Work in Progress Paper

No Author Given

No Institute Given

Abstract. Advanced Side Channel Analyses make use of dimensionality reduction techniques to reduce both the memory and timing complexity of the attack. 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 also raised some issues implied by the Side-Channel context. The PCA provides a set of vectors (the *principalcomponents*) 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 we propose a new technique to automatize and ameliorate the selection of principal components, named *cumulative Explained Local Variable (ELV) selection*. Moreover we present some extensions of the LDA, available in less constraining situations. 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 cumulative 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.

1 Introduction

The measurement of the power consumption or of the electromagnetic irradiations occurring unavoidably during the execution of cryptographic algorithms in constrained electronic devices can reveal information about sensitive variables (*e.g.* cryptographic keys) handled during the computations. The Side Channel (SC) 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 information about interesting sensitive variables over several time samples. Reducing the dimensionality of the data is an important issue for Side-Channel Attacks (SCA). Considering the SC traces as column vectors $\mathbf{x}$ in $\mathbb{R}^D$, the compressing phase might be seen as the application of a function $\varepsilon\colon \mathbb{R}^D \to \mathbb{R}^C$, here called *extractor*.

In this paper we focus 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 algebra 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 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, 13], and in a *supervised* way, i.e. on traces grouped in classes and averaged [1, 7–9, 21], without that the difference of these two approaches, that achieve very different performances, were explicit. Another ambiguity in PCA concerns the choice of the components that must be kept after the dimension reduction: as also remarked by Specht et al. [20], some papers declare that the leading components are those that contain almost all the useful information [1], while others propose to discard the leading components [2]. Specht et al. compared, in a specific attack context, the results obtained by choosing different subsets of consecutive components, starting from some empirically chosen index, and concluded that for their data the optimal result is obtained by selecting a single component, the fourth one (with 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. In this paper we propose a new selection methodology, called *Cumulative ELV Selection*, and we argue about its soundness. Our reasonning is essentially based on the assumption that the leaking information is spread over a few time samples of each trace. This assumption has already been done by Mavroeidis et al. in [16], where the authors also proposed a components selection method. As we will see in this paper, the important difference between their proposal and ours is that we do not discard the information given by the eigenvalues associated to the PCA components.

The introduction of the LDA in SCA literature also revealed some issues: even if declared more meaningful and informative than the PCA method [4, 21], it is often set aside because of a practical constraint; it is subject to the so-called *Small Sample Size problem (SSS)*, i.e. it requires a number of observations (traces) which is higher than the dimension (size) D of them. In some contexts it might be an excessive requirement, which may become unacceptable in many practical contexts where the amount of observations is very limited. Many propositions to circumvent this problem have been made, especially by Pattern Recognition and Face Recognition communities [3, 6, 12, 23]. We find mandatory to test and compare such methods in Side-Channel context in order to draw fair conclusions about the practical application of LDA in SCA contexts.

The paper is organised as follows: in Section 2 we fix notations, recall preliminaries and set up a unified comparison framework to compare different extractors. 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 comparison are showed in Section 5, while conclusions and perspectives follow in Section 6.

2 Preliminaries and Comparison Framework

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. [22], 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 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:

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

The first set of traces, 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, 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* [15] and by the asymptotic guessing entropy, respectively defined as

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

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 PoI Research 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 [17]):

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

$$\begin{aligned} \mathbb{R}^D &\to \mathbb{R}^C \quad \text{with } D \leq C \\ \boldsymbol{x} &\mapsto \varepsilon_C(\boldsymbol{x}) \ . \end{aligned}$$

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. Such extractors can be defined by a $D \times C$ matrix containing as columns 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 $\left((\varepsilon(\mathbf{x}_{n'}))_{n'=1,\dots N'} , (\varepsilon((\mathbf{y}_n)))_{n=1,\dots N} \right)$, i.e. an adversary that applies ε as traces pre-processing phase. The extractor ε is an *effective extractor of PoIs 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.

2.3 Compare Effective Extractors

As we have seen in the introduction, the state of the art proposes different techniques to create extractors: a T-test followed by a thresholding, PCA, LDA, etc. A lot of these tools make use of the profiling traces to provide an extractor; thus, for such techniques the profiling traces are mandatory and cover the double role of feeding the method that generates the extractor, and of modelling the leakage function if the adversary makes use of a profiling stage. It might be interesting, but it is out of the scope of this paper, to study whether all these methods, eventually provided with enough profiling observations, construct effective extractors (meaning that there exists a threshold N, possible high, for which (4) is satisfied).

A much less theoretical issue, but very important from a practical point of view, is the comparison between the proposed methods. Seldom the extractors provided by different methods are equal or similar, and choosing which one is the best one according to the context (amount of noise, specificity of the information leakage, nature of the side channel, etc.) is not a trivial task, especially because a universal criterion to compare different extractors must encompass a lot of parameters. Since the aim of this paper is to effectuate a comparison between different extractors, and some new propositions, to construct extractors of PoI, we are obliged to focus on a given adversary, and to specify the different goals that may be pursued by our attackers.

The Four Criteria. The comparison of the tested methods is based on four criteria, that exploit the guessing entropy as an efficiency measure for an attack whose preprocessing coincides with the extractors to compare. For each criterion let us fix a common threshold β for such a guessing entropy, and all the adversary parameters but the one targeted by the specific criterion. We consider four criteria:

1. *Minimize* N: the best method is the one that achieves $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ with the minimal number of attack traces
2. *Minimize* N': the best method is the one that achieves $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ with the minimal number of profiling traces
3. *Minimize* C: the best method is the one that achieves $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ reducing as much as possible the size of the extracted traces
4. *Minimize the number of PoI*: the best method is the one that achieves $\mathrm{GE}_{\mathcal{A}'} \leq \beta$ exploiting the minimal number of original trace points.

The last criterion can be expressed, for the projecting methods, as the search for a projecting matrix which as much of null rows as possible. In this way the many time samples corresponding to these zero rows, never contribute to the computation of the projected samples. The meaning of this criterion comes from the assumption that only a few time samples leak vulnerable information, and one can be interested in precisely detect only those points, e.g. a security designer.

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, via singular value decomposition, as the eigenvectors of the empirical covariance matrix of data: given a set of data $(\mathbf{x}_i)_{i=1,\dots,N}$ the empirical covariance matrix is given by

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

where $\overline{\mathbf{x}}$ is the empirical mean of data. Let us denote by r the rank of S and by $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_r$ and $\lambda_1, \dots, \lambda_r$ 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 λ_i, for each 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 allows a dimensionality reduction that reinforces such information: that is projecting the data onto the C-dimensional subspace of $\mathbb{R}^D$ spanned by the C leading PCs.

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 $z \in \mathcal{Z}$. An SCA efficiency clearly 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. She 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, 21] 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 dimension 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:

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

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 $\mathbf{S} = \mathbf{M}\mathbf{M}^T$ 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 next section 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 such questions has been given [8], linked to the concept of *explained variance* (or *explained global variance*, EGV for short) of a PC $\boldsymbol{\alpha}_i$:

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

where r is the rank of the covariance matrix $\mathbf{S}$, and λ_j is the eigenvalue associated to the j-th PC $\boldsymbol{\alpha}_j$. The EGV 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 $\mathbf{S}$). By definition of EGV, the sum of all the PCs EGV is equal to 1; that is why this quantity is often multiplied by 100 and expressed as percentage. Choosing the PCs exploiting the EGV 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, tanking for granted that the optimal way to chose PCs is in their natural order. Nevertheless, 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, 20] 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 [19]; 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.

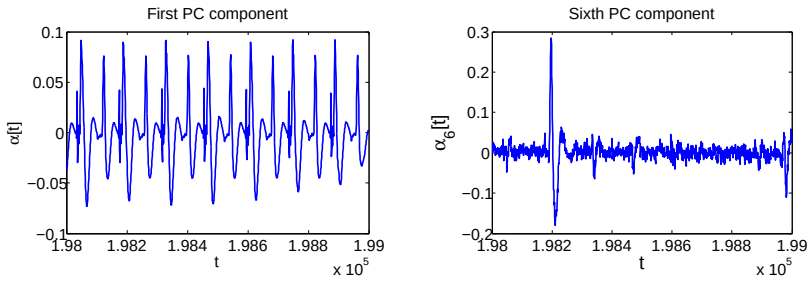

Fig. 1. First and sixth PCs in DPA contest v4 trace set

To the best of our knowledge, until now only one method adapted to SCA context has been proposed to automatically chose PCs [16] 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 [16] propose a measure to evaluate the eigenvectors *localization*.

It is called *Inverse Participation Ratio* (IPR) and is defined as follows:

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

The authors of [16] 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

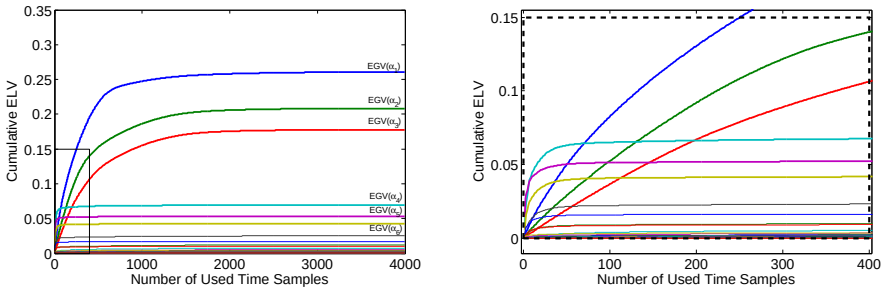

Fig. 2. Cumulative ELV trend of principal components. On the right a zoom of the plot on the left. Data acquisition described in Sec. (2.3).

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).

Let us start by giving some intuition behind our new concept. Thinking to the observations $\mathbf{x}^{\mathrm{T}}$, or to the class-averages $\overline{\mathbf{x}}^{\mathrm{T}}$ in class-oriented PCA case, as

realizations of a random variable $\mathbf{X}$, we have that λ_i is an estimator for the variance of the random variable $\mathbf{X} \cdot \boldsymbol{\alpha}_i$. Developing, we obtain

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

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

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

where $\mathbf{S}_j$ 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*

$$\text{ELV}(i, t) = \frac{\lambda_i \boldsymbol{\alpha}_i[t]^2}{\sum_{j=1}^{r} \lambda_j} = \text{EGV}(i)\boldsymbol{\alpha}_i[t]^2 \ . \tag{13}$$

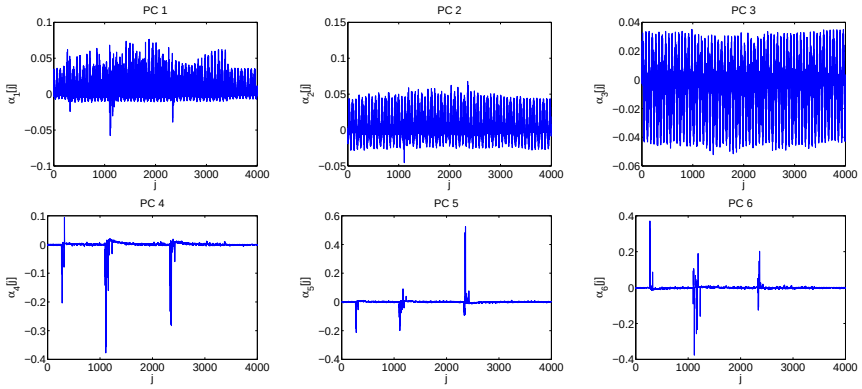

Fig. 3. The first six PCs. Acquisition campaign on an 8-bit AVR Atmega328P (see Sec. 5).

It may be observed that the sum over all the trace samples ELVs of a PC equals the EGV of the considered PC. If we operate such a sum for each PC in a cumulative way and in decreasing order over the coefficients starting from the one that holds the maximal ELV, we obtain a complete description of the trend followed by each component 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 4^{th}, the 5^{th} and the 6^{th} achieve a large part of their final EGVs very quickly (*i.e.* by adding the ELV contributions of much less time samples). This implies that the EGV of the 4^{th}, the 5^{th} and the 6^{th} 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 couples (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 select, within a selected PC, the non-significant points, *i.e.* those with a low ELV, to be setted 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 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) is another statistical tool for dimensionality reduction, which is theoretically more appropriate than PCA for classification problems as SCA, as already observed in [21]. 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*:

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

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*:

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

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:

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

Then, among all eigenvectors $\mathbf{S_W^{-1}S_B}$, $\boldsymbol{\alpha}_1$ 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 first C 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 [21]. 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 literature, points out the same drawback of the LDA, which is 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.[1] The direct consequence of this problem is the singularity of $\mathbf{S_W}$ and the non-applicability of the LDA.

If the LDA has seen 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 solution and chose some of them to introduce, in order to test them over side channel traces.

Fisherface Method The most popular among the solutions to SSS is the so-called *Fisherface* method[2] [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.

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

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

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 $Q = [\mathbf{v}_1, \ldots, \mathbf{v}_{D - \text{rank}(\mathbf{S_W})}]$ be the matrix of vectors that span the null space of $\mathbf{S_W}$. [6] proposes to transform the data $\mathbf{x}$ into $\mathbf{x}' = QQ^T\mathbf{x}$. Such a transformation maintains the original dimension D of the data, but let the new within-class matrix $\mathbf{S'_W} = QQ^T\mathbf{S_W}QQ^T$ be the null $D \times D$ matrix. Afterwards, the method looks for the eigenvectors of the new between-class matrix $\mathbf{S'_B} = QQ^T\mathbf{S_B}QQ^T$. 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^TU.

[1] 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.

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

Direct LDA As the previous, this method, introduced in [23] 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 $D_B = V^T \mathbf{S_B} V$ be the singular value decomposition 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 $V^{\star T} \mathbf{S_B} V^\star$; transforming the data $\mathbf{x}$ into ${D_B^\star}^{1/2} V^{\star T} \mathbf{x}$ 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 $\mathbf{S'_W} = {D_B^\star}^{1/2} V^{\star T} \mathbf{S'_W} V^\star {D_B^\star}^{1/2}$. After storing the C lowest-rank eigenvectors in a matrix $U^\star$, the LDCs obtained via the Direct LDA method are the columns of $V^\star {D_B^\star}^{1/2} U^{\star T}$. Observe that for the first projection, there is no need to compute the big $\mathbf{S_B}$ matrix, because the computational trick of Sec. 3.2 is applicable.

$\mathbf{S_T}$ Spanned Space Method The last variant of LDA that we consider has been proposed in [12] and is actually a variant of the Direct LDA: instead of 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 if 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.

5 Experimental Results

In this section we to effectuate a comparison between the different extractors provided by the PCA and the LDA in association with the different techniques of selection of components. We organized our experiments focusing on a given adversary, whose aim is to optimally choose an extraction method we respect to the four criteria presented in Sec. 2.3. Thus, we consider four scenarios: 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; then, the extensions of LDA presented in Sec. (4.1) are not evaluated.[3] As a consequence, for these three scenarios, we always perform the LDA in a favourable situation, which makes expect the LDA to be more efficient than the PCA in our experiments. Our goal is therefore only to study whether the PCA can be made almost as efficient than the LDA thanks to the component selection methods discussed in Sec. 3.3.

The testing adversary. Our testing adversary attacks an 8-bit AVR microprocessor Atmega328P and acquires power-consumption traces via the ChipWhis-

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

pered platform [18]. The target device stores a secret 128-bits key and performs the first steps of an AES: the reading 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 of the traces equals $D = 3996$. The target sensitive variable is the output of the first Sbox byte, 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: 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 gives birth 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.

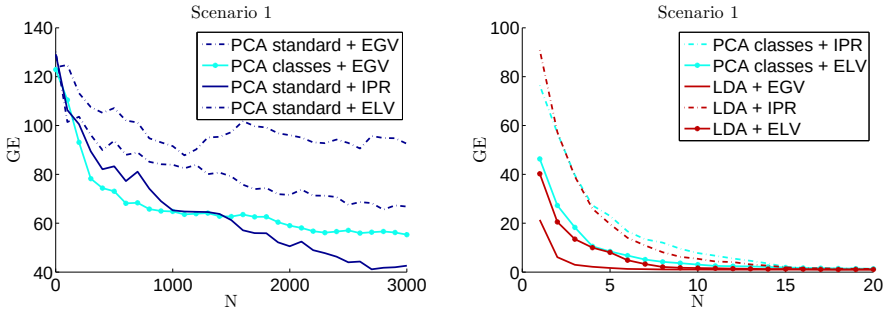

Fig. 4. Guessing Entropy as function of the number of attack traces, for different extraction methods

Scenario 1. To analyse the dependence between our extraction methods, presented in Sec. 3.3, and the number of attack traces N needed to achieve a given GE, we fixed the other parameters as follows: $N_p = 50$ ($N' = 50 * 256$), $C = 3$ and $\#PoI = 3996$ (all points are allowed to participate in the building of PCs and LDCs). The experimental results, depicted in Fig. 4, show that the PCA standard method has very bad performances in SCA, while the LDA is the method that performs best. Concerning the class-oriented PCA, we observe that its performance its comparable to that of LDA, when combined with the selection methods ELV (which performs best) or IPR.

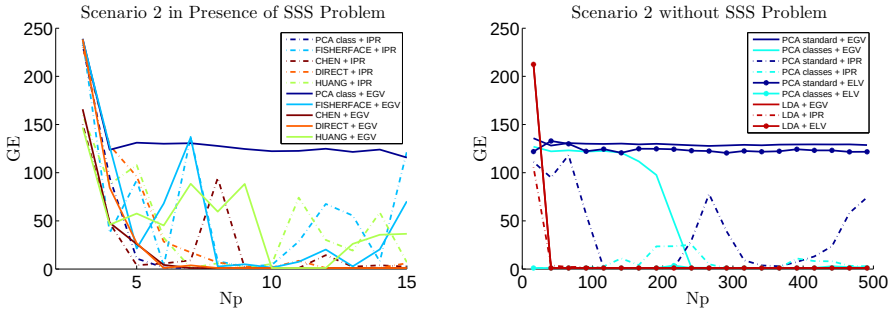

Fig. 5. Guessing Entropy as function of the number of profiling traces per class, for different extraction methods: on the left the LDA is substituted by its extensions to handle the SSS problem.

Scenario 2. Now we test the behaviour of the extraction methods, as function of the number N_p of profiling traces per class available. The number of components C is still fixed to 3, and $\#PoI = 3996$ again. 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 will test the four extensions of LDA presented in Sec. 4.1, associated to either the standard selection, to which we abusively refer as EGV,[4] and to the IPR selection. We compare them to the class-oriented PCA associated to the same selection methods. The ELV selection is not performed in this case because, for some of the techniques extending LDA, 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 LDA technique, so the classical one is performed. Results for this scenario are shown in Fig. 5. It may be noticed that the combination class-oriented PCA + IPR does not suffer the lack of profiling traces; anyway, it is outperformed by the Chen method in association to EGV. The Direct LDA 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 confirm that the standard PCA is not adapted to SCA, even when provided with more profiling traces. As expected, it also confirms that among class-oriented PCA and LDA, LDA 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 do not worsen the efficiency of the attack, which allows adversary to choose C as the maximum value supported by their computational means.

[4] It consists in keeping the first C LDCs, except for the Direct LDA, which asks to keep the last LDCs.

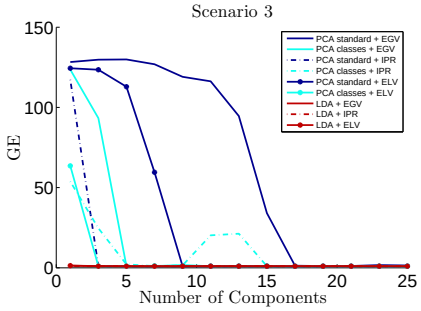

Fig. 6. Guessing Entropy as function of the number of the traces size after reduction. Figure on the right is a zoom of the one on the left.

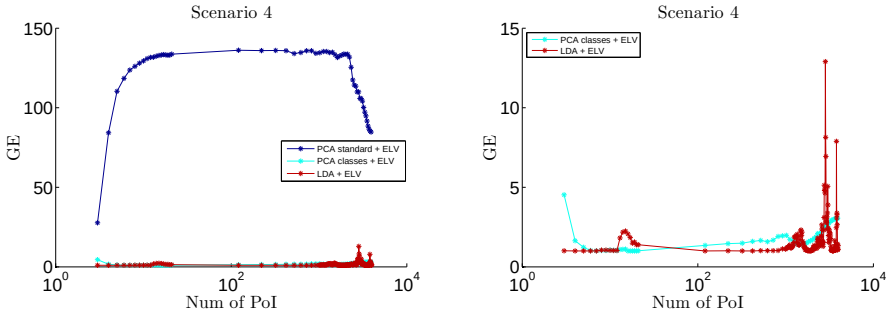

Fig. 7. Guessing Entropy as function of the number of original time samples the dimensionality reduction depends on.

Scenario 4. This is the only 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 to zero. Thus, we select couples *(component, time sample)* in order of decreasing ELV, allowing the presence of only $C = 3$ components and $\#PoI$ globally considered, i.e. we impose that the matrix containing the 3 selected components as columns has exactly $\#PoI$ rows different from the zero vector. 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, that are excellently detected by the LDA, and that adding contribution from other points raises the noise which is then compensated by the contributions of other 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.

		Parameter to minimize			
Method	Selection	N	N' (SSS)	N' (not 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	+		+	★
Chen	EGV		★		
Chen	IPR		+		
Direct LDA	EGV		★		
Direct LDA	IPR		+		
Fisherface			-		
Huang			-		

Table 1. Overview of extractors performances in tested situations. ★ = best method. + = performance comparable to the best method. − = lower performances.

6 Conclusion and Future Developments

In this paper we studied and compared two well-known techniques to construct extractors for SC traces, the PCA and the LDA. In our comparison framework we confirmed what expected by theoretical facts: the LDA method is much more adequate that the PCA one, thanks to its class-distinguishing asset. Aware of the PCA issue of choosing suitable components for SCA, and observed in two different real case contexts, we proposed a new selection method, based on the ELV notion. Thanks to this technique the class-oriented PCA can achieve in some cases (*e.g* with our test traces set) performances comparable to those of the LDA. Moreover, it remains a good alternative to LDA when the SSS problem occurs. Finally, among other proposed alternatives to the LDA in presence of the SSS problem, we show that the Direct LDA and the SW 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, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2006.
2. Lejla Batina, Jip Hogenboom, and JasperG.J. van Woudenberg. Getting more from pca: First results of using principal component analysis for extensive power analysis. In Orr Dunkelman, editor, *Topics in Cryptology CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 383–397. Springer Berlin Heidelberg, 2012.
3. Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, 1997.
4. N. Bruneau, S. Guilley, A. Heuser, D. Marion, and O. Rioul. Less is more: Dimensionality reduction from a theoretical perspective. In *17th Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015)*, to appear, 2015.

5. Suresh Chari, JosyulaR. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski, Cetin K. Koc, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer Berlin Heidelberg, 2003.
6. Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713 – 1726, 2000.
7. Omar Choudary and Markus G Kuhn. Efficient stochastic methods: Profiled attacks beyond 8 bits. *IACR Cryptology ePrint Archive*, 2014.
8. Omar Choudary and Markus G Kuhn. Efficient template attacks. In *Smart Card Research and Advanced Applications*, pages 253–270. Springer, 2014.
9. Thomas Eisenbarth, Christof Paar, and Bjorn Weghenkel. Building a side channel based disassembler. In MarinaL. Gavrilova, C.J.Kenneth Tan, and EdwardDavid Moreno, editors, *Transactions on Computational Science X*, volume 6340 of *Lecture Notes in Computer Science*, pages 78–99. Springer Berlin Heidelberg, 2010.
10. Ronald A Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8(4):376–386, 1938.
11. Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition (2Nd Ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
12. Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of lda. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 29–32 vol.3, 2002.
13. Peter Karsmakers, Benedikt Gierlichs, Kristiaan Pelckmans, Katrien De Cock, Johan Suykens, Bart Preneel, and Bart De Moor. Side channel attacks on cryptographic devices as a classification problem. Technical report, COSIC technical report, 2009.
14. Ke Liu, Yong-Qing Cheng, and Jing-Yu Yang. A generalized optimal set of discriminant vectors. *Pattern Recognition*, 25(7):731–739, 1992.
15. James L Massey. Guessing and entropy. In *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*, page 204. IEEE, 1994.
16. Dimitrios Mavroeidis, Lejla Batina, Twan van Laarhoven, and Elena Marchiori. PCA, eigenvector localization and clustering for side-channel attacks on cryptographic hardware devices. In PeterA. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7523 of *Lecture Notes in Computer Science*, pages 253–268. Springer Berlin Heidelberg, 2012.
17. Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.
18. Colin O'Flynn and Zhizhang David Chen. Chipwhisperer: An open-source platform for hardware embedded security research. In *Constructive Side-Channel Analysis and Secure Design*, pages 243–260. Springer, 2014.
19. TELECOM ParisTech. Dpa contest 4. http://www.DPAcontest.org/v4/.
20. Robert Specht, Johann Heyszl, Martin Kleinsteuber, and Georg Sig. Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution EM measurements. In *Sixth International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE 2015)*, 2015.
21. François-Xavier Standaert and Cedric Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware*

and Embedded Systems CHES 2008, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer Berlin Heidelberg, 2008.

22. François-Xavier Standaert, TalG. Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer Berlin Heidelberg, 2009.
23. Hua Yu and Jie Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001.