Evaluation and Improvement of Generic-Emulating DPA Attacks

Weijia Wang[1](✉), Yu Yu[1](✉), Junrong Liu[1,2], Zheng Guo[1,2], François-Xavier Standaert[3](✉), Dawu Gu[1], Sen Xu[1], and Rong Fu[4]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
{aawwjaa,yyuu,liujr,guozheng,dwgu,xusen0328}@sjtu.edu.cn

[2] Shanghai Viewsource Information Science and Technology Co., Ltd, Shanghai, China

[3] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Louvain-la-Neuve, Belgium
fstandae@uclouvain.be

[4] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

Abstract. At CT-RSA 2014, Whitnall, Oswald and Standaert gave the impossibility result that no generic DPA strategies (i.e., without any *a priori* knowledge about the leakage characteristics) can recover secret information from a physical device by considering an injective target function (e.g., AES and PRESENT S-boxes), and as a remedy, they proposed a slightly relaxed strategy "generic-emulating DPAs" free from the non-injectivity constraint. However, as we show in this paper, the only generic-emulating DPA proposed in their work, namely the SLR-based DPA, suffers from two drawbacks: unstable outcomes in the high-noise regime (i.e., for a small number of traces) and poor performance especially on real smart cards (compared with traditional DPAs with a specific power model). In order to solve these problems, we introduce two new generic-emulating distinguishers, based on lasso and ridge regression strategies respectively, with more stable and better performances than the SLR-based one. Further, we introduce the cross-validation technique that improves the generic-emulating DPAs in general and might be of independent interest. Finally, we compare the performances of all aforementioned generic-emulating distinguishers (both with and without cross-validation) in simulated leakages functions of different degrees, and on an AES ASIC implementation. Our experimental results show that our generic-emulating distinguishers are stable and some of them behave even better than (resp., almost the same as) the best Difference-of-Means distinguishers in simulated leakages (resp., on a real implementation), and thus make themselves good alternatives to traditional DPAs.

1 Introduction

Since the introduction of differential power analysis (DPA) by Kocher et al. [9], the CHES community has been focusing on efficient key recovery techniques

© International Association for Cryptologic Research 2015
T. Güneysu and H. Handschuh (Eds.): CHES 2015, LNCS 9293, pp. 416–432, 2015.
DOI: 10.1007/978-3-662-48324-4_21

by exploiting the physical information (typically the power consumption) captured from the implementation of a (leaking) cryptographic device, resulting in a rich body of literature on power analysis (see, e.g., [1,2,4,7,11,13,15] for an incomplete list). In this paper, we mainly consider non-profiled[1] power analysis techniques. We mention that some non-profiled attacks such as correlation power analysis (CPA) [2] and differential cluster analysis (DCA) [1] make some reasonable device-specific assumption about the power models. However, with the development of chip industry towards smaller technologies, the impact of power variability is becoming more and more significant, which makes common power models (e.g., Hamming weight, bit leakage) much less respected in practice (especially when reaching nanoscale devices) [10].

More recently, various non-profiled strategies such as mutual information analysis (MIA) using an identity power model [7], Kolmogorov-Smirnov (KS) method [13], Cramér-von Mises test method [13], linear regression[2] (LR)-based method [11] and copulas methods [14] were proposed. All these attacks enable to work in a context where no *a priori* knowledge is assumed about the power model, and DPAs of this form were termed as "generic DPAs" in [16]. The authors of [16] showed an impossibility result that all generic DPAs fail to work when applied to an injective target function. Fortunately, they observed that a slight relaxation of the generic condition (with the incorporation of some minimal "non-device-specific intuition") allows to bypass the impossibility result, for which they coined the name "generic-emulating DPAs". They further exemplified this by relaxing LR-based DPA (as a generic DPA) to stepwise linear regression (SLR)-based DPA (as a generic-emulating DPA) and demonstrated its effectiveness for injective target functions in simulation-based experiments.

However, despite the effectiveness on injective target functions, SLR-based DPA suffer from two drawbacks. First, it tends to be unstable for a small number of traces, reflecting a high variance of the outcomes (and thus lower success rates), which is illustrated in Sect. 5.1. Second, there is still a performance gap between SLR-based DPA and traditional one (e.g., CPA or DPA with the bit model), especially on real smart cards (which were not analyzed in [16]). In this paper, we address the above issues and make the following contributions.

First, we introduce in Sect. 3 two alternative generic-emulating distinguishers named lasso-based and ridge-based ones. We show that the new distinguishers enjoy a more stable and better performance than SLR-based ones (see Fig. 4). Intuitively, our improvement benefits from the fact that our distinguishers use a more continuous way to shrink the parameters than SLR-based ones (see Sect. 5.1 for a discussion).

[1] In contrast, a profiled power analysis (such as template attacks [4] and stochastic attacks [15]) takes advantage of an offline learning phase to gain additional information about an identical device (using a known key), which is not always practical.

[2] Unless otherwise specified, an LR-based distinguisher refers to one that uses a full basis of polynomial terms to construct the power model, and we often omit the term "with a full basis" for succinctness. We refer to Sect. 2.3 for a formal definition.

Second, we exploit in Sect. 4 a technique from statistical learning called 'cross-validation' that might be of independent interest (used in the context of profiled DPA in [5]), and show in Sect. 5 that it can be combined with generic-emulating DPAs to improve the performance in general.

Finally, for a comprehensive comparison, we illustrate in Sect. 5 the performances of SLR-based DPA, ridge-based DPA and lasso-based DPA in various settings, e.g., with and without cross-validation, against leakage function of different degrees and on a real smart card. Some of our attacks outperform the best Difference-of-means (DoM) attack[3] in simulation-based experiments and achieve almost the same performance as best DoM attack on real smart cards. Therefore, our results improve the work of [16] and can be considered as taking one concrete step forward towards making generic-emulating DPA practical.

2 Background

2.1 Differential Power Analysis

Following the 'divide-and-conquer' strategy, a DPA attack breaks down a secret key into a number of subkeys of small length and recovers them independently. Let X be a vector of some (partial) plaintext in consideration, i.e., $X = (X_i)_{i \in \{1,\ldots,N\}}$, where N is the number of measurements and X_i corresponds to the (partial) plaintext of i-th measurement. Let k be a hypothesis subkey, let $F_k : \mathbb{F}_2^m \to \mathbb{F}_2^m$ be a target function, where m is the bit length of X_i, and thus the intermediate value $Z_{i,k} = F_k(X_i)$ is called a target and $Z_k = F_k(X) = (Z_{i,k})_{i \in \{1,\ldots,N\}}$ is the target vector obtained by applying F_k to X component-wise.

Let $L : \mathbb{F}_2^m \to \mathbb{R}$ be the leakage function and let T be a vector of power consumptions. We have $T_i = L \circ Z_{i,k^*} + \varepsilon$ and $T = L \circ Z_{k^*} + \varepsilon$, where $\circ$ denotes function composition, k^* is the correct subkey key and ε denotes probabilistic noise. A trace t_i is the combination of power consumption T_i and plaintext X_i, i.e., $t_i = (T_i, X_i)$. Let the function $M : \mathbb{F}_2^m \to \mathbb{R}$ be the power model that approximates the leakage function L, namely, $T \approx M \circ F_{k^*}(X)$, where the noise information is also included in the power model.

In this paper, we assume that $F_k(\cdot)$ is an injective function (e.g., the AES S-Box). With the above definitions and notation, we can describe DPA as follows:

1. Make a subkey guess k and compute the corresponding target value $F_k(X_i)$.
2. Estimate the power consumptions of $F_k(X)$ with the power model $M(\cdot)$, i.e., $M(F_k(X))$.
3. Compute the correlation between the hypothetical $M(F_k(X))$ and the real trace T. The correlation should be highest upon correct key guess (which can be decided after repeating the above for all possible subkey guesses).

[3] Difference-of-means attack is a form of DPA that exploits the leakage of a single bit. It is generally seen as the 'best' attack strategy without *a prior* knowledge about the power model. The best DoM attack refers to the DoM attack in the best scenario, i.e., assuming additional knowledge about which target bit to exploit to achieve the highest correlation (among all possible target bits).

2.2 Generic DPA and its limitations

The generic DPA is defined in [16] with the definitions below:

Definition 1 (Generic Power Model). *The generic power model associated with key hypothesis k is the nominal mapping to the equivalence classes induced by the key-hypothesised target function $F_k(\cdot)$.*

Definition 2 (Generic Compatibility). *A distinguisher is generic-compatible if it is built from a statistic with operate on a nominal scale measurements.*

Definition 3 (Generic DPA). *A generic DPA strategy performs a standard univariate DPA attack using the generic power model paired with a generic-compatible distinguisher.*

Unfortunately, as shown in [16], no efficient generic DPA strategy is able to distinguish the correct subkey k^* from an incorrect hypothetical value k give that F_{k^*} and F_k are both injective. We refer to [16] for the details and proofs.

2.3 From LR-based DPA to Generic-Emulating DPA

As stated in [16] and [3], any leakage function L on input $z \in \mathbb{F}_2^m$ can be represented in the form of $L(z) = \sum_{u \in \mathbb{F}_2^m} \alpha_u z^u$ with coefficients $\alpha_u \in \mathbb{R}$, where $z = Z_{i,k^*}$, z^u denotes monomial $\prod_{j=1}^{m} z_j^{u_j}$, and z_j (resp., u_j) refers to the j^{th} bit of z (resp., u). Therefore, for each subkey hypothesis k, we use a full basis of polynomial terms to construct the power model: $M_k(Z_{i,k}) = \alpha_0 + \sum_{u \in \mathbb{U}} \alpha_u Z_{i,k}^u$, where $\mathbb{U} = \mathbb{F}_2^m \setminus \{0\}$. The degree of the power model is the highest degree of the non-zero terms in polynomial $M_k(Z_{i,k})$. We denote $\boldsymbol{\alpha} = (\alpha_u)_{u \in \mathbb{U}}$ as the vector of coefficients, which is estimated from $\boldsymbol{U}_k = (Z_{i,k}^u)_{i \in \{1,2,\ldots,N\}, u \in \mathbb{U}}$ and T using ordinary least squares, i.e., $\boldsymbol{\alpha} = (\boldsymbol{U}_k^\top \boldsymbol{U}_k)^{-1} \boldsymbol{U}_k^\top T$, where $(Z_{i,k}^u)_{i \in \{1,2,\ldots,N\}, u \in \mathbb{U}}$ is a matrix with (i,u) being row and column indices respectively, and $\boldsymbol{U}_k^\top$ is the transposition of $\boldsymbol{U}_k$. Finally, the goodness-of-fit (denoted as R^2), as a measurement of similarity between $M_k(Z_k)$ and the real power consumption T, can be computed for each M_k which separates the correct key hypothesis from incorrect ones[4]. This method, called Linear Regression-based DPA (LR-based DPA) with a full basis, falls into a special form of generic DPA, and thus it doesn't distinguish correct sub-keys from incorrect ones on injective target functions (see [16]).

To address the issue, generic-emulating DPA additionally exploits the characteristics of power models in practice (by losing a bit of generality) and it makes *a priori* constrain on $\boldsymbol{\alpha}$. As observed in [16], the coefficient vector $\boldsymbol{\alpha}$ is typically sparse for a realistic power model (under correct sub-key). Therefore, SLR-based DPA, as a generic-emulating DPA, starts from a power model with a full basis

[4] We use Pearson's coefficient to measure the goodness-of-fit in this paper, i.e., $R^2 = \rho(T, M_k(Z_k))$, where ρ is the Pearson's coefficient.

$\mathbb{U} = \mathbb{F}_2^m \setminus \{0\}$ and excludes some 'insignificant' terms while keeping all the 'significant' ones in the basis. Then, it measures the goodness-of-fit R^2 to separate the correct sub-key from incorrect ones. Formally,

$$\begin{aligned}\hat{\boldsymbol{\alpha}}^{SLR} \overset{\mathsf{def}}{=} \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} (T_i - M_k(Z_{i,k}))^2, \\ \text{subject to} \sum_{u \in \mathbb{U}} |\mathsf{sign}(\alpha_u)| \leq s,\end{aligned} \tag{1}$$

where the absolute value of signum function $|\mathsf{sign}(\alpha_u)| = 0$ if $\alpha_u = 0$, and otherwise (i.e., $\alpha_u \neq 0$) $|\mathsf{sign}(\alpha_u)| = 1$.

It should be noted that Eq. (1) and the description of SLR-based DPA in [16] are equivalent. The former one follows the definition of stepwise regression in [6], and the latter one more focuses on the algorithmic aspects of SLR-based DPA, and the parameter s come from the p-values in [16].

However, as we will show in Sect. 5.1, SLR-based DPA suffers from two drawbacks: (1) it is not stable for small number of traces; (2) in comparison with traditional DPA, SLR-based DPA has poor performance especially on real implementations. In next sections, we present two alternative generic-emulating distinguishers with more stable and improved performances, as well as a strategy called 'cross-validation' that might be of independent interest.

3 Alternative Generic-Emulating Distinguishers

In this section, we present two new generic-emulating distinguishers: the ridge-based and lasso-based distinguishers. For consistency with [16], we use the same power model as SLR-based DPA, i.e., $M_k(Z_{i,k}) = \alpha_0 + \sum_{u \in \mathbb{U}} \alpha_u Z_{i,k}^u$. It should be noted that (in our terminology) generic-emulating DPAs and generic-emulating distinguishers are not exactly the same, the latter one output the coefficients while the former output key k^* (as its best guess) and the corresponding R^2.

3.1 Ridge-Based Distinguisher

Ridge-based distinguisher shrinks coefficients α_u by explicitly imposing an overall constraint on their size [8]:

$$\begin{aligned}\hat{\boldsymbol{\alpha}}^{ridge} \overset{\mathsf{def}}{=} \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} \Big(T_i - M_k(Z_{i,k})\Big)^2, \\ \text{subject to} \sum_{u \in \mathbb{U}} \alpha_u^2 \leq s.\end{aligned} \tag{2}$$

An equivalent formulation to the above is

$$\hat{\boldsymbol{\alpha}}^{ridge} = \underset{\alpha}{\operatorname{argmin}} \left(\sum_{i=1}^{N} (T_i - M_k(Z_{i,k}))^2 + \lambda \sum_{u \in U} \alpha_u^2 \right), \tag{3}$$

whose optimal solution is given by:

$$\hat{\boldsymbol{\alpha}}^{ridge} = (\boldsymbol{U}_k^{\mathsf{T}}\boldsymbol{U}_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}_k^{\mathsf{T}}T, \tag{4}$$

where matrix $\boldsymbol{I}$ is the $|\mathbb{U}| \times |\mathbb{U}|$ identity matrix, $|\mathbb{U}|$ denotes the cardinality of $\mathbb{U}$ and $\boldsymbol{U}_k$ is defined in Sect. 2.3.

How the Coefficients Shrink in the Ridge-Based Distinguisher?. As described in Sect. 3.1, the ridge-based distinguisher enforces a general constraint $\sum_{u\in\mathbb{U}} \alpha_u^2 < s$ on the coefficients of M_k, but it is not clear how each individual coefficient α_u shrinks (e.g., which coefficient shrinks more than the others). We show an interesting connection between the degree of a term $Z_{i,k}^u$ in M_k (i.e., the Hamming Weight of u) and the amount of shrinkage of its coefficient α_u.

First, we use a technical tool from "principal component analysis" (see, e.g., [8]). Informally, the principal components of $\boldsymbol{U}_k$ are a set of linearly independent vectors obtained by applying an orthogonal transformation to $\boldsymbol{U}_k$, i.e., $\boldsymbol{P} = \boldsymbol{U}_k\boldsymbol{V}$, where the columns of matrix $\boldsymbol{P}$ are called the principal components, and the columns of matrix $\boldsymbol{V}$ are called directions of the (respective) principal components. An interesting property is that columns of P, denoted by $P_1, \ldots, P_{2^m-1}$, have descending variances (i.e., P_1 has the greatest variance). Among the columns of $\boldsymbol{V}$, the first one, denoted V_1 (the direction of P_1), has the maximal correlation to coefficient vector $\boldsymbol{\alpha}$. We refer to [8] for further discussions and proofs.

Then, we further study the correlation between V_1 and $\boldsymbol{\alpha}$, both seen as a vector of $2^m - 1$ components indexed by u. Figures 1 and 2 depict the direction of the first principle component V_1 and the degrees of terms in $\boldsymbol{U}_k$ respectively, and they represent a high similarity (albeit in a converse manner). Quantitatively, the Pearson's coefficient between V_1 and the corresponding vector of degrees is -0.9704, which is a nearly perfect negative correlation.

Finally, given that V_1 is positively correlated to $\boldsymbol{\alpha} = (\alpha_u)_{u\in\mathbb{U}}$ while negatively correlated to their term degrees, we establish the connection that α_u is conversely proportional to the Hamming weight of u. In other words, the more Hamming weight that u has, the less α_u contributes to the power model. Therefore, ridge-based distinguisher is consistent with low-degree power models (e.g., the Hamming weight and bit models) in practice.

3.2 Lasso-Based Distinguisher

The lasso-based distinguisher is similar to the ridge-based one excepted for a different constraint on the parameters [8]:

$$\hat{\boldsymbol{\alpha}}^{lasso} \overset{\mathsf{def}}{=} \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} \Big(T_i - M_k(Z_{i,k})\Big)^2, \tag{5}$$

$$\text{subject to} \sum_{u\in\mathbb{U}} |\alpha_u| \leq s. \tag{6}$$

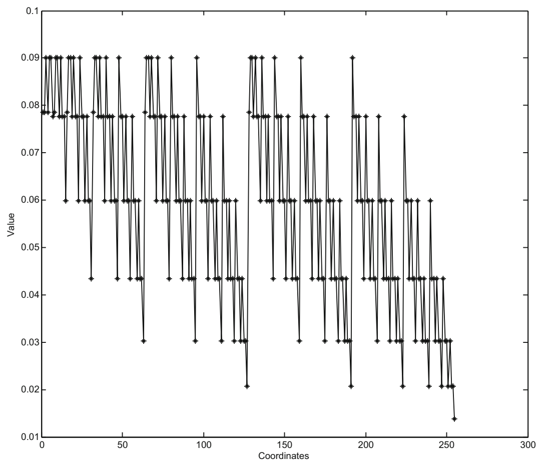


Fig. 1. An illustration of V_1 (the direction of the first principal component of $\boldsymbol{U}_k$).

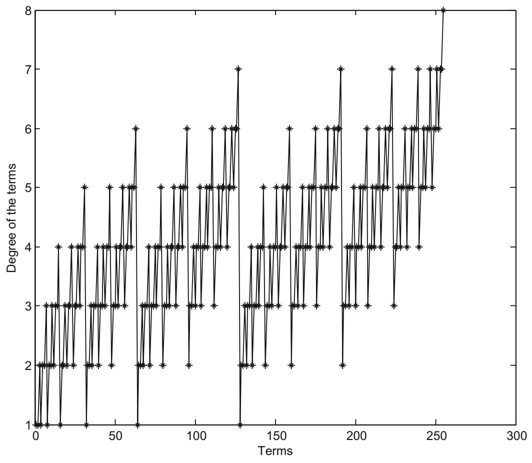


Fig. 2. An illustration of the degrees of the terms in $\boldsymbol{U}_k$.

A subtle but important difference between lasso-based and ridge-based regressions is their ways of shrinking the coefficients. By choosing a sufficiently small s, lasso-based distinguisher will have some of its coefficients exactly stuck to zero and in contrast, the ridge-based distinguisher will only shrink the coefficients (with the amounts of shrinkage conversely proportional to the degrees of terms). Thus, we can consider the lasso-based distinguisher as a tool inbetween SLR-based and ridge-based distinguishers.

Finding the optimal solution for lasso-based distinguishers is essentially a quadratic programming problem. Fortunately, there are known efficient algorithms and we use the "Least Angle Regression" algorithm for this purpose [6] (see Appendix A for full details).

4 Generic-Emulating DPAs with Cross-Validation

In this section, we combine generic-emulating DPA with the K-fold[5] cross-validation technique from statistical learning. We mention that cross-validation was already used for evaluation of side-channel security in the profiled setting [5]. Algorithm 1 shows how to combine generic-emulating DPA with cross-validation.

Algorithm 1. Generic-emulating DPA with cross-validation

```
Require: traces $t_i = \{T_i, x_i\}$ where $i \in \{1, ..., N\}$; the number of parts $K$ ; target
    function $F_k(\cdot)$;
Ensure: $\hat{k}$ as the best guess for the subkey;
 1: for $i = 1$; $i <= N$; $i{+}{+}$ do
 2:     $\mathcal{S}_{x_i} = t_i$
 3: end for
 4: for $i = 1$; $i <= K$; $i{+}{+}$ do
 5:     $\mathcal{C}_i = \{\mathcal{S}_{K*(i-1)+1}, ..., \mathcal{S}_{K*i}\}$
 6: end for
 7: for all $k$ such that $k \in F_n^2$ do
 8:     for $i = 1$; $i <= K$; $i{+}{+}$ do
 9:         Compute the $\boldsymbol{\alpha}$ using the traces in $\mathcal{C}_j$, where $j \in \{1 \ldots K\} \setminus \{i\}$
10:         Calculate the goodness-of-fit $R_i^2$ from $\mathcal{C}_i$
11:     end for
12:     $R_k^2 = (\sum_{i=1}^{K} R_i^2)/K$
13: end for
14: $\hat{k} = \text{argmax}_k R_k^2$
```

As sketched in Fig. 3, the algorithm follows the steps below:

First, we classify the traces into 2^m sets based on the values of the corresponding input, denoted as $\mathcal{S}_{\{0 \ldots 2^m-1\}}$. Otherwise said, all traces in each set correspond to the same value of input (partial plaintext).

Then, we split $\mathcal{S}_{\{1 \ldots 2^m\}}$ into K parts $\mathcal{C}_{\{1 \ldots K\}}$ of roughly equal size. For each part $\mathcal{C}_i$, we compute the coefficients α_i using the rest $K-1$ parts from the trace set, and calculate the goodness-of-fit R_i^2 using the traces in $\mathcal{C}_i$. We then get the average goodness-of-fit $R_k^2 = (\sum_{i=1}^{K} R_i^2)/K$ for the hypothetical subkey k in consideration. Finally, we return the subkey candidate with the highest averaged goodness-of-fit.

For example, let the target function be an AES S-box, let $K = 8$ and use the ridge regression-based distinguisher, and thus the traces are classified into $\mathcal{C}_{\{1 \ldots 8\}}$, where $\mathcal{C}_1 = \{S_1, \ldots, S_{32}\}$, $\mathcal{C}_2 = \{S_{33}, \ldots, S_{64}\}$, $\ldots$, $\mathcal{C}_8 =$

[5] We shall not confuse K with k, where K is a parameter as in the "K-fold cross-validation" and k is a subkey hypothesis.

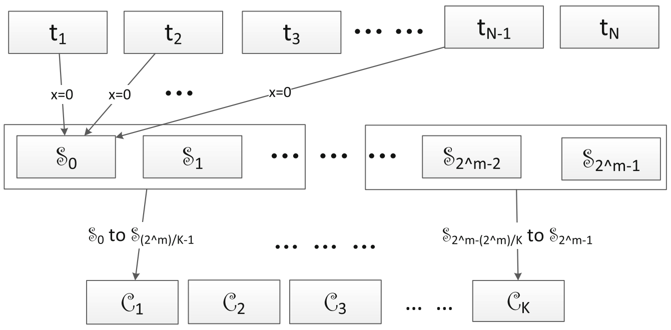


Fig. 3. Generic emulating DPA attack with cross-validation. The traces are divided into 2^m sets $\mathbb{S}_{\{0\dots 2^m-1\}}$, which are in turn categorized into K parts $\mathbb{C}_{\{1\dots K\}}$ to mount cross-validation.

$\{\mathbb{S}_{225}, \dots, \mathbb{S}_{256}\}$. For part $\mathbb{C}_1$, we first compute its coefficients $\boldsymbol{\alpha}$ using the traces from sets $\mathbb{S}_{33}$, $\mathbb{S}_{34}$, ..., $\mathbb{S}_{256}$, and then calculate the goodness-of-fit R^2 using sets $\mathbb{S}_1$, $\mathbb{S}_2$ ... $\mathbb{S}_{32}$, where k is a key hypothesis.

For a leakage function $L : \mathbb{F}_2^m \rightarrow \mathbb{R}$ with input space of size 2^m, the cross-validation technique can determine the coefficients $\boldsymbol{\alpha}$ from traces on only a portion of (rather than the whole) input space. As we will show in Sect. 5.1, the 'Non-Device-Specific' nature of cross-validation allows to relax the LR-based DPA from a generic DPA (with a full basis) to a generic-emulating one by learning the leakage function from a subset of the input space.

5 Experimental Results

In this section, we give experimental results based on both simulation-based environments and real smart cards. In the simulation-based experiments, we first show that SLR-based DPA tends to be unstable for small number of traces. Then, we give a comprehensive comparison between the performance of SLR-based DPA, ridge-based DPA and lasso-based DPA in various settings, e.g., with and without cross-validation, against power models of different degrees and on a real smart card. In particular, some of these attacks beat the best DoM attack (see Footnote 3) in simulation-based experiments and achieve almost the same performance as best DoM attack on real smart cards. This improves the work of [16], where the SLR-based DPA doesn't outperform best DoM attack in simulation-based experiments (and real implementations are not considered in [16]).

In both scenarios, we target the AES-128's first S-box of the first round with an 8-bit subkey (recall that AES-128's first round key is the same as its encryption key). Following [16], we do the following trace pre-processing to facilitate the evaluation: we average the traces based on their the input (an 8-bit plaintext) and use the resulting 256 mean power traces to mount the attack. Since the running time of generic-emulating DPA increases as the number of traces grows, it may become unbearable when we have hundreds of thousands of traces Therefore, it is reasonable to use a few mean power traces instead of a huge number of traces in both simulation-based and real attacks.

The parameters from different distinguishers, e.g., λ, s, and K from ridge-based, lasso-based distinguishers and cross-validation respectively, can also affect the success rate of the attacks. We will directly use the best values for these parameters, $\lambda = 800$, $s = 2$, $K = 7$, which were decided through searching over the space (up to some accuracy) in favor of best success rate. It should be though that the same parameters can be used in the various experimental settings (i.e., the variety of settings doesn't seem to affect the choice of the best parameters significantly).

5.1 Simulation-Based Experiments

SLR-based Distinguishers are Not Stable. By definition, the SLR-based distinguisher keeps only a subset of the terms from the basis. As a result, some 'insignificant' terms that still have some (although not much) contributions to the power model are discarded and it leads to instability of the results especially when the number of traces used in the attacks is small. Rephrased using the terminology of statistical learning, such a subset selection often leads to high variance of the outcomes due to the discretization process (see a discussion in [8]). In this context, the actual coefficients (corresponding to the correct key k^*) of the power model tend to be more evenly distributed since the noise is included in the power model, and the outcome of the SLR-based DPA (R^2 of the correct subkey) will be varying (dependent on which subset of the basis is selected) and thus leads to an unstable outcome with low success rate. In contrast, the ridge-based (and lasso-based) distinguishers only shrink the coefficients of the 'insignificant' terms rather than simply discarding them. In general, the shrinking techniques are continuous and do not suffer much from high variability [8], which makes these distinguishers good alternatives to the SLR-based one.

We use simulated-based experiments to illustrate the above issue. In the case of a fixed leakage function of degree 8 with SNR = 0.1, we use both SLR-based and ridge-based distinguishers to approximate the 255 coefficients of the power model with different trace sets and compute the corresponding variance (of the approximated coefficients). We then repeat this with different set sizes, which are depicted in Fig. 4. The variance of outcomes increases with the noise level[6] (i.e., the decrease of the number of traces), and for the same number of traces the ridge-based distinguisher has a much lower variance of its outcomes than the SLR-based one, and thus has a more stable performance.

[6] By "noise level" we refer to the overall amount of noise by combining all traces rather that the SNR of the measurement environment. In general, increasing the number of traces reduces the noise level, which can be seen by averaging the traces.

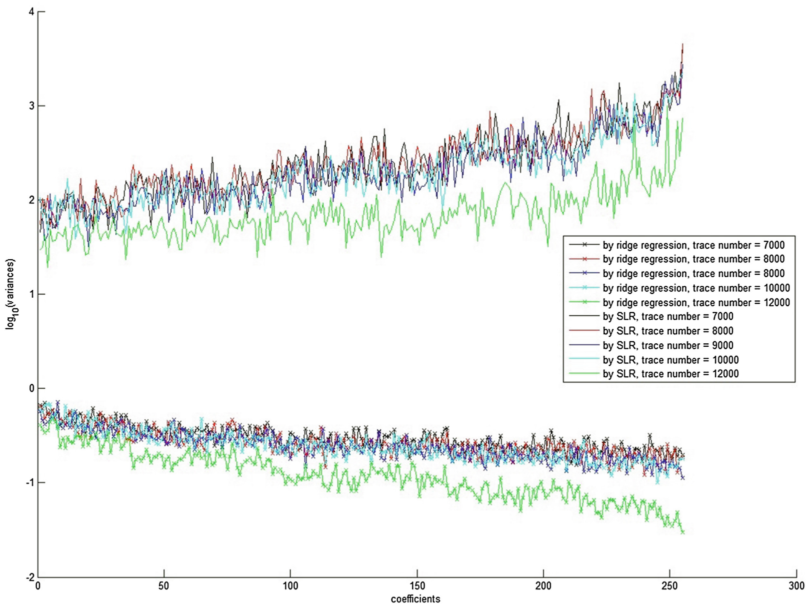


Fig. 4. Variances estimated of the estimated coefficients, for the ridge-based and SLR-based distinguishers, using different numbers of traces.

A Comparison of Various Attacks with Simulation-Based Experiments. Fig. 5 illustrates the (1st order[7]) success rates of all aforementioned DPAs on leakage functions of different degrees, in which we repeat each experiments 100 times (each time with a different random leakage function) to compute the success rates. In addition, we include the best DoM attack and known model DPA as baselines, where the former (resp., latter) is considered as the best traditional DPA attacks without any (resp., full) *a priori* knowledge about the power model. We have the following observations.

First, ridge-based DPA with cross-validation and lasso-based DPA have are among the best attacks in all settings (in particular they outperform the best DoM attacks, and are only less powerful than known power-model DPA). We attribute this to the intuition that generic-emulating DPAs are better suited for power models of moderate and high degrees than traditional DPA. Second, the new generic-emulating DPAs perform better than SLR-based one, which is consistent with the discussion in Sect. 5.1. Third and interestingly, cross-validation improves the performance of ridge-based DPA while it does not

[7] We shall not confuse the t-th order success rate with high order DPA. The t-th order success rate is a generalization of the ordinary success rate [12]. That is, the attack is considered as successful as long as the correct key is ranked among top t in the key candidate list produced by the distinguisher. Note that for t-th order success rate there remains a guessing entropy of $\log_2 t$.

Table 1. The running times for various attacks, where C-V stands for cross-validation.

SLR-based DPA	5.644 s
SLR-based DPA with C-V	26.011 s
Ridge-based DPA	4.290 s
Ridge-based DPA with C-V	16.930 s
Lasso-based DPA	3.308 s
Lasso-based DPA with C-V	3.140 s
Best DoM attack	0.363 s

(and may even worsen) SLR-based and lasso-based DPAs. (It also makes the LR-based DPA work even for injective target function). This fits the intuition that cross-validation cannot be a universal performance enhancer in a non-profiled setting (since its standard use is to avoid overfitting models in the profiled setting). However, experiments show that despite heuristic, its application in a non-profiled setting can be useful.

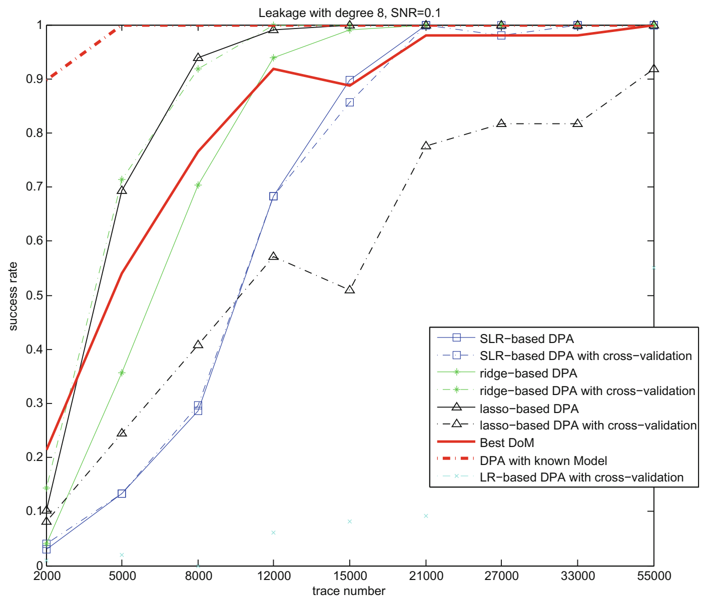
Finally, in order to fully exemplify the power of generic-emulating DPA, we also perform the attacks against some artificial leakage function, in which all low degree terms are discarded. More specifically, we consider the leakage function $L(z) = \sum_{u \in \mathcal{U}} \alpha_u z^u, \forall z \in \mathbb{F}_2^m$, where u is from $\mathbb{F}_2^m$ but excludes those whose Hamming weight is less than or equal to p. We simulate the traces for $p = 4, m = 8$ and show the success rates in Fig. 6. We can see that in this case, the best DoM attack behaves poorly and meanwhile the generic-emulating DPAs are not affected. Admittedly, this leakage case may be unrealistic, but it serves as a good example that generic-emulating DPAs can deal with a wider range of leakage function.

Table 1 below tabulates the running times of all attacks mentioned above (we use 256 averaged single-point traces so this running times also hold for the experiments on smart cards in the next section). We note that SLR-based and lasso-based distinguishers have the longest and shortest running time respectively. In general, cross-validation increases the running time for most (e.g., SLR-based and ridge-based) distinguishers except for the lasso-based one. This is due to that cross-validation actually only operates on a subset of the traces and thus makes the effective input length of the Least Angle Regression algorithm (used by the lasso-based distinguisher) shorter.

5.2 Experiments on Smart Cards

We carry out experiments on an AES microscale ASIC implementation, and measure the power consumptions using a LeCroy waverunner 610Zi digital oscilloscope at a sampling rate of 250 MHz.

The lefthand of Fig. 7 gives the success rates for all attacks discussed above on the real smart card. The experiment shows that cross-validation significantly improves the performance of generic-emulating DPAs on real smart cards.



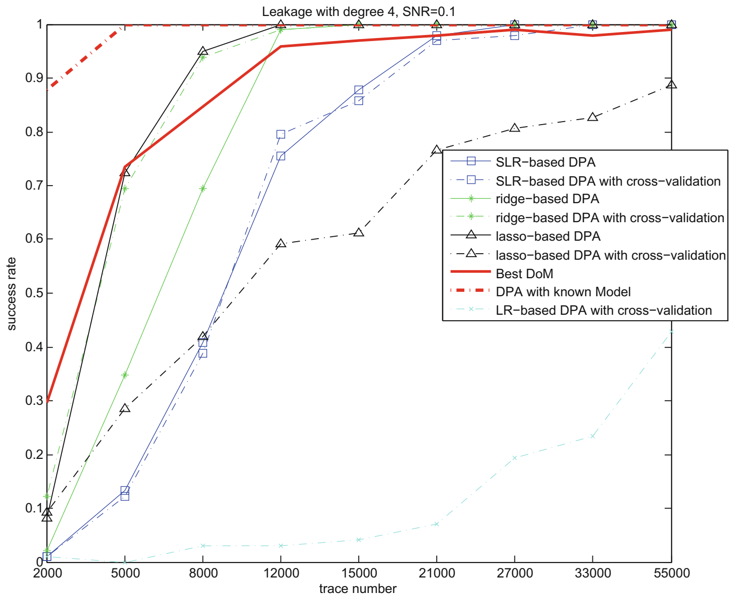


Fig. 5. The success rates of different attacks for different leakages and SNR = 0.1.

In addition, ridge-based DPA with cross-validation is the one with the closest performance to the best DoM DPA. Finally, unlike the simulation based case, the DPAs without cross-validation perform poorly and mostly do not work, and thus we conclude that cross-validation is a useful tool for generic-emulating DPAs

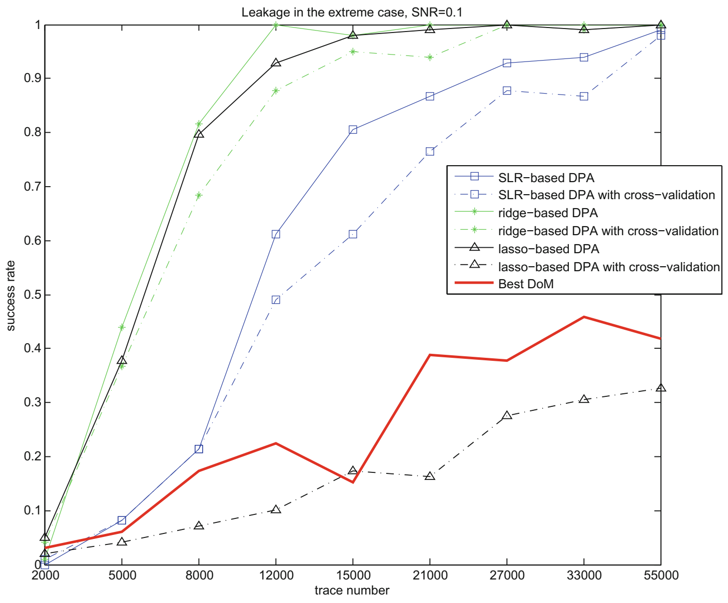


Fig. 6. The success rates of various attacks for an 'artificial' power model.

against real smart cards. As in the previous section, the exact reasons of this behavior are hard to explain due to their heuristic nature, but we can reasonably assume that it is mostly the less regular behavior of real measurements that make cross-validation more useful in this context.

The righthand of Fig. 7 gives the 8th-order success rates for all DPA attacks. This is for a better alignment with the DoM attack. That is, the best DoM attack assumes the knowledge of which target bit (out of 8 candidates) gives the highest correlation, but in practice there is a guessing entropy of $\log_2 8 = 3$ bits

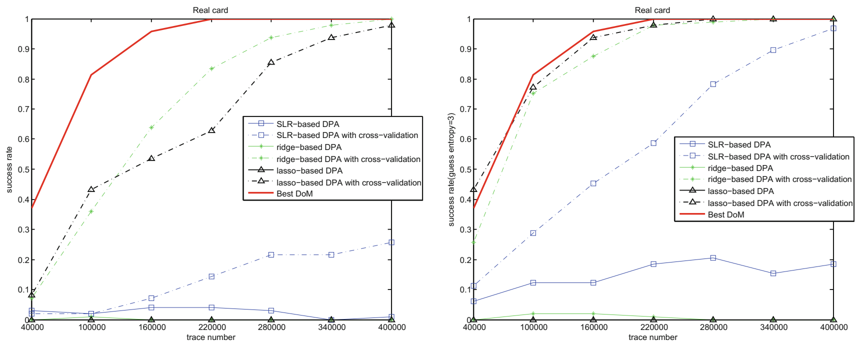


Fig. 7. The success rates of various attacks on a real AES ASIC implementation.

for realistic attackers. Likewise, there is also a guessing entropy of 3 bits from a successful 8th-order key recovery to an ideal (1st-order) one. In this scenario, the result suggests that the performances of ridge-based and lasso-based DPAs (both with cross-validation) are very close to (almost the same as) that of the best DoM. This demonstrates the usefulness and effectiveness of generic-emulating DPA in practice. Namely, they don't loose much in contexts where DoM attacks work best, and they gain a lot in more challenging scenarios.

6 Conclusion

In this paper, we continue the study of [16] on the feasibility, efficiency and limits of generic(-emulating) DPAs. We propose two new generic-emulating distinguishers, i.e., the ridge-based and lasso-based ones. We illustrate that these new distinguishers are more stable compared to the SLR-based one. We also show through both simulation-based and real experiments that our generic-emulating DPAs are practical (as compared with traditional DPAs). In addition, combined with the cross-validation technique, the generic-emulating DPAs demonstrate a significantly improved performance in our attacks against real cryptographic devices.

Acknowledgments. This research work was supported in parts by the National Basic Research Program of China (Grant 2013CB338004) and the European Commission through the ERC project 280141 (CRASH). Yu Yu was supported by the National Natural Science Foundation of China Grant (Nos. 61472249, 61103221). Zheng Guo was supported by the National Natural Science Foundation of China Grant (No. 61402286). F.-X. Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). Dawu Gu was supported by the National Natural Science Foundation of China Grant (No. 61472250), the Doctoral Fund of Ministry of Education of China (No. 20120073110094), the Innovation Program by Shanghai Municipal Science and Technology Commission (No. 14511100300), and Special Fund Task for Enterprise Innovation Cooperation from Shanghai Municipal Commission of Economy and Informatization (No. CXY-2013-35).

A Least Angle Regression

The algorithm to solve the lasso problem using least angle regression is similar to the SLR in [16]. But instead of including coefficients at each step, the coefficients are increased in their joint least squares direction (see below). We describe the algorithm as follows:

1. Standardize the terms $Z_k^u, \forall u \in \mathbb{U}$ to have zero mean and unit variance, where $Z_k^u = (Z_{i,k}^u)_{i \in \{1,2,\ldots,N\}}$.
2. Start with $\dot{T} = T - mean(T)$, an empty set $\mathbb{A}$ and $\alpha_u = 0$ for all $u \in \mathbb{U}$ where $mean(T)$ is the mean of T.
3. Find u satisfying $u = \text{argmax}_u |(Z_k^u)^\mathsf{T} \dot{T}|$, and add u to set $\mathbb{A}$.

4. Increase all the coefficients α_j, in their joint least squares direction, i.e., the direction is: $((\boldsymbol{Z}_k^{\mathbb{A}})^{\mathsf{T}}\boldsymbol{Z}_k^{\mathbb{A}})^{-1}(\boldsymbol{Z}_k^{\mathbb{A}})^{\mathsf{T}}T$, where matrix $\boldsymbol{Z}_k^{\mathbb{A}} = (Z_{i,k}^u)_{i\in\{1,\ldots,N\},u\in\mathbb{A}}$, where $j \in \mathbb{A}$, and take residuals $\dot{T} = T - \sum_{u\in\mathbb{A}} Z_k^u \alpha_u$ along the way until another $\bar{u}$ reaches $\mathrm{argmax}_{\bar{u}}|(Z_k^{\bar{u}})^{\mathsf{T}}\dot{T}|$. In the above process, whenever any coefficients α_l for $l \in \mathbb{A}$ reaches zero, remove l from $\mathbb{A}$. Add $\bar{u}$ to set $\mathbb{A}$.
5. Repeat items 3 and 4 until $\forall u : (Z_k^u)^{\mathsf{T}}\dot{T} = 0$ is satisfied.

References

1. Batina, L., Gierlichs, B., Lemke-Rust, K.: Differential cluster analysis. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 112–127. Springer, Heidelberg (2009)
2. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
3. Carlet, C.: Boolean functions for cryptography and error correcting codes. Boolean Model. Meth. Math. Comput. Sci. Eng. **2**, 257–397 (2010)
4. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: 4th International Workshop Cryptographic Hardware and Embedded Systems - CHES 2002, Redwood Shores, CA, USA, 13–15 August 2002, pp. 13–28, Revised Papers (2002)
5. Durvaux, F., Standaert, F.-X., Veyrat-Charvillon, N.: How to certify the leakage of a chip? In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 459–476. Springer, Heidelberg (2014)
6. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)
7. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 426–442. Springer, Heidelberg (2008)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, vol. 1, Second edn. Springer, New York (2009)
9. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
10. 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, Heidelberg (2011)
11. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 30–46. Springer, Heidelberg (2005)
12. Standaert, F.-X., Malkin, T.G., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (2009)
13. Veyrat-Charvillon, N., Standaert, F.-X.: Mutual information analysis: how, when and why? In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 429–443. Springer, Heidelberg (2009)
14. Veyrat-Charvillon, N., Standaert, F.-X.: Generic side-channel distinguishers: improvements and limitations. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 354–372. Springer, Heidelberg (2011)

15. Whitnall, C., Oswald, E., Mather, L.: An exploration of the Kolmogorov-Smirnov test as a competitor to mutual information analysis. In: Prouff, E. (ed.) CARDIS 2011. LNCS, vol. 7079, pp. 234–251. Springer, Heidelberg (2011)
16. Whitnall, C., Oswald, E., Standaert, F.-X.: The myth of generic DPA..and the magic of learning. In: Benaloh, J. (ed.) CT-RSA 2014. LNCS, vol. 8366, pp. 183–205. Springer, Heidelberg (2014)