z}_j[i]\times\mathbf{s}_i \ . \tag{3}$$

Thus, $\mathbf{y}_j$ is a linear combination of the source signals $\mathbf{s}_i$. If a single coefficient $\mathbf{z}_j[i]$ in (3) is non-zero then the sum contains a single signal $\mathbf{s}_i$ and therefore $\mathbf{y}_j$ corresponds to a row of the signal matrix $\mathbf{S}$ we want to recover (equivalently, $\hat{\mathbf{w}}_j$ corresponds to one row of the unmixing matrix $\mathbf{W}$). In other terms if for every $j\in[1;n]$, the sum in (3) contains a single signal $\mathbf{s}_i$ then $\hat{\mathbf{W}}$ is a good estimation of $\mathbf{W}$ modulo a permutation of the rows order.

Since the $\mathbf{s}_i$ are mutually independent (Assumption 1), this linear combination $\mathbf{y}_j$ tends towards a Gaussian distribution when the number of non-zero coefficients $\mathbf{z}_j[i]$ increases (by *Central Limit Theorem*). Conversely, due to Assumption 2 the vector $\mathbf{y}_j$, viewed as a random variable, becomes least Gaussian when the number of non-zero coefficients $\mathbf{z}_j[i]$ tends towards one. Such a non-Gaussianity of a probability density function (pdf) may for instance be measured thanks to the Kurtosis moments (see Sec. A). Based on this remark, the core idea of ICA is to find, among all possible estimations $\hat{\mathbf{w}}_j$, the one that maximizes the non-Gaussianity of $\hat{\mathbf{w}}_j\cdot\mathbf{X}$. Such a vector would necessarily correspond to a vector $\mathbf{z}$ which has a single nonzero component and the corresponding vector $\mathbf{y}_j = \hat{\mathbf{w}}_j\cdot\mathbf{X}$ should therefore equal one of the source signals $\mathbf{s}_i$.

More formally, ICA is an optimization algorithm that aims (1) at estimating the unmixing matrix $\mathbf{W}$ by maximizing the non-Gaussianity of $\mathbf{y}_j = \hat{\mathbf{w}}_j \cdot \mathbf{X}$, and (2) at afterwards deducing the sources signals. In fact, the optimization landscape for non-Gaussianity in a n-dimensional space of vectors $\hat{\mathbf{w}}_j \in \mathbb{R}^p$ has $2n$ local maxima (two for each source signal corresponding to $+\mathbf{s}_i$ and $-\mathbf{s}_i$). To find all independent source signals, we need to find all these local maxima.

For self-completeness, we list in Appendix A several methods which can be applied to quantify the Non-Gaussianity of a random variable.

2.4 Differences between ICA, Projection Pursuit, PCA and LDA

The Projection Pursuit [25] is a statistical technique that aims at finding the most informative projections of a highly multivariate data. It has been demonstrated in [32] that the most interesting directions are those that show the lowest Gaussianity and this is exactly what the ICA estimation does. Thus, both techniques are remarkably similar and optimize the same criterion despite the fact that they have been developed independently by the Statistics and the Signal Processing communities [38]. Meanwhile, several major differences between these techniques can be pinpointed:

- PP aims at reducing the dimension of the processed data such that only few (*i.e.* mainly one or two) directions are preserved, whereas ICA aims at identifying all source vectors (*i.e.* all directions) with the same dimension as the processed data.
- Unlike ICA, PP makes no assumption about the source signals. Said differently, when ICA assumptions are satisfied, then its estimation returns the independent components of the processed data. Otherwise, what we obtain by applying ICA is the projection pursuit directions.

Regarding PCA [39] and LDA [24], which are widely used in the SCA context for dimensionality reduction and measurement processing [2,5,15,54], several important differences may be noticed. In fact, while PCA aims at finding the most interesting orthogonal projections that maximize the variance of the data, LDA seeks for some directions that maximize the inter-class variance and minimize the intra-class variance of the data. Hence, both techniques exploit the second-order statistic of the processed data unlike ICA that aims rather at estimating higher-order statistics such as the fourth-order cummulant (*i.e.* the Kurtosis) by finding the interesting projections (not necessarily orthogonal) that minimize the Gaussianity of the components [38]. So, PCA and LDA are suitable when the source signals are Gaussian ones and when the signal variance is informative. However, when dealing with strongly non-Gaussian data, the variance may not be the statistic of interest compared to higher-order moments. Indeed, in the ICA model, all timing samples are *a priori* equally important unlike for PCA and LDA where many components will be discarded since judged less informative. Actually, we think that ICA and PCA/LDA are not competitors but complement each other; applying a dimensionality reduction technique (PCA or

LDA) after processing ICA to filter the SCA traces may increase the success of the attack.

2.5 ICA Methods

Several algorithms were developed to perform the ICA estimation. We review in this section the most popular ones.

InfoMax. It is based on a neural network approach which tries to maximize the entropy of the network's output [6,47]. Let us view the observations' matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ as an input layer, then the p-dimensional rows $\mathbf{y}_j$, $j \in [1; n]$, of the matrix $\mathbf{Y}$ defined in Sec. 2.3 satisfies $\mathbf{y}_j = f_j(\hat{\mathbf{w}}_j \cdot \mathbf{X})$, where f_j is some non-linear function and the vectors $\hat{\mathbf{w}}_j$ can be viewed as the weight vectors of the neurons. So, finding the weight matrix $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_j)_{1 \leq j \leq n}$ that maximizes the *negentropy* of $\mathbf{Y}$ for a well chosen set of f_j functions leads to an ICA estimation. The InfoMax approach is equivalent to the maximum likelihood estimation, not detailed in this work for lack of room, which could be also used to estimate the ICA model [13].

FastICA. It is the most commonly used approach to estimate the ICA [34,37]. It is based on the maximization of the negentropy as demonstrated in (9). Indeed, it is faster than the conventional ICA algorithms and can be used to perform projection pursuit as well [33].

Joint Approximate Diagonalization of Eigenmatrices (JADE). It is based on the diagonalization of the cumulant matrices [14]. In fact, the diagonal elements of a cumulant matrix characterize the distribution of a signal, while the off-diagonal elements indicate the statistical dependencies between signals. So, JADE algorithm uses the second and the fourth cumulant matrix. First, the data are transformed into a reduced set of PCA loadings (*i.e.* a diagonalization of the second-order cumulant matrix with a selection of the interesting directions) that are then whitened to have equal variances. Second, the fourth-order cumulant matrix is diagonalized via a rotation matrix (using the Jacobi algorithm) yielding the mixing matrix.

3 Filtering Leakage using ICA

3.1 SCA Model vs. ICA Model

In a side-channel context, the matrix of observations $\mathbf{X} \in \mathbb{R}^{n \times p}$ in (1) is assumed to be related to the manipulation of a sensitive variable Z ranging over some finite set. We recall that the values taken by Z correspond to the output of a processing $\varphi(m, k)$ involving a plaintext m and a secret parameter k. The dimension n of $\mathbf{X}$ corresponds to the number of observations of the manipulation,

while p denotes the length of each observation. It is often assumed that an observation $\mathbf{x}$, viewed as a random variable defined over $\mathbb{R}^p$, is well modelled by a linear combination of two mutually independent parts:

- a part $Z \mapsto \mathcal{D}(Z) \in \mathbb{R}^p$ which is a deterministic function representing the un-noisy leakage on Z during its manipulation by the system and,
- a random part $\mathbf{r}$ representing the noise in the observations and being associated with a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ (in our case we make the classical assumption that $\mathbf{\Sigma}$ is diagonal which essentially implies that the instantaneous noises in the observation vectors are mutually independent).

Hence, the noisy observation of the manipulation of Z can be associated with a random variable $\mathbf{x}$ defined over $\mathbb{R}^p$ as:

$$\mathbf{x} = a_1 \times \mathcal{D}(Z) + a_2 \times \mathbf{r} \ , \tag{4}$$

where (a_1, a_2) are some weighting coefficients defined over $\mathbb{R}^2$ (and a_2 is often assumed to be equal to one). After assuming that the n rows of $\mathbf{X}$ are n independent realizations of the random variable $\mathbf{x}$ defined in (4), it may be checked that $\mathbf{X}$ fits well the ICA model defined in (1) and (2) by setting $\mathcal{D}(Z) = \mathbf{s}_1^{(m)}$ and $\mathbf{r} = \mathbf{s}_2$.[4] So, the ICA noise reduction technique described in previous sections should allow for an easy detection of the interesting components.

Remark 3. The deterministic part $\mathcal{D}(Z)$ is often assumed to be well estimated by a linear combination in $\mathbb{R}$ of the bits of Z. Under this modelling, the different sources $\mathbf{s}_i$ are no longer 2 but $\log_2(Z)$ (*i.e.* composed of the bits of Z and the noise). In this context, the ICA could be used to isolate the noise signal from the other ones.

Remark 4. We stress the fact that one can extend the leakage model defined in (4) to the following one:

$$\mathbf{x} = a_1 \times \mathcal{D}(Z_1) + a_2 \times \mathcal{D}'(Z_2) + a_3 \times \mathbf{r} \ ,$$

where (a_1, a_2, a_3) is a triplet of weighting coefficients defined over $\mathbb{R}$, where $\mathcal{D}(Z_1)$ is the deterministic part of the targeted variable Z_1 and where $\mathcal{D}'(Z_2)$ is the deterministic part of a non-targeted variable Z_2 (aka algorithmic noise). This model can be used, for instance, when an adversary tries his attack on several SBoxes processed in parallel in a hardware setting context.

At this point, it must be observed that, unlike SSA (which transforms individual traces [46]), the ICA cannot be applied on a single observation in our context (*i.e.* on matrices $\mathbf{X}$ with a single row) since at least $n > t$ measurements are generally required to recover t source signals [28]. So according to our modelling where $t = 2$, at least two measurements are required, for each

[4] Note that we used the notation $\mathbf{s}_1^{(m)}$ to alert on the fact that the signal s_1 corresponds to the plaintext m.

possible value z of Z (or equivalently for each possible m), to recover the power consumption of $\mathcal{D}(Z = z) = \mathcal{D}(\varphi(m, k))$. Let us assume that we collected two such power observations by executing the processing two times for a randomly chosen plaintext m. We then get a matrix of observations $\mathbf{X}$ composed of two rows $\mathbf{x_1}$ and $\mathbf{x_2}$) which are realizations of the same random variable defined in (4) and hence satisfy:

$$\mathbf{x_1} = a_{1,1} \times \underbrace{\mathcal{D}(Z = z \text{ where } z = \varphi(m, k))}_{\text{realization of } \mathbf{s}_1^{(m)}} + a_{1,2} \times \underbrace{\mathbf{r}_1}_{\text{realization of } \mathbf{s}_2} \tag{5}$$

and

$$\mathbf{x_2} = a_{2,1} \times \underbrace{\mathcal{D}(Z = z \text{ where } z = \varphi(m, k))}_{\text{realization of } \mathbf{s}_1^{(m)}} + a_{2,2} \times \underbrace{\mathbf{r}_2}_{\text{realization of } \mathbf{s}_2} \tag{6}$$

with $\mathbf{r}_1$ and $\mathbf{r}_2$ being two realizations of the same noise random variable $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. As recalled in Sec. 2, the ICA technique recovers the source signals $\mathcal{D}(Z = z)$ and $\mathbf{r}$ by estimating the unmixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ *s.t.*

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} .$$

3.2 First Approach to Apply ICA in SCA Context

To apply ICA for denoising side-channel measurements, a first approach may consist in using two identical probes to capture the leakage during the execution of a cryptographic implementation. So, for each execution (*e.g.* a plaintext encryption) two measurements (one per probe) are collected satisfying (5) and (6). Then, an ICA algorithm is applied to recover the noise-free signal.

To efficiently apply this approach, both probes must be positioned above the same location of the chip surface to collect the same activity. We believe that this constraint is sometime hard to fulfill in practice and is also highly dependent on the size of the targeted chip under evaluation.

3.3 Second Approach to Apply ICA in SCA Context

We present in Algorithm 1 a second framework for using the ICA technique in order to filter the side-channel measurements.

Our algorithm takes as input the matrix of observations $\mathbf{X}$ (with row elements denoted by $\mathbf{x}_i$) and the corresponding set of plaintexts (or ciphertexts) $\mathcal{M} = \{m_i\}$ used during the execution of the targeted cryptographic operation. The goal is to output a set of noise-free measurements. To do so, for each possible value m of the plaintext we collect all[5] the observations $\mathbf{x}_i$ that have been

[5] Another option could consist in only using a few number of measurements (*e.g.* 100) for each value m in order to speed up the execution of our algorithm.

Algorithm 1 Denoising side-channel traces using ICA technique

Require: $\mathbf{X}$: the noisy measurements dataset and $\mathbf{M}$: the set of plaintexts (or ciphertexts) m such that $Z = \varphi(m, k)$ for some public function φ and some target secret k

Ensure: filtered measurement dataset

1: **for** each value m **do**
2: From $\mathbf{X}$ take the observations $\mathbf{x}_i$ that have been captured during the processing of m and store them in a new observation matrix $\mathbf{X}^{(m)}$
3: **end for**
4: **for** each value m **do**
5: Apply ICA on $\mathbf{X}^{(m)}$ to remove the noise signal ($\mathbf{s}_2$) and keep the genuine signal ($\mathbf{s}_1^{(m)}$)
6: In $\mathbf{X}^{(m)}$, replace each row by $\mathbf{s}_1^{(m)}$
7: **end for**
8: **return** $(\mathbf{X}^{(m)})_m$

captured during the processing of this value (*i.e.* for which $m_i = m$) and we store them in a new observations matrix $\mathbf{X}^{(m)}$. Then, for each of the $\mathbf{X}^{(m)}$, one of the ICA methods described in Sec. 2.5 (*e.g.* FastICA or JADE) is applied to recover the source signals: the noise $\mathbf{s}_2$ and the genuine signal $\mathbf{s}_1^{(m)}$. At this step, the genuine signal can be identified by mere visual inspection and/or by fixing a threshold to distinguish it from the noise signal[6] (an illustration is given in Appendix B). Actually, this phase is quite essential since as discussed in Sec. 2.2 one of the ambiguities of the ICA technique is that the recovered source signals are outputted in a random order. Then, the noise component $\mathbf{s}_2$ is removed and only the genuine signal $\mathbf{s}_1^{(m)}$ is kept. Finally, we replace all the measurements in $\mathbf{X}^{(m)}$ that have been captured during the processing of m by the noise-free signal $\mathbf{s}_1^{(m)}$. Once, we have performed this procedure for all m values, we obtain a set of filtered measurements.

In the sequel, we will rather use the second approach for applying the ICA technique. Our choice was motivated by the fact that it is faster than the first approach and requires fewer measurements.

4 Practical Experiments

4.1 Experimental Setup

Targeted Implementations. To evaluate the efficiency of the ICA framework described in Sec. 3.3, we have targeted two different implementations: (1) a

[6] This threshold is defined for one m value (*e.g.* $m = 0$) and then applied for the other ones. We stress the fact that other approaches could be applied to distinguish the genuine signal from the noise. For instance, one can (1) compute the correlation between the noisy signal and the obtained source signals or (2) apply a dimensionality reduction algorithm (*e.g.* PCA or LDA).

software AES implementation first unprotected and secondly protected by first-order Boolean masking and (2) a hardware unprotected one.

While for the software unprotected AES implementation (running on an ATMega163 micro-controller) the power traces were acquired using our in-house equipment[7], we used the power measurements publicly provided in the website of DPA-contest V2 [56] for the hardware one. The rationale behind using the DPA contest V2 campaign is twofold: (1) to evaluate the efficiency of ICA on a very noisy setup [56] and (2) to ease the reproduction of our results by the side-channel community. Finally, our first-order Boolean masking scheme was implemented on the ChipWhisperer-Lite Board (CW1173) [49] and the power traces were collected using our in-house equipment. The goal is to assess the efficiency of ICA technique in the presence of side-channel countermeasures.

Denoising Setup and Evaluation Metric. Regarding the ICA methods, we have considered mainly the FastICA and the JADE algorithms. The source code of these algorithms are publicly available [1,12]. We have just adapted them to our context (*i.e.* by setting the number of the output components and the dimensions of the processed traces). Once, the traces have been filtered using our framework described in Algorithm 1, we conducted a CPA attack over several independent sets of traces. Then, we have computed the averaged rank of the correct key among all key hypotheses (*aka* the *guessing entropy metric* [55]).

ICA vs. State-of-the-art Denoising Techniques. For the sake of comparison, we have applied the *averaging method*[8], the *Wiener filter* [53] and the SSA technique to filter the power traces of both AES implementations[9]. Moreover, we have performed the CPA attack on noisy traces without preprocessing. The goal was to evaluate the efficiency of ICA w.r.t. the commonly used filtering techniques in side-channel context.

4.2 Unprotected AES Implementation on ATMega163

To fulfill the requirement pointed out in the second part of Section 3.1 to apply the ICA technique, we chose to repeat each acquisition two times with the same AES input. We got 10.000 power traces, aka 5.000 pairs of acquisitions. Then, for a sample size n ranging from 50 to 1000, we ran Algorithm 1 for a subset of our acquisitions such that $|\mathbf{X}| = |\mathbf{M}| = n$ and we filtered the traces by applying one of the denoising techniques described in Section 2.5. To quantify the mean behavior of the algorithm, we repeated each experiment 100 times (for each

[7] A LeCroy WavePro 725Zi oscilloscope with maximum 40 GS/s sampling rate and an active differential probe Lecroy ZD1500 have been used to measure the voltage drop over a 1Ω resistor in the VDD path.

[8] It merely consists in replacing the fifth step in Algorithm 1 by an averaging of the traces in $\mathbf{X}^{(m)}$.

[9] We recall that other filtering techniques exist, *e.g.* the wavelet [20], but are not considered in our work since are heuristic methods.

sample size n and each denoising technique). For an illustration, an exemplary power trace of the implementation and the source signals (*i.e.* the noise and the filtered trace) recovered by the FastICA method are shown in Appendix B.

Regarding the SSA, we followed the approach described in [46]: (1) the window length WL was fixed by applying the rule-of-thumb $WL = [\log(n)^c]$ with $c = 1.5$, and (2) during the reconstruction phase only the second component is used. In fact, it was argued in [46] that the first component usually corresponds to low-frequency noise and thus should not be considered during the reconstruction phase. This observation was confirmed during our experiments.

The efficiency of a CPA attack targeting the first AES SBox[10] after each filtering technique is depicted in Fig. 1 with respect to the number of traces before denoising (as described below, the CPA efficiency has been averaged over 100 experiments).

From Fig. 1, the following observations may be emphasized:

- the CPA attack performs well when the traces are filtered using ICA techniques (*i.e.* either FastICA or JADE). In fact, one can see that less traces are needed to disclose the good value of the key when ICA is applied to filter the traces.
- when the SSA is used to denoise the measurements, the gain in terms of SNR is low (compared to the ICA techniques) which translates into a small (or even no) gain in terms of number of traces needed to discover the key with respect to those needed when no preprocessing is done. This can be explained by the fact that SSA is a heuristic tool and that the results are highly dependent on the choice made to set the window length and/or to select the components standing for the useful information. Indeed, in [46], authors have argued that the selection of the most informative components may be simply done by a mere visual inspection of the obtained singular spectrum. However, this ad-hoc approach is subject to errors due to biased selection of the appropriate components. The same conclusion holds for the choice of the window length for constructing the trajectory matrix. Despite the fact that some rules and guidelines exist, the optimal choice is highly dependent on the processed data [58].
- regarding the use of the averaging method and the Wiener filter to denoise the traces, the related attack results are less efficient compared to those obtained when ICA is applied.

4.3 Unprotected AES Implementation on FPGA

For this second scenario, we performed a similar evaluation with the minor difference that we have first estimated the SNR of the traces before and after applying the FastICA technique for denoising. This choice was motivated by the fact that the DPA contest V2 traces are more noisy compared to these acquired on the ATMega163 micro-controller. Let us recall that the leakage satisfies

[10] We stress the fact that same results were obtained when targeting the other SBoxes and are not shown here for lack of room.

$\mathbf{x} = a_1 \times \mathcal{D}(Z) + a_2 \times \mathbf{r}$, then it is well known that the *instantaneous* SNR (*i.e.* the SNR for each of p coordinates of the observation vector) is a p-dimensional vector such that its ith coordinate is defined as:

$$\mathrm{SNR}[i] = \frac{\mathbb{V}_Z[\mathbb{E}[\mathbf{x}[i] \mid Z]]}{\mathbb{E}_Z[\mathbb{V}[\mathbf{x}[i] \mid Z]]} \ . \tag{7}$$

Remark 5. By definition of $\mathbf{x}$, it may be checked that, for every z, $\mathbb{E}[\mathbf{x} \mid Z = z]$ equals $a_1 \times \mathbb{E}[\mathcal{D}(z)] + a_2 \times \mathbb{E}[\mathbf{r}]$, and hence that $\mathbb{V}_Z[\mathbb{E}[\mathbf{x}[i] \mid Z]]$ equals $a_1^2 \mathbb{V}[\mathcal{D}(Z)[i]]$ if the noise $\mathbf{r}$ is independent of Z (which are classical and reasonable assumptions). On the other hand, it can be checked that, for every z, $\mathbb{V}[\mathbf{x}[i] \mid Z = z]$ equals $a_2^2 \mathbb{V}[\mathbf{r}[i]]$. Consequently, (7) is equivalent to $\mathrm{SNR}[i] = \frac{a_1^2}{a_2^2} \frac{\mathbb{V}[\mathcal{D}(Z)[i]]}{\mathbb{V}[\mathbf{r}[i]]}$, under the independent and additive noise assumption. This can be rephrased as the ratio between the variance of the information and the variance of the noise.

Remark 6. In [7], the authors propose to use the *Normalized Inter-Class Variance* (NICV) instead of the SNR. This essentially replaces the denominator in (7) by $\mathbb{V}[\mathbf{x}[i]]$, that is $a_2^2 \mathbb{V}[\mathbf{r}[i]] + a_1^2 \mathbb{V}[\mathcal{D}(Z)[i]]$ since the noise is considered independent of Z. Eventually, this gives $\mathrm{NICV}[i] = \frac{1}{\mathrm{SNR}[i]+1}$.

So, to obtain a first intuition about the efficiency of the ICA as a denoising technique we have compared the obtained SNR with and without applying the FastICA. For the sake of comparison, we also added the SNR when the averaging technique is applied. The results are shown on Fig. 2.

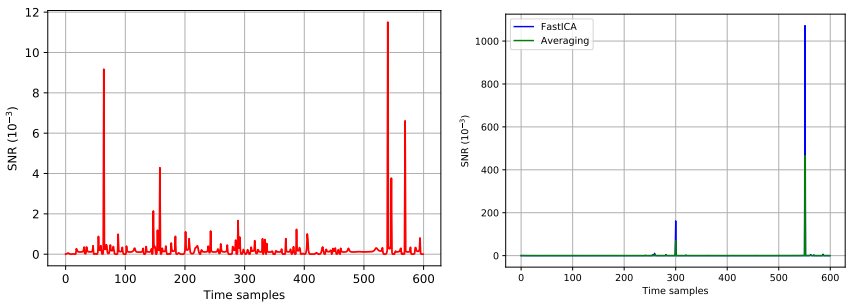


Fig. 2: Signal-to-Noise Ratio estimation without filtering (left) and when applying the FastICA and averaging techniques (right).

From Fig. 2, one can conclude that the SNR gain is close to a factor of 100. In general, higher SNR should translate into a successful attack requiring much less traces. To confirm this claim, we have performed a CPA attack by targeting the output of the first AES SBox. For the sake of comparison, we considered the same

filtering techniques as those used in the first case study (Sec. 4.2). Regarding the SSA, the window length has been set using the previously described rule-of-thumb and only the second component was selected for the re-construction phase. The attack results for each filtering technique are depicted in Fig. 3 (left-hand side).

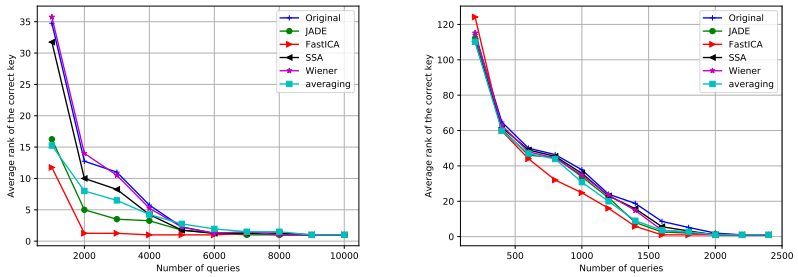


Fig. 3: Evolution of the correct key rank (y-axis) according to an increasing number of traces (x-axis) for each filtering technique when targeting the first AES SBox (unprotected implementation at left-hand side and protected one at right-hand side).

As expected the connection between the SNR gain and the number of traces needed for a successful attack is confirmed. In fact, when applying the FastICA technique less traces are needed to recover the good value of the key (*i.e.* 1.000 traces instead of 3.000 for the non-preprocessed traces). Actually, in several works [44,45] the relation between the number of traces required to achieve 90% of success rate for the CPA attack ($N_{90\%}$) and the SNR has been exhibited and, for every coordinate $i \in [1; p]$, it rewrites:

$$N_{90\%} \approx \frac{2\beta_{90\%}^2}{\mathrm{SNR}[i]} \ , \tag{8}$$

where $\beta_{90\%}$ is a quantile of a normal distribution for the 2-sided confidence interval [45]. So, (8) confirms our experimental findings, the higher the SNR is, the less traces are required to succeed a CPA attack. Regarding the SSA, the averaging method and the Wiener filter, the gains are not that large.

4.4 Masked AES Implementation on the ChipWhisperer-Lite Board (ATMega 128)

We focus in this section on the practical evaluation of the ICA against a first-order Boolean masking scheme[11] implemented on the ChipWhisperer-Lite board

[11] Particular attention has been paid on the implementation to ensure that no first-order leakage occurred.

(CW1173) [49]. To do so, we have acquired a set of power measurements standing for the loading of the masks and the processing of the first AES round. For our attack phase, we assumed that the leaking points related to the loading of the masks are known. Then, we have performed a second-order CPA attack with centered product as a combination function [51]. The attack results when applying different filtering techniques are depicted in Fig. 3 (right-hand side).

From Fig. 3 (right-hand side), one can conclude that the FastICA is more efficient than the other tested denoising techniques. Noticeably, the gain in terms of number of traces needed to succeed a second-order CPA attack is not very high (as it was the case for the second scenario)[12]. This could be explained by the fact that the noise level of the ChipWhisperer-Lite board is quite low.

5 Conclusion and Perspectives

In this work, we proposed an in-depth study of the application of ICA in side-channel context. In particular, we discussed the relationship between the ICA and the commonly used preprocessing techniques (*e.g.* PCA, LDA and projection pursuit). Then, we proposed a framework to use the ICA as a preprocessing technique to reduce the noise level of side-channel measurements. Finally, we validated its interest in three different scenarios. Namely, we considered an unprotected software AES implementation, the noisy traces of the DPA contest v2 and a first-order Boolean masking implementation. The obtained results have shown that the ICA introduces a significant SNR gain which implies a gain in terms of the number of measurements required to succeed a side-channel attack.

References

1. Python implementation of FastICA algorithm. `http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html`.
2. C. Archambeau, É. Peeters, F.-X. Standaert, and J.-J. Quisquater. Template Attacks in Principal Subspaces. In *CHES*, volume 4249 of *LNCS*, pages 1–14. Springer, October 10-13 2006. Yokohama, Japan.
3. J. Balasch, B. Gierlichs, O. Reparaz, and I. Verbauwhede. *DPA, Bitslicing and Masking at 1 GHz*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
4. L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F.-X. Standaert, and N. Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *J. Cryptology*, 24(2):269–291, 2011.
5. L. Batina, J. Hogenboom, and J. G. J. van Woudenberg. *Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis*, pages 383–397. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
6. A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, Nov. 1995.

[12] On other protected implementations, we observed that the gain with ICA techniques is more important. However, we cannot communicate information related to these implementations and the tested chips since these are confidential IPs.

7. S. Bhasin, J.-L. Danger, S. Guilley, and Z. Najm. NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage. In *International Symposium on Electromagnetic Compatibility" (EMC '14 / Tokyo)*. IEEE, May 12-16 2014. Session OS09: EM Information Leakage. Hitotsubashi Hall (National Center of Sciences), Chiyoda, Tokyo, Japan.
8. S. Bhasin, J.-L. Danger, S. Guilley, and Z. Najm. Side-channel Leakage and Trace Compression Using Normalized Inter-class Variance. In *Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '14, pages 7:1–7:9, New York, NY, USA, 2014. ACM.
9. L. Bohy, M. Neve, D. Samyde, and J. jacques Quisquater. Principal and independent component analysis for crypto-systems with hardware unmasked units. In *Proceedings of e-Smart 2003*, 2003.
10. É. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA.
11. E. Cagli, C. Dumas, and E. Prouff. *Kernel Discriminant Analysis for Information Extraction in the Presence of Masking*, pages 1–22. Springer International Publishing, Cham, 2017.
12. J. F. Cardoso. Python and Matlab implementations of JADE algorithm. `https://github.com/camilleanne/pulse/blob/master/jade.py` and `http://perso.telecom-paristech.fr/~cardoso/Algo/Jade/jadeR.m`.
13. J. F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, April 1997.
14. J. F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings F - Radar and Signal Processing*, 140(6):362–370, Dec 1993.
15. O. Choudary and M. G. Kuhn. *Efficient Template Attacks*, pages 253–270. Springer International Publishing, Cham, 2014.
16. P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994.
17. P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 1st edition, 2010.
18. C. C. Consortium. Common Criteria (*aka* CC) for Information Technology Security Evaluation (ISO/IEC 15408), 2013.
Website: `http://www.commoncriteriaportal.org/`.
19. T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.
20. N. Debande, Y. Souissi, M. A. Elaabid, S. Guilley, and J.-L. Danger. Wavelet Transform Based Pre-processing for Side Channel Analysis. In *HASP*, pages 32–38. IEEE, December 2nd 2012. Vancouver, British Columbia, Canada. DOI: 10.1109/MICROW.2012.15.
21. N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: A deflation approach. *Signal Process.*, 45(1):59–83, July 1995.
22. A. A. Ding, C. Chen, and T. Eisenbarth. *Simpler, Faster, and More Robust T-Test Based Leakage Detection*, pages 163–183. Springer International Publishing, Cham, 2016.
23. F. Durvaux and F.-X. Standaert. *From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces*, pages 240–262. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
24. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

25. J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 23(9):881–890, Sept. 1974.
26. S. Gao, H. Chen, W. Wu, L. Fan, W. Cao, and X. Ma. *My Traces Learn What You Did in the Dark: Recovering Secret Signals Without Key Guesses*, pages 363–378. Springer International Publishing, Cham, 2017.
27. D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom. Ecdsa key extraction from mobile devices via nonintrusive physical side channels. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1626–1638, New York, NY, USA, 2016. ACM.
28. P. Georgiev and F. J. Theis. *Blind Source Separation of Linear Mixtures with Singular Matrices*, pages 121–128. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
29. B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis. In *CHES, 10th International Workshop*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, August 10-13 2008. Washington, D.C., USA.
30. B. Gierlichs, K. Lemke-Rust, and C. Paar. Templates vs. Stochastic Methods. In *CHES*, volume 4249 of *LNCS*, pages 15–29. Springer, October 10-13 2006. Yokohama, Japan.
31. G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side-channel resistance validation, September 2011. NIST Non-Invasive Attack Testing Workshop, `http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf`.
32. P. J. Huber. Projection pursuit. *Ann. Statist.*, 13(2):435–475, 06 1985.
33. A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 273–279. MIT Press, 1998.
34. A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *Trans. Neur. Netw.*, 10(3):626–634, May 1999.
35. A. Hyvärinen. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Processing Letters*, 10(1):1–5, 1999.
36. A. Hyvärinen. Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Comput.*, 11(7):1739–1768, Oct. 1999.
37. A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, 9(7):1483–1492, Oct. 1997.
38. A. Hyvrinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.
39. I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics, 2002. ISBN: 0387954422.
40. C. Jutten and J. Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1 – 10, 1991.
41. P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, London, UK, UK, 1999. Springer-Verlag.
42. T.-H. Le, J. Clediere, C. Servière, and J.-L. Lacoume. Noise Reduction in Side Channel Attack using Fourth-order Cumulant. *IEEE Transaction on Information Forensics and Security*, 2(4):710–720, December 2007. DOI: 10.1109/TIFS.2007.910252.
43. J. Longo, E. De Mulder, D. Page, and M. Tunstall. *SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip*, pages 620–640. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

44. H. Maghrebi, V. Servant, and J. Bringer. *There Is Wisdom in Harnessing the Strengths of Your Enemy: Customized Encoding to Thwart Side-Channel Attacks*, pages 223–243. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
45. S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, `http://www.dpabook.org/`.
46. S. Merino Del Pozo and F.-X. Standaert. *Blind Source Separation from Single Measurements Using Singular Spectrum Analysis*, pages 42–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
47. J.-P. Nadal and N. Parga. Nonlinear neurons in the low-noise limit: a factorial code maximizes information transfer. *Network: Computation in Neural Systems*, 5(4):565–581, 1994.
48. G. R. Naik and W. Wang. *Blind Source Separation: Advances in Theory, Algorithms and Applications*. Springer Publishing Company, Incorporated, 2014.
49. C. O'Flynn and Z. D. Chen. *ChipWhisperer: An Open-Source Platform for Hardware Embedded Security Research*, pages 243–260. Springer International Publishing, Cham, 2014.
50. A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Higher Education, 4 edition, 2002.
51. E. Prouff, M. Rivain, and R. Bévan. Statistical analysis of second order differential power analysis. *IEEE Trans. Comput.*, 58(6):799–811, June 2009.
52. T. Schneider and A. Moradi. *Leakage Assessment Methodology*, pages 495–513. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
53. Y. Souissi, S. Guilley, J.-L. Danger, G. Duc, and S. Mekki. Improvement of power analysis attacks using Kalman filter. In *ICASSP*, IEEE Signal Processing Society, pages 1778–1781. IEEE, March 14-19 2010. Dallas, TX, USA; DOI: 10.1109/ICASSP.2010.5495428.
54. F.-X. Standaert and C. Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, August 10–13 2008. Washington, D.C., USA.
55. F.-X. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.
56. TELECOM ParisTech SEN research group. DPA Contest (2^{nd} edition), 2009–2010. `http://www.DPAcontest.org/v2/`.
57. J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker. Improving Differential Power Analysis by Elastic Alignment. In A. Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2011.
58. R. Wang, H.-G. Ma, G.-Q. Liu, and D.-G. Zuo. Selection of window length for singular spectrum analysis. *Journal of the Franklin Institute*, 352(4):1541 – 1560, 2015.

A Estimating the Non-Gaussianity

Several techniques exist to measure the non-Gaussianity of $\mathbf{y}_j = \hat{\mathbf{w}}_j \cdot \mathbf{X}$. We list hereafter, the most commonly used ones.

In this section, we shall assume that $p = 1$, for simplicity reasons, and thus the vectors $\mathbf{y}_j$ are one dimensional and we associate them to the random variables Y_j with $j \in [1; n]$.

A.1 Kurtosis.

The *Kurtosis* is the fourth-order cumulant defined for a univariate random variable Y_j by:

$$\text{kurt}(Y_j) = \mathbb{E}[Y_j^4] - 3 \cdot \mathbb{E}[Y_j^2]^2 \ .$$

It may be checked that the Kurtosis is zero for a Gaussian random variable and is not null (either negative or positive) for most non-Gaussian random variables. Thus, the non-Gaussianity may be measured by computing the absolute value or the square of Kurtosis.

Herein we provide a toy example on how to use the Kurtosis to apply the ICA. Let $\mathbf{X} = (X_1, X_2)$ and $\mathbf{S} = (S_1, S_2)$ be respectively the matrix of observations and sources satisfying the ICA model of (1). Assume that the independent components S_1 and S_2 have the respective Kurtosis values $\text{kurt}(S_1)$ and $\text{kurt}(S_2)$. Our goal is to recover one of the independent sources. Let $Y_j = \hat{\mathbf{w}}_j \cdot \mathbf{X} = \hat{\mathbf{w}}_j \cdot \mathbf{A} \cdot \mathbf{S} = \mathbf{z}_j \cdot \mathbf{S} = z_{j,1} \cdot S_1 + z_{j,2} \cdot S_2$. Using the additive property of the Kurtosis for independent variables, we get:

$$\begin{aligned} \text{kurt}(Y_j) &= \text{kurt}(z_{j,1} \cdot S_1) + \text{kurt}(z_{j,2} \cdot S_2) \\ &= z_{j,1}^4 \cdot \text{kurt}(S_1) + z_{j,2}^4 \cdot \text{kurt}(S_2) \ . \end{aligned}$$

On the other hand, we recall that Y_j has been beforehand whitened (*i.e.* $\mathbb{E}[Y_j^2] = 1$) as explained in Sec. 2.2. Said differently, $\mathbb{E}[Y_j^2] = z_{j,1}^2 + z_{j,2}^2 = 1$. Geometrically, this means that the vector $\mathbf{z}_j$ is constrained to the unit circle on the 2-dimensional plane. The optimization problem is now: find the maxima of the function $|\text{kurt}(Y_j)| = |z_{j,1}^4 \cdot \text{kurt}(S_1) + z_{j,2}^4 \cdot \text{kurt}(S_2)|$ under the constraint $z_{j,1}^2 + z_{j,2}^2 = 1$. Several studies (*e.g.* [17,21]) have solved this problem and have shown that the maxima are the points when exactly one of the elements of $\mathbf{z}_j$ is zero and the other one is non-zero; because of the unit circle constraint, the non-zero element is actually equal to ± 1. Thus, these points are exactly those such that Y_j equals one of the independent components $\pm S_1$ or $\pm S_2$.

A.2 Negentropy.

A second measure of non-Gaussianity is given by the *negentropy*. It measures the difference in entropy between a given distribution and the Gaussian distribution with the same mean and variance. In fact, a fundamental result of information theory is that a Gaussian variable has the largest entropy among all random

variables of equal variance [19,50]. Thus, the entropy can be used as a measure of non-Gaussianity. The negentropy J of a random variable Y_j is defined as:

$$J(Y_j) = \mathbb{H}[Y_{gauss}] - \mathbb{H}[Y_j] \ ,$$

where Y_{gauss} is a Gaussian random variable with the same covariance matrix as Y_j. The negentropy is (1) always positive, (2) invariant by any linear invertible transformation of the variable [16] and (3) equal to zero if and only if Y_j is Gaussian.

The negentropy is the optimal estimator of non-Gaussianity. However, its processing is computationally hard. In fact, this processing requires pdfs estimation which has already been shown to be tricky in several papers (see for instance [4]). To deal with this issue, one of the most common methods is based on the maximum-entropy principle [35] and is defined by:

$$J(Y_j) \simeq \sum_{i=1}^{d} k_i [\mathbb{E}[G_i(Y_j)] - \mathbb{E}[G_i(Y_{gauss})]]^2 \ , \tag{9}$$

where k_i are some positive constants and the functions G_i are some non-quadratic functions (*aka* contrast functions). The variables Y_j and Y_{gauss} are assumed to be of zero mean and unit variance [35]. The most challenging task is now to choose the set of constant values $(k_i)_{1 \leq i \leq d}$ and the appropriate contrast functions G_i to perform an optimal and fast approximation. According to [35], the following choices of G_i provide a good compromise:

$$\begin{aligned} G_i(u) &= \tfrac{1}{b} log(cosh(b \cdot u)) \ \text{ or} \\ G_i(u) &= \quad -\exp(-\tfrac{u^2}{2}) \ , \end{aligned}$$

where b is some constant value in $[1; 2]$, often taken equal to one.

A.3 Mutual Information.

Another approach for the non-Gaussianity estimation inspired by information theory is the minimization of mutual information. In fact, the mutual information I between n mutually independent random variables Y_1, Y_2, ..., Y_n is defined as:

$$\mathrm{I}\,(Y_1; Y_2; \cdots; Y_n) = \sum_{i=1}^{n} \mathbb{H}[Y_i] - \mathbb{H}[\mathbf{Y}] \ ,$$

where $\mathbf{Y} \doteq (Y_1, Y_2, \cdots, Y_n)$. An important property of the mutual information [19,50] is that for an invertible linear transformation $\hat{\mathbf{W}}$ defined over $\mathbb{R}^{n \times n}$ *s.t.* $\mathbf{Y} = \hat{\mathbf{W}} \cdot \mathbf{X}$ we have $\mathbb{H}[\mathbf{Y}] = \mathbb{H}[\mathbf{X}] - \log(\det(\hat{\mathbf{W}}))$. We then deduce that the previous equation rewrites:

$$\mathrm{I}\,(Y_1; Y_2; \cdots; Y_n) = \sum_{i=1}^{n} \mathbb{H}[Y_i] - \mathbb{H}[\mathbf{X}] - \log(\det(\hat{\mathbf{W}})) \ , \tag{10}$$

where det(.) is the determinant function.

Now, let us assume that the Y_i have unit variance, then the covariance matrix of $\mathbf{Y}$ is expressed as:

$$\mathbb{E}[\mathbf{Y} \cdot \mathbf{Y}^\intercal] = \hat{\mathbf{W}} \cdot \mathbb{E}[\mathbf{X} \cdot \mathbf{X}^\intercal] \cdot \hat{\mathbf{W}}^\intercal = \mathbf{I}_n \ ,$$

and its determinant satisfies:

$$\det(\mathbb{E}[\mathbf{Y} \cdot \mathbf{Y}^\intercal]) = \det(\mathbf{I}_n) = \det(\hat{\mathbf{W}}) \cdot \det(\mathbb{E}[\mathbf{X} \cdot \mathbf{X}^\intercal]) \cdot \det(\hat{\mathbf{W}}^\intercal) = 1 \ .$$

This means that $\det(\hat{\mathbf{W}})$ is constant and that the second term in (10) (*i.e.* $\mathbb{H}[\mathbf{X}]$) is also a constant (*i.e.* invariant with respect to $\hat{\mathbf{W}}$). Hence, we have:

$$\mathrm{I}\,(Y_1; Y_2; \ldots; Y_m) = \sum_{i=1}^{n} \mathbb{H}[Y_i] + \text{Constant} \ . \tag{11}$$

So, the minimization of the mutual information $\mathrm{I}\,(Y_1, \cdots, Y_n)$ is achieved by minimizing the entropy of the individual variables Y_i which is equivalent to minimizing their Gaussianity. Moreover, since all Y_i have the same unit variance, their negentropy becomes:

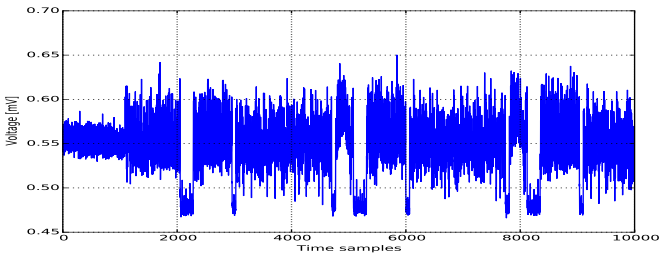
$$J(Y_i) = \mathbb{H}[Y_{gauss}] - \mathbb{H}[Y_i] = C - \mathbb{H}[Y_i] \ , \tag{12}$$

where $C = \mathbb{H}[Y_{gauss}]$ is the entropy of a Gaussian variable with unit variance. By substituting (12) in (11) we get:

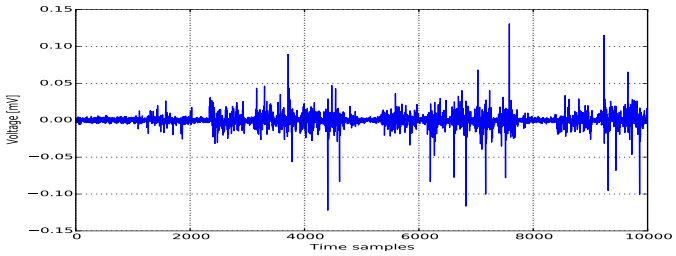
$$\mathbb{H}[Y_1; \ldots; Y_m] = Const - \sum_{i=1}^{n} J(Y_i) \ ,$$

where $Const$ is a constant (that includes all terms C, $\mathbb{H}[\mathbf{X}]$ and $\log(det(\hat{\mathbf{W}}))$. This is the fundamental relation between the mutual information and the negentropy of the variables Y_i. If the mutual information of a set of variables decreases (indicating that the variables are less dependent) then the negentropy increases and the random variables Y_i are less Gaussian. So, finding an invertible transformation $\hat{\mathbf{W}}$ that minimizes the mutual information is equivalent to finding directions in which the negentropy is maximized.

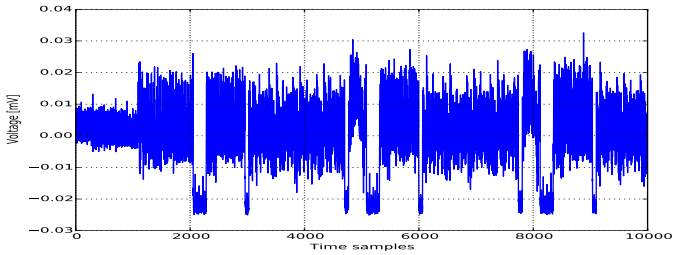
B Example of Trace Denoising based on the FastICA Method



(a) Original trace



(b) Noise signal



(c) Filtered trace

Fig. 4: Unprotected AES implementation: original power trace, noise signal and filtered trace.