Enhancing Dimensionality Reduction Methods for Side-Channel Attacks

Eleonora Cagli, Cécile Dumas, Emmanuel Prouff

Enhancing Dimensionality Reduction Methods
for Side-Channel Attacks

Eleonora Cagli[1,3], Cécile Dumas[1], and Emmanuel Prouff[2,3]

[1] Univ. Grenoble Alpes, F-38000, Grenoble, France
CEA, LETI, MINATEC Campus, F-38054 Grenoble, France
{eleonora.cagli,cecile.dumas}@cea.fr
[2] ANSSI, France emmanuel.prouff@ssi.gouv.fr
[3] UPMC Université Paris 06, Équipe POLSYS, LIP6, F-75005, Paris, France

Abstract. Advanced Side-Channel Analyses make use of dimensionality reduction techniques to reduce both the memory and timing complexity of the attacks. The most popular methods to effectuate such a reduction are the Principal Component Analysis (PCA) and the Linear Discriminant Analysis (LDA). They indeed lead to remarkable efficiency gains but their use in side-channel context also raised some issues. The PCA provides a set of vectors (the *principal components*) onto which project the data. The open question is which of these principal components are the most suitable for side-channel attacks. The LDA has been valorized for its theoretical leaning toward the class-distinguishability, but discouraged for its constraining greed of data. In this paper we present an in-depth study of these two methods, and, to automatize and to ameliorate the principal components selection, we propose a new technique named *cumulative Explained Local Variance (ELV) selection*. Moreover we present some extensions of the LDA, available in less constrained situations than the classical version. We equip our study with a comprehensive comparison of the existing and new methods in real cases. It allows us to verify the soundness of the ELV selection, and the effectiveness of the methods proposed to extend the use of the LDA to side-channel contexts where the existing approaches are inapplicable.

Keywords: Side-channel attacks, dimensionality reduction, principal components analysis, components selection, linear discriminant analysis, explained local variance, small size sample problem

1 Introduction

The measurement of the power consumption or of the electromagnetic irradiations during the execution of cryptographic algorithms in constrained electronic devices can reveal information about sensitive variables (*e.g.* cryptographic keys). The side channel traces are usually acquired by oscilloscopes with a very high sampling rate, which permits a powerful inspection of the component behaviour, but, at the same time, produces high-dimensional data, that spread

the sensitive information over a (sometimes) huge number of time samples. Reducing the dimensionality of the data is an important issue for Side-Channel Attacks (SCA). Considering the side channel traces as column vectors $\mathbf{x}$ in $\mathbb{R}^D$, the compressing phase might be seen as the application of a function $\varepsilon \colon \mathbb{R}^D \to \mathbb{R}^C$, called *extractor* in this paper.

The present work focuses on the so-called *projecting extractors*, *i.e.* those methods that provide extractors ε whose image components are linear combinations of the original data, or equivalently, expressible *via* a matrix multiplication:

$$\varepsilon(\mathbf{x}) = A\mathbf{x} \text{ with } A \in M_{\mathbb{R}}(C, D) \ , \tag{1}$$

where $M_{\mathbb{R}}(C, D)$ denotes the set of real-coefficient matrices of size $C \times D$. In particular we effectuate an in-depth study and a comprehensive comparison between the PCA and the LDA methods [10, 11], and we investigate their exploitability in Side-Channel context. Indeed, PCA and LDA are classical statistical procedures, but the way they have been inherited in SCA domain is somehow ambiguous and opened some issues and questions.

The PCA has been applied both in an *unsupervised* way, i.e. on the whole data [2, 14], and in a *supervised* way, i.e. on traces grouped in classes and averaged [1, 7–9, 22]. The second way implies that, during the training phase, the attacker is able to choose, or at least to know, the secret parameters of the implementation under attack (or a perfect copy of it). As already remarked in [9] and not surprisingly, the complete knowledge assumed in the supervised approach hugely raises performances; we will highlight it in our experiments, and we will concentrate on this powerful kind of approach, leaving the unsupervised case for further studies. The main competitor of PCA in the supervised context is the LDA, that thanks to its class-distinguishability asset, is known to be more meaningful and informative [4, 22] than the PCA method for side channels. Nevertheless, the LDA is often set aside because of its practical constraints; it is subject to the so-called *Small Sample Size problem (SSS)*, i.e. it requires a number of observations (traces) which must be higher than the dimension (size) D of them. In some contexts it might be an excessive requirement, which may become unacceptable in many practical situations where the amount of observations is very limited and the traces size is huge.

One of the open issues in PCA concerns the choice of the components that must be kept after the dimension reduction: as already remarked by Specht et al. [21], some papers declare that the leading components are those that contain almost all the useful information [1, 8], while others propose to discard the leading components [2]. In a specific attack context, Specht et al. compares the results obtained by choosing different subsets of consecutive components, starting from some empirically chosen index. They conclude that for their data the optimal result is obtained by selecting a single component, the fourth one, but they give no formal argumentation about this choice. Such a result is obviously very case-specific. Moreover, the possibility of keeping non-consecutive components is not

considered.

Our main contribution consists in proposing a new selection methodology, called *cumulative ELV selection*. We will argue about the generality and the soundness of this methodology and show that it can raise the PCA performances, making them close to those of the LDA, even in the supervised context. This makes PCA an interesting alternative to LDA in those cases where the LDA is inapplicable. The reasonning behind the ELV selection methodology is essentially based on the observation that, for secure implementations, the leaking information, if existing, is spread over a few time samples of each trace. This observation has already been met by Mavroeidis et al. in [17], where the authors also proposed a components selection method. As we will see in this paper, the main difference between their proposal and ours is that we do not discard the information given by the eigenvalues associated to the PCA components, but we synergistically exploit such information and the observation met. For the sake of comparison, we also analyse many propositions to circumvent the SSS problem that have been made in literature, especially by Pattern Recognition and Face Recognition communities [3, 6, 13, 24]. The gain given by these techniques does not outperform the PCA method equipped with our ELV selection.

The paper is organised as follows: in Section 2 we fix notations, recall preliminaries and formalize the context. Section 3 presents the PCA, and handles the choice of components problem, introducing the ELV selection method. In Section 4 the LDA method is presented, together with different methodologies to avoid the SSS problem. Experiments and comparisons are showed in Section 5, while conclusions and perspectives follow in Section 6.

2 Preliminaries and Formalization

2.1 Preliminaries

In the following, bold block capitals $\mathbf{M}$ denote matrices and Greek or Latin bold lower cases, $\boldsymbol{\alpha}$ or $\mathbf{x}$, denote real column vectors. The i-th entry of a vector $\mathbf{x}$ is indicated by $\mathbf{x}[i]$.

A side-channel key recovery adversary, being inspired by the model proposed by Standaert et al. [23], corresponds to a 5-tuple $\mathcal{A} = (\mathrm{A}, \tau, m, N', N)$, where A is an algorithm or a procedure with time complexity τ and memory complexity m, that takes as input two sets of measurements of respective sizes N' and N. The algorithm A returns a vector of key candidates. Since the goal of the attack is to distinguish the right key $k^\star$ in a set $\mathcal{K}$ of candidates, the output vector, called *guessing vector* $\mathbf{g}$, sorts such candidates in decreasing order with respect to their likelihood:

$$\mathrm{A}\colon \left((\mathbf{x}_i)_{i=1,\ldots N'}, (\mathbf{y}_j)_{j=1,\ldots N}\right) \mapsto \mathbf{g} = [\mathbf{g}[1], \ldots, \mathbf{g}[|\mathcal{K}|]] \ . \tag{2}$$

The first set of input traces $(\mathbf{x}_i)_{i=1,\dots N'}$, here called *profiling set*, is optional, and corresponds to measurements obtained from a profiling device, identical to the device under attack but with full access to the public and secret parameters. The second set of traces $(\mathbf{y}_j)_{j=1,\dots N}$, called *attack set*, corresponds to measurements acquired from the device under attack, parametrized by a key which will be the target of the attack.

An interesting tool to assess the soundness of an adversary is given by the *guessing entropy* [16] and by the asymptotic guessing entropy, respectively defined as

$$\mathrm{GE}_{\mathcal{A}(N)} = \mathbb{E}\left[i\colon \mathbf{g}[i] = k^\star\right] \quad \text{and} \quad \mathrm{GE}_{\mathcal{A}}^\infty = \lim_{N\to\infty} \mathrm{GE}_{\mathcal{A}(N)} \,, \tag{3}$$

where $\mathcal{A}(N)$ denotes the adversary $\mathcal{A}$ with its fifth parameter fixed to N.

A trace $\mathbf{x}$ can be seen as an element in $\mathbb{R}^D$, and its size or dimension D, that depends on a lot of factors (e.g. the instruments setup or the cryptographic algorithm under attack), usually ranges between some thousands and some hundreds of thousands. Nevertheless, only few points of the trace depend on the secret target key. A preliminary step of an attack therefore generally consists in the extraction of the so-called Points of Interest (PoI) from the rough traces. By definition, the latter points are those which depend on both the secret target parameter and on some given public data (a necessary condition to perform *differential* attacks). This extraction represents a non-trivial concrete obstacle for the practical performances of an attack.

2.2 Formalization: Extractors and Fundamental Property

To formalize the problem of the research of PoIs, we remark that in general an attack is composed of four fundamentally different phases:

1. Instruments calibration and traces acquisitions (to build the profiling and attack sets)
2. [Optional] trace pre-processing
3. [Optional] profiling (useful to model the leakage function)
4. Key discrimination: a statistical test, or a statistical distinguisher, is processed over the traces to discriminate key candidates

In this scheme the research of PoI is part of the traces pre-processing. We will formalize it as the application of a function, called *extractor* (by analogy to the notion of randomness extractor [18]):

Definition 1. *Let $\boldsymbol{x} \in \mathbb{R}^D$ represents an observation. An* Extractor *is any function of the form:*

$$\varepsilon_C\colon \mathbb{R}^D \to \mathbb{R}^C \quad \text{with } C \leq D$$
$$\boldsymbol{x} \mapsto \varepsilon_C(\boldsymbol{x}) \,.$$

Notation 1. *The dimension C of an extractor will be omitted if there is no ambiguity or if it is not needed in the context.*

Example 1. A special family of extractors that is widely studied in last years, is the one constituted by the *linear* or *projecting* extractors, *i.e.* those for which each sample in the reduced space $\mathbb{C}$ is a linear combination of samples of the original space. By analogy to (1), such extractors can be defined as $C \times D$ matrices, whose rows are the coefficients to use for the C linear combinations.

Obviously, not any extractor ε is suitable to soundly realise the traces pre-processing of an attack; for example the restriction over a random coordinate, i.e. $\varepsilon(\mathbf{x}) = x[r]$, r being random, is hardly a good candidate for an extractor. For this reason an adversary might aim to only consider extractors that satisfy the following fundamental property:

Property 1 (Effective Extractors). Let $\mathcal{A}$ be an adversary and $\mathrm{GE}_{\mathcal{A}(N)}$ be its guessing entropy, when no trace processing phase is effectuated. Let ε be an extractor, and $\mathcal{A}'$ be an adversary that coincides with $\mathcal{A}$ but whose algorithm A is fed with the sets $\left((\varepsilon(\mathbf{x}_i))_{i=1,\ldots N'}, (\varepsilon((\mathbf{y}_j)))_{j=1,\ldots N} \right)$, i.e. an adversary that applies ε as traces pre-processing phase. The extractor ε is an *effective extractor with respect to N* only if, for any $T \geq N$, we have:

$$\mathrm{GE}_{\mathcal{A}'(T)} \leq \mathrm{GE}_{\mathcal{A}(T)} . \tag{4}$$

In practice this property guarantees that the application of ε does not discard the informative parts of the traces, those that make $\mathcal{A}$ achieve its guessing entropy.

3 Principal Component Analysis

3.1 Principal Component Analysis, the classical statistical tool

The Principal Component Analysis (PCA) [10] is a statistical technique for data dimensionality reduction. It looks for the so-called *Principal Components* (PCs for short), which are vectors that form an orthonormal basis for $\mathbb{R}^D$ (*i.e.* these vectors have norm equal to 1 and are orthogonal to each other). Such PCs are computed as the eigenvectors of the empirical covariance matrix of data: given a set of data $(\mathbf{x}_i)_{i=1,\ldots,N}$, the empirical covariance matrix is given by:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathsf{T}} , \tag{5}$$

where $\overline{\mathbf{x}}$ is the empirical mean of data. Let us denote by r the rank of S and by $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_r$ and $\lambda_1, \ldots, \lambda_r$ its eigenvectors and the corresponding eigenvalues, respectively. We assume that the $\boldsymbol{\alpha}_i$ are listed in the decreasing order of the values λ_i. It can be shown that each λ_i equals the empirical variance of the data

projected onto the corresponding PC $\boldsymbol{\alpha}_i$. Since the data variability is associated to the amount of information, transforming data over the basis provided by the PCs leads to a dimensionality reduction that reinforces the information: such a dimensionality reduction is obtained by projecting the data onto the C-dimensional subspace of $\mathbb{R}^D$ spanned by the C leading PCs, or equivalently by constructing the matrix A of (1) storing as rows the C leading PCs, transposed.

3.2 Principal Component Analysis, the *Class-Oriented Version*

In SCA context, the useful information part contained in data is the one that allows to discriminate observations linked to different intermediate computations. Let us denote by $Z = e(P, K)$ the target intermediate variable, that depends on both a secret variable K and on a public one P, and that takes values $z \in \mathcal{Z}$. The side-channel attack efficiency depends on the ability of the involved extractor to amplify the distinguishability between traces associated to different z.

During the profiling phase the attacker is assumed to know the value z of the sensitive variable handled during each acquisition. He can therefore assign the *class z* to each profiling trace (in analogy with the pattern recognition terminology), obtaining the labelled profiling set $(\mathbf{x}_i^z)_{i=1,\dots,N_z}$, where N_z is the number of traces belonging to the class z. This knowledge is very useful to construct a good class-distinguishing extractor, but the classical PCA does not exploit it. For this reason in SCA literature [1, 7–9, 22] a *class-oriented* version of PCA is often used instead of the classical one. Let $\overline{\mathbf{x}}^z$ be the empirical mean of traces belonging to the same class z. The class-oriented version of the PCA consists in applying the PCA dimensionality reduction to the set $(\overline{\mathbf{x}}^z)_{z \in \mathcal{Z}}$, instead of applying it directly to the traces $\mathbf{x}_i^z$. This implies that the empirical covariance matrix will be computed using only the $|\mathcal{Z}|$ average traces. Equivalently, in case of *balanced* acquisitions (N_z constant for each class z), it amounts to replace the covariance matrix $\mathbf{S}$ of data in (5) by the so-called *between-class* or *inter-class scatter matrix*, given by:

$$\mathbf{S_B} = \sum_{z \in \mathcal{Z}} N_z (\overline{\mathbf{x}}^z - \overline{\mathbf{x}})(\overline{\mathbf{x}}^z - \overline{\mathbf{x}})^\intercal . \tag{6}$$

Remark that $\mathbf{S_B}$ coincides, up to a multiplicative factor, to the covariance matrix obtained using the class-averaged traces.

Performing PCA (or LDA as we will see in next section) always requires to compute the eigenvectors of some symmetric matrix $\mathbf{S}$, essentially obtained by multiplying a matrix $\mathbf{M}$ with its transposed (e.g. for class-oriented PCA we have $\mathbf{M} = [\sqrt{N_{z_1}}(\overline{\mathbf{x}}^{z_1} - \overline{\mathbf{x}}), \sqrt{N_{z_2}}(\overline{\mathbf{x}}^{z_2} - \overline{\mathbf{x}}), \dots]$). Let $\mathbf{M}$ have dimension $D \times N$, and suppose $N \ll D$ (which occurs for example in class-oriented PCA , since $N = |Z|$). Then, the matrix $\mathbf{S} = \mathbf{MM}^\intercal$ has rank at most N. Moreover, rows of $\mathbf{M}$ are often linearly dependent (as in our example since they are forced to have

zero mean), so the rank of $\mathbf{S}$ is actually strictly less than N, giving us at most $N-1$ eigenvectors.

A practical problem when D is large, which happens e.g. when attacking RSA, is represented by the computation and the storage of the $D \times D$ matrix $\mathbf{S}$. Archambeau et al. [1] proposed a method that circumvents this issue, allowing computing the eigenvectors of low-rank big-dimensional symmetric matrices without computing and storing such matrices. In Sec. 4 we will observe in which cases such a method can be applied to LDA and for which LDA variants.

3.3 The Open Question: Choosing the Components to Keep

The introduction of the PCA method in SCA context (either in its classical or class-oriented version) has raised some important questions: *how many* principal components and *which ones* are sufficient/necessary to reduce the trace size (and thus the attack processing complexity) without losing important discriminative information?

Until now, an answer to the questions above has been given in [8], linked to the concept of *explained variance* (or *explained global variance*, EGV for short) of a PC $\boldsymbol{\alpha}_i$:

$$\mathrm{EGV}(\boldsymbol{\alpha}_i) = \frac{\lambda_i}{\sum_{k=1}^{r} \lambda_k} \ , \tag{7}$$

where r is the rank of the covariance matrix $\mathbf{S}$, and λ_j is the eigenvalue associated to the j-th PC $\boldsymbol{\alpha}_j$. $\mathrm{EGV}(\boldsymbol{\alpha}_i)$ is the variance of the data projected over the i-th PC (which equals λ_i) divided by the total variance of the original data (given by the trace of the covariance matrix $\mathbf{S}$, *i.e.* by the sum of all its non-zero eigenvalues). By definition of EGV, the sum of all the EGV values is equal to 1; that is why this quantity is often multiplied by 100 and expressed as percentage. Exploiting the EGV to choose among the PCs consists in fixing a wished *cumulative explained variance* β and in keeping C different PCs, where C is the minimum integer such that

$$\mathrm{EGV}(\boldsymbol{\alpha}_1) + \mathrm{EGV}(\boldsymbol{\alpha}_2) + \cdots + \mathrm{EGV}(\boldsymbol{\alpha}_C) \geq \beta \ . \tag{8}$$

However, if the adversary has a constraint for the reduced dimension C, the EGV notion simply suggests to keep the first C components, taking for granted that the optimal way to chose PCs is in their natural order. This assumption is not always confirmed in SCA context: in some works, researchers have already remarked that the first components sometimes contain more noise than information [2, 21] and it is worth discarding them. For the sake of providing a first example of this behaviour on publicly accessible traces, we applied a class-oriented PCA on 3000 traces from the DPA contest v4 [20]; we focused over a small 1000-dimensional window in which, in complete knowledge about masks and other countermeasures, information about the first Sbox processing leaks (during the first round). In Fig. 1 the first and the sixth PCs are plotted. It may be noticed that the first component indicates that one can attend a high variance

by exploiting the regularity of the traces, given by the clock signal, while the sixth one has high coefficients localised in a small time interval, very likely to signalize the instants in which the target sensitive variable leaks.
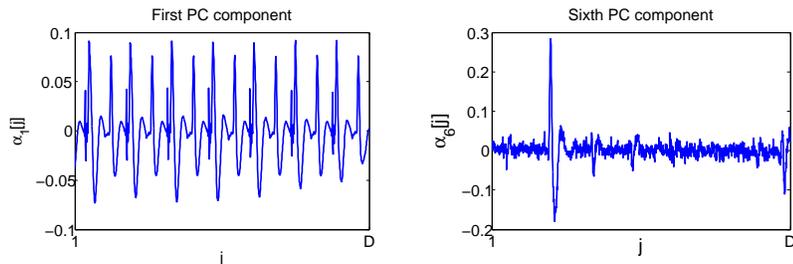

Fig. 1: First and sixth PCs in DPA contest v4 trace set (between time samples 198001 and 199000)

To the best of our knowledge, a single method adapted to SCA context has been proposed until now to automatically choose PCs [17] while dealing with the issue raised in Fig. 1. It is based on the following assumption:

Assumption 1. *The leaking side-channel information is localised in few points of the acquired trace.*

In the rest of the paper, we conduct our own analyses under Assumption 1 that we think to be reasonable in SCA contexts where the goal of the security developers is to minimize the number of leaking points. Under this assumption, the authors of [17] use for side-channel attack purposes the *Inverse Participation Ratio* (IPR), a measure widely exploited in Quantum Mechanics domain (see for example [12]). They propose to use such a score to evaluate the eigenvectors *localization*. It is defined as follows:

$$\text{IPR}(\boldsymbol{\alpha}_i) = \sum_{j=1}^{D} \boldsymbol{\alpha}_i[j]^4 \ . \tag{9}$$

The authors of [17] suggest to collect the PCs in decreasing order with respect to the IPR score.

The selection methods provided by the evaluation of the EGV and of the IPR are somehow complementary: the former is based only on the eigenvalues associated to the PCs and does not consider the form of the PCs themselves; the latter completely discards the information given by the eigenvalues of the PCs, considering only the distribution of their coefficients. One of the contributions of the present paper is to propose a new selection method, that builds a bridge between the EGV and the IPR approaches. As we will argue, our method, based

on the so-called *explained local variance*, does not only lead to the construction of a new selection criterion, but also permits to modify the PCs, choosing individually the coefficients to keep and those to discard.

3.4 The Explained Local Variance Selection Method

The method we develop in this section is based on a compromise between the variance provided by each PC (more precisely its EGV) and the number of time samples necessary to achieve a consistent part of such a variance. To this purpose we introduce the concept of *Explained Local Variance* (ELV).
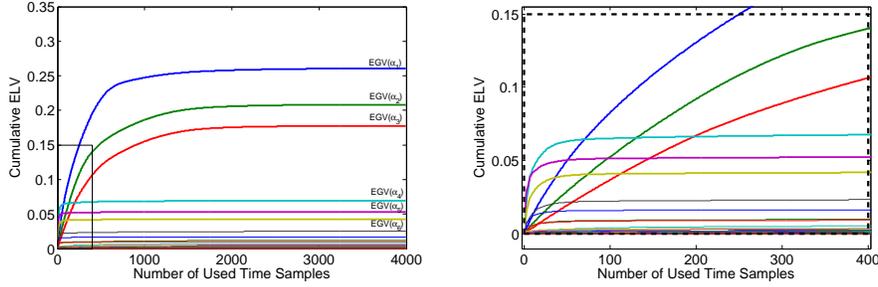

Fig. 2: Cumulative ELV trend of principal components. On the right a zoom of the plot on the left. Data acquisition described in Sec. 5.

Let us start by giving some intuition behind our new concept. Thinking to the observations $\mathbf{x}^{\mathsf{T}}$, or to the class-averages $\overline{\mathbf{x}}^{\mathsf{T}}$ in class-oriented PCA case, as realizations of a random variable $\mathbf{X}^{\mathsf{T}}$, we have that λ_i is an estimator for the variance of the random variable $\mathbf{X}^{\mathsf{T}} \cdot \boldsymbol{\alpha}_i$. Developing, we obtain

$$\lambda_i = \hat{\text{var}}(\sum_{j=1}^{D} \mathbf{X}^{\mathsf{T}}[j]\boldsymbol{\alpha}_i[j]) = \sum_{j=1}^{D}\sum_{k=1}^{D} \hat{\text{cov}}(\mathbf{X}^{\mathsf{T}}[j]\boldsymbol{\alpha}_i[j], \mathbf{X}^{\mathsf{T}}[k]\boldsymbol{\alpha}_i[k]) = \tag{10}$$

$$= \sum_{j=1}^{D} \boldsymbol{\alpha}_i[j] \sum_{k=1}^{D} \boldsymbol{\alpha}_i[k]\hat{\text{cov}}(\mathbf{X}^{\mathsf{T}}[j], \mathbf{X}^{\mathsf{T}}[k]) = \sum_{j=1}^{D} \boldsymbol{\alpha}_i[j](\mathbf{S}_j^{\mathsf{T}} \cdot \boldsymbol{\alpha}_i) = \tag{11}$$

$$= \sum_{j=1}^{D} \boldsymbol{\alpha}_i[j]\lambda_i\boldsymbol{\alpha}_i[j] = \sum_{j=1}^{D} \lambda_i\boldsymbol{\alpha}_i[j]^2 \tag{12}$$

where $\mathbf{S}_j^{\mathsf{T}}$ denotes the j-th row of $\mathbf{S}$ and (12) is justified by the fact that $\boldsymbol{\alpha}_i$ is an eigenvector of $\mathbf{S}$, with λ_i its corresponding eigenvalue. The result of this computation is quite obvious, since $\| \boldsymbol{\alpha}_i \| = 1$, but it evidences the contribution of each time sample in the information held by the PC. This makes us introduce the following definition:

Definition 2. *The* Explained Local Variance *of a PC* $\boldsymbol{\alpha}_i$ *in a sample* j, *is defined by*

$$\mathrm{ELV}(\boldsymbol{\alpha}_i, j) = \frac{\lambda_i \boldsymbol{\alpha}_i[j]^2}{\sum_{k=1}^r \lambda_k} = \mathrm{EGV}(\boldsymbol{\alpha}_i)\boldsymbol{\alpha}_i[j]^2 \ . \tag{13}$$
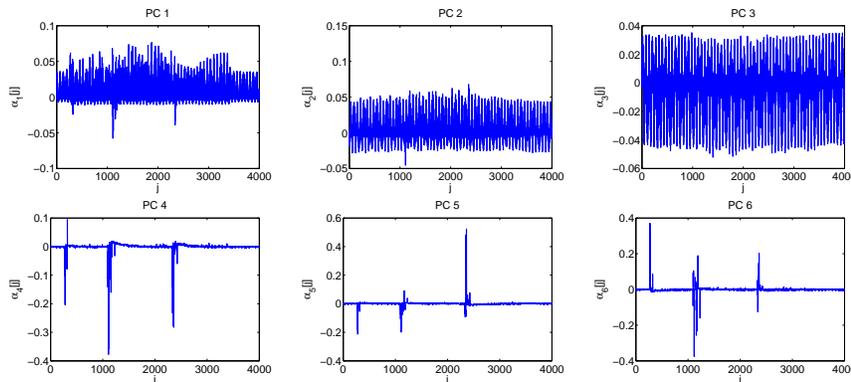

Fig. 3: The first six PCs. Acquisition campaign on an 8-bit AVR Atmega328P (see Sec. 5).

Let $\mathcal{J} = \{j_1^i, j_2^i, \ldots, j_D^i\} \subset \{1, 2, \ldots, D\}$ be a set of indexes sorted such that $\mathrm{ELV}(\boldsymbol{\alpha}_i, j_1^i) \geq \mathrm{ELV}(\boldsymbol{\alpha}_i, j_2^i) \geq \cdots \geq \mathrm{ELV}(\boldsymbol{\alpha}_i, j_D^i)$. It may be observed that the sum over all the $\mathrm{ELV}(\boldsymbol{\alpha}_i, j)$, for $j \in [1, \ldots, D]$, equals $\mathrm{EGV}(\boldsymbol{\alpha}_i)$. If we operate such a sum in a cumulative way following the order provided by the sorted set $\mathcal{J}$, we obtain a complete description of the trend followed by the component $\boldsymbol{\alpha}_i$ to achieve its EGV. As we can see in Fig. 2, where such cumulative ELVs are represented, the first 3 components are much slower in achieving their final EGV, while the 4th, the 5th and the 6th achieve a large part of their final EGVs very quickly (*i.e.* by adding the ELV contributions of much less time samples). For instance, for $i = 4$, the sum of the $\mathrm{ELV}(\boldsymbol{\alpha}_4, j_k^4)$, with $k \in [1, \ldots, 30]$, almost equals $\mathrm{EGV}(\boldsymbol{\alpha}_4)$, whereas the same sum for $i = 1$ only achieves about the 15% of $\mathrm{EGV}(\boldsymbol{\alpha}_1)$. Actually, the EGV of the 4th, the 5th and the 6th component only essentially depends on a very few time samples. This observation, combined with Assumption 1, suggests that they are more suitable for SCA than the three first ones. To validate this statement, it suffices to look at the form of such components (Fig. 3): the leading ones are very influenced by the clock, while the latest ones are well localised over the leaking points.

Operating a selection of components *via* ELV, in analogy with the EGV, requires to fix the reduced space dimension C, or a threshold β for the cumulative ELV. In the first case, the maximal ELVs of each PC are compared, and the C components achieving the highest values of such ELVs are chosen. In the second case, all pairs (PC, time sample) are sorted in decreasing order with respect to

their ELV, and summed until the threshold β is achieved. Then only PCs contributing in this sum are selected.

We remark that the ELV is a score associated not only to the whole components, but to each of their coefficients. This interesting property can be exploited to further remove, within a selected PC, the non-significant points, *i.e.* those with a low ELV. In practice this is done by setting these points to zero. That is a natural way to exploit the ELV score in order to operate a kind of *denoising* for the reduced data, making them only depend on the significant time samples. In Sec. 5 (scenario 4) we test the performances of an attack varying the number of time samples involved in the computation of the reduced data, and showing that such a denoising processing might impact significantly.

4   Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [11] is another statistical tool for dimensionality reduction, which is theoretically more appropriate than PCA for classification problems, such as SCA, as already observed in [4, 22]. Indeed it seeks for linear combinations of data that characterize or separate two or more classes, not only spreading class centroids as much as possible, like the class-oriented PCA does, but also minimizing the so-called *intra-class variance*, i.e. the variance shown by data belonging to the same class.

Description. Applying LDA consists in maximizing the so-called *Rayleigh quotient*:

$$\boldsymbol{\alpha}_1 = \operatorname{argmax}_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}^{\mathsf{T}} \mathbf{S_B} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^{\mathsf{T}} \mathbf{S_W} \boldsymbol{\alpha}} \ , \tag{14}$$

where $\mathbf{S_B}$ is the *between-class scatter matrix* already defined in (6) and $\mathbf{S_W}$ is called *within-class* (or intra-class) *scatter matrix*:

$$\mathbf{S_W} = \sum_{z \in \mathcal{Z}} \sum_{i=1}^{N_z} (\mathbf{x}_i^z - \overline{\mathbf{x}}^z)(\mathbf{x}_i^z - \overline{\mathbf{x}}^z)^{\mathsf{T}}. \tag{15}$$

Remark 1. Let $\mathbf{S}$ be the the global covariance matrix of data, also called *total scatter matrix*, defined in (5); we have the following relationship between $\mathbf{S_W}, \mathbf{S_B}$ and $\mathbf{S}$:

$$\mathbf{S} = \frac{1}{N}(\mathbf{S_W} + \mathbf{S_B}) \ . \tag{16}$$

It can be shown that the vector $\boldsymbol{\alpha}_1$ which maximizes (14) must satisfy $\mathbf{S_B} \boldsymbol{\alpha}_1 = \lambda \mathbf{S_W} \boldsymbol{\alpha}_1$, for a constant λ, *i.e.* has to be an eigenvector of $\mathbf{S_W^{-1} S_B}$. Moreover, for any eigenvector $\boldsymbol{\alpha}$ of $\mathbf{S_W^{-1} S_B}$, with associated eigenvalue λ, the Rayleigh quotient equals such a λ:

$$\frac{\boldsymbol{\alpha}^{\mathsf{T}} \mathbf{S_B} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^{\mathsf{T}} \mathbf{S_W} \boldsymbol{\alpha}} = \lambda \ . \tag{17}$$

Then, among all eigenvectors of $\mathbf{S_W^{-1}S_B}$, $\boldsymbol{\alpha}_1$ must be the leading one.

The computation of the eigenvectors of $\mathbf{S_W^{-1}S_B}$ is known under the name of *generalized eigenvector problem*. The difficulty here comes from the fact that $\mathbf{S_W^{-1}S_B}$ is not guaranteed to be symmetric. Due to this non-symmetry, $\boldsymbol{\alpha}_1$ and the others eigenvectors do not form an orthonormal basis for $\mathbb{R}^D$, but they are anyway useful for classifications scopes, as in SCA. Let us refer to them as *Linear Discriminant Components* (LDCs for short); as for PCs we consider them sorted in decreasing order with respect to their associated eigenvalue, which gives a score for their informativeness, see (17). Analogously to the PCA, the LDA provides a natural dimensionality reduction: one can project the data over the C first LDCs. As for PCA, this choice might not be optimal when applying this reduction to side-channel traces. For the sake of comparison, all the selection methods proposed for the PCA (EGV, IPR and ELV) will be tested in association to the LDA as well.

In the following subsection we will present a well-known problem that affects the LDA in many practical contexts, and describe four methods that circumvent such a problem, with the intention to test them over side-channel data.

4.1   The Small Sample Size Problem

In the special case in which the matrix $\mathbf{S_B}$ is invertible, the generalized eigenvalue problem is convertible in a regular one, as in [22]. On the contrary, when $\mathbf{S_B}$ is singular, the simultaneous diagonalization is suggested to solve such a problem [11]. In this case one can take advantage by the singularity of $\mathbf{S_B}$ to apply the computational trick proposed by Archambeau et al., see Sec. (3.2), since at most $r = \text{rank}(\mathbf{S_B})$ eigenvectors can be found.

If the singularity of $\mathbf{S_B}$ does not affect the LDA dimensionality reduction, we cannot say the same about the singularity of $\mathbf{S_W}$: SCA and Pattern Recognition literatures point out the same drawback of the LDA, known as the *Small Sample Size problem* (SSS for short). It occurs when the total number of acquisitions N is less than or equal to the size D of them.[4] The direct consequence of this problem is the singularity of $\mathbf{S_W}$ and the non-applicability of the LDA.

If the LDA has been introduced relatively lately in the SCA literature, the Pattern Recognition community looks for a solution to the SSS problem at least since the early nineties. We browsed some of the proposed solutions and chose some of them to introduce, in order to test them over side channel traces.

---

[4] It can happen for example when attacking an RSA implementation, where the acquisitions are often huge (of the order of 1,000,000 points) and the number of measurements may be small when the SNR is good, implying that a good GE can be achieved with a small N.

Fisherface Method The most popular among the solutions to SSS is the so-called *Fisherface* method[5] [3]. It simply relies on the combination between PCA and LDA: a standard PCA dimensionality reduction is performed to data, making them pass from dimension D to dimension $N - |\mathcal{Z}|$, which is the general maximal rank for $\mathbf{S_W}$. In this reduced space, $\mathbf{S_W}$ is very likely to be invertible and the LDA therefore applies.

$\mathbf{S_W}$ Null Space Method It has been introduced by Chen et al. in [6] and exploits an important result of Liu et al. [15] who showed that Fisher's criterion (14) is equivalent to:

$$\boldsymbol{\alpha}_1 = \mathrm{argmax}_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}^\intercal \mathbf{S_B} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\intercal \mathbf{S_W} \boldsymbol{\alpha} + \boldsymbol{\alpha}^\intercal \mathbf{S_B} \boldsymbol{\alpha}} \ . \tag{18}$$

The authors of [6] point out that such a formula is upper-bounded by 1, and that it achieves its maximal value, *i.e.* 1, if and only if $\boldsymbol{\alpha}$ is in the null space of $\mathbf{S_W}$. Thus they propose to first project data onto the null space of $\mathbf{S_W}$ and then to perform a PCA, *i.e.* to select as LDCs the first $|\mathcal{Z}| - 1$ eigenvectors of the between-class scatter matrix of data into this new space. More precisely, let $Q = [\mathbf{v}_1, \ldots, \mathbf{v}_{D-\mathrm{rank}(\mathbf{S_W})}]$ be the matrix of vectors that span the null space of $\mathbf{S_W}$. [6] proposes to transform the data $\mathbf{x}$ into $\mathbf{x}' = QQ^\intercal \mathbf{x}$. Such a transformation maintains the original dimension D of the data, but let the new within-class matrix $\mathbf{S'_W} = QQ^\intercal \mathbf{S_W} QQ^\intercal$ be the null $D \times D$ matrix. Afterwards, the method looks for the eigenvectors of the new between-class matrix $\mathbf{S'_B} = QQ^\intercal \mathbf{S_B} QQ^\intercal$. Let U be the matrix containing its first $|\mathcal{Z}| - 1$ eigenvectors: the LDCs obtained via the $\mathbf{S_W}$ null space method are the columns of $QQ^\intercal U$.

Direct LDA As the previous, this method, introduced in [24], privileges the low-ranked eigenvectors of $\mathbf{S_W}$, but proposes to firstly project the data onto the rank space of $\mathbf{S_B}$, arguing the fact that vectors of the null space of $\mathbf{S_B}$ do not provide any between-class separation of data. Let $D_B = V^\intercal \mathbf{S_B} V$ be the diagonalization of $\mathbf{S_B}$, and let $V^\star$ be the matrix of the eigenvectors of $\mathbf{S_B}$ that are not in its null space, *i.e.* whose eigenvalues are different from zero. Let also $D_B^\star$ denotes the matrix $V^{\star\intercal} \mathbf{S_B} V^\star$; transforming the data $\mathbf{x}$ into $D_B^{\star 1/2} V^{\star\intercal} \mathbf{x}$ makes the between-class variance to be equal to the $(|\mathcal{Z}| - 1) \times (|\mathcal{Z}| - 1)$ identity matrix. After this transformation the within-class variance assumes the form $\mathbf{S'_W} = D_B^{\star 1/2} V^{\star\intercal} \mathbf{S'_W} V^\star D_B^{\star 1/2}$. After storing the C lowest-rank eigenvectors in a matrix $U^\star$, the LDCs obtained via the Direct LDA method are the columns of $V^\star D_B^{\star 1/2} U^{\star\intercal}$.

$\mathbf{S_T}$ Spanned Space Method The last variant of LDA that we consider has been proposed in [13] and is actually a variant of the Direct LDA: instead of

---

[5] The name is due to the fact that it was proposed and tested for face recognition scopes.

removing the null space of $\mathbf{S_B}$ as first step, this method removes the null space of $\mathbf{S_T} = \mathbf{S_B} + \mathbf{S_W}$. Then, denoting $\mathbf{S'_W}$ the within-class matrix in the reduced space, the reduced data are projected onto its null space, i.e. are multiplied by the matrix storing by columns the eigenvectors of $\mathbf{S'_W}$ associated to the null eigenvector, thus reduced again. A final optional step consists in verifying whether the between-class matrix presents a non-trivial null-space after the last projection and, in this case, in effectuating a further projection removing it.

Remark 2. Let us remark that, from a computational complexity point of view (see [13] for a deeper discussion), the least time-consuming procedure among the four proposed is the Direct LDA, followed by the Fisherface and the $\mathbf{S_T}$ Spanned Space Method, that have a similar complexity. The $\mathbf{S_W}$ Null Space Method is in general much more expensive, because the task of removing the $\mathbf{S_W}$ null space requires the actual computation of the $(D \times D)$-dimensional matrix $\mathbf{S_W}$, *i.e.* the computational trick proposed by Archambeau et al. [1], see Sec. 3.2 is not applicable. On the contrary, the other three methods take advantage of such a procedure, reducing drastically their complexity.

5 Experimental Results

In this section we compare the different extractors provided by the PCA and the LDA in association with the different techniques of components selection. Defining an universal criterion to compare the different extractors would not make sense since the latter one should encompass a lot of parameters, sometimes opposite, that vary according to the context (amount of noise, specificity of the information leakage, nature of the side channel, etc.). For this reason we will choose to split our comparisons according to four contexts depending on the final goal pursued by the attacker:

1. *Minimize N*: achieve $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ with the minimal number of attack traces, with β a fixed threshold, common to the four goals
2. *Minimize N'*: achieve $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ with the minimal number of profiling traces
3. *Minimize C*: achieve $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ reducing as much as possible the size of the extracted traces
4. *Minimize #PoI*: achieve $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ exploiting the minimal number of original trace points.

An attack scenario has been defined for each of the goals above: in each one, three of the four parameters $N, N', C, \#PoI$ are fixed and one varies. For those in which N' is fixed, the value of N' is chosen high enough to avoid the SSS problem, and the extensions of LDA presented in Sec. 4.1 are not evaluated. [6] This choice of N' will imply that the LDA is always performed in a favourable situation, which makes expect the LDA to be particularly efficient for these experiments. Consequently, for the scenarios in which N' is high, our goal is to study whether the PCA can be made almost as efficient as the LDA thanks to the component selection methods discussed in Sec. 3.3.

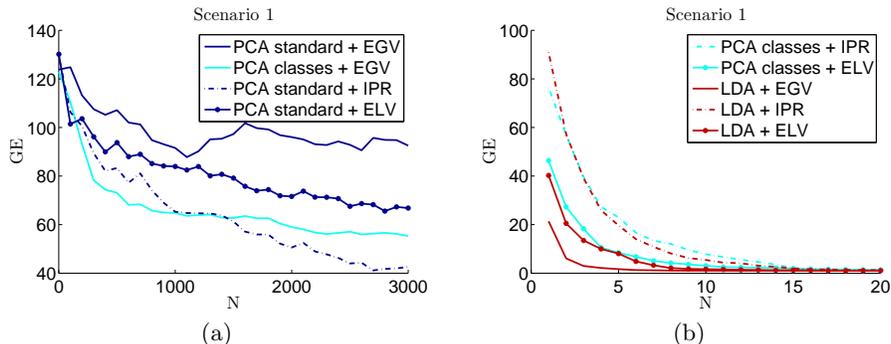[6] This study is let open for an extended version of this paper.

Fig. 4: Guessing Entropy as function of the number of attack traces for different extraction methods. All Guessing Entropies are estimated as the average rank of the right key over 100 independent experiments.

The testing adversary. Our testing adversary attacks an 8-bit AVR microprocessor Atmega328P and acquires power-consumption traces via the ChipWhisperer platform [19].[7] The target device stores a secret 128-bit key and performs the first steps of an AES: the loading of 16 bytes of the plaintext, the AddRoundKey step and the AES Sbox. It has been programmed twice: two different keys are stored in the device memory during the acquisition of the profiling and of the attack traces, to simulate the situation of two identical devices storing a different secret. The size D of the traces equals 3996. The sensitive target variable is the output of the first Sbox processing, but, since the key is fixed also during the profiling phase, and both Xor and Sbox operations are bijective, we expect to detect three interesting regions (as those high-lighted by PCs 4, 5 and 6, in Fig. 3): the reading of the first byte of the plaintext, the first AddRoundKey and the first Sbox. We consider an *identity classification* leaking function (i.e. we make minimal assumption on the leakage function), which implies that the 256 possible values of the Sbox output yields to 256 classes. For each class we assume that the adversary acquires the same number N_p of traces, *i.e.* $N' = N_p \times 256$. After the application of the extractor ε, the trace size is reduced to C. Then the attacker performs a Bayesian Template Attack [5], using C-variate Gaussian templates. This choice comes from the information-theoretic optimality of such an attack which, exploiting the maximum likelihood parameters estimation, yields to an unbiased comparison between the extractors.

Scenario 1. To analyse the dependence between the extraction methods presented in Sections 3 and 4 and the number of attack traces N needed to achieve a given GE, we fixed the other parameters as follows: $N_p = 50$ ($N' = 50 \times 256$), $C = 3$ and $\#PoI = 3996$ (all points are allowed to participate in the building of the PCs and of the LDCs). The experimental results, depicted in Fig. 4(a)-(b),

---

[7] This choice has been done to allow for reproducibility of the experiments.

show that the PCA standard method has very bad performances in SCA, while the LDA outperforms the others. Concerning the class-oriented PCA, we observe that its performance is close to that of LDA when combined with the selection methods ELV (which performs best) or IPR.

Scenario 2. Now we test the behaviour of the extraction methods as function of the number N_p of available profiling traces per class. The number of components C is still fixed to 3, $\#PoI = 3996$ again and the number of attack traces is $N = 100$. This scenario has to be divided into two parts: if $N_p \leq 15$, then $N' < D$ and the SSS problem occurs. Thus, in this case we test the four extensions of LDA presented in Sec. 4.1, associated to either the standard selection, to which we abusively refer as EGV, [8] or to the IPR selection. We compare them to the class-oriented PCA associated to EGV, IPR or ELV. The ELV selection is not performed for the techniques extending LDA, since for some of them the projecting LDCs are not associated to some eigenvalues in a meaningful way. On the contrary, if $N_p \geq 16$ there is no need to approximate the LDA technique, so the classical one is performed. Results for this scenario are shown in Fig. 5. It may be noticed that the combinations class-oriented PCA + ELV/IPR select exactly the same components, for our data, see Fig. 5(e) and do not suffer from the lack of profiling traces. They are slightly outperformed by the $\mathbf{S_W}$ Null Space method associated with the EGV, see Fig.5(d). The Direct LDA (Fig. 5(b)) method also provides a good alternative, while the other tested methods do not show a stable behaviour. The results in absence of the SSS problem (Fig.5(f)) confirm that the standard PCA is not adapted to SCA, even when provided with more profiling traces. It also shows that among class-oriented PCA and LDA, the class-oriented PCA converges faster.

Scenario 3. Let C be now variable and let the other parameters be fixed as follows: $N = 100, N_p = 200, \#PoI = 3996$. Looking at Fig. 6, we might observe that the standard PCA might actually well perform in SCA context if provided with a larger number of kept components; on the contrary, a little number of components suffices to the LDA. Finally, keeping more of the necessary components does not worsen the efficiency of the attack, which allows the attacker to choose C as the maximum value supported by his computational means.

Remark 3. In our experiments the ELV selection method only slightly outperforms the IPR. Nevertheless, since it relies on more sound and more general observations, *i.e.* the maximization of explained variance concentrated into few points, it is likely to be more robust and less case-specific. For example, in Fig. 5(f) it can be remarked that while the class-oriented PCA + ELV line keeps constant on the value 0 of guessing entropy, the class-oriented PCA + IPR is sometimes higher than 0.

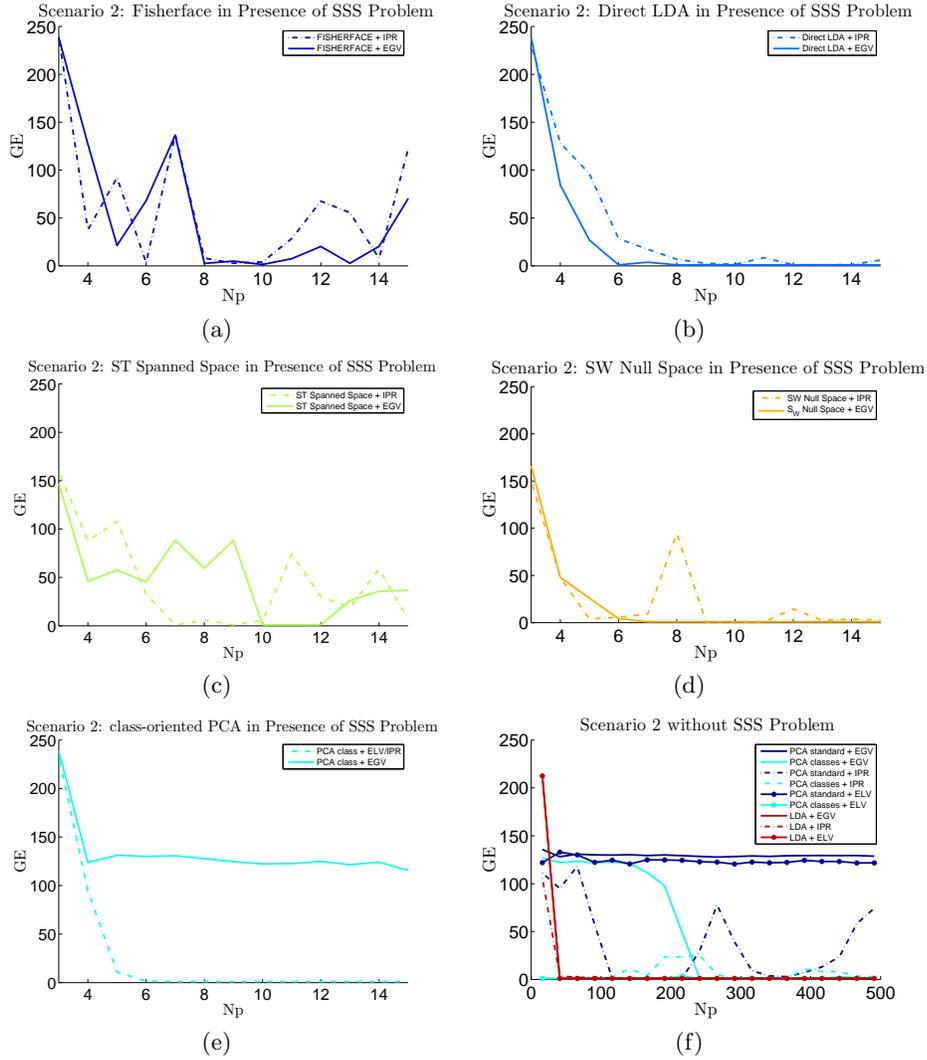[8] It consists in keeping the C first LDCs (the C last for the Direct LDA)

Fig. 5: Guessing Entropy as function of the number of profiling traces. Figures (a)-(d): methods extending the LDA in presence of SSS problem; Figure (e): class-oriented PCA in presence of the SSS problem; Figure (f): number of profiling traces high enough to avoid the SSS problem.
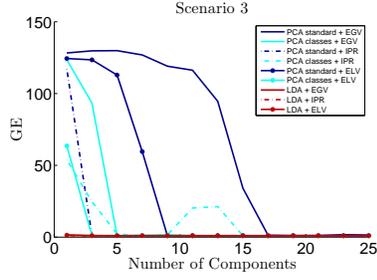
Fig. 6: Guessing Entropy as function of the number of the traces size after reduction

			Parameter to minimize		
Method	Selection	N	N' (SSS)	N' ($\neg$SSS)	C
PCA standard	EGV	-		-	-
PCA standard	ELV	-		-	-
PCA standard	IPR	-		-	+
PCA class	EGV	-	-	-	-
PCA class	ELV	+	★	★	+
PCA class	IPR	+	★	+	-
LDA	EGV	★		+	★
LDA	ELV	+		+	★
LDA	IPR	+		+	★
$\mathbf{S_W}$ Null Space	EGV		★		
$\mathbf{S_W}$ Null Space	IPR		+		
Direct LDA	EGV		★		
Direct LDA	IPR		+		
Fisherface			-		
$\mathbf{S_T}$ Spanned Space			-		

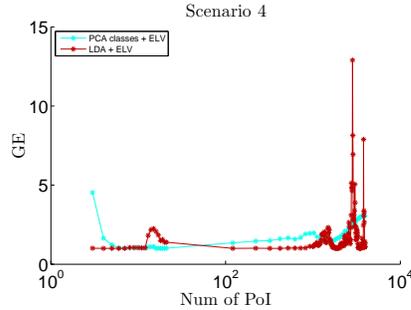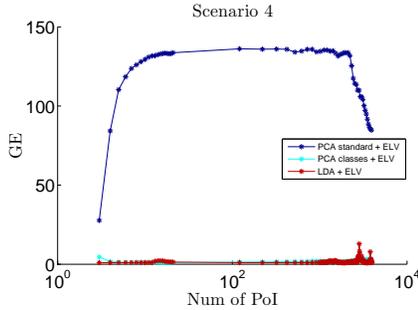**Table 1:** Overview of extractors performances in tested situations.

Fig. 7: Guessing Entropy as function of the number of time samples contributing to the extractor computation.

An overview of the results of our comparison in scenarios 1, 2 and 3 is depicted in Table 1: depending on the adversary purpose, given by the parameter to minimize, a ★ denotes the best method, a + denotes a method with performances close to those of the best one and a − is for methods that show lower performances. Techniques introduced in this paper are highlighted by a grey background. For example we remark that the class-oriented PCA takes advantage of the association with our ELV selection of components, achieving optimal performances when the goal is to minimize the number of profiling traces N'. As expected, when there are no constraints over N', the LDA outperforms the other methods; however, even in this case which is very favourable to the LDA, the class-oriented PCA equipped with the ELV selection has an efficiency which is close to that of the LDA.

Scenario 4. This is the single scenario in which we allow the ELV selection method to not only select the components to keep but also to modify them, keeping only some coefficients within each component, setting the other ones to zero. We select pairs *(component, time sample)* in decreasing order of the ELV

values, allowing the presence of only $C = 3$ components and $\#PoI$ different times samples: *i.e.*, we impose that the matrix A defining the extractor (see (1)) has $C = 3$ rows (storing the 3 chosen components, transposed) and exactly $\#PoI$ non-zero columns. Looking at Fig. 7 we might observe that the LDA allows to achieve the maximal guessing entropy with only 1 PoI in each of the 3 selected components. Actually, adding PoIs worsen its performances, which is coherent with the assumption that the vulnerable information leaks in only a few points. Such points are excellently detected by the LDA. Adding contribution from other points raises the noise, which is then compensated by the contributions of further noisy points, in a very delicate balance. Such a behaviour is clearly visible in standard PCA case: the first 10 points considered raise the level of noise, that is then balanced by the last 1000 points.

6 Conclusions

In this paper we studied and compared two well-known techniques to construct extractors for side-channel traces, the PCA and the LDA. The LDA method is more adequate than the PCA one, thanks to its class-distinguishing asset, but more expensive and not always available in concrete situations. We deduced from a general consideration about side-channel traces, i.e. the fact that for secured implementations the vulnerable leakages are concentrated into few points, a new methodology for selecting components, called ELV. We showed that the class-oriented PCA, equipped with the ELV, achieves performances close to those of the LDA, becoming a cheaper and valid alternative to the LDA. Being our core consideration very general in side-channel context, we believe that our results are not case-specific. Finally, among other alternatives to the LDA in presence of SSS problem proposed in Pattern Recognition literature, we showed that the Direct LDA and the $\mathbf{S_W}$ Null Space Method are promising, as well.

References

1. C. Archambeau, E. Peeters, F.-X. Standaert, and J.-J. Quisquater. Template attacks in principal subspaces. In Louis Goubin and Mitsuru Matsui, editors, *CHES 2006*, volume 4249 of *LNCS*, pages 1–14. Springer Berlin Heidelberg, 2006.
2. 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. In Orr Dunkelman, editor, *CT-RSA 2012*, volume 7178 of *LNCS*, pages 383–397. Springer Berlin Heidelberg, 2012.
3. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, July 1997.
4. N. Bruneau, S. Guilley, A. Heuser, D. Marion, and O. Rioul. Less is more: Dimensionality reduction from a theoretical perspective. In T. Güneysu and H. Handschuh, editors, *CHES 2015*, pages 22–41. Springer, 2015.
5. S. Chari, J. R. Rao, and P. Rohatgi. Template attacks. In Burton S. Kaliski, Cetin K. Koc, and Christof Paar, editors, *CHES 2002*, volume 2523 of *LNCS*, pages 13–28. Springer Berlin Heidelberg, 2003.

6. L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713 – 1726, 2000.

7. O. Choudary and M. G. Kuhn. Efficient stochastic methods: Profiled attacks beyond 8 bits. *IACR Cryptology ePrint Archive*, 2014.

8. O. Choudary and M. G. Kuhn. Efficient template attacks. In *CARDIS 2014*, pages 253–270. Springer, 2014.

9. T. Eisenbarth, C. Paar, and B. Weghenkel. Building a side channel based disassembler. In M. L. Gavrilova, C. J. K. Tan, and E. D. Moreno, editors, *Transactions on Computational Science X*, volume 6340 of *LNCS*, pages 78–99. Springer Berlin Heidelberg, 2010.

10. R. A Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8(4):376–386, 1938.

11. K. Fukunaga. *Introduction to Statistical Pattern Recognition (2Nd Ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

12. T. Guhr, A. Müller-Groeling, and H. A. Weidenmüller. Random-matrix theories in quantum physics: common concepts. *Physics Reports*, 299(4):189–425, 1998.

13. R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the small sample size problem of LDA. In *Pattern Recognition, 2002*, volume 3, pages 29–32 vol.3, 2002.

14. P. Karsmakers, B. Gierlichs, K. Pelckmans, K. De Cock, J. Suykens, B. Preneel, and B. De Moor. Side channel attacks on cryptographic devices as a classification problem. Technical report, COSIC technical report, 2009.

15. K. Liu, Y.-Q. Cheng, and J.-Y. Yang. A generalized optimal set of discriminant vectors. *Pattern Recognition*, 25(7):731–739, 1992.

16. J. L. Massey. Guessing and entropy. In *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*, page 204. IEEE, 1994.

17. D. Mavroeidis, L. Batina, T. van Laarhoven, and E. Marchiori. PCA, eigenvector localization and clustering for side-channel attacks on cryptographic hardware devices. In P. A. Flach, T. De Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7523 of *LNCS*, pages 253–268. Springer Berlin Heidelberg, 2012.

18. N. Nisan and D. Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.

19. C. O'Flynn and Z. D. Chen. Chipwhisperer: An open-source platform for hardware embedded security research. In *COSADE 2014*, pages 243–260. Springer, 2014.

20. TELECOM ParisTech. Dpa contest 4. http://www.DPAcontest.org/v4/.

21. R. Specht, J. Heyszl, M. Kleinsteuber, and G. Sigl. Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution em measurements. In S. Mangard and A. Y. Poschmann, editors, *COSADE 2015*, volume 9064 of *Lecture Notes in Computer Science*, pages 3–19. Springer International Publishing, 2015.

22. F.-X. Standaert and C. Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES 2008*, volume 5154 of *LNCS*, pages 411–425. Springer Berlin Heidelberg, 2008.

23. F.-X. Standaert, T. G. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 443–461. Springer Berlin Heidelberg, 2009.

24. H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001.