Ridge-Based Profiled Differential Power Analysis

Weijia Wang[1], Yu Yu[1,3,4](✉), François-Xavier Standaert[2], Dawu Gu[1],
Xu Sen[1], and Chi Zhang[1]

[1] School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai, China
{aawwjaa,yyuu,dwgu,push.beni,liujr,guozheng}@sjtu.edu.cn
[2] ICTEAM/ELEN/Crypto Group,
Université catholique de Louvain, Louvain-la-Neuve, Belgium
fstandae@uclouvain.be
[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[4] Westone Cryptologic Research Center, Beijing, China

Abstract. Profiled DPA is an important and powerful type of side-channel attacks (SCAs). Thanks to its profiling phase that learns the leakage features from a controlled device, profiled DPA outperforms many other types of SCA and are widely used in the security evaluation of cryptographic devices. Typical profiling methods (such as linear regression based ones) suffer from the overfitting issue which is often neglected in previous works, i.e., the model characterizes details that are specific to the dataset used to build it (and not the distribution we want to capture). In this paper, we propose a novel profiling method based on ridge regression and investigate its generalization ability (to mitigate the overfitting issue) theoretically and by experiments. Further, based on cross-validation, we present a parameter optimization method that finds out the most suitable parameter for our ridge-based profiling. Finally, the simulation-based and practical experiments show that ridge-based profiling not only outperforms 'classical' and linear regression-based ones (especially for nonlinear leakage functions), but also is a good candidate for the robust profiling.

Keywords: Side-channel attack · Profiled DPA · Linear regression · Ridge regression · Cross-validation

1 Introduction

Side-channel attacks (SCAs) exploit the physical information leaked from the implementation of a cryptographic algorithm, and they are usually more powerful than brute-force attacks or classical cryptanalytic techniques that target at the mathematical weakness of the underlying algorithm. Differential power analysis (DPA), proposed by Kocher et al. [15], is a form of side-channel attack that efficiently recovers the secret key from multiple (typically noisy) power consumption measurements (on different plaintexts). Profiled DPA (e.g., [3,20,24]) adds a profiling phase (prior to the online exploitation phase) to the original

H. Handschuh (Ed.): CT-RSA 2017, LNCS 10159, pp. 347–362, 2017.
DOI: 10.1007/978-3-319-52153-4_20

DPA and can be considered as a powerful class of power analysis. The profiling phase learns the leakage function from the power consumption of a training device, and it can significantly enhance the performance of the subsequent online exploitation phase, namely, the key recovery attack mounted against a similar target device. We will focus on the profiling phase in this paper.

Chari et al. [3] proposed the first profiled DPA called template attacks, whose profiling phase is based on multivariate Gaussian templates. We refer to the profiling phase of templates attacks as classical profiling (following the terminology in [24]). Later Schindler et al. [20] proposed a very promising profiled DPA that uses linear regression (LR) as its profiling method (referred to as LR-based profiling hereafter). Compared with classical profiling, LR-based profiling builds up a model more efficiently with less number of measurements and it allows a trade-off between the profiling and online exploitation phases: more measurements used in the profiling phase, less measurements needed in the exploitation phase [8,22,24]. However, the LR-based profiling suffers from the overfitting issue in practice. That is, noisy measurements in the profiling phase can result in a model that describes mostly the noise instead of the actual leakage function. Thus, the LR-based profiling may need more measurements than necessary. We mention other profiling methods those based on agglomerative hierarchical clustering [25], K-means [25] and different machine learning methods such as SVM [12,14,16], random forests [16,17], neural networks [18,19], which enjoy additional features or are more useful for specific data structures or have an overhead for the time complexity. We are not extending this line of research any further.

In this paper, we propose a new profiling method (named ridge-based profiling) based on ridge regression. By imposing a constraint on the coefficients of linear regression, ridge regression is a good alternative to linear regression with better performance on noisy data [11]. As the constraint (described by a parameter) affects the performance of ridge-based profiling, we apply the K-fold cross-validation to find out the most suitable constraint (i.e., the optimized parameter) for ridge-based profiling. We also conduct experiments of the above parameter optimization in settings of various noise levels. Our results suggest that the optimized parameter is related to the noise level of measurements (i.e., the optimized parameter increases with respect to the noise level).

We analyze the ridge-based profiling both in theory and by experiments. Our theoretical investigation aims to answer the question:

Why, how and when is ridge-based profiling better?

where ‘why’ aims to justify the improvement of ridge-based profiling over LR-based one, ‘how’ and ‘when’ analyze to which extent and under what condition an improvement can be achieved. Then for a comprehensive comparison, we evaluate the performances of classical, LR-based and ridge-based profiling in simulation-based experiments on various settings, which shows the improvement of ridge-based profiling and confirm the theoretical analysis, At last, we conduct the practical experiments on the FPGA implementation. The results are consistent to the ones of simulation-based experiments, and furthermore, they show that the ridge-based profiling can tolerate (some) differences between profiling and exploitation traces, resulting in a type of robust profiling [25]. Therefore, on one hand, our results can be considered as an improvement of [3,20,24]. And on the other hand, we extend the related works which applied the stepwise and ridge regressions to the non-profiled setting [23,26].

2 Background

Following the 'divide-and-conquer' strategy, a profiled 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., $X = (X_i)_{i \in \{1,\dots,n\}}$, 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 $\mathrm{F}_k : \mathbb{F}_2^m \to \mathbb{F}_2^m$ be a target function, where m is the bit length of X_i, and thus the intermediate value $Z_{i,k} = \mathrm{F}_k(X_i)$ is called a target and $Z_k = \mathrm{F}_k(X) = (Z_{i,k})_{i \in \{1,\dots,N\}}$ is the target vector obtained by applying F_k to X component-wise.

The leakage of a target can be scattered over several points in a measurement's power consumption. Let $\mathrm{L}^j : \mathbb{F}_2^m \to \mathbb{R}$ be the leakage function at jth point and let T_i be a vector of power consumption points whose target is Z_{i,k^*}. We have $T_i^j = \mathrm{L}^j \circ Z_{i,k^*} + \varepsilon_j$ and $T^j = \mathrm{L} \circ Z_{k^*} + \varepsilon_j$, where $\circ$ denotes function composition, k^* is the correct subkey key and ε_j 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 $\mathrm{M}^j : \mathbb{F}_2^m \to \mathbb{R}$ be the model that approximates the determinate part of leakage function L^j, namely, $T_i^j \approx \mathrm{M}^j \circ F_{k^*}(X_i) + \varepsilon_j$.[1] The model is obtained by learning from the profiled device in the profiling phase.

Profiled DPA can be divided into two phases: profiling phase and online exploitation phase. In the rest of this section, we recall these two phases. Our presentation is largely based on the (excellent) introduction provided in [24].

2.1 Profiling Phase

The aim of the profiling phase is to 'learn' the leakage functions L^j and the noises ε_j for all the points. We briefly introduce classical and LR-based profilings below.

Classical profiling. Classical profiling is the profiling phase of template attacks [3] and it views the leakage of each intermediate value as a vector of random values following the multivariate Gaussian distribution, i.e., $T_z \sim \mathrm{N}(\mu_z, \Sigma_z)$, where T_z is the power consumption (points) given the associated intermediate target being z. The adversary 'learns' the physical leakages by finding the $p \times 1$ sample mean $\hat{\mu}_z$ and the $p \times p$ sample covariance $\hat{\Sigma}_z$ for all the target z on the profiling device. Finally, the intermediate value-conditioned leakages is $\mathrm{N}(\hat{\mu}_z, \hat{\Sigma}_z)$ for the intermediate value z. As suggested in [4], we assume the noise distribution of

[1] We often omit the superscript 'j' in L^j, M^j and ε^j for succinctness.

different intermediate targets to be equal and use the same covariance estimates (across all intermediate targets).

Linear regression-based profiling. LR-based profiling [20] uses the stochastic model of the following form: $M(Z_i) = \alpha_0 + \sum_{u \in \mathbb{F}_2^m} \alpha_u Z_i^u + \varepsilon$, where coefficients $\alpha_u \in \mathbb{R}$, $Z_i = Z_{i,k^*}$, 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). The degree of the model is the highest degree of the non-zero terms in polynomial $M(Z_i)$. Define the set $\mathbb{U}_d = \{u | u \in \mathbb{F}_2^m, \mathrm{HW}(u) \leq d\}$ (where HW : $\mathbb{F}_2^m \rightarrow \mathbb{Z}$ is the Hamming weight function), then we denote $\boldsymbol{\alpha}_d = (\alpha_u)_{u \in \mathbb{U}_d}$ as the vector of coefficients with degree d, which is estimated from $\boldsymbol{U}_d = (Z_i^u)_{i \in \{1,2,\ldots,N\}, u \in \mathbb{U}_d}$ and T using ordinary least squares, i.e., $\boldsymbol{\alpha}_d = (\boldsymbol{U}_d^{\mathrm{T}} \boldsymbol{U}_d)^{-1} \boldsymbol{U}_d^{\mathrm{T}} T$, where $(Z_i^u)_{i \in \{1,2,\ldots,N\}, u \in \mathbb{U}}$ is a matrix with (i,u) being row and column indices respectively, and $\boldsymbol{U}_d^{\mathrm{T}}$ is the transposition of $\boldsymbol{U}_d$.

In the LR-based profiling phase, the adversary chooses the degree of model and calculates the coefficients $\boldsymbol{\alpha}$ of the profiling device. Then, the $p \times p$ sample covariance $\hat{\Sigma}$ is computed assuming the noise distributions are identical for various values of intermediate. Finally, the intermediate value-conditioned leakages is $\mathrm{N}(\hat{\alpha}_0 + \sum_{u \in \mathbb{U}_d} \hat{\alpha}_u z_i^u, \hat{\Sigma})$ for the intermediate value z.

2.2 Online Exploitation Phase

Bayesian key recovery. If the covariance matrix is symmetric and positive definite, a p-dimensional multivariate Gaussian distribution $\mathrm{N}(\mu, \Sigma)$ has the following density function:

$$f(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)\right) \ . \quad (1)$$

Therefore, we can describe Bayesian key recovery as follows:

1. Acquire n traces (T_i, X_i), each of p points, for $1 \leq i \leq n$ from the target device.
2. Make a subkey guess k and compute the corresponding intermediate target $Z_{i,k} = F_k(X_i)$ for $1 \leq i \leq n$.
3. Calculate the log likelihood: $\prod_{i=1}^{n} \log(f_{i,k}(T_i))$, where $f_{i,k}(\cdot)$ is the density function associated with the intermediate target $Z_{i,k}$.
4. The log likelihood should be maximum upon correct key guess (which can be decided after repeating the above for all possible subkey guesses).

Correlation DPA. Correlation DPA employs a simple (univariate) online exploitation strategy, and it finds the subkey guess under which the correlation between the determinate part of the template (e.g., $\mathrm{M}_{classical}(z) = \hat{\mu}_z$ in 'classical' profiling and $\mathrm{M}_{LR}(z) = \hat{\alpha}_0 + \sum_{u \in \mathbb{F}_2^m} \hat{\alpha}_u z_i^u$ in LR-based profiling) and the (univariate) leakage is maximized, namely,

$$k_{guess} = \operatorname*{argmax}_{k} \rho(\mathrm{M}(Z_{i,k}), T_i) \quad (2)$$

where ρ is the Pearson's coefficient.

3 Ridge-Based Profiling

In this section, we introduce our ridge-based profiling and give a formal analysis. We consider only the deterministic part of the model, and meanwhile the sample variance $\hat{\Sigma}$ is considered the same way as LR-based profiling.

3.1 Construction

Our new profiling (for each power consumption point) can be see as a generalization of LR-based profiling by explicitly imposing penalty on the coefficients' size, formally,

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_d^{ridge} \overset{\mathsf{def}}{=} \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} \left(T_i - \mathrm{M}_d^{ridge}(Z_i) \right)^2, \\ \text{subject to} \sum_{u \in \mathbb{U}_d} \alpha_u^2 \leq s. \end{aligned} \tag{3}$$

An equivalent formulation to above is (see [11] for detailed derivation):

$$\hat{\boldsymbol{\alpha}}_d^{ridge} = \underset{\alpha}{\operatorname{argmin}} \left(\sum_{i=1}^{N} \left(T_i - \mathrm{M}_d^{ridge}(Z_i)\right)^2 + \lambda \sum_{u \in \mathbb{U}_d} \alpha_u^2 \right), \tag{4}$$

whose optimal solution is given by:

$$\hat{\boldsymbol{\alpha}}_d^{ridge} = (\boldsymbol{U}_d^{\mathrm{T}} \boldsymbol{U}_d + \lambda \boldsymbol{I}_d)^{-1} \boldsymbol{U}_d^{\mathrm{T}} T, \tag{5}$$

where $\mathbb{U}_d$, $\boldsymbol{U}_d$ and Z_i are defined in Sect. 2.1, matrix $\boldsymbol{I}_d$ is the $|\mathbb{U}_d| \times |\mathbb{U}_d|$ identity matrix and $|\mathbb{U}_d|$ denotes the cardinality of $\mathbb{U}_d$.

Parameter optimization. As illustrated above, there is an undetermined parameter (i.e., λ), the choice of which affects the performance of the profiling. For each power consumption point, we propose a method to choose the optimized parameter based on the K-fold[2] cross-validation technique from statistical learning. We mention that cross-validation was already used in the field of side-channel attack (for different purposes), such as evaluation of side-channel security [6] and unprofiled DPA [23]. Algorithm 1 finds the optimized parameter using cross-validation, where we omit the subscript d (the degree) for succinctness.

We sketch the algorithm below. We first choose a set of candidate parameters (up to some accuracy), and then split profiling traces into K parts $\mathcal{C}_{\{1...K\}}$ of roughly equal size. For each part $\mathcal{C}_i$, we compute the coefficients $\boldsymbol{\alpha}_{\lambda,i}$ using the remaining $K-1$ parts from the trace set, and calculate the goodness-of-fit $R_{\lambda,i}$ using the traces in $\mathcal{C}_i$, which is a measurement of similarity between estimated power consumption and the actual power consumption T.[3] We then get the

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

[3] We use the coefficient of determination to measure the goodness-of-fit in this paper, i.e., $R = \sum_{i=1}^{N_t}(\hat{T}_i - T_i)^2 / \sum_{i=1}^{N_t}(T_i - \sum_{i=1}^{N_t} T_i)^2$, where $\hat{T}$ is the estimated power consumption and N_t is the trace number in $\mathcal{C}_i$.

Algorithm 1. finding the optimized parameter

```
Require: profiling traces $t_i = \{T_i, x_i\}$ where $i \in \{1, ..., N\}$; the number of parts $K$;
    the true key $k^*$; the set of candidate parameters $\Lambda$;
Ensure: $\hat{\lambda}$ as the optimized parameter for the subkey;
for $i = 1$; $i <= K$; $i$++ do
    $\mathcal{C}_i = \{t_{K*(i-1)+1}, ..., t_{K*i}\}$
end for
for all $\lambda$ such that $\lambda \in \Lambda$ do
    for $i = 1$; $i <= K$; $i$++ do
        Compute the $\boldsymbol{\alpha}_{\lambda,i}$ using the traces in $\mathcal{C}_j$, where $j \in \{1...K\} \setminus \{i\}$
        Calculate the goodness-of-fit $R_{\lambda,i}$ from $\mathcal{C}_i$
    end for
    $R_\lambda = (\sum_{i=1}^{K} R_{\lambda,i})/K$
end for
$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} R_\lambda$
```

average goodness-of-fit $R_\lambda = (\sum_{i=1}^{K} R_{\lambda,i})/K$ for the each candidate parameter λ in consideration. Finally, we return the parameter with the highest averaged goodness-of-fit.

3.2 Theoretical Analysis

In this sub-section, we investigate the improvement of ridge-based profiling (over LR-based one) theoretically. We first answer the 'why' and 'how' questions by analyze the sampling variance of model's coefficients. Then we answer the 'when' question by studying the way that the coefficients shrink in the ridge-based profiling.

Why and How is Ridge-Based Profiling Better? For simplicity we consider the univariate leakage, where the leakage of the i-th trace is $T_i = \mathrm{L} \circ Z_{i,k^*} + \varepsilon$. Since the coefficients learned from the LR-based (resp., ridge-based) profiling determine the model (by definition), varying the coefficients will affect stability of the performance. The variance-covariance matrix of the coefficients learned from the LR-based (resp., ridge-based) profiling are given by [13, Eq. 4.8]:

$$\mathrm{Var}(\boldsymbol{\alpha}_d^{lr}) = (\boldsymbol{U}_d^{\mathrm{T}} \boldsymbol{U}_d)^{-1} \sigma^2 \tag{6}$$

$$\mathrm{Var}(\boldsymbol{\alpha}_d^{ridge}) = \boldsymbol{W} \boldsymbol{U}_d^{\mathrm{T}} \boldsymbol{U}_d \boldsymbol{W} \sigma^2 \tag{7}$$

where $\boldsymbol{W} = (\boldsymbol{U}_d^{\mathrm{T}} \boldsymbol{U}_d + \lambda \boldsymbol{I}_d)^{-1}$ and σ^2 is the variance of noise ε, which is identical for both LR-based and ridge-based profilings.

Without loss of generality, we fix $\sigma^2 = 1$ and the target values to be bytes, then compare $\mathrm{Var}(\boldsymbol{\alpha}_d^{lr})$ to $\mathrm{Var}(\boldsymbol{\alpha}_d^{ridge})$. Figure 1 shows that the variances goes up with the increase of d and the decrease of λ. For the same degree and parameter, the variance learned from ridge-based profiling are much lower than the ones from

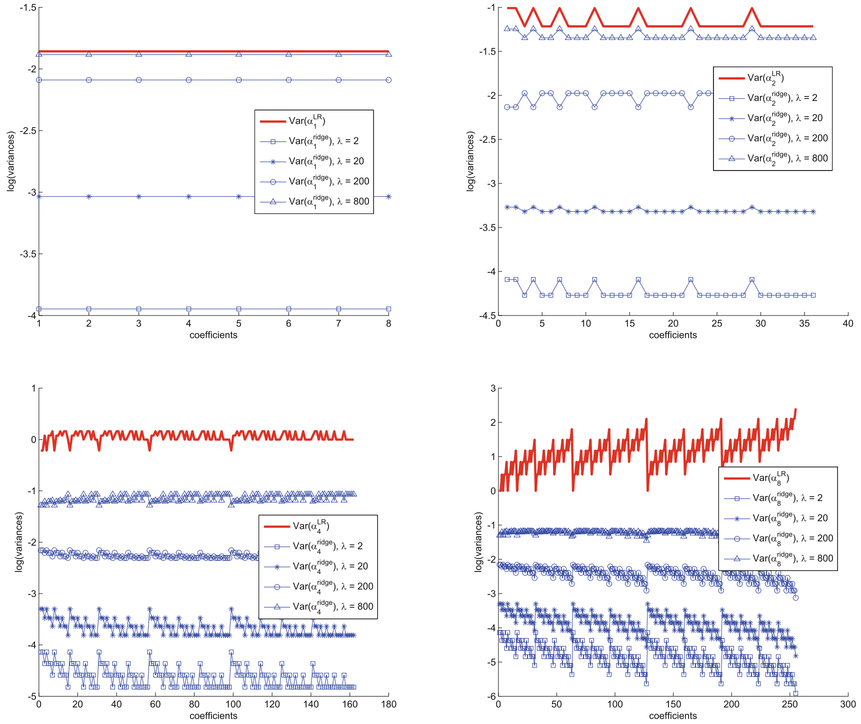


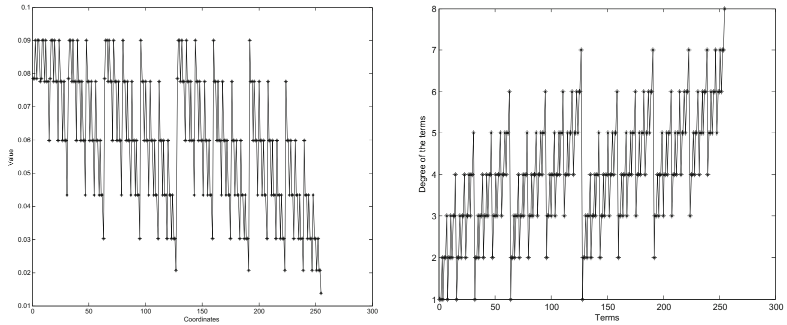
Fig. 1. The variances of the coefficients for different degrees (of the model) and λ. The upper-left, upper-right, lower-left, and lower-right figures correspond to the cases for $d = 1$, $d = 2$, $d = 4$, and $d = 8$ respectively.

LR-based profiling, thus the former has a more stable performance and is less prone to noise. Thus, to avoid overfitting one may use a large λ, but then it may result in a biased model, i.e., the difference between the leakage function and the model becomes more significant, which also decreases performance. Therefore, for best performance we need to choose a judicious value for λ by reaching a tradeoff between bias and coefficients' variance. To this end, we propose to use the cross-validation method in parameter optimization (see Sect. 3.1).

How the Coefficients Shrink in the Ridge-Based Profiling? As described before, the ridge-based profiling 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). In [23], 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 is shown. See the following for a brief introduction and a conclusion of the analysis, and we refer to [23] for more details.

The principal components of $\boldsymbol{U}_d$ are a set of linearly independent vectors obtained by applying an orthogonal transformation to $\boldsymbol{U}_d$, i.e., $\boldsymbol{P}_d = \boldsymbol{U}_d\boldsymbol{V}_d$, where the columns of matrix $\boldsymbol{V}_d$ are called directions of the (respective) principal components. An interesting property is that among the columns of $\boldsymbol{V}_d$, the first one, denoted V_d^1 (the direction of P_d^1), has the maximal correlation to coefficient vector $\boldsymbol{\alpha}_d$. Figure 2(a) and (b) depict the direction of the first principal component V_8^1 and the degrees of terms in $\boldsymbol{U}_8$ respectively, and they represent a high similarity (albeit in a converse manner). Quantitatively, the Pearson's coefficient between V_8^1 and the corresponding vector of degrees is -0.9704, which is a nearly perfect negative correlation. Therefore, we establish the connection that α_u is conversely proportional to the Hamming weight of u. Above analysis is based on the $d = 8$ setting, for the other degrees (1 to 7), similar results can be obtained. To summarize, the more Hamming weight that u has, the less α_u contributes to the model. Therefore, ridge-based distinguisher is consistent with the leakage functions that consist of more low degree terms.

Another observation is that the improvement of ridge-based profiling (over LR-based one) is significant only for non-linear models (used for profiling). We can see that for the model of degree 1 the u(s) of all coefficients have same Hamming weight, and thus every coefficient contributes equally to the model. That is, the coefficients shrink equally in this setting, which leads to comparable performance for both ridge-based and LR-based profilings. However, we stress that the degree of the model (for profiling) is not the same as (and typically no less than) that of the leakage function, and ridge-based profiling can just still enjoy performance improvement for linear leakage functions by setting the degree of model to be greater than 1. We refer to Sect. 4.1, where we will show that the ridge-based profiling outperforms the LR-based one for leakage function of degree 1 and model of degree 4.



(a) The value of V_8^1 (the direction of the first principal component of $\boldsymbol{U}_8$).

(b) The degrees of the terms in $\boldsymbol{U}_8$.

Fig. 2. The similarity between the direction of the first principle component V_8^1 and the degrees of terms in $\boldsymbol{U}_8$

4 Experimental Results

4.1 Simulation-Based Experiments

In this section, we evaluate the ridge-based, LR-based and classical profiling for univariate leakage functions with different degrees and randomized coefficients in the setting of simulated traces. We target at 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). We do the following trace pre-processing to facilitate the profiling: we average the traces based on their the input (an 8-bit plaintext) and use the resulting 256 mean power traces to mount the profiling. This reduces noise and the number of traces needed for profiling (as otherwise the running time goes unnecessarily high with a large number of 'raw' traces).

Finding the Optimized Parameter. At the beginning of ridge-based profiling, the adversary should first find the optimized parameter (i.e., the λ). We evaluate parameter optimization algorithm from Sect. 3.1. We consider the settings whose the degrees (of both leakage function and model) are fixed to 4 and under different signal-noise ratios (SNRs) (0.5, 0.1, 1). Let the set of parameter choices be $\Lambda = \{0.1, 1, 10, 50, 200, 800, 2000, 8000\}$, for which we conduct the parameter optimization algorithm 100 times (each time with a different random leakage function). For a fair comparison, we normalized[4] the averaged goodness-of-fits (of each experiment) and plot them in Fig. 3. We also highlight the mean of the averaged goodness-of-fits with red bold line. This confirms the intuition that the optimized parameter (which corresponds to each setting's minimum averaged goodness-of-fit) decreases with SNR.

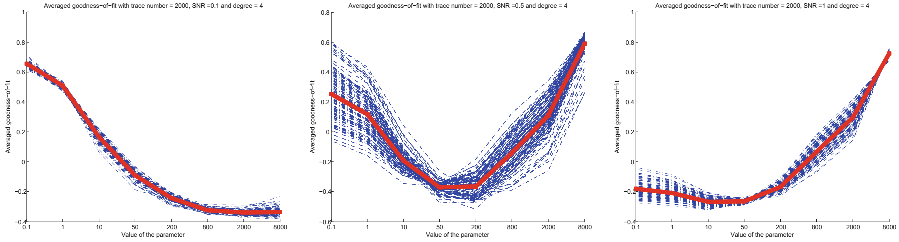


Fig. 3. Averaged goodness-of-fits and their mean values, with SNR = 0.1 (left-hand), 0.5 (middle), 1 (right-hand). (Color figure online)

[4] We apply the averaged goodness-of-fit for normalization, i.e., $\text{norm}(R_\lambda) = (R_\lambda - \text{mean}(R)/(\max(R) - \min(R)))$, where $\text{mean}(R)$ is the average of $\{R_\lambda\}_{\lambda \in \Lambda}$ and $\text{norm}(\cdot)$ is the normalization function.

A Comparison of Different Profilings in Simulation-Based Experiments. We compare different profilings (i.e., classical, LR-based and ridge-based profiling) using two metrics, namely, theoretical information and guessing entropy. The former computes the Perceived Information (PI) [6] between the secret variable and its leakage, and the latter combines the correlation DPA with the model built from one of three different profilings above and mounts the attack 100 times (each time with a different random leakage function) to compute the averaged ranking of the real key.

Figure 4 compares the Perceived Information and guessing entropies (as functions of the number of profiling traces) for different degrees of leakage function. The left-hand three sub-figures show the Perceived Information and the right-hand ones present the guessing entropies. The two sub-figures of the same row correspond to the Perceived Information and guessing entropy for leakage functions of the same degree respectively. Intuitively, the PI is an information theoretic metric that relates to the success rate of a profiled adversary using the estimated model obtained thanks to LR-based or ridge-based regression [5]. So it is the most revealing metric for comparing profiling phases [22]. In particular, the left parts of Fig. 4 exhibit both the informativeness of the model after sufficient profiling (i.e. the final Y axis values) and the efficiency of the profiling (i.e. how fast we converge towards this value). The guessing entropy metric is used as a confirmation that this intuition is matched and could be computed for any number of traces in the exploitation phase. In the profiling phase, we choose the same degree for the model and the leakage function. For all scenarios, the two metrics are consistent: the PI increases and the guessing entropies approaches to 1 with the increase of the number of traces. As clear from the PI figures, the ridge-based profiling performs better than the other two ones in all settings except for the $d = 1$ setting. More precisely, it generally has a better convergence speed, without any significant reduction of the final informativeness. Meanwhile the performance of LR-based profiling lies in between classical and ridge-based ones and it is largely affected by the degree of the leakage function. These observations confirm the theoretical analysis in Sect. 3.2. The guessing entropies computed in function of the number of profiling traces (for a fixed number of attack traces) confirm these trends.

Note that the typical scenario we are interested in is when the adversary has no knowledge about the actual degree of the leakage function for his profiling. In this case, our results show that he may use a conservative estimate about the degree of the model in the profiling phase without loosing efficiency (i.e. speed of convergence). To reflect this case, we also conduct experiments where the estimated degree of the model is higher that its actual value. That is, we simulate the traces with leakage functions of degrees 1 and 2 and then conduct the above experiments assuming a model of degree 4 for profiling. As shown in Fig. 5, the performance of ridge-based profiling is again significantly better. Therefore, our results show that an adversary (or an evaluation laboratory) can simply use a 'conservatively' estimated degree in ridge-based profiling, instead of running an enumeration on its possible values.

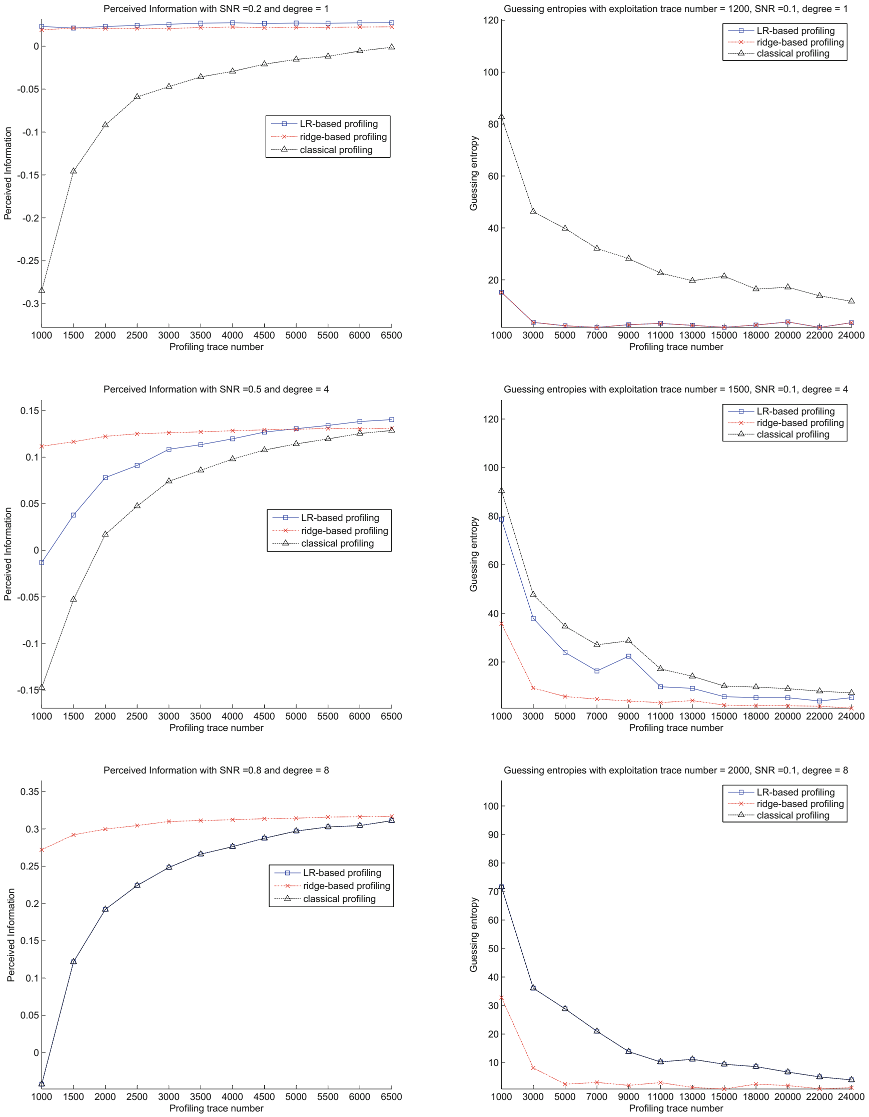


Fig. 4. A comparison of Perceived Information and guessing entropies (in functions of the number of profiling traces) for different degrees of leakage function, where the rows correspond to degrees 1, 4 and 8 respectively.

4.2 Experiments on Real FPGA Implementation

We carry out experiments on the SAKURA-X which running the AES on Xilinx FPGA devices Kintex-7 (XC7K70T/160T/325T). We amplified the signal using a (customized) LANGER PA 303N amplifier, providing 30 dB of gain. Then we measure the (absolute value of) power consumptions of the first round

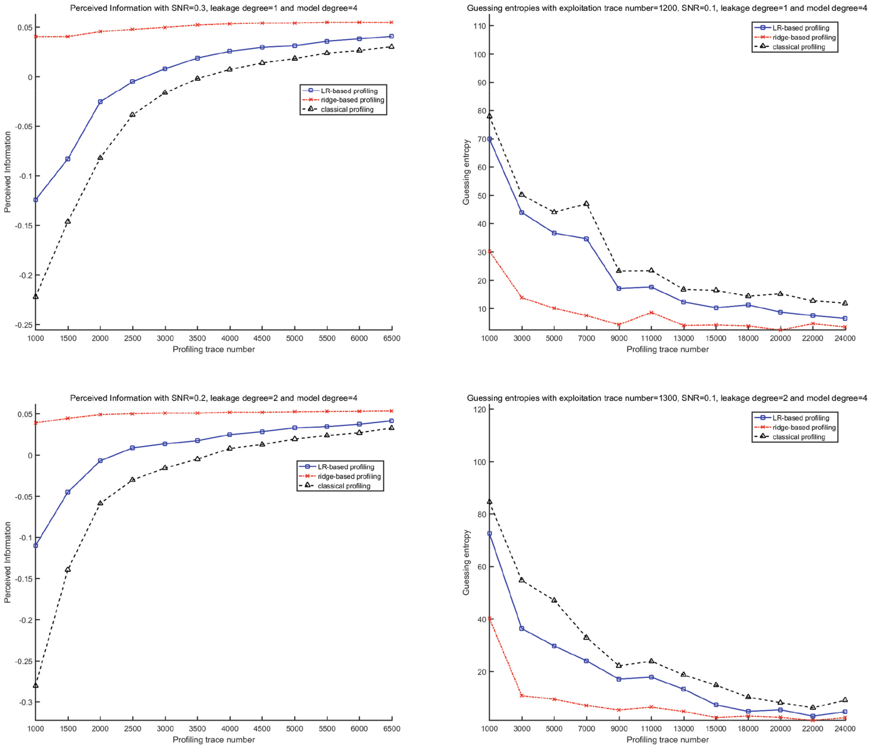


Fig. 5. The Perceived Information and guessing entropies with 'conservatively' degree of model for different numbers of exploitation traces, where the rows correspond to degrees of leakage function 1 and 2 respectively, and the degree of both models is 4.

S-box output, using a LeCroy waverunner 610Zi digital oscilloscope at a sampling rate of 1 GHz. Figure 6 shows the averaged trace[5] of the measurements of first round, we marked the leakage regions of the intermediate variable (i.e., the S-box output) in the figure and target them in our following attacks. We can see that the intermediate variable leaks in both region A and B similarly. Additionally, for each region, we apply the principal component analysis (PCA) to compact measurements [1,2,21], then only target the point of first principal component. And before the profiling, we perform the pre-processing, whose results are 256 mean traces of single point. In the following, to better illustrate the improvement of our new proposed method, we conduct two experiments for two different settings, in which we always profile on points of region A but attack (do the exploitation) on points of different regions.

[5] We shall not confuse the 'averaged trace' with the '256 mean power traces', where the former one is the mean of all the power traces which is only for the presentation of the measurements. And the latter one, as the result of pre-processing, is the means of the traces of same corresponding plaintext.

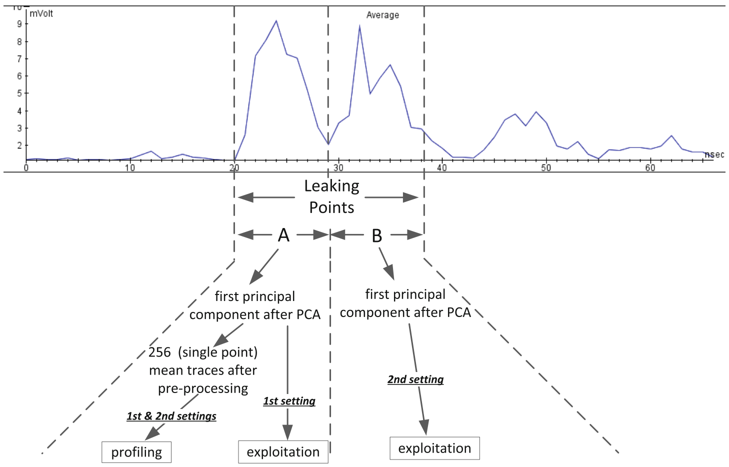
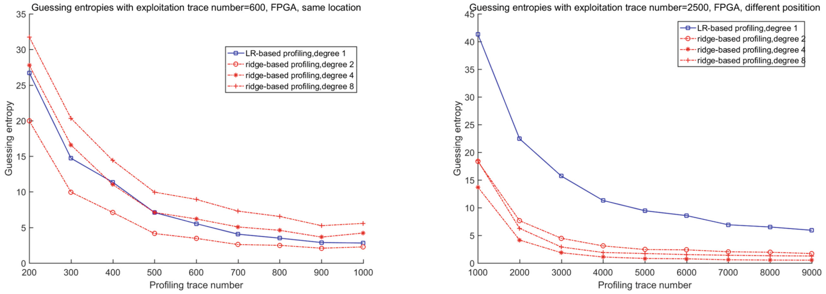


Fig. 6. The average trace of the measurements and the leaking points.

First, we assume a common setting (the 1st setting in Fig. 6) where the profiling and exploitation points are perfect aligned, thus we use the same region (i.e., region A) for both profiling and exploitation. Figure 7(a) shows the guessing entropies (as functions of the number of profiling traces) for ridge-based with different degrees power model in this setting. The parameter (i.e., $\lambda = 8000$) is chosen by means of the cross-validation as simulation-based experiments. We present the guessing entropies of the LR-based profiling with power model of degree 1 as the base line, since (in our attack scenario) it outperforms the LR-based profiling with higher degree as well as the classical one. We can see that the degree of the leakage function of our implementation is around 2. The result shows that (under this implementation) the ridge-based profiling with power model of degree 2 is the best one and perform better than the LR-based one (with power model of degree 1), which is consistent to the results of simulation-based experiments and theoretical analysis.

Further, we conduct another experiments to show that our new method can be used as a type of robust profiling [25], which can tolerate (some) differences between profiling and exploitation traces in a more realistic setting. As shown in Fig. 6 (the 2nd setting), we profile on the points in A and attack (do the exploitation) on the points in B. We aim to show how the miss-alignment of the points affects the ridge-based profiling. Figure 7(b) presents the guessing entropies (as functions of the number of profiling traces) for ridge-based with different degrees power model. We choose a larger parameter $\lambda = 500000$ by using the parameter optimization process in Sect. 3.1. We also add the LR-based profiling (with power model of degree 1) as the base line. The results show that the performance of ridge-based profiling is better than the LR-based one, which means that the performance of the new profiling method is better robust than LR-based one to the distortions between profiling and exploitation points.



(a) standard scenario

(b) scenario of robust profiling

Fig. 7. A comparison of guessing entropy (in functions of the number of profiling traces) for FPGA implementation.

5 Conclusion

In this paper, we propose a new profiled differential power analysis based on ridge regression. Our theoretical analysis and experiments double confirm that the proposed profiling method has better performance than LR-based one by using a more stable (to avoid overfitting) and has a good potential to be a type of robust profiling. In view of the importance of profiled based side-channel analysis in security evaluations, these results show ridge-based profiling can allow laboratories to save significant factors in the number of traces they need to build a satisfying leakage model.

Acknowledgments. This work has been funded in parts by Major State Basic Research Development Program (973 Plan), the European Commission through the ERC project NANOSEC and by the INNOVIRIS project SCAUT. Yu Yu was supported by the National Natural Science Foundation of China Grant (Nos. 61472249, 61572192, 61572149), Science and Technology on Communication Security Laboratory (9140C110203140C11049), and International Science & Technology Cooperation & Exchange Projects of Shaanxi Province (2016KW-038). François-Xavier Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). Dawu Gu was supported by National Natural Science Foundation of China (No. 61472250).

References

1. Archambeau, C., Peeters, E., Standaert, F., Quisquater, J.: Template attacks in principal subspaces. In: Goubin, L., Matsui, M. (eds.) [9], pp. 1–14
2. Batina, L., Hogenboom, J., Woudenberg, J.G.J.: Getting more from PCA: first results of using principal component analysis for extensive power analysis. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 383–397. Springer, Heidelberg (2012). doi:10.1007/978-3-642-27954-6_24
3. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003). doi:10.1007/3-540-36400-5_3

4. Choudary, O., Kuhn, M.G.: Efficient template attacks. In: Francillon, A., Rohatgi, P. (eds.) [7], pp. 253–270. http://dx.doi.org/10.1007/978-3-319-08302-5
5. Duc, A., Faust, S., Standaert, F.-X.: Making masking security proofs concrete. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 401–429. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46800-5_16
6. 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). doi:10.1007/978-3-642-55220-5_26
7. Francillon, A., Rohatgi, P. (eds.): CARDIS 2013. LNCS, vol. 8419. Springer, Cham (2014). http://dx.doi.org/10.1007/978-3-319-08302-5
8. Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. stochastic methods. In: Goubin, L., Matsui, M. (eds.) [9], pp. 15–29
9. Goubin, L., Matsui, M. (eds.): CHES 2006. LNCS, vol. 4249. Springer, Heidelberg (2006)
10. Güneysu, T., Handschuh, H. (eds.): CHES 2015. LNCS, vol. 9293. Springer, Heidelberg (2015). http://dx.doi.org/10.1007/978-3-662-48324-4
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn., vol. 1, pp. 43–94. Springer, New York (2009)
12. Heuser, A., Zohner, M.: Intelligent machine homicide. In: Schindler, W., Huss, S.A. (eds.) COSADE 2012. LNCS, vol. 7275, pp. 249–264. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29912-4_18
13. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)
14. Hospodar, G., Gierlichs, B., Mulder, E.D., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. J. Cryptographic Eng. **1**(4), 293–302 (2011)
15. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). doi:10.1007/3-540-48405-1_25
16. Lerman, L., Bontempi, G., Markowitch, O.: Power analysis attack: an approach based on machine learning. IJACT **3**(2), 97–115 (2014)
17. Lerman, L., Poussier, R., Bontempi, G., Markowitch, O., Standaert, F.-X.: Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis). In: Mangard, S., Poschmann, A.Y. (eds.) COSADE 2014. LNCS, vol. 9064, pp. 20–33. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21476-4_2
18. Martinasek, Z., Hajny, J., Malina, L.: Optimization of power analysis using neural network. In: Francillon, A., Rohatgi, P. (eds.) [7], pp. 94–107. http://dx.doi.org/10.1007/978-3-319-08302-5
19. Quisquater, J., Samyde, D.: Automatic code recognition for smartcards using a kohonen neural network. In: Proceedings of the Fifth Smart Card Research and Advanced Application Conference, CARDIS 2002, November 21–22, 2002, San Jose, CA, USA (2002)
20. 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). doi:10.1007/11545262_3
21. Standaert, F.-X., Archambeau, C.: Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 411–425. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85053-3_26

22. Standaert, F.-X., Koeune, F., Schindler, W.: How to compare profiled side-channel attacks? In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 485–498. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01957-9_30
23. Wang, W., Yu, Y., Liu, J., Guo, Z., Standaert, F., Gu, D., Xu, S., Fu, R.: Evaluation and improvement of generic-emulating DPA attacks. In: Güneysu, T., Handschuh, H. (eds.) [10], pp. 416–432. http://dx.doi.org/10.1007/978-3-662-48324-4
24. Whitnall, C., Oswald, E.: Profiling DPA: efficacy and efficiency trade-offs. In: Bertoni, G., Coron, J.-S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 37–54. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40349-1_3
25. Whitnall, C., Oswald, E.: Robust profiling for DPA-style attacks. In: Güneysu, T., Handschuh, H. (eds.) [10], pp. 3–21. http://dx.doi.org/10.1007/978-3-662-48324-4
26. 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, Cham (2014). doi:10.1007/978-3-319-04852-9_10