Categorising and Comparing Cluster-Based DPA Distinguishers

Xinping Zhou[1,2], Carolyn Whitnall[2], Elisabeth Oswald[2], Degang Sun[1], and Zhu Wang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, P.R. China
[2] Department of Computer Science, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK

Abstract. Side-channel distinguishers play an important role in differential power analysis, where real world leakage information is compared against hypothetical predictions in order to guess at the underlying secret key. A class of distinguishers which can be described as 'cluster-based' have the advantage that they are able to exploit multi-dimensional leakage samples in scenarios where only loose, 'semi-profiled' approximations of the true leakage forms are available. This is by contrast with univariate distinguishers exploiting only single points (e.g. correlation), and Template Attacks requiring concise fitted models which can be overly sensitive to mismatch between the profiling and attack acquisitions. This paper collects together—to our knowledge, for the first time—the various different proposals for cluster-based DPA (concretely, Differential Cluster Analysis, First Principal Components Analysis, and Linear Discriminant Analysis), and shows how they fit within the robust 'semi-profiling' attack procedure proposed by Whitnall et al. at CHES 2015. We provide discussion of the theoretical similarities and differences of the separately proposed distinguishers as well as an empirical comparison of their performance in a range of (real and simulated) leakage scenarios and with varying parameters. Our findings have application for practitioners constrained to rely on 'semi-profiled' models who wish to make informed choices about the best known procedures to exploit such information.

1 Introduction

It is well-established that the extent and accuracy of an attacker's knowledge about the data-dependent functional form of side-channel leakage impacts substantially on the effectiveness of a differential side-channel analysis (DPA)[3]. At one end of the spectrum are detailed, usually multivariate fitted models acquired in a profiling stage during which the attacker has access to a device identical to the target [4]; at the other, are reasoned guesses based on general knowledge of circuit activity, such as Hamming weight or Hamming distance assumptions

Author version of an article to appear at SAC 2017.

[3] The 'P' in DPA stands for Power but the principles of DPA extend equally to other data-dependent observables such as electromagnetic radiation.

[9]. The former can be used to perform Bayesian classification on target traces. These can be highly efficient at recovering secret values in the case that there is a close match between the profiling and the attack acquisitions, but can fail altogether in the presence of discrepancies [5,11]. The latter are most typically used in correlation attacks [3], which succeed as long as the guessed model is reasonably proportional to the true form of the leakage, but are less efficient (or entirely ineffective) the larger the divergence between model and reality [15]. They are also inherently univariate, raising the question of how best to combine relevant information from different points in the traces.

A form of 'semi-profiling', sitting somewhere between the two extremes, is achieved by unsupervised clustering of leakage traces with known intermediates, as proposed by Whitnall et al. at CHES 2015 [16]. This procedure assumes some *a priori* access to measurements from a duplicate device, without necessarily requiring the degree of control over or replicability of the acquisitions assumed in a 'fully profiled' setting. Rather than outputting precise and detailed models, these aim at rough arrangements of intermediate values into similarly-leaking classes, which can be used as 'nominal power models' [17] in cluster- (AKA partition- [13]) based DPA.

Several proposals for cluster-based DPA distinguishers have been made, including the recent Linear Discriminant Analysis (LDA) based attack [8]. However, practitioners have so far had little guidance as to which of these might be preferable for use in real attack scenarios when constrained to rely on 'semi-profiled' nominal models. Most of the experimental investigations done previously have been performed under standard (non-profiled) leakage assumptions such as the Hamming weight, and in leakage scenarios conforming well to those assumptions. Further, each new cluster-based distinguisher has typically been compared against correlation-based DPA (a popular benchmark) rather than against existing proposals of a similar nature; to the best of our knowledge, there does not yet exist a study collecting together all of these conceptually similar methodologies, as we aim to do here. We explore and explain the points on which the different distinguishers differ and, by integrating them within the clustering-based semi-profiling attack procedure of [16], are able to empirically test their performance for a wider range of leakage scenarios and prior knowledge assumptions than previously attempted, thus arriving at a clearer picture of the best options for semi-profiling adversaries and evaluators.

The rest of the paper proceeds as follows: Section 2 covers the preliminaries of DPA generally, cluster-based DPA in particular, and the application of unsupervised clustering for building the semi-profiled power models used by cluster-based DPA. In Section 3 the four distinguishers are empirically tested in one hardware and one software leakage scenario, as parameters vary. We also test them against simulated leakage with increasing levels of Gaussian noise. Section 4 discusses some of the reasons for the difference in performance from a theoretical perspective, and Section 5 concludes.

2 Preliminaries

2.1 Differential Power Analysis

We consider a 'standard DPA attack' scenario as defined in [10], and briefly explain the underlying idea as well as introduce the necessary terminology here. We assume that the power consumption $\mathbf{P} = \{P_1, ..., P_T\}$ of a cryptographic device (as measured at time points $\{1, ..., T\}$) depends, for at least some $\boldsymbol{\tau} \subset \{1, ..., T\}$, on some internal value (or state) $F_{k^*}(X)$ which we call the *target*: a function $F_{k^*} : \mathcal{X} \to \mathcal{Z}$ of some part of the known plaintext—a random variable $X \overset{R}{\in} \mathcal{X}$—which is dependent on some part of the secret key $k^* \in \mathcal{K}$. Consequently, we have that $P_t = L_t \circ F_{k^*}(X) + \varepsilon_t, t \in \boldsymbol{\tau}$, where $L_t : \mathcal{Z} \to \mathbb{R}$ describes the data-dependent leakage function at time t and ε_t comprises the remaining power consumption which can be modeled as independent random noise (this simplifying assumption is common in the literature—see, again, [10]). The attacker has N power measurements corresponding to encryptions of N known plaintexts $x_i \in \mathcal{X}$, $i = 1, \ldots, N$ and wishes to recover the secret key k^*. The attacker can accurately compute the internal values as they would be under each key hypothesis $\{F_k(x_i)\}_{i=1}^N, k \in \mathcal{K}$ and uses whatever information he possesses about the true leakage functions L_t to construct a prediction model (or models) $M_t : \mathcal{Z} \to \mathcal{M}_t$.

A distinguisher D is some function which can be applied to the measurements and the hypothesis-dependent predictions in order to quantify the correspondence between them, the intuition being that the predictions under a correct key guess should give more information about the true trace measurements than an incorrect guess. For a given such comparison statistic, D, the *theoretic* attack vector is $\mathbf{D} = \{D(L \circ F_{k^*}(X) + \varepsilon, M \circ F_k(X))\}_{k \in \mathcal{K}}$, and the *estimated* vector from a practical instantiation of the attack is $\hat{\mathbf{D}}_N = \{\hat{D}_N(L \circ F_{k^*}(\mathbf{x}) + \mathbf{e}, M \circ F_k(\mathbf{x}))\}_{k \in \mathcal{K}}$ (where $\mathbf{x} = \{x_i\}_{i=1}^N$ are the known inputs and $\mathbf{e} = \{e_i\}_{i=1}^N$ is the observed noise). Then the attack is *o-th order theoretically successful* if $\#\{k \in \mathcal{K} : \mathbf{D}[k^*] \leq \mathbf{D}[k]\} \leq o$ and *o-th order successful* if $\#\{k \in \mathcal{K} : \hat{\mathbf{D}}_N[k^*] \leq \hat{\mathbf{D}}_N[k]\} \leq o$.

2.2 Cluster-Based Distinguishers

Differential Cluster Analysis Differential Cluster Analysis (DCA) was proposed by Batina et al. in [2]. The main idea of DCA is that the hypothesised cluster arrangement $(M \circ F_k(X)))$ arising from the correct key guess conforms with the real power consumption, so that the between-cluster variance (or the sum of the variances within each cluster) as the separation criterion would be maximum (or minimum) when compared with the cluster arrangements arising under other key hypotheses. The distinguisher score can be expressed as:

$$D_{DCA}(k) = \sum_{m \in M} \sum_{t \in \tau'} \mathrm{var}(\{\mathrm{P}_{t,i} | M \circ F_k(x_i) = m\})^2 \tag{1}$$

where $\{P_{t,i}\}_{i=1}^N$ is the power traces, $\boldsymbol{\tau}'$ is the attacker's best knowledge about $\boldsymbol{\tau}$ (one hopes that $\boldsymbol{\tau}' \cap \boldsymbol{\tau} \neq \emptyset$), M is a nominal approximation (taking values in $\mathcal{M}$)

for the leakage output by a power model, and $n_m = \#\{x_i | M \circ F_k(x_i) = m\}$, i.e. the number of observations in the trace set for which the predicted cluster label is m. An alternative separation criterion, also suggested in [2], is the variance ratio of [13]:

$$D_{\text{DCA-VR}}(k) = \frac{\sum\limits_{t \in \boldsymbol{\tau}'} \text{var}(\{P_{t,i}\}_{i=1}^{N})^2}{\frac{1}{N} \sum\limits_{m \in \mathcal{M}} n_m \sum\limits_{t \in \boldsymbol{\tau}'} \text{var}(\{P_{t,i} | M \circ F_k(x_i) = m\})^2}, \tag{2}$$

First Principal Components Analysis Principal component analysis (PCA) is a popular method for unsupervised dimensionality reduction. An $N \times T$ matrix is orthogonally transformed so that the T columns in the new matrix are linearly uncorrelated and sorted in decreasing order of variance. By construction, the columns are the eigenvectors of the covariance matrix, sorted according to the size (largest to smallest) of the corresponding eigenvalues $\lambda_1, \ldots, \lambda_T$. The first $q < T$ of these columns maximise (w.r.t. all other $N \times q$ transformations) the total variance preserved whilst minimising the mean squared reconstruction error $\sum_{i=q+1}^{T} \lambda_i$. The hope is that all of the 'important' information will be concentrated into a small number of components.

First Principal Components Analysis (FPCA) as a distinguisher for SCA is proposed by Souissi et al. in [12]. The procedure is to sort the total power consumption $\{P_{t,i}\}_{i=1}^{N}$ into different clusters $\{\{P_{t,i}\} | M \circ F_k(x_i) = m, t \in \boldsymbol{\tau}'\}$ under the key hypothesis k and power model M[4]. Mean vectors are computed within each cluster and combined into a matrix upon which PCA is subsequently performed. The FPCA distinguisher score is defined as the sum of the first m eigenvalues $\lambda_1, \ldots, \lambda_T$ associated with the PCA transformation.

Linear Discriminant Analysis Linear Discriminant Analysis (LDA) is another widely-used—in this case, supervised—dimensionality reduction method. It seeks the directions along which the projected data displays large between-cluster distances and small within-cluster distances. Suppose the original $N \times T$ size data, which is already sorted into p different clusters with j^{th} $(1 \leq j \leq p)$ cluster $\mathbf{C}_j$ has n_j vectors ($\sum\limits_{j=1}^{p} n_j = N$). The mean vector of of the whole data is μ and the mean vector of jth cluster is μ_j. The projection direction ω is given by,

$$S_B \omega = \lambda S_W \omega \tag{3}$$

where $S_B = \sum_{j=1}^{p} N_j (\mu_j - \mu)^T (\mu_j - \mu)$, $S_W = \sum_{j=1}^{p} \sum\limits_{x \in \mathbf{C}_j} (x - \mu_j)^T (x - \mu_j)$ represents the between-cluster scatter matrix and within-cluster scatter matrix respectively (for details see [6]). Performing LDA amounts to calculating the

[4] Because the hypothesised class labels are used to perform FPCA, it is no longer 'unsupervised' relative to that information.

generalized eigenvalues $\lambda_1, \ldots, \lambda_{T'}$ (from largest to smallest and $T' \leq T$) and the corresponding generalized eigenvectors eigenvector $\omega_1, \ldots, \omega_{T'}$.

The use of LDA as a DPA distinguisher is proposed by Mahmudlu et al. in [8]. Similar in procedure to FPCA, LDA-based DPA operates as follows: arrange the power consumption traces into clusters according to the key hypothesis and the power model; perform LDA on the labeled clusters; extract the first (largest) generalized eigenvalue as the distinguisher score for the key hypothesis.

2.3 Unsupervised Clustering for Semi-Profiled Power Models

Unsupervised clustering for robust semi-profiled power models was proposed by Whitnall et al. in [16]. The idea is to learn meaningful groupings of known intermediates displaying similar leakage characteristics. It can be regarded as a tradeoff between a non-profiled power model which can be easily used for attacks but might not precisely describe the power consumption and the profiled power model which can precisely describe the power consumption but might not be easily used in attacks. The procedure for semi-profiled modelling is as follows. First, sort the N_p w-width (subset of $\boldsymbol{\tau}$) profiling traces into different clusters according to the intermediate value $F_{k^*}(x_i)$ (F, k^*, x_i are known in the profiling phase). Second, the mean vector of each cluster is used to represent the cluster and PCA is performed to concentrate the relevant leakage information into fewer dimensions. Finally, an unsupervised clustering method such as K-means or hierarchical clustering is used to learn a partitioning on the reduced data. Thus, the intermediate values are mapped onto K cluster labels. This is then the power model, which can be paired with any cluster-based distinguisher (i.e. one which operates on a so-called 'nominal' model) in a (multivariate) attack phase.

3 Performance Evaluation

As demonstrated in [16], the parameters have some influence on the performance of the distinguishers. For the purpose of comprehensive comparison, we investigate the performance of the clustering distinguishers under different realizations of these parameters in this paper:

- The number of profiling power traces N_p used to profile the power model.
- The window width of profiling traces w_p and the window width of attacking traces w_a.
- The number of clusters K.

We also experiment with different leakage settings. We evaluate the performance of the clustering distinguishers on traces acquired from two unprotected implementations of AES—one software, running on an ARM microcontroller (10,000 traces total); one hardware, designed for an RFID-type system (5,000 traces total). Our chosen evaluation metric is the mean rank of the correct subkey [14].

3.1 Software Scenario

Influence of number of profiling traces N_p First, we consider the influence of the profiling sample size on the performance of the clustering distinguishers. For the software implementation, the attack intermediate value is the output of the first S-box. We denote the DCA distinguisher, variance ratio-based DCA distinguisher, FPCA distinguisher, LDA distinguisher by DCA, DCA-VR, FPCA and LDA respectively in the experimental results graphs hereafter. Since N_p is the only parameter under test here, we fix the window width of profiling and attack traces to 20, and restrict the number of clusters K to be no larger than 10, allowing the clustering procedure to test different values of K and choose the one producing the largest mean *silhouette index* (SI) as per [16].[5] Fig. 1 shows the guessing entropies of different clustering distinguishers under the clustering power models as profiled using different numbers of samples.

We first observe that the LDA distinguisher—the most recent to have been introduced for the purposes of side-channel key recovery—is actually less efficient than its predecessors, for all tested profiling sample sizes. A particular drawback of LDA is that it needs a certain number (and spread) of attack traces to return a meaningful distinguishing score; if samples are too small to evidence within-group scatter then the computations entail division by zero, leading to eigenvalues of 'infinite' value. We assign the maximum rank in such instances, which amounts to concluding that the attack has returned no information about the subkey. Table 1 reports the scale of the problem, which especially diminishes the ability for the distinguisher to succeed in attack sample sizes of 25 or smaller, regardless of the size of the profiling sample. In Section 4 we examine this phenomenon in more detail and explain why it is an inevitable feature of LDA.

As for the other distinguishers, when the profiling sample size is not sufficient (e.g. 200), DCA-VR (to our knowledge, the earliest of the three, dating back to 2008 [13]) appears to achieve fractionally better outcomes than DCA and FPCA. But for larger profiling samples (sufficient to profile the power model more accurately), the results of DCA, DCA-VR, and FPCA are almost the same.

Influence of window widths w_p and w_a We then test the influence of the widths of the profiling and attack trace windows (w_p and w_a respectively). As is clear from the previous subsection, more profiling traces will lead to better results, so for this part of the analysis we fix the number of profiling traces at 4000. Again, the number of clusters is not assigned but is constrained to be no larger than 10. The values of w_p and w_a we test are $\{4, 10, 20, 40\}$.

[5] The silhouette index is defined for the i^{th} object as $S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$, where a_i is the average distance from the i^{th} object to the other objects in the same cluster, and b_i is the minimum (over all clusters) average distance from the i^{th} object to the objects in a different cluster.

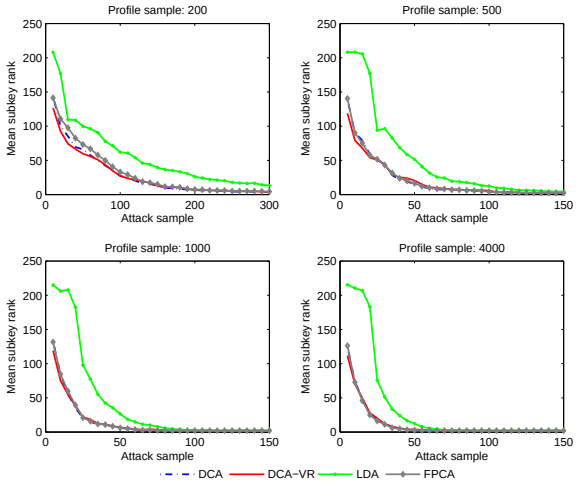

Fig. 1. Mean subkey rank of clustering distinguishers against the software implementation, as profiling sample size varies. (Window width: 20; reps: 100).

		Attack sample					
		5	**10**	**15**	**20**	**25**	**...**
Profile sample	**200**	145	189	200	152	0	0
	500	194	203	202	157	0	0
	1000	204	201	200	157	0	0
	4000	207	203	198	158	0	0

Table 1. Mean number of 'infinite' scores returned by the LDA distinguisher as profile and attack sample sizes vary (reps: 100, width: 20).

First, we consider the scenario in which the width of the profiling trace window is equal to that of the attacking trace window ($w_p = w_a$). The results are shown in Fig. 2. It seems that the DCA-VR is the most stable distinguisher as the window widths vary. The efficiencies of the DCA and FPCA distinguishers are almost equal. Both of them perform better when the widths become wider, in contrast with the LDA distinguisher, which performs worse as the widths increase.

Next, we focus on the scenario in which the width of the profiling window is *not* equal to that of the attacking window. Although we test all 4×4 pairwise combinations, for the purposes of presentation we focus on profiling widths w_p 4 and 20, in each case varying w_a as previously. The results are shown in Fig. 3 and Fig. 4. We observe that the DCA-VR performs best when the profiling window is narrow. The profiling window width has more of an impact than the attacking window width for the DCA-VR, DCA and FPCA distinguishers according to these two figures (the same holds for the remaining figures which are not presented here due to space restrictions). However, this is opposite for

the LDA distinguisher, which is affected more by the window width of the attack traces than that of the profiling traces.

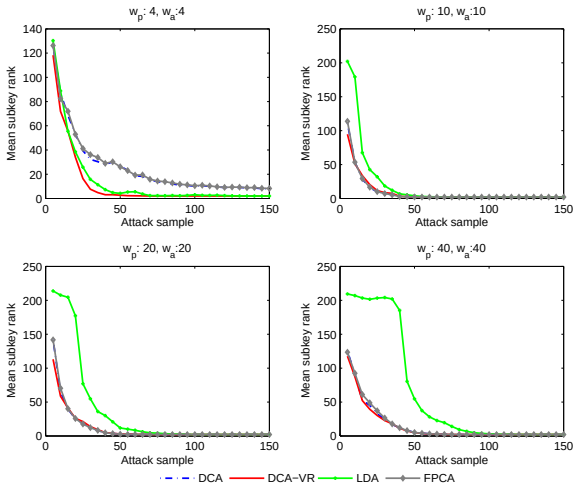

Fig. 2. Mean subkey rank of clustering distinguishers against software implementation, as window widths vary (reps: 100, w_p: window width of profiling traces, w_a: window width of attacking traces).

Influence of number of clusters K In the above subsections, rather than fixing the number of clusters K we let the clustering algorithm choose the number for each power model according to the SI. However, an 'optimal' clustering according to the SI need not necessarily imply optimality with regards to DPA performance. We therefore next explore how varying the number of clusters (from 2 to 8) affects the performance of the cluster-based distinguishers. As before, we fix the number of profiling traces at 4000, and we fix the profiling and attacking window widths at 20. The result is shown in Fig. 5. We clearly see that DCA-VR still performs best whatever the value of K. The performance of DCA is almost the same as that of FPCA, and both decrease as K increases. By contrast, the value of K seems to barely influence the performance of DCA-VR and LDA.

3.2 Hardware Scenario

Influence of number of profiling traces N_p We now move to consider the practical performance of the cluster-based distinguishers in the hardware setting. Preliminary investigations of the data acquired from the hardware implementation revealed considerable variation in the exploitability of the different S-boxes; we picked one (S-box 14) which was more amenable to attack in order to report

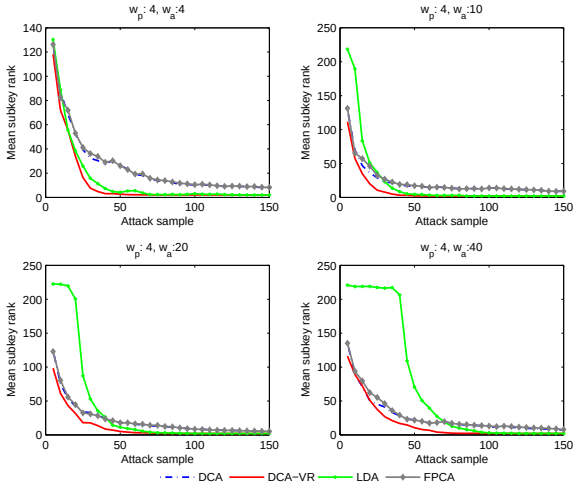

Fig. 3. Mean subkey rank of clustering distinguishers against software implementation, for a profiling window of width 4 as attacking window widths vary (reps: 100, w_p: window width of profiling traces, w_a: window width of attacking traces).

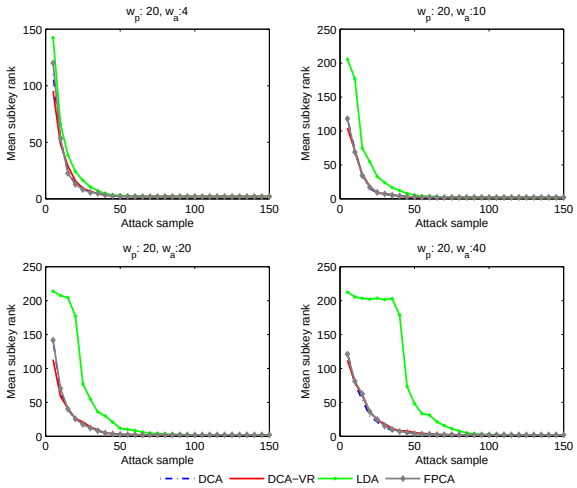

Fig. 4. Mean subkey rank of clustering distinguishers against software implementation, for a profiling window of width 20 as attacking window widths vary (reps: 100, w_p: window width of profiling traces, w_a: window width of attacking traces).

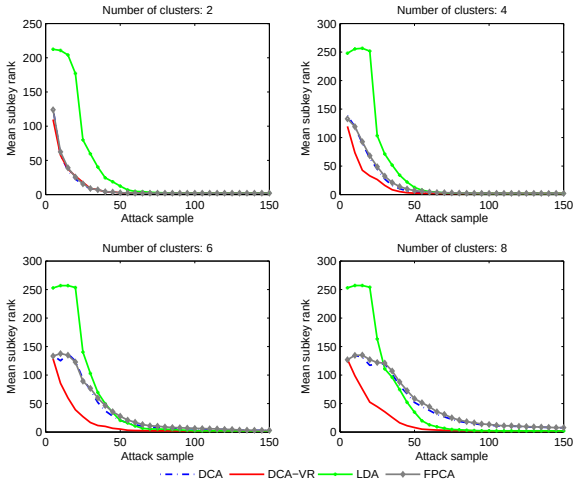

Fig. 5. Mean subkey rank of clustering distinguishers against software implementation, as the numbers of clusters varies (reps: 100, window width of profiling traces: 20, window width of attacking traces: 20).

interesting (but clearly not definitive) results. We first investigate the influence of the number of profiling traces N_p on the performance of the distinguishers. As done in the software scenario, we fix the window width (to 10 this time, owing to the coarser granularity of leakages from hardware, which typically runs in fewer clock cycles), and allow the cluster algorithm to select up to 10 clusters according to the SI. Fig. 6 shows the experimentally observed performance of these distinguishers given different numbers of profiling traces to profile the power model. Unlike the result in the software scenario, the DCA-VR is no longer the most efficient distinguisher. However, LDA still performs the least efficiently. Besides, as in the software scenario, distinguishing scores of 'infinite' value are frequently returned when the sample size is small; as before we interpret such outcomes as a failure to deduce anything about the key.

Influence of window widths w_p and w_a As before, we investigate the influence of window width on the performance of cluster-based distinguishers against hardware leakages. The power model is profiled using 4000 power traces with the number of clusters constrained to be no larger than 10, just as in the software scenario. The values of w_p and w_a we test are $\{4, 10, 20, 40\}$. First, we fix the attack window width w_a equal to the profiling window width w_p. The experimental result is indicated in Fig. 7. Then, the profiling window width w_p is fixed at 4 and then 10, while the attacking window width w_a is allowed to vary. The results are presented in Figs. 8 and 9.

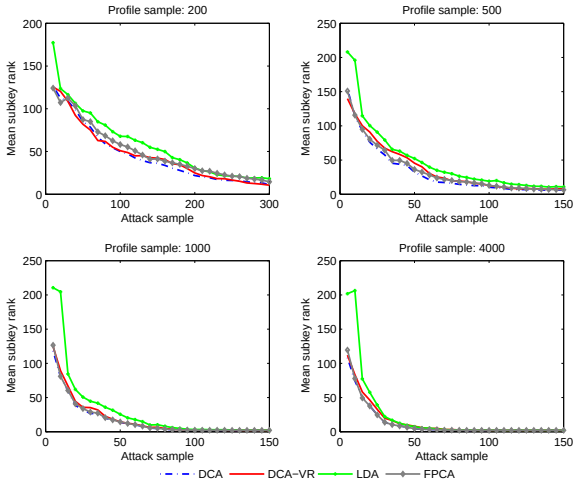

Fig. 6. Mean subkey rank of clustering distinguishers against the hardware implementation, as profiling sample size varies. (Window width: 10; reps: 100).

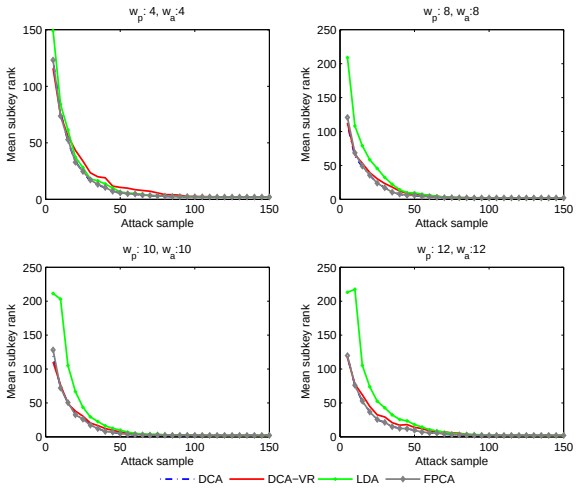

Fig. 7. Mean subkey rank of clustering distinguishers against hardware implementation, as window widths vary (reps: 100, w_p: window width of profiling traces, w_a: window width of attacking traces).

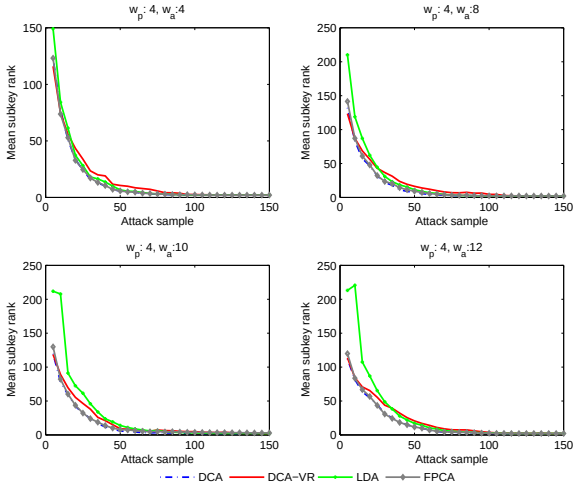

Fig. 8. Mean subkey rank of clustering distinguishers against hardware implementation, for a profiling window of width 4 as attacking window widths vary (reps: 100, w_p: window width of profiling traces, w_a: window width of attacking traces).

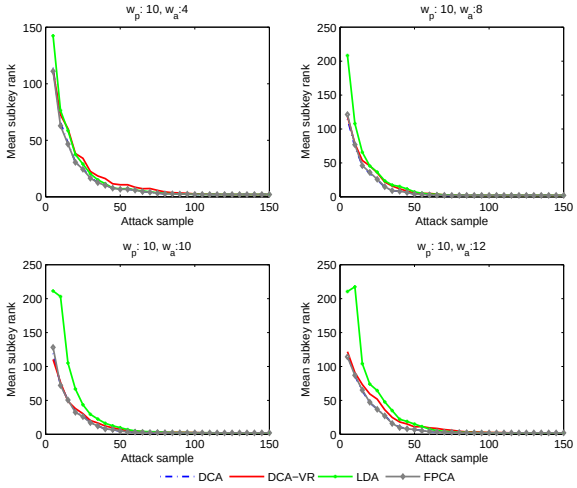

Fig. 9. Mean subkey rank of clustering distinguishers against hardware implementation, for a profiling window of width 10 as attacking window widths vary (reps: 100, w_p: window width of profiling traces, w_a: window width of attacking traces).

Influence of the number of clusters K Fig. 10 shows the distinguishers' performance when the power models are constructed to have different (fixed) numbers of clusters (keeping the window widths at 10). We observe that DCA, DCA-VR, and LDA distinguishers are stable as the number of clusters changes, with the relative performance summarised as DCA>DCA-VR>LDA. The performance of FPCA deteriorates as the number of clusters increases, just as in the software setting.

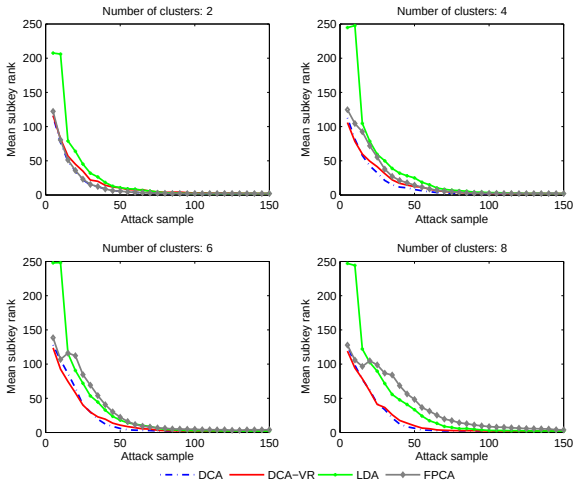

Fig. 10. Mean subkey rank of clustering distinguishers against hardware implementation, as the numbers of clusters varies (reps: 100, window width of profiling traces: 10, window width of attacking traces: 10).

3.3 Influence of Noise

Since LDA has been promoted as especially useful in scenarios exhibiting high levels of noise [8], we now explore the performance of all four distinguishers as noise increases. To do this, we simulate traces by adding Gaussian noise of increasing magnitude to the Hamming weight of intermediate value.

From Fig. 11 it can be observed that FPCA is detrimentally affected by the increase of noise, but the poor performance of LDA relative to DCA and DCA-VR is unchanged as the noise level increases. This is explained in part by the PCA dimensionality reduction step that all of the distinguishers share: LDA may have an advantage over methods that *don't* pre-process leakages to strengthen the signal, but among known approaches following a similar procedure it remains less efficient than the alternatives. Besides, the result of FPCA under Hamming weight model (9 clusters) also confirms the previous finding that it is affected by the number of cluster more (see Fig. 5 and Fig. 10).

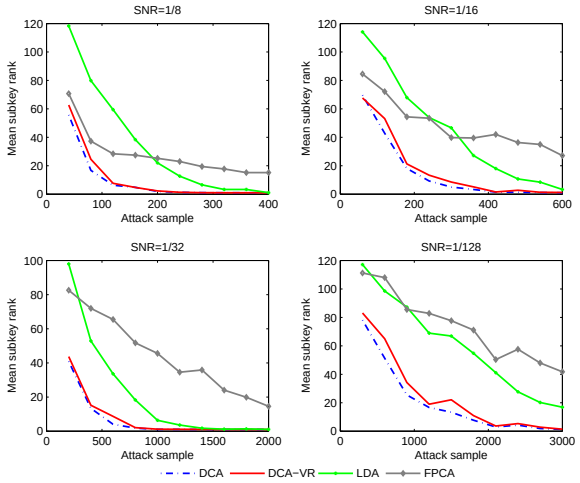

Fig. 11. Mean subkey rank of clustering distinguishers against simulated traces as noise varies (reps: 100, window width: 20).

4 Discussion

In this section, we unpack some of the theoretical similarities and differences of the cluster-based distinguishers.

4.1 Similarities

The basic operating procedure is the same for all four of the distinguishers considered: first partition the traces into different clusters $\{\mathbf{C}_j\}_{j=1}^{p}$ according to the key guess and the power model, then compute an indicator of 'cluster quality' to evaluate the extent to which the particular key guess produces a good partition. This strategy takes advantage of the fact that, for a correct key guess, the arrangement produced by the power model should correspond with the true cluster structure of the leakage measurements, so that the indicator value stands out by comparison with the wrong key guesses.

Specifically, for DCA, if the partition is correct, all traces within one cluster $\mathbf{C}_j$ would be 'close' to each other. Thus the indicator – the sum of the variances of each cluster – would be *low* for the correct key candidate. The DCA-VR is another kind of DCA, the indicator is the ratio between the overall variance and the weighted mean of the variances of each cluster, which would be *high* for the correct key candidate. FPCA exploits the fact that if the partition is correct then the mean traces within each cluster $\mathbf{C}_j$ are well separated from each other. Performing PCA on these mean traces finds the directions along which they exhibit the greatest dispersion. Since the eigenvalues associated with the projection directions measure this dispersion, the first (i.e. the largest) of these is chosen as the indicator; it should be maximal under the correct key

guess. Similarly, LDA finds the directions along which the clusters have small within-cluster scatter and large between-cluster scatter; the ratio of the latter to the former is the indicator in this instance and should (again) be largest for the correct key.

4.2 Differences

LDA We can see from all the experimental results that the LDA distinguisher's performance is much poorer than that of the others when the attack sample size is small (e.g. in the top left figure of Fig. 1, the mean subkey rank of LDA is about 200, compared with about 150 for the other distinguishers, given 5 attacking traces). As explained before, the reason for this is essentially that a certain number of observations are needed before the indicator can be properly computed. From equation (3) we are reminded that the indicator used by the LDA distinguisher, λ, is the eigenvalue of matrix $S_W^{-1}S_B$, where $S_W = \sum_{j=1}^{p} \sum_{x \in \mathbf{C}_j} (x-\mu_j)^T (x-\mu_j)$. Let Σ_j be the covariance matrix of $\mathbf{C}_j$. We get that $\sum_{x \in \mathbf{C}_j} (x-\mu_j)^T (x-\mu_j) = (n_j - 1)\Sigma_j$. When the number of traces in the j^{th} cluster n_j is smaller than the width of the traces w_a, the covariance matrix is a singular matrix. In this case, the S_W, as the sum of a number of singular matrices, might be still a singular matrix, in which case its inverse does not exist. Therefore, LDA is not well-suited to attack small sized samples. It can be useful in the scenario that the trace window width is small, but it seldom outperforms its (pre-dated) rivals.

DCA vs. FPCA The indicator of the DCA distinguisher in Section 2.2 can be rewritten as follows:

$$D_{DCA}(k) = \sum_{j=1}^{p} n_j ||\mu_j - \mu||^2 \tag{4}$$

where the symbols are defined as previously, and $||\cdot||^2$ denotes the squared Euclidean norm ($||z_1, z_2, ..., z_k||^2 = \sum_{i=1}^{k} z_i^2$). Equation (1) exploits the within-cluster variance; Equation (4) exploits the between-cluster variance. Since the sum of within-cluster variance and between-cluster variance is constant, minimizing (1) is exactly equivalent to maximizing (4). The indicator of FPCA λ is given by $\Sigma\omega = \lambda\omega$, where Σ is the covariance matrix of $\{\mu_j\}_{j=1}^{p}$.

$$\Sigma = \sum_{j=1}^{p} (\mu_j - \mu)^T (\mu_j - \mu) \tag{5}$$

Thus, both FPCA and DCA are related to the between-cluster variance. In the ideal environment[6] , their performances are almost identical.

[6] For the software implementation, the influence of noise is relatively small.

DCA vs. DCA-VR From Equations (1) and (2) it can be seen that the only material difference between DCA and DCA-VR is that DCA takes the total variance of each cluster while DCA-VR takes the weighted mean of the variances of each cluster, because the numerator of Equation (2) $\sum_{t \in \boldsymbol{\tau}'} \mathrm{var}(\{P_{t,i}\}_{i=1}^{N})^2$ is constant across all key hypotheses (i.e. no matter what the partition). In DCA-VR, two variables are monitored: the partition, and the cardinality of each cluster n_j. Under a correct key guess, these are both correct; otherwise, they are both wrong. So, in an ideal environment, the true key hypothesis is more clearly distinguished from the alternatives than by DCA, which only monitors the partition; in other words, DCA-VR is a 'reinforced' DCA that benefits from (correct or otherwise) information on the cluster sizes. However, for the hardware implementation, the noise is large and the power model is not as precise as in the software one, which leads to a non-ideal environment. The error on the cluster variance induced by the noise and the partition would be amplified by the weighting according to n_j. Thus, against the hardware implementation, the performance of DCA-VR is slightly less efficient than DCA.

5 Conclusion

Our empirical comparison of the various different suggestions for cluster-based DPA has revealed that the variance ratio (DCA-VR) – to our knowledge, the earliest proposal, dating back to Standaert et al. in 2008 [13] – consistently either *is*, or at least closely rivals, the best performing distinguisher of its type. This is observed across the two example scenarios tested and as implementation parameters vary. By contrast, FPCA and DCA (which are conceptually very close) perform strongly in some scenarios (especially in the case of hardware leakages, where they marginally outperform DCA-VR) but are less robust to changes in parameters. The most recent proposal, LDA, is disadvantaged by the requirement for a certain minimum number of data points before the distinguishing scores can be computed, and is typically less efficient and less robust than its competitors, even in high noise scenarios where it has been especially advocated for use. We therefore conclude that, whilst it is interesting to seek out alternative means of exploiting semi-profiled leakage information, for the time being it would seem that established methodologies remain the best option for practitioners.

Acknowledgements and Disclaimer The authors would like to thank Thomas Korak, Thomas Plos and Michael Hutter at TU Graz for supplying us with data from the TAMPRES project [1,7]. This work was supported by the National Natural Science Foundation of China (No.61372062), by the European Union's H2020 Programme under grant agreement number ICT-731591 (REASSURE). No research data was created for this paper.

References

1. Tampres: Tamper resistant sensor nodes. http://www.tampres.eu, 2009-2013
2. Batina, L., Gierlichs, B., Lemke-Rust, K.: Differential cluster analysis. In: Cryptographic Hardware and Embedded Systems-CHES 2009, pp. 112–127. Springer (2009)
3. Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J.J. (eds.) Proceedings of CHES 2004. LNCS, vol. 3156, pp. 135–152. Springer Berlin / Heidelberg (2004)
4. Chari, S., Rao, J., Rohatgi, P.: Template Attacks. In: Kaliski, B., Koç, Ç., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 51–62. Springer Berlin / Heidelberg (2003)
5. Elaabid, M., Guilley, S.: Portability of templates. J. Cryptographic Engineering 2(1), 63–74 (2012)
6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(2), 179–188 (1936)
7. Korak, T., Plos, T., Hutter, M.: Attacking an AES-enabled NFC tag: Implications from design to a real-world scenario. In: International Workshop on Constructive Side-Channel Analysis and Secure Design. pp. 17–32. Springer (2012)
8. Mahmudlu, R., Banciu, V., Batina, L., Buhan, I.: LDA-Based Clustering as a Side-Channel Distinguisher (2016)
9. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer (2007)
10. Mangard, S., Oswald, E., Standaert, F.X.: One for all–all for one: unifying standard differential power analysis attacks. IET Information Security 5(2), 100–110 (2011)
11. Renauld, M., Standaert, F.X., Veyrat-Charvillon, N., Kamel, D., Flandre, D.: A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 109–128. Springer (2011)
12. Souissi, Y., Nassar, M., Guilley, S., Danger, J.L., Flament, F.: First principal components analysis: a new side channel distinguisher. In: International Conference on Information Security and Cryptology. pp. 407–419. Springer (2010)
13. Standaert, F.X., Gierlichs, B., Verbauwhede, I.: Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected cmos devices. In: International Conference on Information Security and Cryptology. pp. 253–267. Springer (2008)
14. Standaert, F.X., Malkin, T.G., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 443–461. Springer (2009)
15. Whitnall, C., Oswald, E.: A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. J. Cryptographic Engineering 1(2), 145–160 (August 2011)
16. Whitnall, C., Oswald, E.: Robust profiling for DPA-style attacks. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 3–21. Springer (2015)
17. Whitnall, C., Oswald, E., Standaert, F.X.: The Myth of Generic DPA...and the Magic of Learning. In: Benaloh, J. (ed.) CT-RSA. LNCS, vol. 8366, pp. 183–205. Springer (2014)