PCA, Eigenvector Localization and Clustering for Side-Channel Attacks on Cryptographic Hardware Devices

Dimitrios Mavroeidis, Lejla Batina, Twan van Laarhoven, and Elena Marchiori

Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands

Abstract. Spectral methods, ranging from traditional Principal Components Analysis to modern Laplacian matrix factorization, have proven to be a valuable tool for a wide range of diverse data mining applications. Commonly these methods are stated as optimization problems and employ the extremal (maximal or minimal) eigenvectors of a certain input matrix for deriving the appropriate statistical inferences. Interestingly, recent studies have questioned this "modus operandi" and revealed that useful information may also be present within low-order eigenvectors whose mass is concentrated (localized) in a small part of their indexes. An application context were localized low-order eigenvectors have been successfully employed is "Differential Power Analysis" (DPA). DPA is a well studied side-channel attack on cryptographic hardware devices (such as smart cards) that employs statistical analysis of the device's power consumption in order to retrieve the secret key of the cryptographic algorithm. In this work we propose a data mining (clustering) formulation of the DPA process and also provide a theoretical model that justifies and explains the utility of low-order eigenvectors. In our data mining formulation, we consider that the key-relevant information is modelled as a "low-signal" pattern that is embedded in a "high-noise" dataset. In this respect our results generalize beyond DPA and are applicable to analogous low-signal, hidden pattern problems. The experimental results using power trace measurements from a programmable smart card, verify our approach empirically.

1 Introduction

Spectral Clustering [17] is a popular data mining paradigm that is currently considered both as an effective practical tool for data analysis and also an active area of research. The common characteristic of Spectral Clustering methods is that they employ the spectrum (eigenvectors and eigenvalues) of certain input matrices as a central component for deriving the required data inferences. For instance, Spectral Clustering for Normalized Cut optimization [17] employs the (extremal) eigenvectors of the normalized Laplacian matrix for deriving the data clustering structure. Due to their common association with Trace optimization problems, most spectral methods employ extremal (maximal or minimal) eigenvectors for drawing the necessary inferences.

Recent studies have illustrated that low-order (not extremal) eigenvectors may also be useful for detecting interesting structures within data. More precisely, in [8, 7] it was demonstrated that low-order eigenvectors that are localized (i.e. contain a large number

of zero or near-zero entries), can be effectively used for detecting small, local and well connected clusters. In a global sense these clusters may be difficult to detect since they are small and do not correspond to a clustering objective optimum. Interestingly, the concept of eigenvector (or more generally eigenfunction) localization is well known in several scientific applications such as Quantum Mechanics, DNA data and astronomy (see [8, 7, 18] and references therein).

Low-order, localized eigenvectors have been successfully employed in the context of Differential Power Analysis (DPA) [16, 5]. DPA is a side-channel attack which involves statistical analysis of a cryptographic device's power consumption. Cryptographic algorithms are nowadays typically implemented in software or hardware on (small) physical devices that interact with and are influenced by their environments. These devices provide unintended output channels, called side channels. In general, these types of information leakages may be linked either to the types of operations that the cryptographic algorithm is performing, or to the data, i.e., the keys being processed. This makes them a very powerful tool for trying to extract the secret key. Using DPA, an adversary can obtain secret keys by analyzing power consumption measurements from multiple cryptographic operations performed by a vulnerable smart card or other device.

In this work we propose a novel data mining (clustering) formulation of the DPA process and also introduce a theoretical model that illustrates the utility and semantics of low-order eigenvectors. In our data mining formulation, we consider that the key-relevant information is modelled as a "low-signal" pattern that is embedded in a "high-noise" dataset. The essential property that allows for the detection of these low-signal patterns, is that they depend on a very small number of features. The embedding of such patterns in a high-noise dataset will result in the localization of the relevant eigenvectors thus making these patterns detectable, even though they are "buried" in the middle or lower part of the matrix's spectrum. In this respect our results generalize beyond DPA and are applicable to analogous low-signal/hidden pattern problems. Consequently, we employ Inverse Participation Ration (IPR) measure that can effectively select the appropriate low-order eigenvectors even in cases where standard countermeasures, that aim in hiding the key-relevant patterns from the device's power consumption, are employed.

2 Differential Power Analysis and Clustering

2.1 Differential Power Analysis (DPA)

Small embedded devices such as smart cards and mobile phones have become omnipresent in our lives as they are used daily in financial transactions, access control, mobile payments, etc. Hence, the security of such devices relies on the security protocols that are founded on standard cryptographic algorithms and in particular on their implementations. The weaknesses of cryptographic implementations are typically explored via side-channel information, e.g. power consumption of a device, which can be monitored and statistically analyzed to retrieve cryptographic secret keys of the device.

Side-channel attacks are the main security threat for smart cards since the first academic publications by Kocher et al. [14, 15]. Different sources of side-channel data,

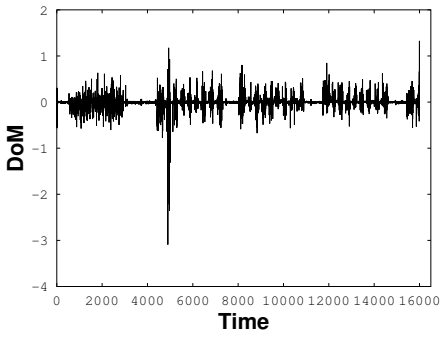


Fig. 1. Difference between the cluster centroids (*y*-axis) for the correct key of 1DES_wo_ctrmsr dataset described in Section 6

such as electromagnetic emanation [21, 12], timing [14], sound, and temperature have been used for successful side-channel attacks (for a general overview see e.g. [19]). Nowadays, every device used for the applications of embedded security e.g. a smart card, an RFID tag, a mobile phone is considered to be a suitable target for the side-channel attackers to recover the secret key in indirect way. Successful attacks were performed on some (still) widely used commercial devices with security functionalities such as the KeeLoq-based remote keyless entry systems, the contactless Mifare DESFire card and the most recent attack on Atmel CryptoMemory cards [2].

In the late 90's Kocher et al. showed that, by measuring the power consumption of a smart card, one can retrieve information about the secret keys inside a tamper-proof device. The main observation is that the device's power consumption depends on the data being processed. This makes power traces correlated to certain intermediate variables, which, when directly depending on some key bits and some known data e.g. plaintext, allow for key recovery by simply using basic statistical tools such as the Distance of Means (DoM) test. This original approach is called Differential Power Analysis (DPA) [15]. The details of the DoM test can be found in [15], but as a general description, DoM considers a grouping (clustering) of the collected power traces for each candidate key and then selects the correct key (best clustering) when a clear peak is observed in the difference between the cluster centroids. DoM distinguisher is illustrated in Figure 1, where (in the *y* axis) the difference between the cluster centroids for the correct key is presented (we used 1DES_wo_ctrmsr dataset described in Section 6). The *x* axis contains the features (which is time in our application context, since power traces are collected over discrete time intervals).

Other known distinguishers include Pearson correlation coefficient (Correlation Power Analysis (CPA) [6]), Spearman's rank correlation [3], Mutual Information Analysis (MIA) [13] etc. Ideas from unsupervised learning were used for a distinguisher based on cluster analysis, so called Differential Cluster Analysis (DCA) [4]. CPA is still considered the first choice of the attacker, especially in the cases where power consumption is linearly dependent on Hamming weight (or distance) of the data. Although Pearson

DPA with DoM distinguisher	Clustering
Possible Subkeys	Each subkey corresponds to a clusterings of Power Traces (in two groups)
DoM measure	Difference between cluster centroids
Correct Key≡ Peak in DoM	Best clustering ≡ Largest absolute value distance between centroids in just one or for a small number of features

Table 1. Analogies between DPA and Clustering

correlation solely tests for equidistance of cluster centroids, i.e. linearity, this limitation is proven to be not so restrictive in most of the devices used today [3].

2.2 DPA as Clustering

The acute reader may have already observed certain analogies between DoM based DPA (i.e. DPA with the Distance of Means distinguisher) and clustering. The analogy is based on the fact that the DoM distinguisher defines a clustering (in two groups) of the set of power traces for each possible key and consequently determines the secret key using a quality measure that is based on the difference between the two cluster centroids. More precisely, the secret key (equiv. best clustering) is determined by the largest peak in the difference between the cluster centroids, i.e. the largest absolute value distance (between the cluster centroids) in a small number of features. The analogies between DPA (with DoM distinguisher) and clustering are summarized in Table 1.

Up to this point, the clustering perspective of DoM based DPA is simply a "terminology rephrase" and the novel algorithmic insights that it provides may not be directly evident. As we will analyze lateron in detail, the power of the clustering formulation of DPA can be exploited in the cases where noise prevents DoM from identifying the correct key. The possible failures of DoM can be attributed to input data noise or to the use of countermeasure techniques that aim in hiding the key-relevant information from the power consumption. In the presence of high noise levels (that can be possibly due to the use of countermeasures), the input data must be appropriately pre-processed/analyzed such that the DoM distinguisher identifies the correct key. This can be achieved if we appropriately identify the structure of the signal that we wish to preserve in the data and the structure of the noise that we wish to remove.

Based on the clustering perspective of DPA we can define the following noise/signal formulation:
Signal→Clustering structures that "depend" on few features
Noise→Clustering structures that "depend" on many features

Intuitively the term "depend on few features" attempts to capture the fact that the DoM distinguisher uses the peak of the absolute value difference between the two cluster centroids *in a small subset of the available features*. Formally the term "depend on few features" is defined in Section 3.3 and it is taken to mean that the clustering objective does not change if we remove many features.

Using the clustering perspective of DPA and the afore noise/signal formulation we can state that the goal of a successful data mining algorithm in this application context is *to identify the clustering structures that "depend" on few features, or analogously to remove the clustering structures that "depend" on many features*. Naturally, due to noise or data-structure reasons, there may exist other clusterings that "depend" on a small number of features but are not relevant to the correct key. However, as the empirical results verify in Section 6 the effective identification of these clustering structures (even at the potential cost of including non key-relevant clusterings) can enhance the effectiveness of the DoM method.

Having illustrated the clustering perspective of DPA and its potential utility for identifying the correct key, we will move on to illustrate the relevance of low-order localized eigenvectors of PCA for identifying the low-level signal within the data i.e. the relevant clustering structures that "depend" on a small number of features.

3 PCA, Localization and Clustering

3.1 PCA and Clustering

PCA is a very popular data preprocessing technique that is commonly understood as a data-reduction/approximation method. In principle, PCA maximizes the data variance in the reduced space and works by projecting the data matrix to the principal components (dominant eigenvectors) of the feature-covariance matrix. The intimate connections between PCA and clustering (often referred to as projected clustering, or subspace clustering), have been studied by several authors (such as [10], [20]).

In order to clarify the connections between PCA and clustering we briefly illustrate the results of [10]. Let X denote an input matrix that is $n \times m$, where n is the number of instances (power traces in our context) and m is the number of features (time scale in our context) and let also X_{fc} denote the feature centered data matrix (i.e. from each column of X the mean value is subtracted). If we consider the Singular Value of $X_{fc} = U\Sigma V^T$ it is easy to verify the following:

- The sample Covariance matrix can be written as: $Cov = \frac{1}{n-1} X_{fc}^T X_{fc}$.
- The right singular vectors of X_{fc} (columns of matrix V) are also the eigenvectors of matrix Cov.
- The singular values σ_i of X_{fc} are equal to $\sigma_i = \sqrt{\lambda_i (n-1)}$ where λ_i denotes the i^{th} eigenvalue of the sample covariance matrix Cov.

The main result of [10] states that the dominant left singular vector u_1 of X_{fc} can be regarded as a "continuous" clustering solution to the $K = 2$-means clustering problem (i.e. the elements of u_1 can be regarded as continuous approximations to the discrete cluster assignments of the instances). The connection to PCA projections becomes evident if we write u_1 as a projection of the input matrix X_{fc} to the dominant eigenvector of the sample Covariance matrix: $u_1 = X_{fc} v_1 / \sigma_1 = X_{fc} v_1 / \sqrt{(n-1)\lambda_1}$, where v_1 is the dominant right singular vector and σ_1 and λ_1 are as defined in the afore enumeration. One can obtain a discrete cluster solution using various discretization strategies of u_1. A simple discretization approach is to assign all instances i with $u_1(i) > mean(u_1)$ to

cluster A and the rest to cluster $\overline{A}$. As it is rigorously analysed in [10] this simple process can be regarded as a continuous approximation solution to the $(K = 2)$-means objective function. Analogous results are also obtained for $K > 2$.

The results of [10] regarding the relationships between PCA and $(K = 2)$-means can be summarized in the following two lemmas:

Lemma 1 (Continuous Clustering Solution, equation 5, Theorem 2.2 in [10]). *A continuous clustering solution of the $(K = 2)$-means clustering problem can be derived by the dominant left singular vector of the feature-centered input matrix X_{fc}.*

Lemma 2 (Quality of the Continuous Clustering Solution, equation 6, Theorem 2.2 in [10]). *The quality of the continuous clustering solution is estimated (in a lower bound sense) by the dominant singular value of the feature-centered input matrix X_{fc}.*

Analogously to the afore lemmas, we can consider that the rest (low-order) left singular vectors of the feature-centered input matrix X_{fc} provide us with approximate clusterings of lower quality (with respect to the K-means objective), since they correspond to smaller singular values.

As we have analyzed in the previous section, in DoM based DPA we are interested in finding the 2-way clustering structure of the power traces that corresponds to the secret key. The relevance of PCA is not immediately evident since the dominant eigenvector(s) of PCA correspond to clustering structures that optimize the K-means clustering objective. This objective employs equally all the features and is not relevant to the criterion that is used for identifying the secret key. In the subsequent sections, the relevance of low-order PCA eigenvectors will be illustrated. These eigenvectors will correspond to low-quality (or at least non-optimal) clustering structures (according to the K-means objective) but will have certain properties (localization in a small number of features) that are crucial for identifying the secret key of the cryptographic device.

3.2 Eigenvector Localization

The key property that will allow us to detect the key-relevant clustering structures is localization. The term "eigenvector localization" is used to refer to cases where the majority of the mass of an eigenvector is located in a small number of entries (i.e. most of the eigenvector entries are zero and near-zero). This phenomenon has been observed in several diverse application areas such as DNA single-nucleotide polymorphism data, spectral and hyperspectral data in astronomy (see [8],[7],[18] and references therein). Eigenvector localization is commonly measured using the *Inverse Participation Ratio* [8],[7] that is defined as:

$$IPR(v) = \sum_{i=1}^{m} v(i)^4 \tag{1}$$

Where v is the eigenvector (of length m) whose localization we want to measure. It should also be noted that this definition assumes that the eigenvectors are normalized (i.e. $\sum_{i=1}^{m} v(i)^2 = 1$). The higher the value of IPR the more localized the eigenvector is.

It can be observed that IPR is related to the fourth moment in statistics and can effectively measure the level of concentration of the eigenvector values. For example if the

values of an eigenvector are equally scattered in all its indexes, i.e. $v = (\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}, ..., \frac{1}{\sqrt{m}})$ then the *IPR* will be $IPR(v) = 1/m$. While, in the extreme case where there is only one non-zero entry, i.e. $v = (0, .., 1, ..., 0)$ the *IPR* value will be $IPR(v) = 1$. Naturally, localization can be quantified with various other measures, such as the simple sum of absolute values. However, in the context of this work we focus solely on the IPR measure since it is consistent with the relevant work and also works effectively in practice (as observed in experiments Section 6). A deeper study on the different potential measures for quantifying eigenvector localization constitutes an interesting topic for further work.

3.3 Localized Principal Eigenvectors and Clustering Structures that Depend on Few Features

In Section 3.1 we have illustrated the relevance of PCA and its dominant eigenvectors for obtaining (continuous) approximate clusterings that optimize the K-means objective. We will now move on and show that the projection of a (feature centered) data matrix to a low-order localized eigenvector corresponds to a (continuous) clustering solution that has lower quality (with respect to the K-means objective) and "depends" only on a small number of features.

Intuitively, the term "depends on a small number of features" means that the quality of the clustering does not change if we remove the features where the eigenvector is localized. This is formally shown in the following proposition:

Proposition 1. *Let X_{fc} be an $n \times m$ matrix (n=# instances, m=# features). If σ_l, u_l and v_l are the l^{th} singular value and vectors of X_{fc} with $v_l(i) = 0$ for all $i \in A \subset \{1, ..., m\}$, then σ_l,u_l and v_l will also be singular value and vectors of $X_{fs} = X_{fc}\mathbf{diag}(u)$ where $u(i) = 0$ for all $i \in A$, $u(i) = 1$ for all $i \in \overline{A}$ and $\mathbf{diag}(u)$ denotes a diagonal matrix with vector u in its diagonal.*

Proof. In our proof we will initially show that v_l is an eigenvector of $X_{fs}^T X_{fs}$ and that u_l is an eigenvector of $X_{fs}X_{fs}^T$.

We have that $X_{fs}^T X_{fs} v_l = \mathbf{diag}(u) X_{fc}^T X_{fc} \mathbf{diag}(u) v_l$. Now since $v_l(i) = 0$ and $u(i) = 0$ for the same i, we have that $\mathbf{diag}(u)v_l = v_l$. Thus we can write:
$\mathbf{diag}(u)X_{fc}^T X_{fc}\mathbf{diag}(u)v_l = \mathbf{diag}(u)X_{fc}^T X_{fc} v_l$. Now since v_l is a right singular vector of X_{fc} we can derive that:
$\mathbf{diag}(u)X_{fc}^T X_{fc} v_l = \mathbf{diag}(u)X_{fc}^T(\sigma_l u_l) = \sigma_l \mathbf{diag}(u)X_{fc}^T u_l = \sigma_l^2 \mathbf{diag}(u)v_l = \sigma_l^2 v_l$

Above, we have shown that v_l is an eigenvector of $X_{fs}^T X_{fs}$ with corresponding eigenvalue σ_l^2 and thus v_l is also a right singular vector of X_{fs} with corresponding singular value σ_l.

Now for u_l we can write $X_{fs}X_{fs}^T u_l = X_{fc}\mathbf{diag}(u)\mathbf{diag}(u)^T X_{fc}^T u_l = X_{fc}\mathbf{diag}(u)(\sigma_l v_l) = \sigma_l X_{fc}\mathbf{diag}(u)v_l = \sigma_l X_{fc} v_l = \sigma_l^2 u_l$

Above, we have shown that u_l is an eigenvector of $X_{fs}X_{fs}^T$ with corresponding eigenvalue σ_l^2 and thus u_l is also a left singular vector of X_{fs} with corresponding singular value σ_l.

Since u_l and v_l correspond to the same singular value they will effectively be a singular vector pair of X_{fs} with singular value σ_l. **QED**

Informally the afore proposition can be summarized, using the same notation, as follows:

- Let X_{fc} be an input matrix with a localized low-order right singular vector v_l (recall that this is also an eigenvector of the feature-covariance matrix).
- The corresponding clustering solution will be u_l, the l^{th} left singular vector of X_{fc} or equivalently, the projection of the input data matrix to v_l.
- The (continuous) quality of this clustering solution will be equal to the l^{th} singular value of X_{fc}.
- If we apply feature selection and remove the features where v_l is localized (in the Proposition this matrix is denoted as $X_{fs} = X_{fc}\mathbf{diag}(u)$), then the matrix X_{fs} will have σ_l, u_l and v_l as signular values and vectors.
- This effectively means, that the quality σ_l of the (continuous) clustering u_l is the same for both X_{fc} (original input matrix) and X_{fs} (matrix after feature selection).

The analysis presented in this Section should have clarified the utility of localized eigenvectors of a feature covariance matrix for identifying clusters that "depend on few features". However, one question that still remains is *Why (and when) should we expect these localized eigenvectors to appear in the spectrum of a Covariance matrix.* This is a valid and important question since an $n \times m$ matrix can at most have $min(n,m)$ singular vectors. These can be much less than the number of possible clusterings of the n instances.

4 Why (and when) are the Eigenvectors of PCA Localized

We should initially state that our motivation to study localized eigenvectors in the context of DPA was based on [5] where it was demonstrated that low-order localized eigenvectors do appear in the spectrum of the Covariance matrix and also carry important information related to the secret key used by the cryptographic device. Based on these findings we will study in this section a simple model that can explain the appearance of these useful, cluster related localized eigenvectors.

Our model considers a high noise "unstructured" matrix and a low-signal rank-1 update that "inserts" the relevant clustering to the unstructured data matrix. The terms "high-noise" vs. "low-signal" are employed to stress that the singular values of the "unstructured matrix" are larger than the singular value of the rank-1 update. This eventually means that these cluster structures would be "buried" in the low-order spectrum of the resulting data matrix and would not be detectable using a standard spectral clustering technique.

In mathematical terms our model can be stated as:

$$X_{input} = X_{unstruct} + \gamma u_{cl} v_{cl}^T \tag{2}$$

Where X_{input} and $X_{unstruct}$ are (instance$\times$variable) matrices ($n \times m$), u_{cl} is an $n \times 1$ vector that contains the instance (power trace) clustering structure[1], v_{cl} is the localized

[1] $u_{cl}(i) = \sqrt{n_2/nn_1}$ if i belongs to cluster A_1 and $u_{cl}(i) = -\sqrt{n_1/nn_2}$ if i belongs to cluster A_2. n_1 and n_2 are the cluster sizes of A_1 and A_2 respectively and $n = n_1 + n_2$ is the total number of instances.

$m \times 1$ vector with $||v_{cl}||_2 = 1$ that contains only a small number of non-zero indexes (i.e. it is localized) and γ is the "power" of the signal that is small as compared to the largest eigenvalues of $X_{unstruct}$. Based on this model, we will inspect whether the localized right singular vectors of X_{input} can be effectively used for detecting the "hidden" clustering structure u_{cl} that depends on a small number of features.

It can be observed that the spectrum of X_{input} will be determined by the correlation of u_{cl} to the left singular vectors of $X_{unstruct}$ and also by the correlations of v_{cl} to the right singular vectors of $X_{unstruct}$. In our simulations, we will consider that the structures u_{cl} and v_{cl} are not already present in the unstructured data matrix. This is achieved by considering that the left and right singular vectors of $X_{unstruct}$ are random orthogonal matrices (Haar distribution) [22, 9], thus the vectors u_{cl} and v_{cl} will exhibit a (uniformly) random correlation with the singular vectors of $X_{unstruct}$. Moreover, in order to simulate the $X_{unstruct}$ matrix, we employ the actual singular values that were observed in the dataset 1DES_RandomDelays, which is described in the Experiments Section 6.

Based on the above, we define $X_{unstruct}$ as follows:

$$X_{unstruct} = U_{un} \Sigma V_{un}^T$$

where U_{un} and V_{un}^T are uniform random orthogonal matrices [22, 9] and Σ contains the singular values of dataset 1DES_RandomDelays.

Based on this definition of $X_{unstruct}$ we have experimented with the two parameters of the model, the γ parameter and the number of features that determine the cluster structure (i.e. the number of zeros in v_{cl}). The simulations illustrate that when the size of γ is "sufficiently large" and also when the cluster structure depends on "sufficiently few" features, we can effectively use the IPR measure for detecting the hidden clustering structure. The terms "sufficiently large" and "sufficiently few" are explored within the experiments[2].

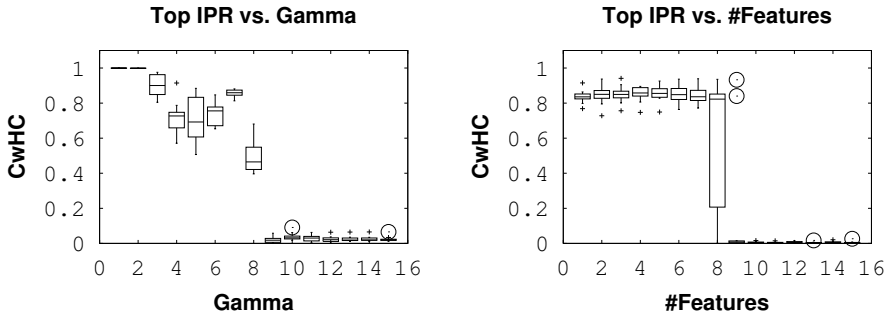
In Figure 2(a) we have fixed the level of localization (i.e. number of zeros in the in v_{cl} of formula 2), and we have used multiple γ values, while in Figure 2(b) we have fixed γ and have varied the localization of the hidden clustering structure. More precisely, in Figure 2(a) we have fixed v_{cl} at 10 Features, and the initial value for the γ parameter was taken to be equal to the maximum singular value of $X_{unstruct}$ divided by 2. Consecutively, in each run we have divided γ by 2, i.e. in the 8^{th} run we set γ to be equal to the maximum singular value of $X_{unstruct}$ divided by 2^8.

In Figure 2(b), we have fixed the γ value to be equal to the maximum singular value of $X_{unstruct}$ divided by 2^7 and we have taken the initial number of non-zero element in v_{cl} to be 60. Consecutively, in each run we have set the number of features as multiples of 60, i.e. in the 5th run the number of features (non-zero elements) in v_{cl} are taken to be $5 \cdot 60$.

In both Figures 2(a),2(b), in order to detect the hidden clustering structure, we have executed the following procedure:

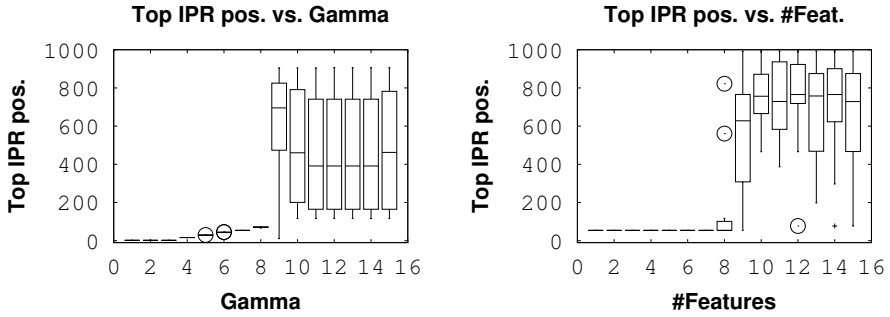
– We compute the SVD of $X_{input} = U_{input} \Sigma_{input} V_{input}^T = X_{unstruct} + \gamma u_{cl} v_{cl}^T$

[2] In order to facilitate the reproduction of empirical results, the relevant model simulation code is available at the website of the correspondence author: `http://sites.google.com/site/mavroeid/`



(a) IPR vs. Gamma (b) IPR vs. # Features

Fig. 2. Effectiveness of top IPR vector for detecting hidden clustering structure



(a) Position of top IPR vs. Gamma (b) Position of top IPR vs. # Features

Fig. 3. Position of top IPR vector

- We compute the IPR of the right singular vectors (columns of V_{input})
- We select the left singular vector that corresponds to the top IPR right singular vector and measure its correlation with the hidden clustering structure using the inner product. (*CwHC* measure, *x* axis in Figure 2)

The afore procedure is repeated 10 times and Figure 2 reports the resulting boxplots. Interestingly, Figure 2 illustrates that even when we insert a very low-signal clustering structure (i.e. when γ is 2^7 times smaller than the maximum singular value of $X_{unstruct}$), that is localized with less than 10% of the available features, IPR can effectively detect the hidden clustering structure.

Figure 3 illustrates the positions (ordered according to singular values) of the top IPR vectors that are used for finding the hidden clustering structure in Figure 2. Again boxplots are presented based on the 10 simulations of X_{input}.

5 Related Work

To the extent of our knowledge, low-order localized eigenvectors of graph adjacency and Laplacian matrices have received little attention within the data mining community ([8, 7] and references therein). As compared to these approaches the main distinctive characteristic of our work is that it focuses on an (instance$\times$feature) input matrix and interprets localized low-order eigenvectors as clusterings that depend on a small number of features.

Recently, there were a few works showing the potential of PCA for side-channel analysis [1, 5]. The impact of low-order eigenvectors in attacking smart card platforms was demonstrated in experiments but the theoretical reasoning on the effectiveness of the new method was lacking [5]. Our work fills in this gap in proving that the previous study was not an isolated and random experiment but there is indeed a lot of potential in exploring similar techniques that are already known for machine learning researchers.

6 Experiments

6.1 Data Description

To perform side-channel analysis some hardware equipment is required, being part of a measurement set-up. A typical set-up for measurements includes a smart card (on which the target algorithm is implemented), a smart card reader, an oscilloscope for the acquisition of power traces and a PC for the analysis and key recovery. All our experiments are performed using a programmable card from Atmel (ATMega163+24C256). The card contains an Atmel 8-bit AVR RISC-based microcontroller that combines 16KB of programmable flash memory, 1KB SRAM and 512B EEPROM. Single and Triple DES implementations that we considered were all software implementations, with or without countermeasures. An implementation of triple DES consists of three DES algorithms where the new key is 112 bits long because the first and last DES have the same key.

The first datasets called 1DES_wo_ctrmsr and 3DES_wo_ctrmsr are single DES and triple-DES implementations without countermeasures. However, cryptographic algorithms are in practice often implemented with one or more countermeasures rendering side-channel analysis. The countermeasures are commonly divided into two groups: masking or hiding [19]. Masking countermeasures are based on masking data or the key (or both) by adding a random value to the input that is created to make the intermediate variables data- and key-independent complicating in this way DPA substantially. Hiding countermeasures aim at making a side-channel e.g. power consumptions unrelated to the data processed for example by burying the signal into noise. For the measurements of simple DES with countermeasures we employed random delays (1DES_RandomDelays), which introduces random wait states during execution of the algorithm. In this way the effect of misaligned traces is obtained.

We have also employed data masking countermeasures 3DES_DataMasking for triple DES, so the inputs are masked with a random value and unmasked at the end to get the correct output.

As our experiments use DES and triple DES, we give some details of the algorithms next. The Data Encryption Algorithm (DES) was invented in the 70's by IBM and used

1DES_wo_ctrmsr	DES without countermeasures
1DES_RandomDelays	DES with random delays countermeasure
3DES_wo_ctrmsr	3DES without countermeasures
3DES_DataMasking	3DES with data masking countermeasure

Table 2. Datasets used in experiments. In order to facilitate the reproduction of empirical results, all datasets are available at the website of the correspondence author: http://sites.google.com/site/mavroeid/

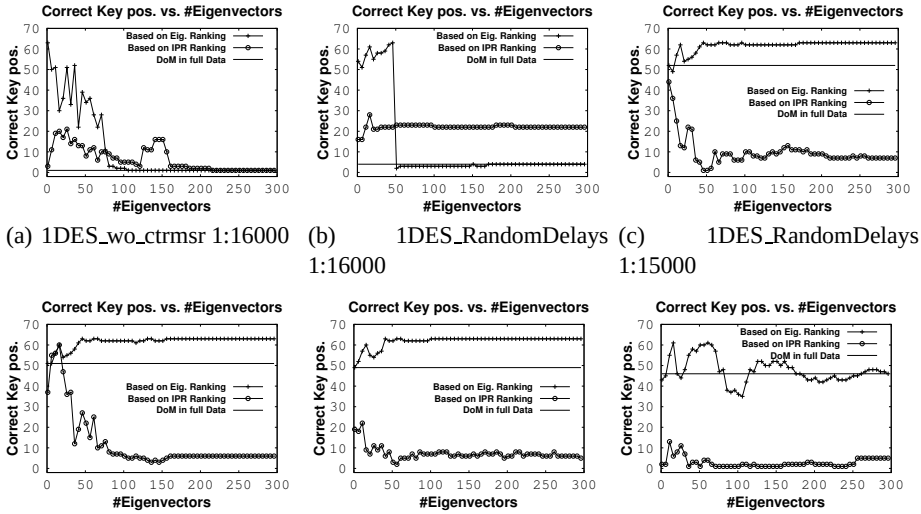
as the main encryption standard for more than two decades. Due to the short key, DES is nowadays mainly used as a part of the triple DES algorithm (3DES), but nevertheless attacking DES remains the first step in the side-channel analysis of 3DES.

DES uses a 64-bit key and it operates on 64-bit blocks of plaintext as input and returns blocks of 64 bits as output (called ciphertext) after 16 rounds. Each round has several operation, but the most important one is the non-linear function f that contains 8 substitution boxes $(S_1,\ldots,S_8)$, so-called S-boxes. The S-boxes have a 6-bit input, and a 4-bit output derived by a table look-up. The most important property for side-channel analysis is that the 6-bit inputs, and hence also 4-bit outputs of S-boxes, depend only on plaintexts and 6 bits of the key. In other words, knowing plaintexts and making a hypothesis on the 6-bit subkeys, we can monitor only one S-box to learn the right hypothesis on the subkey (out of 2^6 possibilities). By monitoring, we mean computing the S-box outputs and a function of the inputs, hereafter called the selection function as in [15] (for known plaintexts and a key guess) and correlating the values to the power consumption measured during this computation. The reasoning behind lies in the simple fact that power consumed by a device depends on the data being processed.

The procedure for the data acquisition and the key recovery, consists of several steps explained below. We use the example of DES to list the steps.

- Denote the selection function by $f(x,k)$ where k is typically a small part of the key (here 6 bits). For a large number (say n) of randomly generated plaintexts, we can compute the value of f for each 6 bits of a subkey. At the same time we collect power measurements of the device while performing the algorithm.
- Each power trace contains the values at m time points i.e. samples. Hence, we can create a matrix A of size nxm that contains the power traces corresponding to different plaintexts.
- Next step is to calculate the hypothetical values of f for every possible subkey k. In the case of DES there are 64 different subkeys for each S-box. As the selection function we simply choose one bit of the S-box output e.g. MSB (or LSB) so the function $f(x,k)$ can be either 0 or 1. Based on this value we sort all power traces into two sets, say S_0 and S_1. Note that this will result in a power-trace clustering for each possible subkey.
- We apply distance of means (DoM) test to the S_0 and S_1 partition plotting each so-called differential traces that is computed as:

$$d = \overline{S_1} - \overline{S_0}.$$



(a) 1DES_wo_ctrmsr 1:16000 (b) 1DES_RandomDelays 1:16000 (c) 1DES_RandomDelays 1:15000

(d) 1DES_RandomDelays 1:12000 (e) 1DES_RandomDelays 1:8000 (f) 1DES_RandomDelays 3000:5000

Fig. 4. Correct Key Position based on DoM method for single-DES

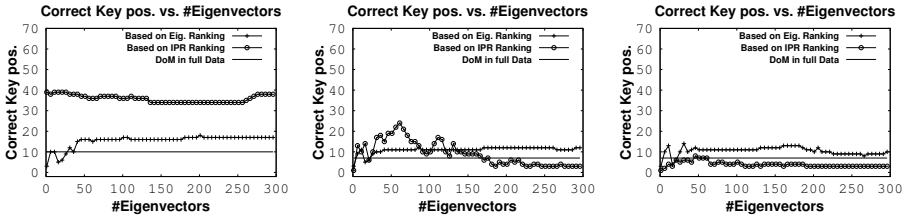
Out of 64 differential traces we select the one with the highest peaks at certain time sample as the correct key.

6.2 Empirical Results

In order to evaluate the effectiveness of IPR and localized eigenvectors for retrieving the correct key, we adopt the following procedure:

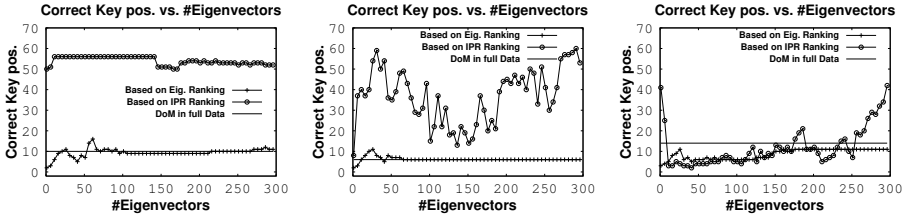
- We compute the SVD of the centered input $n \times m$ (power trace $\times$ time) matrix $X_{fc} = U\Sigma V^T$.
- We compute the IPR of the right singular vectors (i.e. the columns of V).
- We compute projections based on PCA and IPR. I.e. We compute $Pr_{(PCA)}(k) = X_{fc}V_{1:k}V_{1:k}^T$ and $Pr_{(IPR)}(k) = X_{fc}V_{IPR(1:k)}V_{IPR(1:k)}^T$, where $V_{1:k}$ contains the dominant right singular vectors (that correspond to the largest singular values), while $V_{IPR(1:k)}$ contains the top IPR singular vectors.
- We compare the results obtained by the DoM measure in the full data, in $Pr_{(PCA)}$ and $Pr_{(IPR)}$. I.e. for each input matrix we compute the DoM measure and we rank the keys according to the largest observed peak. Since we know for these datasets the correct key, we can observe whether the IPR or PCA preprocessing improves the position of the correct key.

In Figure 4 we evaluate the effectiveness of IPR in single-DES datasets (1DES_wo_ctrmsr, 1DES_RandomDelays). The y-axis of Figure 4 reports the position of the correct key,



(a) 3DES_wo_ctrmsr 1:16000 (b) 3DES_wo_ctrmsr 1:12000 (c) 3DES_wo_ctrmsr 2900:84000

Fig. 5. Correct Key Position based on DoM method for 3DES_wo_ctrmsr



(a) 3DES_DataMasking 1:16000 (b) 3DES_DataMasking 1:15000 (c) 3DES_DataMasking 1:12000

Fig. 6. Correct Key Position based on DoM method for 3DES_DataMasking

based on the full dataset and also using PCA and IPR preprocessing. The optimal result is to obtain a *position=1* for the correct key. Moreover, knowing that the largest peak in these datasets (at the aligned data) occurs around time 4900 (in accordance to DPA terminology, these are called interesting points), we have investigated the effect of IPR and PCA as we zoom to time 4900 (the original time scale is 1-16000).

In the single-DES dataset without countermeasures 1DES_wo_ctrmsr (Figure 4(a)), the full-data matrix already retrieves the correct key in top-position. Moreover, we can observe that PCA and IPR preprocessing converge rather quickly to position=1 with PCA achieving a faster convergence while IPR being more effective when a very small number of eigenvectors are used.

In the single-DES dataset with random delays 1DES_RandomDelays (Figures 4(b), 4(c),4(d),4(e),4(f) for different time-intervals), the full-data matrix fails to retrieve the correct key and in most cases (with the exception of the 1-16000 time interval) ranks it very low. On the other hand IPR is very effective in the majority of experiments with the notable exception of the 1-16000 time interval. More precisely, we can observe that the performance of IPR dramatically improves as we zoom to time 4900 (interest-point).

Notably, when we focus on the 3000-5000 time scale IPR consistently retrieves the correct key in top positions.

In the triple-DES dataset without countermeasures DES_wo_ctrmsr (Figures 5(a), 5(b),5(c)), the full-data matrix places the correct key in the 10th position. We can again observe that the IPR ranking requires an adequate zoom into the interest-point for effectively identifying the correct key. A similar behaviour is also observed in triple-DES with Data Masking.

7 Discussion

In this work we have presented a thorough theoretical framework that illustrates the relevance of low-order localized eigenvectors in the application context of DPA. In our analysis, we have considered that the key-relevant information is modelled as a "low-signal", localized pattern that is embedded in a "high-noise" dataset. In this respect our results generalize beyond DPA and are applicable to analogous low-signal/hidden pattern problems.

Based on the empirical results one can identify two interesting further work topics that can potentially enhance the practical effectiveness of our framework. The first is related to the automatic determination of the appropriate number of eigenvectors that are needed in order to optimize the position of the correct key. The successful tackling of this issue requires further statistical analysis of the IPR measure in order to be able to identify statistically significant localized eigenvectors and not simple noise artifacts.

The second issue is related to the determination of the appropriate time-interval that maximizes the effectiveness of IPR and localization. As discussed above, to maximize the effect of IPR, we rely on the ability of "zooming" into the traces i.e. we assume the points with maximal leakage (also sometimes called interesting points) being known to some accuracy. This is a standard assumption for DPA, and there are several methods proposed in the relevant literature. For example, an attacker can simply apply the correlation with input data analysis [11]. Similarly, one can use template attacks [1] to learn more about the points of interest. It constitutes an interesting issue for further work to explore the required tightness of the time-interval that can guarantee the effectiveness of IPR and localization.

References

1. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.
2. Josep Balasch, Benedikt Gierlichs, Roel Verdult, Lejla Batina, and Ingrid Verbauwhede. Power analysis of atmel cryptomemory - recovering keys from secure eeproms. In *Proceedings of the 12th conference on Topics in Cryptology*, CT-RSA'12. Springer-Verlag, 2012.
3. Lejla Batina, Benedikt Gierlichs, and Kerstin Lemke-Rust. Comparative Evaluation of Rank Correlation Based DPA on an AES Prototype Chip. In *Information Security, 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008. Proceedings*, volume 5222 of *Lecture Notes in Computer Science*, pages 341–354. Springer, 2008.

4. Lejla Batina, Benedikt Gierlichs, and Kerstin Lemke-Rust. Differential cluster analysis. In *CHES '09: Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, pages 112–127, Berlin, Heidelberg, 2009. Springer-Verlag.
5. Lejla Batina, Jip Hogenboom, and Jasper 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*, volume 7178 of *Lecture Notes in Computer Science*, pages 383–397. Springer, 2012.
6. E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. *Cryptographic Hardware and Embedded Systems CHES 2004*, 3156:16–29, 2004.
7. Mihai Cucuringu, Vincent D. Blondel, and Paul Van Dooren. Extracting spatial information from networks with low-order eigenvectors. *CoRR*, abs/1111.0920, 2011.
8. Mihai Cucuringu and Michael W. Mahoney. Localization on low-order eigenvectors of data matrices. *CoRR*, abs/1109.1355, 2011.
9. Persi Diaconis and Mehrdad Shahshahani. The subgroup algorithm for generating uniform random variables. *Probability in the Engineering and Informational Sciences*, 1(01), 1987.
10. Chris H. Q. Ding and Xiaofeng He. *K*-means clustering via principal component analysis. In *ICML*, 2004.
11. Paul N. Fahn and Peter K. Pearson. IPA: A new class of power attacks. In *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 173–186. Springer, 1999.
12. K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, number 2162 in Lecture Notes in Computer Science. Springer-Verlag, 2001.
13. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*. Springer, 2008.
14. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. In *Advances in Cryptology: Proceedings of CRYPTO'96*, number 1109 in Lecture Notes in Computer Science, pages 104–113. Springer-Verlag, 1996.
15. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology: Proceedings of CRYPTO'99*, number 1666 in Lecture Notes in Computer Science, pages 388–397. Springer-Verlag, 1999.
16. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, London, UK, UK, 1999. Springer-Verlag.
17. Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.
18. Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.
19. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
20. P. Meinicke and H. Ritter. Local pca learning with resolution-dependent mixtures of gaussians. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 1, pages 497–502, 1999.
21. J.-J. Quisquater and D. Samyde. ElectroMagnetic Analysis (EMA): Measures and Couter-Measures for Smard Cards. In *Smart Card Programming and Security (E-smart 2001)*, volume 2140 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
22. G W Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980.