A Novel Use of Kernel Discriminant Analysis as a Higher-Order Side-Channel Distinguisher

Xinping Zhou[1,2,⋆], Carolyn Whitnall[3], Elisabeth Oswald[3], Degang Sun[1,2], and Zhu Wang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, P.R. China
[2] School of Cyber Security, University of Chinese Academy of Sciences, P.R. China
{zhouxinping, sundegang, wangzhu}@iie.ac.cn
[3] Department of Computer Science, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK
{Carolyn.Whitnall, Elisabeth.Oswald}@bristol.ac.uk

Abstract. Distinguishers play an important role in Side Channel Analysis (SCA), where real world leakage information is compared against hypothetical predictions in order to guess at the underlying secret key. However, the direct relationship between leakages and predictions can be disrupted by the mathematical combining of d random values with each sensitive intermediate value of the cryptographic algorithm (a so-called "d-th order masking scheme"). In the case of software implementations, as long as the masking has been correctly applied, the guessable intermediates will be independent of any one point in the trace, or indeed of any tuple of fewer than $d + 1$ points. However, certain $d + 1$-tuples of time points may jointly depend on the guessable intermediates. A typical approach to exploiting this data dependency is to pre-process the trace – computing carefully chosen univariate functions of all possible $d + 1$-tuples – before applying the usual univariate distinguishers. This has a computational complexity which is exponential in the order d of the masking scheme. In this paper, we propose a new distinguisher based on Kernel Discriminant Analysis (KDA) which directly exploits properties of the mask implementation without the need to exhaustively pre-process the traces, thereby distinguishing the correct key with lower complexity. Experimental results for 2nd and 3rd order attacks (i.e. against 1st and 2nd order masking) verify that the KDA is an effective distinguisher in protected settings.

Keywords: Kernel Discriminant Analysis, Higher-order Side Channel Analysis, Side Channel Distinguisher

1 Introduction

Protecting sensitive information from attacks exploiting the physically observable characteristics of cryptographic devices in operation has been a key aim for

⋆ This work was done while the author was in the Department of Computer Science, University of Bristol.

vendors and evaluation labs ever since the devastating effectiveness and simplicity of such attacks began to become apparent with the work of Kocher et al. in the late 1990s [11]. Software countermeasures such as masking [8] successfully disrupt the relationship between sensitive intermediate values and single points of observed leakage – precisely the trace feature that Differential Power Analysis (DPA) in particular targets[4]. However, *tuples* of points of size greater than the number of masks d can still *jointly* depend on the sensitive intermediates. This gives rise to so-called 'higher order' DPA [14], which typically proceeds by combining multiple points via some (non-linear) pre-processing function before applying a standard DPA distinguisher – essentially treating the pre-processed traces in a univariate manner, albeit with an exponential (in d) increase in the impact of noise relative to a 'first order' attack [22].

Aside from the greater data complexity implied by the inflated noise, higher-order attacks are also hampered by the increasing difficulty of locating the leaking tuples. The computational complexity of an 'exhaustive search' approach – in which all possible point combinations are computed and analysed – grows exponentially with d. Heuristics exist to reduce the search problem by placing informed restrictions on the regions of the trace to be iteratively explored [9] but, precisely because of their heuristic nature, these do not guarantee to find the best (or indeed any) exploitable combinations. A recent proposal (presented at Cardis 2016 [6]) aims to bypass the need for explicit enumeration of the $(d+1)$-tuples without recourse to heuristics, using Kernel Discriminant Analysis (KDA) [15].

KDA is a generalisation of Linear Discriminant Analysis (LDA), a statistical method to find linear combinations of features (i.e. variables in a dataset, or points in a trace) that characterise class separations. In particular, it outputs projection directions that maximise the ratio of between-group to within-group scatter, so that 'interesting' variation may be concentrated into a reduced-dimension space for further analysis. LDA has been promoted as one of a number of methods to extract sensitive data dependent features from side-channel traces for some years (beginning with [24], to the best of our knowledge). However, because it only finds linear combinations, it is unable to locate the types of *joint* data dependencies exhibited by traces which have been protected by software masking. By contrast, the 'kernel trick' employed by KDA allows to implicitly map the data into a higher dimensional feature space within which to perform the discriminant analysis, thereby extracting non-linear combinations of the sort that (in the case of DPA) *do* yield sensitive information on further analysis. Because the mapping of the tuple candidates need not be computed explicitly (by contrast with the preprocessing required by established higher order DPA methodologies), its complexity is polynomial, rather than exponential, in d.

However, another recent development in the literature has been to demonstrate the direct applicability of LDA as a side-channel *distingisher*, not just a pre-processing method. In this capacity, it shares the advantages of other

[4] Hardware masking schemes also exist, which process shares in parallel but shift the exploitable leakage into higher moments of the (univariate) trace distributions[18].

'partition-based' [25] (aka 'nominal power model' based [30]) distinguishers – namely that it operates needing only a clustering of the intermediates into similarly leaking classes, rather than (e.g.) a proportional approximation of the leakage as would be necessary for a correlation DPA attack. It is therefore natural to suppose that KDA can similarly be extended for use as a distinguisher, with the same flexibility advantages over higher-order correlation DPA that LDA has over first-order correlation, as well as the reduction in complexity with respect to d. Indeed, in the following, we confirm that this is the case – KDA can be used, not just to locate the interesting leakage prior to an attack, but as a side-channel distinguisher in its own right. We show how to achieve this, provide experimental validation of the effectiveness of our methodology, and reason about its potential as well as its drawbacks.

1.1 Outline

The rest of the paper proceeds as follows. Section 2 covers the preliminaries of (higher-order) SCA, LDA and KDA. In Section 3 we describe the natural connection between KDA and the higher-order SCA problem, and present a methodology to extract sensitive information using KDA, before going on to experimentally verify its effectiveness. Section 4 discusses the efficiency and advantages (and drawbacks) of our proposed approach, and Section 5 concludes the paper.

2 Preliminaries

2.1 Differential Power Analysis

We consider a 'standard DPA attack' scenario as defined in [13], 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, [13]). 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 Masking

Since the scale of the threat of (first-order) DPA began to emerge [11], many countermeasure schemes have been proposed. The principle behind masking is to split the sensitive intermediate values $s = F_{k^*}(x)$ into $d+1$ shares $(r_0, ..., r_d \in \mathcal{Z})$ satisfying the relation[5]

$$s = r_0 \otimes r_1 \otimes ... \otimes r_d$$

where the $\otimes$ operation is the bitwise addition (or XOR) in the common case of Boolean masking. One of the shares, e.g. r_0, is sometimes referred to as the 'masked variable', with the other shares, $(r_1, ..., r_d)$ then viewed as the 'masks'. For a masking scheme to be sound, it is usually required that the masks are uniformly and independently generated from $\mathcal{Z}$. In the case of software implementations, which we focus on here, the shares are processed in sequence so that side-channel leakages are distributed across multiple points in the measured traces.

Classical Higher-Order DPA In the case of a masked implementation, the leakage of the shares corresponding to the sensitive value s is

$$\mathbf{l} = (l_0, l_1, ..., l_d)$$

where

$$\begin{aligned} l_0 &= L_0 \circ (s \oplus r_1 \oplus \ldots \oplus r_d) + \varepsilon_0 \\ l_i &= L_i \circ (r_i) + \varepsilon_i, \qquad \text{for } 1 \leq i \leq d. \end{aligned}$$

It can be seen that no single component of the leakage $\mathbf{l}$ directly relies on s. The first order distinguisher will be unable to learn anything about the secret key k^* in this case.

During a higher-order DPA, an attacker extracts information about k^* by monitoring the leakage of the unknown shares. Generally speaking, the dth order masking scheme can be attacked by a $(d+1)$th order attack. Since it is difficult for the attacker to precisely determine the location of l_i, we assume that ℓ time point candidates can be discovered for each share by some reverse engineering or *a*

[5] This relation exists implicitly even when it doesn't manifest directly in the cryptographic algorithm.

priori knowledge about the masked implementation. Thus, $(d+1)$-tuples of ℓ time points are available for analysis. To analyse the $(d+1)\ell$ time points by classical higher-order DPA, a 'combination function' is required – although it has been observed that such an approach inevitably incurs loss of information [17,27]. The most popular combination function is probably the normalised product, shown in [19] to be the optimal choice in the idealised setting of a correlation attack against Hamming weight leakage with Gaussian noise; other proposals include the absolute difference [14], and some more complex expressions involving sine functions [17].

Note that the combination function operates as a pre-processing procedure on all possible $(d+1)$-tuples of time points. This implies ℓ^{d+1} computations, resulting in ℓ^{d+1} points for analysis via a first order distinguisher D (typically correlation [3]) paired with a power model (recall Section 2.1).

2.3 Kernel Discriminant Analysis

Linear Discriminant Analysis Linear Discriminant Analysis (LDA) is a widely-used (supervised) dimensionality reduction method. It seeks the directions along which the projection of a dataset displays large between-cluster distances and small within-cluster distances. Suppose $\mathbf{P}_i$ is row vector of a matrix $\mathbf{P} \in \mathbb{R}^{N \times U}$ with labels $\mathbf{m} \in \mathbb{R}^{N \times 1}$; then the LDA problem amounts to finding ω to maximize $J(\omega)$ in (1):

$$J(\omega) = \frac{\omega^T S_B \omega}{\omega^T S_W \omega} \tag{1}$$

This procedure is equivalent to solving (2)

$$S_B \omega = \lambda S_W \omega \tag{2}$$

where S_B and S_W represent the between-cluster and within-cluster scatter matrices given by (3) and (4) respectively

$$S_B = \sum_{m \in \mathcal{M}} n_m \left(\frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i - \frac{1}{N} \sum_{i=1}^{N} \mathbf{P}_i \right)^T \left(\frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i - \frac{1}{N} \sum_{i=1}^{N} \mathbf{P}_i \right) \tag{3}$$

$$S_W = \sum_{m \in \mathcal{M}} \sum_{m_i = m} \left(\mathbf{P}_i - \frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i \right)^T \left(\mathbf{P}_i - \frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i \right) \tag{4}$$

where $n_m = \#\{i | m_i = m\}$, i.e. the number of observations in the data set for which the label is m (for details see [10]). Performing LDA amounts to calculating the generalized eigenvalues $\lambda_1, \ldots, \lambda_U$ (ordered from largest to smallest) and the corresponding generalized eigenvectors $\omega_1, \ldots, \omega_U$.

The applications of LDA to SCA are two-fold. On the one hand, it can be used for dimensionality reduction, addressing the problem of 'interesting point

selection' by (hopefully) projecting relevant leakage information along a small number of directions prior to further analysis [24,5]. It has been shown to be the optimal strategy for this purpose – at least in the case of unprotected implementations, where the leakage of sensitive intermediates resides in the marginal distributions of single trace points [4]. The procedure is as follows: sort the total power consumption $\{\mathbf{P}_i\}_{i=1}^N$ into different clusters $\{\{\mathbf{P}_i\}\,|M \circ F_k(x_i) = m\}$ under the correct key k^* and power model M; perform LDA on the labeled clusters; extract the eigenvectors $\omega_1, \ldots, \omega_u \quad (u \leq U)$ corresponding with the first u largest eigenvalues, i.e. the u 'best' projected directions. Projecting the data along these u directions produces a dataset of lower dimension but with minimal information loss.

On the other hand, it has also recently been proposed for use directly as a DPA distinguisher [12]. To this end it operates as follows: sort the total power consumption $\{\mathbf{P}_i\}_{i=1}^N$ into different clusters $\{\{\mathbf{P}_i\}\,|M \circ F_k(x_i) = m\}$ under the key hypothesis k and power model M; perform LDA on the labeled clusters; extract the first (largest) generalized eigenvalue as the distinguisher score for the key hypothesis. 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.

Discriminant Analysis with Kernels LDA can be used to find optimal linear mappings of high dimensional data but is not applicable when the relevant information is known to be contained in non-linear combinations of points, as is the case (e.g.) for side-channel leakages of masked implementations. To extend LDA to the non-linear case, we consider the problem in a feature space $\mathcal{F}$ induced by some mapping function (this mapping process is implicit as will be seen in the following subsection), $\Phi : R^n \to \mathcal{F}$. KDA [15] is used to find non-linear directions by first mapping the data non-linearly by Φ into some feature space $\mathcal{F}$ within which to compute linear discriminants, thus implicitly yielding a non-linear discriminant in the input space. To find such a discriminant, the goal (1) is replaced with:

$$J(\omega') = \frac{\omega'^T S_B^\Phi \omega'}{\omega'^T S_W^\Phi \omega'} \tag{5}$$

where $\omega' \in \mathcal{F}$ and S_B^Φ and S_W^Φ are the corresponding matrices in $\mathcal{F}$.

$$S_B^\Phi = \sum_{m \in \mathcal{M}} n_m \Big(\frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i^\Phi - \frac{1}{N} \sum_{i=1}^N \mathbf{P}_i^\Phi\Big)^T \Big(\frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i^\Phi - \frac{1}{N} \sum_{i=1}^N \mathbf{P}_i^\Phi\Big) \tag{6}$$

$$S_W^\Phi = \sum_{m \in \mathcal{M}} \sum_{m_i = m} (\mathbf{P}_i^\Phi - \frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i^\Phi)^T (\mathbf{P}_i^\Phi - \frac{1}{n_m} \sum_{m_i = m} \mathbf{P}_i^\Phi) \tag{7}$$

where $\mathbf{P}_i^\Phi$ is $\Phi(\mathbf{P}_i)$ projection of $\mathbf{P_i}$ on $\mathcal{F}$ by Φ. For a properly chosen Φ, an inner product $< \cdot, \cdot >$ can be defined on $\mathcal{F}$, which makes for a so-called 'reproducing

kernel Hilbert space',

$$K(\mathbf{x}, \mathbf{y}) =< \Phi(\mathbf{x}), \Phi(\mathbf{y}) >$$

where K is known as the kernel function. Widely-used kernel functions include the Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = exp(-||\mathbf{x} - \mathbf{y}||^2/c)$ ($||\cdot||$ is the 2-norm), and the polynomial kernel $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^{d'}$, for positive constants c and d' satisfying Mercer's condition [20], as defined in [21].

Procedure of KDA Generally, given a labelled data set $\{\mathbf{P}_i\}_{i\in[1,N]}$, the corresponding labels m_i (for simplicity, we use the notation $\mathbf{P}_i^{m_i}$ to denote the label of ith data sample is m_i), and the kernel function $K(\mathbf{x}, \mathbf{y})$, the KDA procedure can be briefly summarised as follows (for more details about the derivation, see [15]):

1. Calculate the between-class scatter matrix $\mathbf{M} \in R^{N\times N}$

$$\mathbf{M} = \sum_{m\in\mathcal{M}} n_m(\mathbf{M}_m - \mathbf{M}_*)(\mathbf{M}_m - \mathbf{M}_*)^T$$

   where $\mathbf{M}_m$ and $\mathbf{M}_*$ are $N \times 1$ size column vectors given by

$$(\mathbf{M}_m)_j = \frac{1}{n_m} \sum_{m_i=m} K(\mathbf{P}_j, \mathbf{P}_i^{m_i})$$

$$(\mathbf{M}_*)_j = \frac{1}{N} \sum_{i=1}^{N} K(\mathbf{P}_j, \mathbf{P}_i).$$

2. Then calculate the within-class scatter matrix $\mathbf{N} \in R^{N\times N}$ given by

$$\mathbf{N} = \sum_{m\in\mathcal{M}} \mathbf{K}_m(\mathbf{I}_{n_m} - \mathbf{1}_{n_m})\mathbf{K}_m^T$$

   where $\mathbf{K}_m$ is an $N \times n_m$ matrix with $(\mathbf{K}_m)_{ij} = K(\mathbf{P}_i, \mathbf{P}_j^{m_j=m})$ (this is the kernel matrix for class m), $\mathbf{I}_{n_m}$ is the $n_m \times n_m$ size identity matrix, and $\mathbf{1}_{n_m}$ is the $n_m \times n_m$ matrix with all entries $1/n_m$.
3. The eigenvalues $\lambda_1^{'}, \ldots, \lambda_{N'}^{'}$ ($N' \leq N$) and the eigenvectors $\alpha_1, \ldots, \alpha_{N'}$ can be extracted by solving

$$\mathbf{N}^{-1}\mathbf{M}\alpha_i = \lambda_i^{'}\alpha_i. \tag{8}$$

4. Since the matrix $\mathbf{N}$ may be singular, it needs regularizing prior to Step 3, which is done by setting

$$\mathbf{N} = \mathbf{N} + \mu\mathbf{I}$$

   for some positive μ.
5. Then the projection of $\mathbf{P}$ onto ω_i' is given by

$$< \omega_i', \Phi(\mathbf{P}) >= \sum_{j=1}^{N} \alpha_i(j)K(\mathbf{P}_j, \mathbf{P}). \tag{9}$$

Generally, KDA is used to transform U-dimensional data into C-dimensional data by taking the C eigenvectors $\alpha_1, \ldots, \alpha_C$ with the C largest eigenvalues and using Equation (9) for the projection step.

KDA has been introduced to SCA as a tool for information extraction[6] in the presence of masking [6]. The authors sort the training data set into different clusters according to the sensitive intermediate value under a known power model, the plaintext and the known key. Then KDA is performed on these clusters to calculate the eigenvectors and corresponding eigenvalues. The two eigenvectors with the two largest eigenvalues are chosen as the projection directions (i.e. C is set to be 2) and used to transform profiling and attack acquisitions prior to performing a template attack.

3 Methodology

In this section, we introduce our proposed distinguisher to the setting of masked implementations, and analyse the method theoretically and empirically.

Due to the successful removal of sensitive intermediate values by masking, classical higher-order side-channel attacks typically proceed by first transferring the original trace points into a new space using a non-linear combination function (CF)[7]. Then, 'first order' distinguisher scores are computed in the new space. In fact, the KDA method combines these two processes (summarised in Fig. 1 and Fig. 2) without performing the non-linear mapping explicitly (the kernel trick embeds it implicitly).

$$R^{(d+1)\ell} \xrightarrow{CF} R^{\ell^{d+1}} \xrightarrow{D} k^*$$

Fig. 1. Classical higher-order SCA.

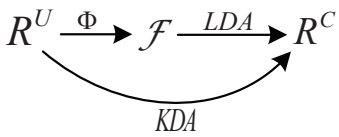

Fig. 2. Process of KDA.

[6] Information extraction is typically understood to refer collectively to the similar but non-identical tasks of dimensionality reduction and interesting point selection.

[7] The combination functions mentioned in Subsection 2.2 all are non-linear.

The calculations of between-class scatter matrix $\mathbf{M}$ and the within-class matrix $\mathbf{N}$ are based on the category (see procedure of KDA). And the maximum eigenvalue in Equation (8) can be regarded as an indicator of dispersion degree of between-class and within-class (as can be seen in Equation (1,2,5)). Thus, based on different (correct or wrong) categories, the dispersion degrees will differ. Therefore, the largest KDA eigenvalue functions as an effective distinguisher for attacks against masked intermediates.

3.1 General Approach

Let $\{x_i\}_{i=1}^{N}$ be the known plaintexts (or ciphertexts) associated with a set of trace measurements $\{\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N\}$ each containing $d\ell$ time points as a dth-order masked implementation encrypts (or decrypts) the x_i. The power model mapping is $M : \mathcal{Z} \to \mathcal{M}$ and the kernel function is chosen as[8] $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^{d'}$ (in this paper we set the degree d' of the polynomial kernel function equal to the number of shares $d+1$ into which each sensitive intermediate is divided). $\mathcal{D}$ denote the KDA distinguisher.

1. For each key hypothesis $k \in \mathcal{K}$, do the following:
   (a) Calculate the intermediate value $z_i = F_k(x_i)$ for each plaintext.
   (b) Map z_i to a power model prediction m_i, given by $M(z_i)$.
   (c) Compute the between-class scatter matrix $\mathbf{M}$ and the within-class scatter matrix $\mathbf{N}$, and regularize $\mathbf{N}$ by $\mathbf{N} = \mathbf{N} + \mu\mathbf{I}$.
   (d) Eigen-decompose the matrix $\mathbf{N}^{-1}\mathbf{M}$. Return the largest eigenvalue as the distinguisher score $\mathcal{D}_k$ for k.
2. Rank the pairs $(k, \mathcal{D}_k)$ according to $\mathcal{D}_k$.
3. Output the key hypothesis k with the largest $\mathcal{D}_k$ as the best guess on the true subkey.

3.2 Theoretical Rationale

In this subsection, we reason about the 'soundness' of the KDA distinguisher. We consider a distinguisher to be 'sound' if, given a sufficient sample of leakages and a 'meaningful' power model[9], it reduces the entropy on the unknown secret key. From an empirical perspective, soundness can be confirmed by observing a reduction in the mean key rank as the number of traces increases.

The essence of KDA is to transfer the raw data into a new, higher-dimensional space via an implicit non-linear projection, then compute the linear discriminant in the new space. This parallels the process of higher-order attacks in SCA. Thus, the soundness of KDA derives from 1) the effectiveness of the implicit projection, and 2) the effectiveness of LDA as a distinguisher in the first-order scenario.

[8] We only test this one example kernel function in our analysis; others, such as Gaussian kernel, are also available and may be effective.

[9] I.e. one that approximates some true aspect of the leakages; see, e.g. [30].

The effectiveness of the projection has been recently demonstrated by the successful use of KDA to extract exploitable side-channel information [6]. Meanwhile, Mahmudlu *et al.* [12] have shown that the largest eigenvalue, which measures the (optimised) between- to within-scatter matrix ratio under a particular key guess, is typically higher for a correct guess (which produces a meaningful labelling on the traces) than an incorrect one (which produces a random labelling), thereby functioning as an effective distinguishing score.

It therefore seems reasonable to expect our proposed KDA distinguisher to be sound; the following experiments are designed to verify this.

3.3 Experimental Validation

We here present the outcomes of several experiments on simulated leakages and (in the case of second-order attacks only) on traces from real implementations to verify the soundness of KDA distinguisher.

We simulate multivariate leakages pertaining to shared intermediates in the presence of Gaussian noise. The basic principle is to add multivariate Gaussian noise ε to the hypothetical data-dependent consumption of the intermediate z,

$$l = M(z) + \varepsilon_G \tag{10}$$

where M is the leakage model (chosen to be the Hamming weight for the following), and z is the intermediate value. The Gaussian noise ε_G has zero mean and a covariance $\mathbf{\Sigma}$ given by,

$$\mathbf{\Sigma} = \mathbf{Q} * \rho * \mathbf{Q} \tag{11}$$

where $\mathbf{Q}$ is a diagonal matrix whose diagonal elements are the noise standard deviation σ and ρ is a co-correlation matrix estimated from real power traces.

For a dth order masked implementation, we simulate a trace of $d+1$-tuples of ℓ points with the secret key k^* as follows:

1. Generate $d+1$ random numbers $(x, r_1, r_2, ..., r_d)$ (the first random number, x, is the plaintext; the rest are masks).
2. For the first ℓ points in the trace, the intermediate values are the output of the XOR between the sensitive intermediate values $s = Sbox(x \oplus k^*)$ and the masks.
3. For the ith $(2 \leq i \leq d+1)$ sub-part of the trace, the intermediate value is r_{i-1}.

The Hamming weights of these intermediates are computed and additively combined with simulated noise samples of the specified Gaussian structure.

The real power traces are taken from the DPA Contest v4 [1]. The target is an 8-bit AVR microcontroller Atmega163 embedded in a smartcard. It contains 16Kb of in-system programmable flash, 512 bytes of EEPROM, 1Kb of internal SRAM and 32 general purpose working registers. The smartcard is read using a simple reader interface mounted on SASEBO-W board and controlled by Xilinx Spartan-VI FPGA. The traces are acquired using a LeCroy WaveRunner 6100A

oscilloscope using an EM probe. The acquisition bandwidth is 200 MHz and the sampling rate FS = 500 MS/s.

In the following experiments, the kernel function is $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^{d+1}$.

Second-order attacks First, we perform second-order attacks on the simulated leakage. To keep the running time reasonable, we simulate 200,000 traces, and set ℓ to 5 so that the traces are 10 time points long. The noise deviation σ of the trace is set equal to 1, and μ for the KDA regularisation is set to be 100,000[10]. We use the second-order KDA distinguisher with the Hamming weight power model to attack the traces, the results of all key candidate distinguisher scores are shown in Fig.3. The red line indicates the correct key. We can clearly see that, from 800 traces on, the distinguisher score associated with the correct key gradually separates from the scores for the alternative candidates, standing out first from 1500 traces onwards.

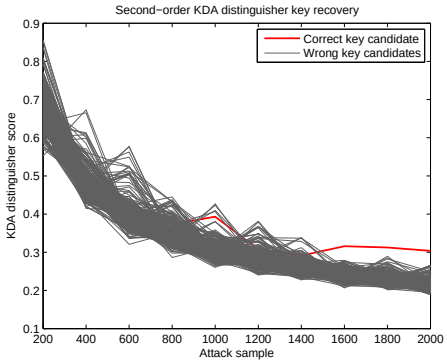

Fig. 3. Second-order attack with KDA on the simulated masked implementation leakage, with $\sigma = 1$.

Fig. 3 just shows the example result of a single trial; it cannot be interpreted as a stable indicator of the typical behaviour of the KDA distinguisher. To evaluate the performance of KDA we repeat the experiment multiple times (randomly selecting from the pool of 200,000 traces in each repetition). Our chosen evaluation metric is the Guessing Entropy [26], estimated from the average rank of the correct subkey. The result can be seen in Fig. 4. The red line indicates the second-order KDA distinguisher using a Hamming weight power model. The mean rank of the correct key decreases as the number of attack samples increases, converging to 1 after about 1900 traces.

Additionally, to further extend the experiments on KDA, we drop the Hamming weight assumption and investigate the performance of KDA when the at-

[10] $\mu = 100,000$ might not be the optimal one; we leave the optimisation μ as further work.

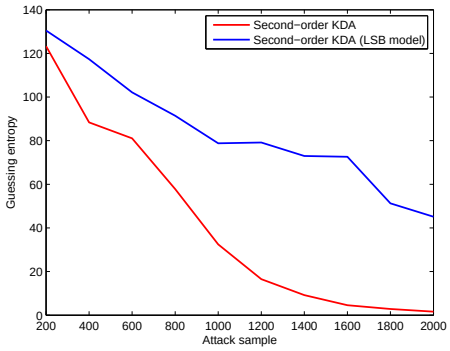

Fig. 4. Guessing entropy of second-order attack with KDA on the simulated masked implementation leakage, with $\sigma = 1$. (reps: 100)

tack samples are simply partitioned according to the least significant bit of the intermediate value[11].Thus, the traces are separated into two clusters, labelled 0 and 1. The results are represented by the blue line in Fig. 4: the mean rank of the correct key decreases as the number of traces increases, implying that the KDA distinguisher remains sound under this simpler power model.

Second, we test the performance of KDA against the real power traces from the DPA contest v4. The mask scheme implemented is the Rotating S-boxes Masking (RSM; for details, see [16]). The RSM scheme involves random masks and random offsets. There already exist several methods to attack these traces, as shown on the website [1]; we don't promote our KDA distinguisher as the optimal one, we simply make use of the data as a scenario in which to demonstrate its effective performance. We focus only on the second-order attack. It is a characteristic of the RSM scheme that the output of the masked S-box and the masked value of next sub-plaintext have the same mask, so that their XOR result can remove the mask. In detail, the first part is $MSbox(x_i \oplus k \oplus r_{i+offset})$, and the second part is $x_{i+1} \oplus r_{i+1+offset}$ where $MSbox$ is the masked sbox, i is the index of the sub-plaintext, $offset$ is a random number, and r is a mask table. According to the description of the RSM algorithm, the first part can be expressed as

$$MSbox(x_i \oplus k \oplus r_{i+offset}) = Sbox(x_i \oplus k) \oplus r_{i+1+offset}$$

Hence, the XOR result of the two parts is $Sbox(x_i \oplus k) \oplus x_{i+1}$ which, although slightly different to the intermediates targeted in the simulated leakage scenario, can be computed for each given key guess.

We choose 10 time points for each part as guided by a preliminary investigation of the traces. As for the previous experiment, we ran the second-order KDA distinguisher 100 times with randomly selected sub-samples of the traces. The guessing entropy results are presented in Fig. 5. We observe that, for a

[11] Sometimes referred to as the 'LSB model'.

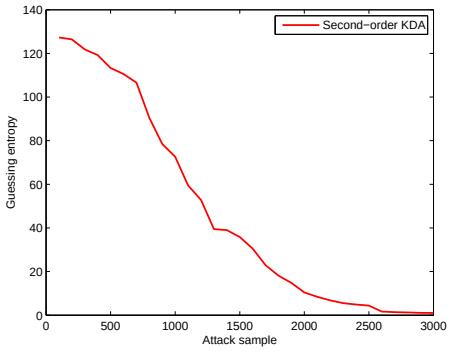

Fig. 5. Guessing entropy of second-order attack with KDA on DPA v4. (reps: 100)

sufficient number of power traces, KDA (with a Hamming weight power model) can successfully recover the secret key.

Third-order attacks In the previous subsection, we verified that KDA can indeed be used as an effective distinguisher for second-order attack. In this subsection, we attempt to extend the KDA into a higher-order attack.

We test the performance of third-order KDA in the simulated leakage scenario. We once more simulate 200,000 traces, but this time with a standard deviation of 0.01. We run third-order KDA to attack the traces 100 times; the guessing entropy of the correct key is indicated in Fig. 6. It decreases as the attack sample increases, as before, converging eventually to 1.

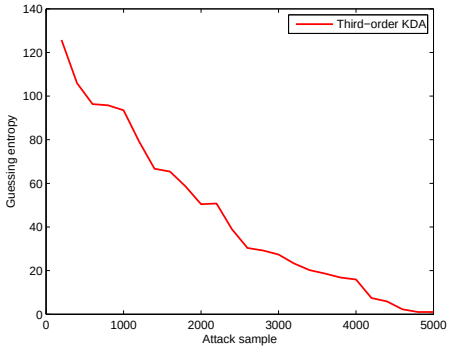

Fig. 6. Guessing entropy of third-order attack with KDA on the simulated masked implementation leakage, with $\sigma = 0.01$. (resp: 100)

4 Discussion

4.1 Complexity Analysis

Let N be the number of power traces, ℓ be the length of the sub-part relevant to the mask in the trace, and d the number of masks. Thus, for simplicity, the length of the trace is $(d+1)\ell$. The analysis which follows includes computational complexity and space complexity.

According to the general approach introduced in Section 3.1 and the KDA procedure in Section 2.3, for a key guess, the KDA distinguisher needs to compute two parts: the kernel between-class scatter matrix $\mathbf{M}$ and within-class scatter matrix $\mathbf{N}$, and the eigenvalue decomposition for $\mathbf{N}^{-1}\mathbf{M} \in R^{N\times N}$. The kernel scatter matrices need $N^2(d+1)\ell$ calculations, and the eigenvalue decomposition of $N \times N$ matrix needs $\mathcal{O}(N^3)$ computations ($\frac{9}{2}N^3$ to be more precise [7]). So the total computational complexity of the KDA distinguisher is $\mathcal{O}(N^2(N+(d+1)\ell))$. The memory usage in KDA is to store the two kernel scatter matrices, so the space complexity is $2N^2$.

The classical higher-order DPA attack first requires preprocessing the original data, incurring ℓ^{d+1} (e.g. 10^8 in the case of a 3rd-order masking with 100 time points for each tuple) calculations (subtraction or multiplication according to the combination function) for each trace. So the whole computation for the preprocessing is $N\ell^{d+1}$. Then the computation (via some first-order distinguisher) of 'similarity' between each column of the $N \times \ell^{d+1}$ matrix and the hypothetical power consumption vector requires $N \times \ell^{d+1}$ calculations in total. Therefore, the computational complexity of classical higher-order DPA is $\mathcal{O}(N\ell^{d+1})$. The main memory usage in classical higher-order DPA is to store the preprocessed traces, which implies a space complexity of $N\ell^{d+1}$.

We can see that the computation complexity of the KDA distinguisher is polynomial in N, d, and ℓ. It can still be optimized by the method of using regularized regression to avoid the eigenvalue decomposition in KDA, although the speed up ratio is 27 times [7]. However, the computational complexity of classical higher-order DPA is not only polynomial in N, but also exponential in d. The space complexity of KDA, which depends polynomially on N, represents another advantage over classical higher-order DPA, which requires additional exponential in ℓ space. If the mask order d is high and the ℓ is large, the computation of classical higher order DPA would be extremely high. When only time is considered, if $N(N+(d+1)\ell) < \ell^{d+1}$ given N, ℓ, d, then the KDA distinguisher becomes a better choice for the higher-order attack.

4.2 Flexible Power Model

Like other clustering-based distinguishers [2,12,23], the KDA distinguisher can be performed using different power models. In the dimensionality reduction setting, 256-class, 9-class (Hamming weight), and 3-class KDA have been investigated [6]. In Section 3, we investigated the binary power model that was first used in the seminal power analysis paper [11], as well as the 9-class Hamming

weight model, for different masking orders; the attacks succeeded in all tested cases.

An especially appealing feature of clustering-based distinguishers is that, unlike classical higher-order DPA, they do not rely on the power model that they use to be *proportionally* approximate to the true leakage; any meaningful *partition* on the intermediate value will suffice. This property suggests KDA as an ideal candidate for use in conjunction with the robust 'semi-profiled' modelling proposed by Whitnall *et al.* at CHES 2015 [29]. The extension of their strategy to higher-order attacks via the KDA distinguisher would be an interesting avenue to explore in future work.

4.3 Limitations and Possibilities

As a baseline against which to compare the key recovery performance of KDA, we also tested higher-order correlation DPA using the 'normalised product' combining function (the best among tested alternatives in typical leakage scenarios [19]) with a Hamming weight power model. In fact, these correlation attacks substantially outperformed the KDA distinguisher in terms of the number of traces required to converge to a guessing entropy of 1. At 2nd and 3rd orders, they were also considerably faster to run.

While the relatively poor trace efficiency is disappointing, it is not surprising given the idealised (Hamming weight or close to Hamming weight) nature of the leakage scenarios tested so far. It is well known that correlation-based attacks perform very efficiently when provided with good proportional approximations of power data dependencies, while the advantages of 'partition'-based [25] (a.k.a./ 'nominal power model'-based [30]) DPA distinguishers only emerge as the true leakage increasingly diverges from standard model assumptions [28]. An interesting avenue for future work will therefore be to deploy the KDA distinguisher in scenarios where higher order correlation DPA is likely to struggle.

We should also stress that our experiments thus far have been *proof of concept*, with no attempt (yet) to optimise for KDA parameters, which may make a substantial difference to the performance of the distinguisher. In particular, it was shown in the dimensionality reduction setting that the quality of the projected traces is influenced by the value of the regularisation parameter μ (Section 4.2 in [6]). This is something we plan to explore in future work, along with alternatives to the polynomial kernel function (such as the Gaussian kernel).

The relatively slow computation time indicates that the overheads of the eigenvalue decomposition dominate at 2nd and 3rd orders, so that the complexity advantages of KDA may only begin to emerge as d increases beyond 3. Establishing the threshold at which KDA becomes computationally preferable to classical higher-order DPA is another interesting avenue for further investigation.

5 Conclusions and Future Perspectives

Following recent separate proposals to extend LDA to the task of directly recovering secret keys from unprotected implementations, and to use KDA for

the extraction of points of (joint) interest from masked implementations, we have taken the logical next step of extending KDA likewise for application as a distinguisher. We have shown the natural common ground between higher-order DPA and the operation of KDA, and reasoned about the soundness of a KDA-based distinguisher from a theoretical perspective, before verifying its effectiveness empirically. Complexity analysis reveals a substantial advantage of KDA (polynomial in the number of traces and the order of the masking scheme) over higher-order DPA (exponential in the order of the masking scheme).

Although the theoretic complexity advantages of KDA do not translate into practical advantages in our proof-of-concept 2nd and 3rd order experiments, there remains considerable scope for enhancing the methodology and for deploying it in scenarios less vulnerable to classical higher-order DPA. The latter include yet higher masking orders and alternative masking forms, as well as data-dependencies which do not conform nicely to standard assumptions. These represent worthwhile avenues for further investigation. We also anticipate that fine-tuning the parameters (in particular, the regularization factor μ) and exploring alternative kernel functions will have a positive impact on performance.

Acknowledgements The authors would like to thank Daniel P. Martin for the fruitful discussions on the complexity analysis. This work was supported by the National Natural Science Foundation of China (No.61372062) and by the EPSRC (EP/N011635/1).

References

1. DPA Contest v4. `http://www.dpacontest.org/v4/`
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: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 16–29. Springer (2004)
4. Bruneau, N., Guilley, S., Heuser, A., Marion, D., Rioul, O.: Less is More – Dimensionality Reduction from a Theoretical Perspective. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 22–41. Springer (2015)
5. Cagli, E., Dumas, C., Prouff, E.: Enhancing dimensionality reduction methods for side-channel attacks. In: International Conference on Smart Card Research and Advanced Applications. pp. 15–33. Springer (2015)
6. Cagli, E., Dumas, C., Prouff, E.: Kernel discriminant analysis for information extraction in the presence of masking. In: International Conference on Smart Card Research and Advanced Applications. pp. 1–22. Springer (2016)
7. Cai, D., He, X., Han, J.: Efficient kernel discriminant analysis via spectral regression. In: Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. pp. 427–432. IEEE (2007)
8. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards Sound Approaches to Counteract Power-Analysis Attacks, pp. 398–412. Springer Berlin Heidelberg", Berlin, Heidelberg (1999), `\url{http://dx.doi.org/10.1007/3-540-48405-1_26}`

9. Durvaux, F., Standaert, F.X., Veyrat-Charvillon, N., Mairy, J.B., Deville, Y.: Efficient Selection of Time Samples for Higher-Order DPA with Projection Pursuits. In: Revised Selected Papers of the 6th International Workshop on Constructive Side-Channel Analysis and Secure Design - Volume 9064. pp. 34–50. COSADE 2015, Springer-Verlag New York, Inc., New York, NY, USA (2015)
10. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics 7(2), 179–188 (1936)
11. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Advances in Cryptology–CRYPTO'99. pp. 789–789. Springer (1999)
12. Mahmudlu, R., Banciu, V., Batina, L., Buhan, I.: LDA-Based Clustering as a Side-Channel Distinguisher (2016)
13. 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)
14. Messerges, T.S.: Using Second-Order Power Analysis to Attack DPA Resistant Software, pp. 238–251. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
15. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop. pp. 41–48. IEEE (1999)
16. Nassar, M., Souissi, Y., Guilley, S., Danger, J.L.: RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012. pp. 1173–1178. IEEE (2012)
17. Oswald, E., Mangard, S.: Template attacks on masking—resistance is futile. In: Cryptographers' Track at the RSA Conference. pp. 243–256. Springer (2007)
18. Peeters, E., Standaert, F.X., Donckers, N., Quisquater, J.J.: Improved higher-order side-channel attacks with fpga experiments. In: CHES. vol. 3659, pp. 309–323. Springer (2005)
19. Prouff, E., Rivain, M., Bevan, R.: Statistical analysis of second order differential power analysis. IEEE Trans. Computers 58(6), 799–811 (2009)
20. Saitoh, S., Sawano, Y.: Theory of reproducing kernels and applications, vol. 44. Springer (2016)
21. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural computation 10(5), 1299–1319 (1998)
22. Schramm, K., Paar, C.: Higher Order Masking of the AES, pp. 208–225. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
23. 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)
24. Standaert, F.X., Archambeau, C.: Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 411–425. Springer (2008)
25. 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)
26. 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)

27. Standaert, F.X., Veyrat-Charvillon, N., Oswald, E., Gierlichs, B., Medwed, M., Kasper, M., Mangard, S.: The world is not enough: Another look on second-order DPA. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 112–129. Springer (2010)
28. Whitnall, C., Oswald, E.: A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. J. Cryptographic Engineering 1(2), 145–160 (August 2011)
29. Whitnall, C., Oswald, E.: Robust profiling for DPA-style attacks. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 3–21. Springer (2015)
30. 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)