Ridge-based DPA: Improvement of Differential Power Analysis For Nanoscale Chips

Weijia Wang, Yu Yu, François-Xavier Standaert, Junrong Liu, Zheng Guo and Dawu Gu

***Abstract*—Differential power analysis (DPA), as a very practical type of side-channel attacks, has been widely studied and used for the security analysis of cryptographic implementations. However, as the development of chip industry leads to smaller technologies, the leakage of cryptographic implementations in nanoscale devices tends to be nonlinear (i.e., leakages of intermediate bits are no longer independent) and unpredictable. These phenomena make some existing side-channel attacks not perfectly suitable, i.e., decreasing their performance and making some common used prior power models (e.g., Hamming weight) to be much less respected in practice.**

To solve above issues, we introduce the regularization process from statistical learning to the area of side-channel attack and propose the ridge-based DPA. We also apply the cross-validation technique to search for the most suitable value of the parameter for our new attack methods. Besides, we present theoretical analyses to deeply investigate the properties of ridge-based DPA for nonlinear leakages.

We evaluate the performance of ridge-based DPA in both simulation-based and practical experiments, comparing to the state-to-the-art DPAs. The results confirm the theoretical analysis. Further, our experiments show the robustness of ridge-based DPA to cope with the difference between the leakages of profiling and exploitation power traces. Therefore, by showing a good adaptability to the leakage of the nanoscale chips, the ridge-based DPA is a good alternative to the state-to-the-art ones.

***Index Terms*—Side-channel attack, Differential Power Analysis, Linear regression, Ridge regression, Cross-validation**

I. Introduction

Side-channel attacks (SCAs) exploit the physical information leaked from the implementation of a cryptographic algorithm, and they are usually more powerful than classical cryptanalytic techniques that target at the mathematical weakness of the underlying algorithm. Differential power analysis (DPA), proposed by Kocher et al. [1], is a form of widely used side-channel attack that efficiently recovers the secret key from multiple (typically noisy) power consumption measurements.

Weijia Wang is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. Email: aawwjaa@sjtu.edu.cn.

Yu Yu is with School of the Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He is also with the State Key Laboratory of Cryptology P.O. Box 5159, Beijing 100878, China, and Westone Cryptologic Research Center, Beijing 100070, China. Email: yyuu@sjtu.edu.cn.

François-Xavier Standaert is with the ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium. Email: fstandae@uclouvain.be.

Junrong Liu is with the Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. Email: liujr@sjtu.edu.cn.

Zheng Guo is with the Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. Email: guozheng@sjtu.edu.cn.

Dawu Gu is with the Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. Email: dwgu@sjtu.edu.cn.

Fig. 1 shows the process of DPA. Following the 'divide-and-conquer' strategy, a DPA attack breaks down the secret key into a number of subkeys of small length, and recovers them independently. Using the exploitation method (e.g., Pearson's coefficient [2]), DPA tries to find the association between the power consumption of the target device and the estimated leakage of a specific intermediate variable $z_{x,k}$ (e.g., the output of S-Box). The attackers exhaustively test all possible values of each subkey and find the one whose corresponding estimated leakages are most similar to the real power consumption. The estimated leakage is determined by the value of intermediate variable under the power model $\mathrm{M}()$ (we refer to Section II for the formal definitions), which is a prior knowledge of the target device. Generally based on the way to obtain the power model, DPA can be divided into profiled and non-profiled ones.

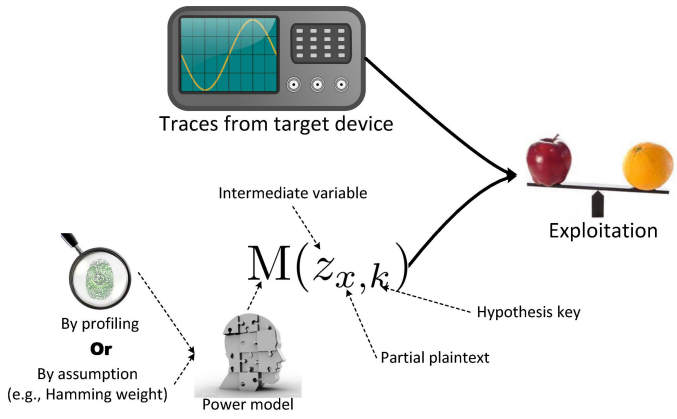

Fig. 1. Differential Power Analysis.

Profiled DPA. Chari et al. [3] proposed the first profiled DPA called template attack, whose profiling phase is based on the multivariate Gaussian template. We refer to the profiling phase of the templates attack as classical profiling (following the terminology in [4]). Later Schindler et al. [5] proposed a very promising profiled DPA that uses linear regression 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 fewer number of measurements, and it allows a tradeoff between the profiling and online exploitation phases: more measurements used in the profiling phase, fewer are needed in the exploitation phase.

Non-profiled DPA. Non-profiled DPA doesn't rely on any profiling device and recovers the secret key only by the target device. These attacks can be achieved by either of following ways:

1) They can make some reasonable device-specific assumptions (such as Hamming weight and bit model) about the leakage function, resulting in the Kocher's original DPA [1], correlation power analysis (CPA) [2], differential cluster analysis (DCA) [6] and so on.
2) Some other non-profiled DPAs are extended from the profiled ones: different models are built 'on-the-fly' for different hypothesis keys and the goodness-of-fit is expected to be minimal for the correct key. A typical example is the non-profiled LR-based DPA [5], [7], [4], [8]. Further, the works of [8] and [9] show that this type of extension may result in a kind of generic or generic-emulating DPA, and the non-profiled DPA we proposed in this paper also belongs to this type.

With the development of chip industry towards smaller technologies, the impact of power variability is becoming more and more significant, which leads to some issues for the existing DPAs [10]. First, it becomes increasingly difficult to produce two chips with the same behavior, leading to the difference of power consumptions between the profiling and exploitation devices. This issue was firstly tackled in [11], but more concerning on the misalignment of the traces of profiling and exploitation. Second, the degree of the leakage function tends to be nonlinear. This issue impacts both profiled and non-profiled DPAs. For profiled DPA, the nonlinear leakage function has more variables thus enhances the profiling's difficulty, which may lead to the overfitting issue in practice. That is, noisy measurements in the profiling phase can lead to a model that describes mostly the noise instead of the actual leakage function. For the non-profiled DPA, common power models may be no longer valid. To tackle this issue, various non-profiled strategies such as mutual information analysis (MIA) [12], Kolmogorov-Smirnov (KS) method [13], Cramér-von Mises test method [13], [5], copulas method [9] and the non-profiled LR-based DPA enables to work in a context where no *a priori* knowledge is assumed about the power model, and the DPAs of this form were termed as "generic DPAs" in [8]. However, the authors of [8] 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 the LR-based DPA (as a generic DPA) to the stepwise linear regression (SLR)-based DPA (as a generic-emulating DPA) and demonstrated its effectiveness for injective target functions in simulation-based experiments. Nevertheless, as shown in [14], despite its theoretical merit, there is still a performance gap between SLR-based DPA and traditional ones in practice.

Our contributions. In this paper, we provide a practical solution to tackle the power variability issue in (both profiled and non-profiled) DPA attacks against nanoscale chips.

First, we propose a new profiling method (named ridge-based profiling) based on the ridge regression (also called as the Tikhonov regularization), which imposes a constraint (or penalty) on the coefficients of linear regression. Ridge regression is a good alternative to linear regression with a better performance on noisy data [15]. We extend the ridge-based profiling to the non-profiled case (in the same manner as the non-profiled LR-based DPA) and get a practical generic-emulating DPA. By building pre-computed tables, the running time of ridge-based profiling can be very efficient. Further, based on the idea of [16], [17] and [18], we can generalize the (both profiled and non-profiled) ridge-based DPA to the high-order setting.

Second, in profiled case, as the constraint (described by a parameter) affects the performance of the ridge-based profiling, we apply the K-fold cross-validation to find out the most suitable constraint (i.e., the optimal parameter). We also experiment on the above parameter optimization in settings with various noise. Our results suggest that the optimal parameter is related to the noise of the measurements (i.e., the optimal parameter is increased with respect to the noise level).

Third, mainly by studying some properties (such as the bias-variance tradeoff) of the ridge regression, we investigate in theory the improvement of ridge-based profiling. Our theoretical study aims to answer the questions:
When is ridge-based DPA most useful, how to best apply it, and why?

Finally, we conduct both simulation-based and practical experiments to evaluate our new methods. In the experiments, our ridge-based profiling outperforms classical and LR-based ones for nonlinear leakage functions, and shows a good potential to be a robust profiling that is suitable to the settings where profiling and attacking devices are not perfectly identical. Meanwhile, the non-profiled ridge-based DPA performs better than the best and averaged Difference-of-Means DPAs [1], and thus make itself a good alternative to traditional DPAs.

This work is based on the previous conference papers [14] and [19]. We highlight below the novel aspects and extensions incorporated into this manuscript:

- We synthesize the works of two papers and provide a systematic solution to the power variability issue of DPA against nanoscale chips.
- We present a generalization to the high-order SCA for the ridge-based DPA.
- In regard to the time complexity, we present an implementation for our new methods based on pre-computed tables. We prove that our implementation is generally equivalent to the ideal one with the increase of trace number.
- We provide an analysis for ridge-based profiling based on the bias-variance tradeoff.
- We provide more comprehensive experiments where:
  1) We use the metric of guessing entropy.
  2) We provide the simulation-based experiments in the setting where the profiling and exploitation devices are different.

[1]Difference-of-means attack is a form of DPA that exploits the leakage of each single bit. It is generally seen as the 'best' attack strategy without *a prior* knowledge about the power model. The averaged and best DoM attack refers to the DoM attack averaging the correlation of each bit and the bit corresponding to the highest (assuming additional knowledge about the best target bit) correlation respectively.

3) The practical experiments are based on the FPGA implementations.
4) We compare the ridge-based profiling with the cluster-based one [11] in the setting of robust profiling.
5) We present the performance of ridge-based DPAs under the FPGA-based multivariate settings.

Outline. The rest of this paper is organized as follows. In Section II, we provide the background to the work. In Section III-A, we present our new methods. The theoretical analyses are described in Section IV. The experimental results in different scenarios are provided in Section V. Finally, we conclude the paper in Section VI.

II. Background

Let X be a vector of some partial plaintexts 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 ith measurement. Let k be a hypothesis subkey, and let $\mathrm{F}_k() : \mathbb{F}_2^m \to \mathbb{F}_2^m$ be a target function (e.g., the Sbox output), where m is the bit length of X_i, and thus $Z_{i,k} = \mathrm{F}_k(X_i)$ is called an intermediate value [2] and $Z_k = \mathrm{F}_k(X) = (Z_{i,k})_{i\in\{1,\ldots,N\}}$ is the vector of intermediate values obtained by applying $\mathrm{F}_k()$ to X component-wise. The target functions considered in this paper are injective.

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 intermediate value is Z_{i,k^*}. We have $T_i^j = \mathrm{L}^j(Z_{i,k^*})$ and $T^j = \mathrm{L}^j(Z_{k^*})$, where k^* is the correct subkey. 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 deterministic part of leakage function $\mathrm{L}^j()$, namely, $T_i^j \approx \mathrm{M}^j(F_{k^*}(X_i))$. [3] The model is obtained by either learning in the profiling phase or some reasonable device-specific assumptions.

Polynomial representation of the leakage function. As stated in [8] and [20], any deterministic part of the leakage function $\mathrm{L}()$ on input $Z_i \in \mathbb{F}_2^m$ can be represented in the algebraic normal form. That is, we have $\mathrm{L}(Z_i) = \alpha_0 + \sum_{u\in\mathbb{F}_2^m} \alpha_u Z_i^u + \varepsilon$, where coefficients $\alpha_u \in \mathbb{R}$, $Z_i = Z_{i,k^*}$, Z^u denotes monomial $\prod_{j=1}^m Z_j^{u_j}$, Z_j (resp., u_j) refers to the j^{th} bit of Z (resp., u), and ε denotes probabilistic noise. The degree of the leakage function is the highest degree of the non-zero terms in polynomial $\mathrm{L}(Z_i)$.

A. Profiling methods

As mentioned above, profiled DPA is made up of two phases: profiling phase and online exploitation phase. We recall these two phases in this and next sub-sections respectively. The aim of the profiling phase is to learn the deterministic parts of leakage function and the noise ε for all the points to get the power model (i.e., $\mathrm{M}()$). Meanwhile, the aim of the online exploitation phase is to recover the key using the power model. Our presentation is largely based on the (excellent) introduction provided in [4].

Classical profiling. We call the profiling phase of template attacks [3] as classical profiling, which views the leakages 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 value being $z = Z_{i,k}$. The adversary 'learns' the physical leakages by computing the $p \times 1$ sample mean $\hat{\mu}_z$ and the $p \times p$ sample covariance $\hat{\Sigma}_z$ for all the intermediate values z on the profiling device. Finally, the intermediate value-conditioned leakages subject to the Gaussian distribution $\mathrm{N}(\hat{\mu}_z, \hat{\Sigma}_z)$ for z. As suggested in [21], we assume the noise distribution of different intermediate targets to be equal and use the same covariance estimates (across all intermediate targets).

Linear regression-based profiling. LR-based profiling [5] uses the stochastic model that is represented by polynomial like the leakage function: $\mathrm{M}(Z_i) = \alpha_0 + \sum_{u\in\mathbb{U}_d} \alpha_u Z_i^u + \varepsilon$, where set $\mathbb{U}_d = \{u | u \in \mathbb{F}_2^m, \mathrm{HW}(u) \leq d\}$ (where $\mathrm{HW} : \mathbb{F}_2^m \to \mathbb{Z}$ 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., $\hat{\boldsymbol{\alpha}}_d = (\boldsymbol{U}_d^{\mathrm{T}}\boldsymbol{U}_d)^{-1}\boldsymbol{U}_d^{\mathrm{T}}T$ [4], 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 $\hat{\boldsymbol{\alpha}}$ of the profiling device. Then, the $p \times p$ sample covariance $\hat{\Sigma}$ is computed assuming the noise distributions are identical for various intermediate values. Finally, the intermediate value-conditioned leakages subject to the Gaussian distribution $\mathrm{N}(\hat{\alpha}_0 + \sum_{u\in\mathbb{U}_d} \hat{\alpha}_u Z_i^u, \hat{\Sigma})$ for the intermediate value Z.

B. (online) Exploitation methods

Bayesian key recovery. A p-dimensional multivariate Gaussian distribution $\mathrm{N}(\mu, \Sigma)$ has the following density function:

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

It should be noted that, to achieve a more robust profiled attack and avoid the numerical problems (see, e.g., [21]), we suggest removing the estimated term $\frac{1}{(2\pi)^{p/2}\|\Sigma\|^{1/2}}$. Hence, we have the robust density function:

$$\mathrm{f}(x) = \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right) . \tag{2}$$

Therefore, we can mount the Bayesian key recovery by calculating the log likelihoods:

$$k_{guess} = \underset{k}{\mathrm{argmax}} \sum_{i=1}^{n} \log(\mathrm{f}_{i,k}(T_i)) , \tag{3}$$

[2] Normally, we name the output of a target function as the intermediate variable (as a random variable), whose value is the intermediate value.

[3] We often omit the superscript 'j' in $\mathrm{L}^j()$ and $\mathrm{M}^j()$ for succinctness.

[4] We omit the subscript 'd' in $\hat{\boldsymbol{\alpha}}_d$ for succinctness in the rest of the paper.

where $f_{i,k}()$ is the density function associated with the intermediate value $Z_{i,k}$.

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

$$k_{guess} = \underset{k}{\operatorname{argmax}} \, \rho(\mathrm{M}(Z_{i,k}), T_i) \ , \tag{4}$$

where ρ is the Pearson's coefficient.

C. Generic DPA and its limitations

The generic DPA is defined in [8]. In regard to the nature of the power model, authors in [8] apply the widely-accepted concept of "levels of measurement" to define the level of models: direct, proportional, ordinal and nominal models. The direct model is obtained by direct approximation of the actual power consumption, which is built by the methods such as the classical profiling and LR-based profiling. Proportional model is less demanding and a good approximation for the leakage function up to proportionality. Examples include the Hamming weight model for the correlation power analysis [2]. Ordinal model approximates the leakage function up to ordinality, which is less demanding again. At last, nominal model, or called as the generic (power) model, has the least demanding, and it approximates the leakage function up to nominality only.

We recall 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-hypothesized target function $F_k()$.

Definition 2 (Generic compatibility): A distinguisher is generic-compatible if it is built from a statistic which operates on 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 [8], no efficient generic DPA strategy is able to distinguish the correct subkey k^* from an incorrect hypothetical value k given that $F_{k^*}()$ and $F_k()$ are both injective. We refer to [8] for the details and proofs.

D. From (non-profiled) LR-based DPA to generic-emulating DPA

For each subkey hypothesis k, we use a full basis of polynomial terms to construct the power model: $\mathrm{M}_k(Z_{i,k}) = \alpha_0 + \sum_{u \in \mathbb{U}} \alpha_u Z_{i,k}^u + \varepsilon$, where $\mathbb{U} = \mathbb{F}_2^m \setminus \{0\}$. The goodness-of-fit (denoted as R), as a measurement of similarity between $\mathrm{M}_k(Z_{i,k})$ and the real power consumption T, can be computed for each $\mathrm{M}_k()$ which separates the correct key hypothesis from incorrect ones using the linear regression "on-the-fly". Normally, we use Pearson's coefficient to measure the goodness-of-fit, i.e., $R^2 = \rho(T, \mathrm{M}_k(Z_k))$. This method, called non-profiled LR-based DPA (with a full basis), falls into a special form of generic DPA, since it doesn't depend on any device-specific assumptions. Thus, it doesn't distinguish correct subkey from incorrect ones on injective target functions (see [8]).

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* constraint on $\hat{\boldsymbol{\alpha}}$. [8] presents the first generic-emulating DPA: stepwise linear regression (SLR)-based DPA, it excludes some 'insignificant' terms while keeping all the 'significant' ones in the basis. SLR-based profiling follows the method named backward elimination: Starting with all terms, the method tests the exclusion of each term based on the goodness-of-fit, and excludes the term (if any, we called it as the insignificant term) whose loss gives the most statistically insignificant deterioration of the goodness-of-fit. Then this process is repeated until no further term can be excluded without a statistically significant loss of goodness-of-fit. See [15] for more details.

However, as shown in [14], SLR-based DPA suffers from two drawbacks: (1) it is not stable for the small number of traces; (2) in comparison with traditional DPAs, SLR-based DPA has poor performance especially on real implementations. Thus it is hardly practically usable in the attacking of the real cryptographic devices.

III. Ridge-based profiling and its application to the profiled and Non-profiled DPAs

A. Construct

In this sub-section, we only consider the deterministic part of the model, and meanwhile, the sample variance $\hat{\Sigma}$ can be obtained in the same way as LR-based profiling.

Our new profiling method (for each power consumption point) also represents the power model by polynomial, and it can be seen as a generalization of LR-based profiling by explicitly imposing a constraint on the coefficients' size, formally,

$$\begin{aligned} \hat{\boldsymbol{\alpha}} \overset{\mathsf{def}}{=} \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{i=1}^{N} \Big(T_i - \mathrm{M}_d(Z_i) \Big)^2 , \\ \text{subject to } \sum_{u \in \mathbb{U}_d} \hat{\alpha}_u^2 \leq s \ . \end{aligned} \tag{5}$$

An equivalent formulation to above is (see [15] for detailed deduction):

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left(\sum_{i=1}^{N} \left(T_i - \mathrm{M}_d(Z_i) \right)^2 + \lambda \sum_{u \in \mathbb{U}_d} \alpha_u^2 \right) \ , \tag{6}$$

whose optimal solution is given by:

$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{U}_d^{\mathrm{T}} \boldsymbol{U}_d + \lambda \boldsymbol{I}_d)^{-1} \boldsymbol{U}_d^{\mathrm{T}} T \ , \tag{7}$$

where $\mathbb{U}_d$, $\boldsymbol{U}_d$ and Z_i are defined in Section II-A, matrix $\boldsymbol{I}_d$ is the $|\mathbb{U}_d| \times |\mathbb{U}_d|$ identity matrix, and $|\mathbb{U}_d|$ denotes the cardinality of $\mathbb{U}_d$. There is a one-to-one correspondence between the parameters s and λ. Generally, a larger s corresponds to a smaller λ, and vice versa.

The extending to non-profiled case. Ridge-based profiling can be extended to the non-profiled case in the same way as the construction of non-profiled LR-based DPA (that is,

building the model under all hypothesis keys and finding the one corresponding to the highest goodness-of-fit), resulting in a new and practical generic-emulating DPA.

B. Implementation details

Due to the limited value space of the partial inputs, the rows from the matrix $\boldsymbol{U}_d$ are probably repeated, thus the running time of Equation (7) would become unnecessary long with the increase of the trace number. In the case that the partial inputs X are randomly distributed, we do the following trace pre-processing to facilitate the attack (which is also conducted in [8]): we average the traces based on their m-bits values of partial inputs, and use the resulting 2^m mean power traces to mount the attack. To this point, the input of the profiling is always 2^m mean power traces, thereby we can simply pre-compute the matrices $\boldsymbol{A}_{d,k,\lambda'} = (\boldsymbol{V}_{d,k}^{\mathrm{T}}\boldsymbol{V}_{d,k} + \lambda'\boldsymbol{I}_d)^{-1}\boldsymbol{V}_{d,k}^{\mathrm{T}}$ for all the possible partial keys k, where $\boldsymbol{V}_{d,k} = (\acute{Z}_{i,k}^{u})_{i\in\{0,1,2,\ldots,2^m-1\},u\in\mathbb{U}_{d,k}}$ and $\acute{Z}_{i,k} = \mathrm{F}_k(i)$. In such a way Equation (7) could be rewritten as:

$$\hat{\boldsymbol{\alpha}}' = \boldsymbol{A}_{d,k,\lambda'}T', \tag{8}$$

where T' is the vector of 2^m mean power consumptions that are corresponding to the partial input from 0 to 2^m-1 respectively.

In the following, we prove that $\hat{\boldsymbol{\alpha}}'$ is approximately equal to $\hat{\boldsymbol{\alpha}}$ with the increase of the number of traces. Since the order of the traces doesn't affect the value of $\hat{\boldsymbol{\alpha}}$, we consider the case that n partial inputs X are in the increasing order from 0 to 2^m-1. As the inputs are randomly distributed, we have the following deductions:

$$\begin{aligned}
&\lim_{n\to\infty}(\boldsymbol{U}_d^{\mathrm{T}}\boldsymbol{U}_d + \lambda\boldsymbol{I}_d)^{-1} \\
&= \lim_{n\to\infty}(\frac{n}{2^m}\boldsymbol{V}_{d,k}^{\mathrm{T}}\boldsymbol{V}_{d,k} + \lambda\boldsymbol{I}_d)^{-1} \\
&= \lim_{n\to\infty}\frac{2^m}{n}(\boldsymbol{V}_{d,k}^{\mathrm{T}}\boldsymbol{V}_{d,k} + \frac{2^m}{n}\lambda\boldsymbol{I}_d)^{-1} \ ,
\end{aligned} \tag{9}$$

and

$$\lim_{n\to\infty}\boldsymbol{U}_d^{\mathrm{T}}T = \lim_{n\to\infty}\frac{n}{2^m}\boldsymbol{V}_{d,k}^{\mathrm{T}}T' \ . \tag{10}$$

By applying Equations (9) and (10) to Equation (7), we have:

$$\begin{aligned}
&\lim_{n\to\infty}\hat{\boldsymbol{\alpha}} \\
&= \lim_{n\to\infty}(\boldsymbol{U}p_d^{\mathrm{T}}\boldsymbol{U}_d + \lambda\boldsymbol{I}_d)^{-1}\boldsymbol{U}_d^{\mathrm{T}}T \\
&= \lim_{n\to\infty}\frac{2^m}{n}(\boldsymbol{V}_{d,k}^{\mathrm{T}}\boldsymbol{V}_{d,k} + \frac{2^m}{n}\lambda\boldsymbol{I}_d)^{-1}\frac{n}{2^m}\boldsymbol{U}_d^{\mathrm{T}}T' \\
&= \lim_{n\to\infty}\hat{\boldsymbol{\alpha}}' \ ,
\end{aligned} \tag{11}$$

where $\frac{2^m}{n}\lambda = \lambda'$.

Therefore, considering the univariate setting and combining with the correlation key recovery introduced in Section II-B, we exemplify in algorithm 1 and 2 the application of ridge-based profiling to (both profiled and non-profiled) DPA attacks. It should be noted that when facing the multivariate setting, we can either (1) transform the points into one by using pre-processing methods such as principal component analysis (PCA); or (2) apply other exploitation methods that are suitable to the multiple points, such as Bayesian key recovery.

Algorithm 1 Profiled ridge-based DPA

Require: profiling traces $t_i = \{T_i, x_i\}$ where $i \in \{1, ..., n_{prof}\}$; exploitation traces $\bar{t}_i = \{\bar{T}_i, x_i\}$ where $i \in \{1, ..., n_{expl}\}$; the pre-computed tables $\boldsymbol{A}_{d,k^*,\lambda}$, $\boldsymbol{V}_{d,k}$ for $k \in \{0, ..., 2^m-1\}$; the true key k^*;
Ensure: the key guessing k_{guess}
1: **Profiling phase:**
2: Let T'_{prof} be the vector of 2^m mean power points of T_i for $i \in \{1, ..., n_{prof}\}$
3: $\hat{\boldsymbol{\alpha}} = \boldsymbol{A}_{d,k^*,\lambda}T'_{prof}$
4: **Exploitation phase:**
5: Let T'_{expl} be the 2^m mean power points of $\bar{T}_i$ for $i \in \{1, ..., n_{expl}\}$
6: **for** $k = 0$; $k < 2^m$; $k{+}{+}$ **do**
7: $R_k = \rho(\boldsymbol{V}_{d,k} \times \hat{\boldsymbol{\alpha}}, T'_{expl})$
8: **end for**
9: return $k_{guess} = \mathrm{argmax}_k\, R_k$

Algorithm 2 Non-profiled ridge-based DPA

Require: traces $t_i = \{T_i, x_i\}$ where $i \in \{1, ..., n\}$; the pre-computed tables $\boldsymbol{A}_{d,k,\lambda}$, $\boldsymbol{V}_{d,k}$ for $k \in \{0, ..., 2^m-1\}$;
Ensure: the key guessing k_{guess}
1: Let T' be the vector of 2^m mean power points of T_i for $i \in \{1, ..., n\}$
2: **for** $k = 0$; $k < 2^m$; $k{+}{+}$ **do**
3: $\hat{\boldsymbol{\alpha}} = \boldsymbol{A}_{d,k,\lambda}T'$
4: $R_k = \rho(\boldsymbol{V}_{d,k} \times \hat{\boldsymbol{\alpha}}, T')$
5: **end for**
6: return $k_{guess} = \mathrm{argmax}_k\, R_k$

C. Generalization to high-order DPA

In this sub-section, we generalize the (both profiled and non-profiled) ridge-based DPA to the context of high-order side-channel attack that targets the leakages of several intermediate variables. It is mounted against the implementation with masking technique (see e.g., [22], [23], [24], [25] for an incomplete list of related studies), in which the sensitive intermediate variable z is randomly encoded into o variables $(z^1, \ldots, z^o)$, such that $\mathrm{Enc}(z) = (z^1, \ldots, z^o)$, where $\mathrm{Enc}() : \mathbb{F}_2^m \to (\mathbb{F}_2^m)^o$ is the encoding function, and every $(o-1)$-tuple of $\{z^1, \ldots, z^o\}$ is independent of z. In [16], [17] and [18], the second-order LR-based DPA has been studied and the authors showed that it is a good alternative to some solutions such as the second-order CPA with a combination function. In the following, we introduce the high-order ridge-based profiling based on the same strategy.

Suppose that the power consumption points of o intermediates for i-th trace are $(T_i^1, \ldots, T_i^o)$, we combine these o power consumption points (for each trace) using the centered product

combination function [5]: $T_i = \prod_{j\in(1,...,o)}(T_i^j - \mathrm{E}(T^j))$. Thus we can build the models and mount the attack with the ridge-based profiling using the vectors of combined power consumptions and the sensitive variable Z, resulting in high-order ridge-based DPA. It should be noted that we can still apply the pre-processing of Section III-B to this high-order case.

D. Searching for the optimal parameters

As illustrated above, there is an undetermined parameter (i.e., λ), the choice of which affects the performance of the profiling. For profiled ridge-based DPA, we propose a method to choose the optimal parameter based on the K-fold [6] cross-validation technique from statistical learning. We mention that the cross-validation was already used in the field of side-channel attack (for different purposes), such as evaluation of side-channel security [29]. Algorithm 3 finds the optimal parameter using cross-validation, where we omit the subscript d (the degree) for succinctness.

The algorithm is sketched 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 $\hat{\boldsymbol{\alpha}}$ using the remaining $K-1$ parts, and calculate the goodness-of-fit $R_{\lambda,i}$ using the traces in $\mathcal{C}_i$. We then get the average goodness-of-fit $R_\lambda = (\sum_{i=1}^{K} R_{\lambda,i})/K$ for each candidate parameter λ in consideration. Finally, we return the parameter with the highest averaged goodness-of-fit.

Algorithm 3 Searching for the optimal 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 Λ;
Ensure: $\hat{\lambda}$ as the optimal parameter for the subkey;
1: **for** $i = 1$; $i \leq K$; $i{+}{+}$ **do**
2: $\quad\mathcal{C}_i = \{t_{K*(i-1)+1}, ..., t_{K*i}\}$
3: **end for**
4: **for all** λ such that $\lambda \in \Lambda$ **do**
5: $\quad$**for** $i = 1$; $\leq K$; $i{+}{+}$ **do**
6: $\quad\quad$Compute the $\hat{\boldsymbol{\alpha}}$ using the traces in $\{\mathcal{C}_j\}_{j\in\{1...K\}\setminus\{i\}}$
7: $\quad\quad$Calculate the goodness-of-fit $R_{\lambda,i}$ from $\mathcal{C}_i$
8: $\quad$**end for**
9: $\quad R_\lambda = (\sum_{i=1}^{K} R_{\lambda,i})/K$
10: **end for**
11: $\hat{\lambda} = \mathrm{argmax}_\lambda\, R_\lambda$

For the non-profiled case, theoretically, λ tunes the tradeoff between the generality and practicability: the smaller λ is, the more generic the method will be, but it may result in worse performance at the same time. Fortunately, as shown in Section V-A1, this parameter is not very sensitive to the attack settings, and the same value of the parameter can be used in various experimental settings.

[5]The combination function is defined as $\mathrm{C}() : \mathbb{R}^o \to \mathbb{R}$ and it combine the leakages of o intermediate variables into a real number. Centered product combination function, first proposed in [22], shows the best performance in DPA [26], [27], [28] and is proved to be optimal for higher-order attacks in very noisy scenarios.

[6]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.

IV. Theoretical analysis

In this section, we investigate the improvement of the ridge-based profiling (over LR-based and SLR-based ones) theoretically. We first answer the 'why' and 'how' questions by analyzing the bias and variance of the model. Then we answer the 'when' question by studying the way that the coefficients shrink corresponding to the penalty factor in the ridge-based profiling.

A. How to apply ridge-based DPA profiling and why it is more effective?

For simplicity, we consider the univariate leakage, where the leakage of the i-th trace is $T_i = \mathrm{L}(Z_{i,k^*})$. Since the coefficients learned from the LR-based (resp., ridge-based, SLR-based) profiling determine the model (by definition), varying the coefficients will affect the performance.

1) Analysis for profiled DPA based on the bias-variance tradeoff: The bias-variance tradeoff comes from the field of statistical learning to measure two sources of the generalization error, where the bias reflects the difference between the leakage function and the model built by profiling methods, and the variance reflects the variability of a model prediction for the profiling traces. The goal of the profiling is to reduce both bias and variance. However, as shown in the theory of statistical learning (see, e.g., [15, Section 2.9]), it is impossible to achieve the minimum for both of them simultaneously. For example, high-variance profiling methods (e.g., LR-based profiling) may be able to represent an accurate model, but are at the risk of overfitting to noisy traces. We mention that bias-variance tradeoff has been introduced into the field of profiled attack [30], and we apply the similar analysis strategy to the profiled ridge-based DPA.

The variance-covariance matrix and biases of the coefficients built by the ridge-based profiling is given by [31, Equation 4.5 and 4.8]:

$$\mathrm{Bias}(\hat{\boldsymbol{\alpha}}) = \lambda \boldsymbol{W}\boldsymbol{\alpha}, \tag{12}$$

$$\mathrm{Var}(\hat{\boldsymbol{\alpha}}) = \boldsymbol{W}\boldsymbol{U}_d^{\mathrm{T}}\boldsymbol{U}_d\boldsymbol{W}\sigma^2, \tag{13}$$

where $\boldsymbol{\alpha}$ are the real coefficients, $\boldsymbol{W} = (\boldsymbol{U}_d^{\mathrm{T}}\boldsymbol{U}_d + \lambda\boldsymbol{I}_d)^{-1}$, and σ^2 is the variance of noise ε, which can be seen as a constant. The formulae will degenerate into the ones corresponding to LR-based profiling when we set the parameter λ to 0. We can see from the formulae that, both biases and variances are related to the value of the parameter λ, and the biases are also related to the real coefficients of the leakage function. Moreover, the variances are increased with the noise, whereas the biases are independent of it.

Without loss of generality, we all the values of the real coefficients as 1's, then compute the averaged bias and variance of model's output with different λ's and degrees of the leakage function. We show in Fig. 2 the bias-variance tradeoff for different degrees of the model. We can see that, as expected, the (averaged) bias is reduced and variance is increased in

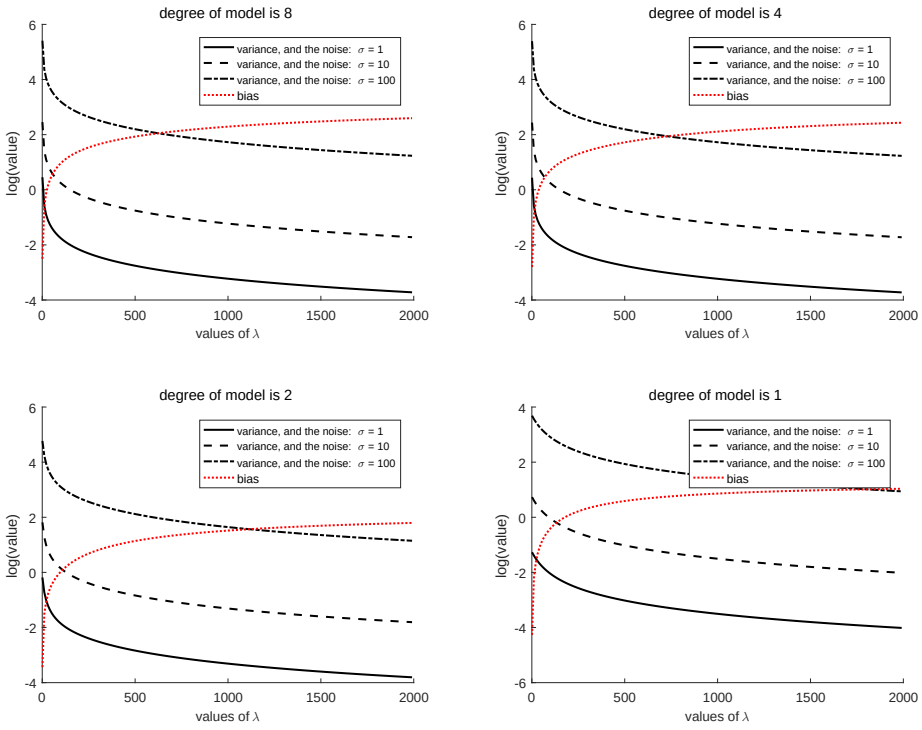

Fig. 2. The bias-variance tradeoff of ridge-based profiling for different degrees (of the model). The upper-left, upper-right, lower-left, and lower-right figures correspond to the cases for $d = 8$, $d = 4$, $d = 2$, and $d = 1$ respectively.

relation to the value of λ, and the intersection of bias and variance curves implies the optimal choice of the λ. The results indicate that, in profiled ridge-based DPA, the best choice for λ increases with the increase of noise. Further, the values of bias and variance at $\lambda = 0$ indicate the better performance of ridge-based profiling than the LR-based one. However, it should be noted that, our analysis only considers the error rate of the model built by the profiling, which does not always relate to the separation between correct and incorrect hypothesis keys. Meanwhile, as the bias depends on the real coefficients, the optimal choice of the λ (related to the intersection of bias and variance curves) indicated in this sub-section is not numerically applicable to other different attack settings. To this end, we propose to use the cross-validation method to search for the most suitable parameter (see Section III-D).

2) Comparing ridge-based profiling with the SLR-based one: By definition, the SLR-based profiling keeps only a subset of the terms of the basis, which is essentially a discrete process and may not fully characterize the real leakage function whose insignificant coefficients are smaller but non-zero. As a result, some 'insignificant' terms that still have some (although not much) contributions to the power model may be discarded, and it leads to instability (compared with the ridge-based profiling) of the results especially when the number of traces used in the attacks is small. (see the discussions in [15], [30]).

Then we compare the variances of the outputs of ridge-based and SLR-based profiling, which affect their stability. In the case of a fixed leakage function of degree 8 with the signal-noise-ratio (SNR)$= 0.1$ [7], we use both SLR-based and ridge-based profiling 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. 3. The variance of outcomes is increased with the noise level [8] (i.e., the decrease of the number of traces), and for the same number of traces the ridge-based profiling has a much lower variance of its outcomes than the SLR-based one, and thus has a more stable performance. Please refer to Appendix A for the comparison between ridge-based and SLR-based profilings by bias-variance tradeoff.

B. How do the coefficients shrink in the ridge-based profiling?

As described in Section III-A, the ridge-based profiling enforces a general constraint $\sum_{u\in\mathbb{U}} \hat{\alpha}_u^2 < s$ on the coefficients of the power model, but it is not clear how each individual coefficient $\hat{\alpha}_u$ 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 the power model (i.e., the Hamming Weight of u) and the amount of shrinkage of its coefficient $\hat{\alpha}_u$.

First, we use a technical tool from principal component analysis (see, e.g., [15]). Informally, the principal components of $\boldsymbol{U}_k$ are a set of linearly independent vectors obtained by

[7] SNR is a measure to compare the level of a desired signal with the level of background noise, which we follow the definition: $SNR = \frac{\sigma_{\text{signal}}}{\sigma_{\text{noise}}}$.

[8] 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.

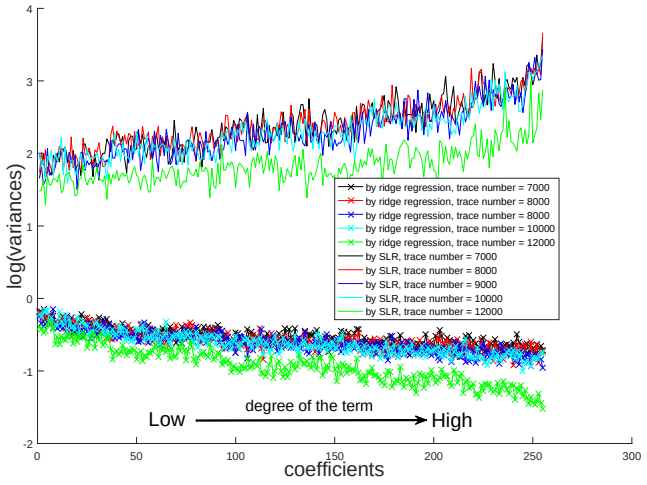

Fig. 3. Variances estimated of the estimated coefficients, for the ridge-based and SLR-based profilings, using different numbers of traces.

applying an orthogonal transformation to $\boldsymbol{U}_k$, i.e., $\boldsymbol{P} = \boldsymbol{D}\boldsymbol{U}_k$, where the columns of matrix $\boldsymbol{P}$ are called the principal components, and columns of P, denoted by P_1, ..., P_{2^m-1}, have descending variances. An interesting property is that P_1 (which has the greatest variance) has the maximal correlation to the shrinkage amounts of coefficients vector $\hat{\boldsymbol{\alpha}}$. We refer to [15] for further discussions and proofs.

Then we can investigate the shrinkage amounts of coefficients vector by studying the first principal components of $\boldsymbol{U}_k$. As shown in Fig. 4, the shrinkage amounts show a high similarity to the degrees of the terms in $\boldsymbol{U}_k$. In such a way, we establish the connection that $\hat{\alpha}_u$ is conversely proportional to the Hamming weight of u. In other words, the more Hamming weight that u has, the less $\hat{\alpha}_u$ contributes to the power model. Therefore, ridge-based profiling is consistent with low-degree power models (e.g., the Hamming weight and bit models) in practice.

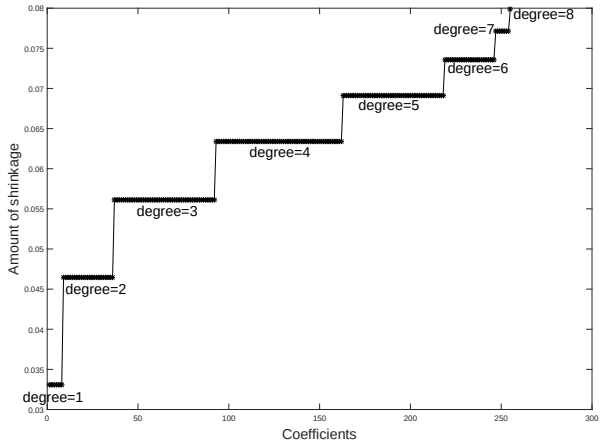

Fig. 4. The shrinkage amounts of coefficients vector, which are characterized by the elements in P_1.

V. Experimental Results

In this section, we evaluate our new profiled and non-profiled DPAs presented in Algorithm 1 and 2 and compare them with the state-of-the-art attacks. We target at the AES-128's first S-box of the first round with an 8-bit subkey (recall that the AES-128's first round key is the same as its encryption key). We use the guessing entropy as the metric for the evaluation by running the experiments (with different inputs) 500 times to compute the averaged ranking of the real key.

A. Simulation-based experiments

In this sub-section, our evaluation is based on univariate leakage with different degrees (of leakage function) and randomized coefficients sampled from -1 to 1 in the setting of simulated traces.

1) Searching for the optimal parameter.: At the beginning of profiled ridge-based DPA, the adversary should first find the optimal parameter (i.e., the λ). For this purpose, we evaluate the parameter optimization algorithm in Section III-D. We consider the settings whose degrees (of both leakage function and model) and trace number are 4 and 2000 respectively. Under different SNRs (0.1, 0.5 and 1), we 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 [9] the averaged goodness-of-fits (of each experiment) and plot the mean of them in Fig. 5. This confirms the intuition that (in profiled case) the optimal parameter (which corresponds to each setting's minimum averaged goodness-of-fit) increases with SNR. It should be noted that the optimal parameter considered in this sub-section is based on the 2^m mean traces, thus the optimal parameter is also highly related to the trace number.

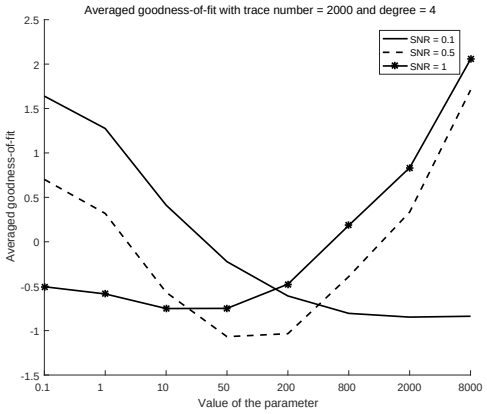

Fig. 5. Averaged goodness-of-fits and their mean values, with SNR = 0.1, 0.5 and 1.

For the non-profiled case, we show that the optimal choice of λ is not very sensitive to noise. Fig. 6 shows the guessing entropies for different noises by varying parameter λ. We can see that, the guessing entropies for all noise settings in general decrease to zero at (almost) the same value for λ (roughly $\lambda = 800$). This indicates that the same value of the parameter

[9] We apply the residual sum of squares for normalization, i.e., $\mathrm{norm}(R^2_\lambda) = (R^2_\lambda - \mathrm{mean}(R^2)/(\max(R^2) - \min(R^2)))$, where $\mathrm{mean}(R^2)$ is the average of $\{R^2_\lambda\}_{\lambda\in\Lambda}$ and $\mathrm{norm}()$ is the normalization function.

can be used in various experimental settings, and thus the attacker does not need to optimize this parameter during the non-profiled attacks.

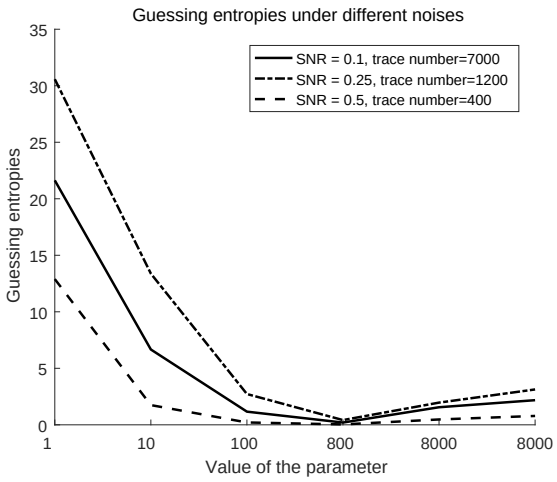

Fig. 6. The guessing entropies of non-profiled ridge-based DPA for different SNRs by varying parameter λ.

2) Profiled case: We combine the correlation key recovery with the model built from different profilings and mount the attacks to compute the guessing entropies. Fig. 7 shows the results. In the profiling phase, we choose the degree of the model to be same as the one of leakage function, and set $\lambda = 400$. The ridge-based profiling performs better than the other two ones in all settings unless when $d = 1$. 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 Section IV.

As introduced in [10], the variability issue may lead to the difference of leakage between profiling and exploitation devices. We show that our new method can be used as a type of robust profiling [11], which can tolerate (some) differences between profiling and exploitation traces in a more realistic setting. To simulate the difference, we randomly choose the coefficients (in the leakage function) for profiling from the range of -1 to 1, and randomize the ones (i.e., coefficients) for exploitation by adding values sampled from the normal distribution $\mathrm{N}(0, 0.3)$. We also implement the clustered-based profiled DPA [11], and compare with our methods. We use the K-means clustering method (that performs better in the settings of our experiments than hierarchical clustering) and select the number of clusters (equals to 3) to be the one producing the highest value of guessing entropy. As shown in Fig. 8, the performance of ridge-based profiling is the best except when the order of leakage function is 1. Therefore, we conclude that the ridge-based profiling is more robust and more suitable in the nanoscale scenario than the LR-based and classical ones. We also find that, in our experiment settings, ridge-based profiling significantly outperforms the clustered-based one, which we attribute to the power models built by different profiling methods. As discussed in [11] and [8], the clustered-based profiling only outputs a nominal power model — a labeling of distinct leakage classes. Meanwhile, the ridge-based (together with LR-based and classical) profiling outputs a direct power model, which characters the overall distribution of leakage, and thus provides more information to the exploitation phase. We mention that the guessing entropies for the clustered-based profiling generally decrease to zero only with much more exploitation and profiling traces, and please refer to Appendix B for results of this (more exploitation and profiling traces) setting.

To give a clear view on the performance of the ridge-based profiling, we present the guessing entropies of the (profiled) ridge-based DPA with different values of the parameter λ. That is, we fix the numbers of profiling and exploitation traces, and vary the values of λ. We present the results in Fig. 9. We can see that, on the whole, the performance of ridge-based profiling relies on the value of λ. More important, the logarithmic coordinate for the parameter indicates that the attacker can use a coarse-grained set of candidate parameters in the search of the optimal parameter.

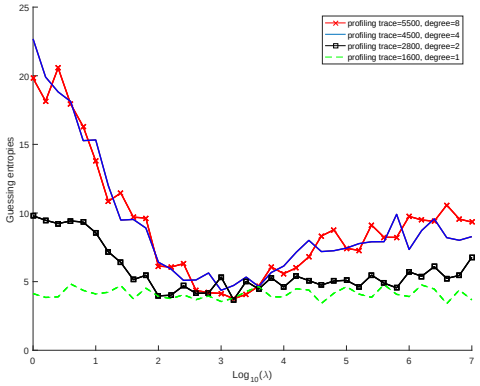

Fig. 9. The guessing entropies profiled DPAs with increasing value of λ, exploitation trace=1700, SNR = 0.1

Another typical scenario we are interested in is that the adversary has no knowledge about the actual degree of the leakage function. In this case, he may use a conservative estimate of the degree of the leakage function in the profiling phase without losing efficiency (i.e., the speed of convergence). To reflect this case, we also experiment where the estimated degree of the model is higher than its actual value. That is, we simulate the traces with leakage functions of degrees 1 and 2 and then conduct the experiments assuming a model of degree 4 for profiling. As shown in Fig. 10, the performance of ridge-based profiling is again significantly better. Therefore, our results show that an attacker (or an evaluation laboratory) can simply use a conservatively estimated degree in ridge-based profiling, instead of running an enumeration of its possible values.

3) Non-profiled case: In the non-profiled case, we compare the non-profiled ridge-based DPA with the best and averaged DoM attacks, which are considered as the best traditional DPA attacks without any knowledge about the leakage function. Additionally, to give a better comparison, we present the guessing entropies for the LR-based DPA with first order basis, which assumes the independent leakages of intermediate bits and is neither generic nor generic-emulating one. We

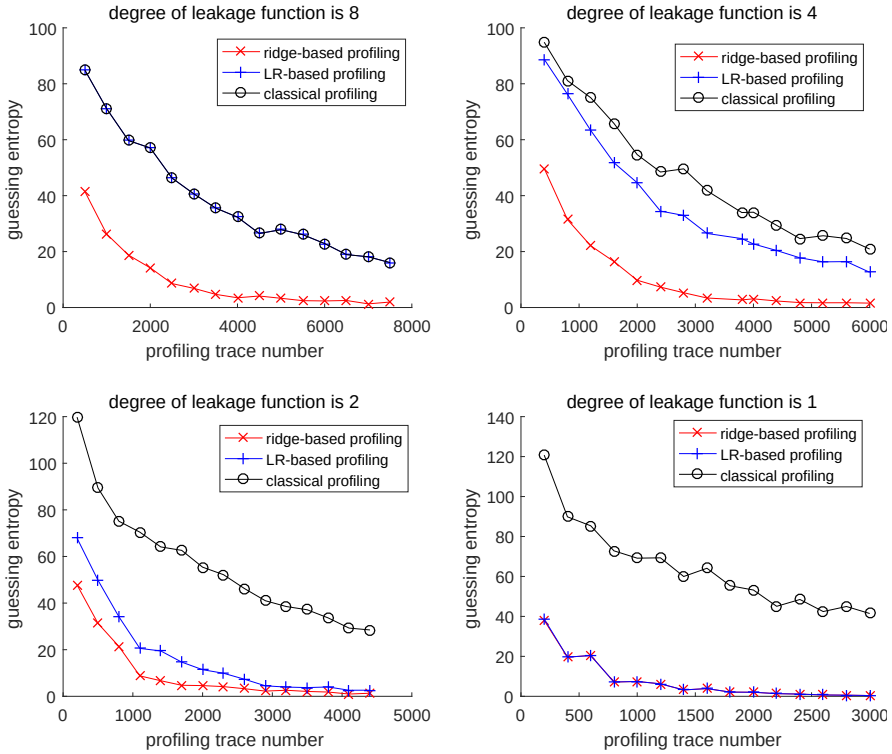

Fig. 7. The guessing entropies of profiled DPAs in the ideal setting, and SNR = 0.1. We increase the number of profiling traces and fix the exploitation trace number as 2200. The upper-left, upper-right, lower-left, and lower-right figures correspond to degrees 8, 4 and 2 and 1 respectively.

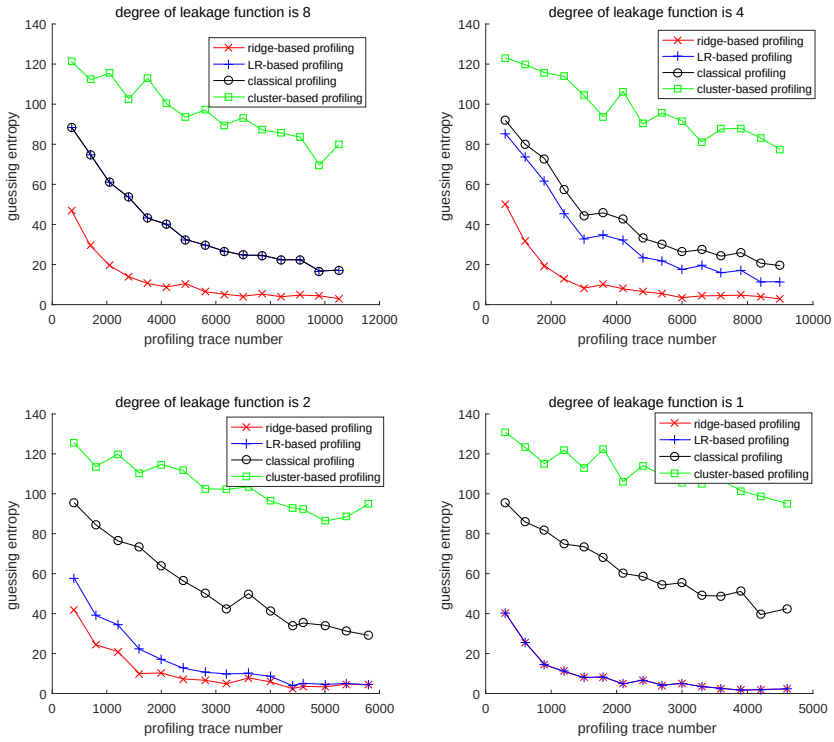

Fig. 8. The guessing entropies of profiled DPAs in a robust setting, and SNR = 0.1. We increase the number of profiling traces and fix the exploitation trace number as 2200. The upper-left, upper-right, lower-left, and lower-right figures correspond to degrees 8, 4 and 2 and 1 respectively.

conduct the experiments in the settings where the degrees of the leakage function are 8, 4, 2. In the experiments, we simply choose the parameter λ that produces the lowest value of guessing entropy, i.e., $\lambda = 800$, which was decided through an exhaustive search over the space (up to some accuracy). We also suggest using this value of λ in other non-profiled attack

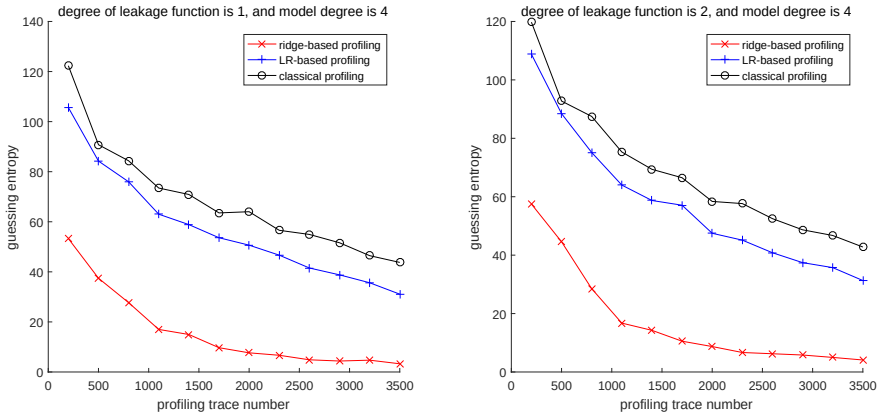

Fig. 10. The guessing entropies profiled DPAs with conservative degrees of model for different numbers of profiling traces, and the exploitation trace number is 1700, SNR = 0.1.

settings. In this paper, for the generality of the attacks, we use the full basis of the model (i.e., the model degree is 8) for all of our non-profiled experiments.

As shown in Fig. 11, The performance of LR-based DPA (with first order basis) is very similar to that of the averaged DoM DPA. More important, the guessing entropy of ridge-based DPA is generally better than the best and averaged DoM attack in all settings. With the increase of the leakage function's degree, the performance of best and averaged DoM attacks become worse, and meanwhile, the one of ridge-based DPA does not change much. We attribute this to the intuition that ridge-based DPA is better suited for power models of high degrees than traditional DPAs.

In order to fully exemplify the power of ridge-based DPA, we also perform the attacks against some artificial leakage functions, in which all low degree terms are discarded. More specifically, we consider the leakage function $\mathrm{L}(z) = \sum_{u \in \mathcal{U}_p} \alpha_u z^u, \forall z \in \mathbb{F}_2^m$, where $\mathcal{U}_p$ is a subset of $\mathbb{F}_2^m$ but excludes those whose Hamming weights are smaller than or equal to p. We simulate the traces for $p = 4, m = 8$ and show the guessing entropies in the lower-right part of Fig. 11. We can see that, in this case, the best and averaged DoM attack behaves poorly, and meanwhile, the ridge-based DPA is not affected. Admittedly, this leakage case may be unrealistic, but it serves as a good example that ridge-based DPA can deal with a wider range of leakage functions.

B. FPGA-based experiments

We carried out experiments on the SAKURA-X which is running the AES on a Xilinx FPGA device Kintex-7 (XC7K70T/160T/325T). We amplified the signal using a (customized) LANGER PA 303N amplifier, providing 30 dB of gain. Then we measured the (absolute value of) power consumptions of the first round S-box output, using a LeCroy WaveRunner 610Zi digital oscilloscope at a sampling rate of 1 GHz. Fig. 12 shows the averaged trace [10] of the measurements of the first round, we mark the leakage regions of the target intermediate variable (i.e., the S-box output) in the Fig. and target them in our following attacks. We can see that the intermediate variable leaks in both regions A and B similarly. Additionally, for each region, we apply the PCA to compact measurements [32], [33], [34]. Please refer to Appendix C for the results using linear discriminant analysis (LDA) pre-processing (compared with that using PCA pre-processing).

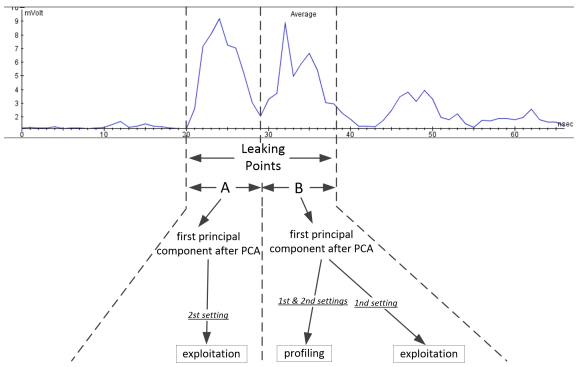

Fig. 12. The average trace of the measurements and the leaking points.

1) Profiled case: In the following, to better illustrate the improvement of ridge-based profiling, we conduct two univariate experiments for different settings, in which we always profile on points of region B but attack (do the exploitation) on points of regions A or B.

First, we assume an ideal univariate (by only targeting the point of first principal component) setting (the 1st setting in Fig. 12) where the profiling and exploitation points are perfectly aligned, thus we use the same region (i.e., region B) for both profiling and exploitation. The left-hand of Fig. 13 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., $\lambda = 50000$) is chosen by mean of the cross-validation like simulation-based experiments. We present the guessing entropies of the LR-based profiling with power model of different degrees as baselines. The results are consistent with the ones of simulation-based experiments and theoretical analysis. As the

[10]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.

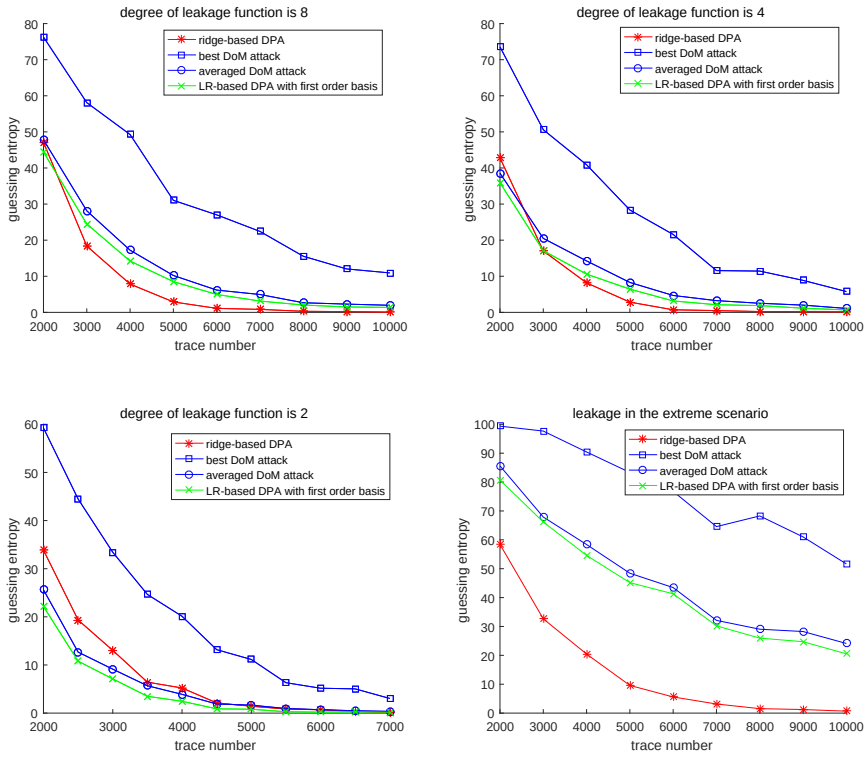

Fig. 11. The guessing entropies of non-profiled DPAs, and SNR = 0.1. The upper-left, upper-right, lower-left, and lower-right figures correspond to degrees 8, 4, 2 and the extreme case respectively.

ridge-based profiling with power models of degrees 4 and 8 are the best ones, we can see that the FPGA implementation in our consideration has nonlinear leakages. On the other hand, the LR-based profiling with degree 1 outperforms that of the higher degrees, which shows that, in the leakage function of our case, the coefficients of the lower degree terms may be more significant than the ones of the higher terms.

Further, we conduct another univariate experiment to show that our new method can be used as a type of robust profiling [11]. As shown in Fig. 12 (the 2nd setting), we profile on the points in B and attack (do the exploitation) on the points in A. Doing this, we aim to show how the deviation of the leakage points affects the ridge-based profiling. The right-hand of Fig. 13 presents the guessing entropies (as functions of the number of profiling traces) for ridge-based with different degrees power model. We choose a same parameter $\lambda = 50000$ as the ideal setting. For clustered-based profiling, we use the K-means clustering method and select the number of clusters (equals 3) to be the one producing the highest value of guessing entropy. We also add the LR-based profilings as the baselines. The results show that the performance of ridge-based profiling is better than the LR-based ones, 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. The results of the clustered-based profiling are consistent with the ones in the simulation-based experiments, and please refer to Appendix B for results in the setting of more exploitation and profiling traces.

At last, we consider the multivariate setting, that is, we apply the PCA to compress measurements into several points (i.e., 1 to 3 points), and conduct the profiling and exploitation (by Bayesian key recovery) in the 1st setting (i.e., same location for profiling and exploitation). In Fig. 14, we the present guessing entropies of the profiled ridge-based (and LR-based) DPA on multiple leakage points. Our experimental results show that the attacks with a single point enjoy the best performance. It indicates that, in our FPGA-based experiments, the first principal component has the largest leakage, and adding more component may introduce more noise. Moreover, even in the multivariate setting, the ridge-based profiling is still better than the LR-based one with same number of points.

2) Non-profiled case: In the non-profiled case, we target on the region B in Fig. 12, and set the parameter λ to 800. In Fig. 15, in univariate setting, we compare the ridge-based DPA with the DoM attacks and the LR-based DPA with first order basis. We can see that the ridge-based DPA outperforms the others, and the LR-based DPA with first order basis performs almost identical to the averaged DoM DPA, which are consistent with the simulation-based experiments. Therefore, the ridge-based DPA is more universally adaptable and can be a good alternative to the traditional (non-profiled) DPAs that rely on the common assumption on the device.

Further, we show the multivariate setting in Fig. 16. We can see that, the (non-profiled) ridge-based DPAs perform very similarly for different numbers of points, and they also outperform the LR-based one for any numbers of points.

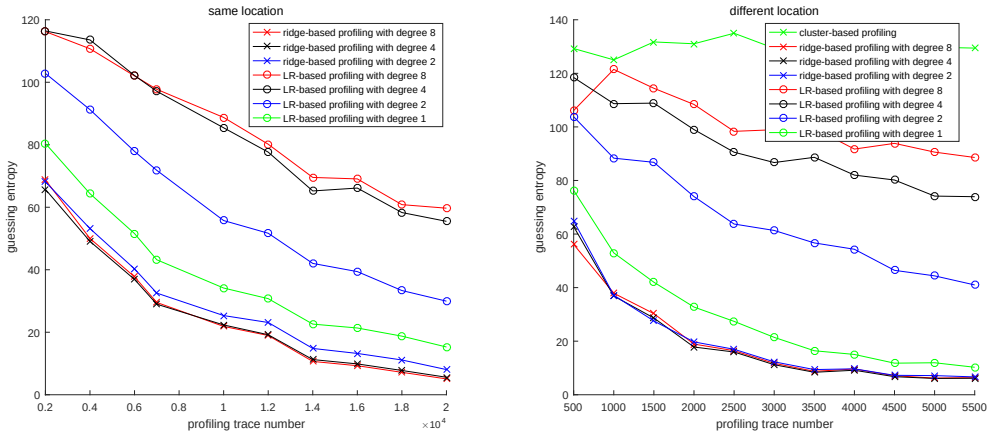

Fig. 13. The guessing entropies of the profiled DPAs in the FPGA-based experiments. Left: 1st setting. Right: 2nd setting

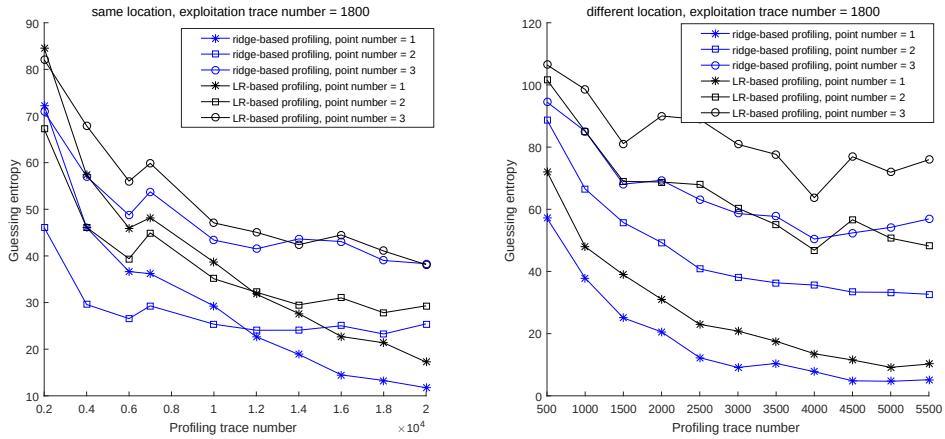

Fig. 14. The guessing entropies of the multivariate profiled DPAs in the FPGA-based experiments. Left: 1st setting. Right: 2nd setting. The model degrees are 2 and 4 for the 1st and 2nd setting respectively.

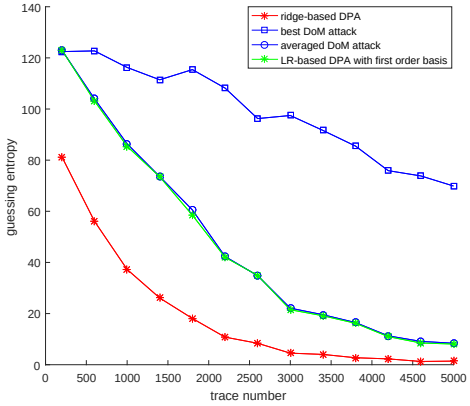

Fig. 15. The guessing entropies of the non-profiled DPAs in the FPGA-based experiments.

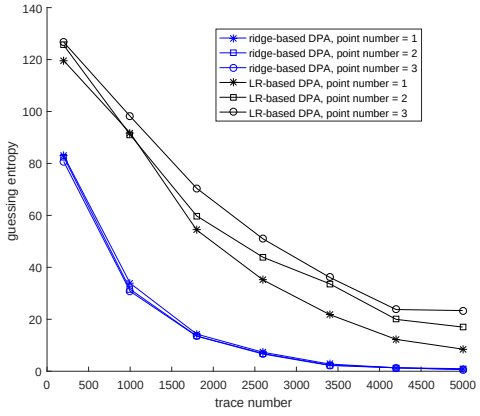

Fig. 16. The guessing entropies of the multivariate non-profiled DPAs in the FPGA-based experiments

VI. Conclusion

In this paper, by applying the ridge regression, we tackle the power variability issue faced by the DPA attacks. By both theory and experiments, we illustrate that our new methods perform better than the state-of-the-art ones and are more universally adaptable to the nanoscale chips.

APPENDIX

A. Comparison between ridge-based and SLR-based profilings by bias-variance tradeoff

By simulation-based experiments, we compute the bias-variance tradeoff of SLR-based and ridge-based profilings. We set the degree of models to 4, and repeat the experiments 200 times with randomly generated coefficients to compute the average values of bias and variance. As shown in Fig. 17, the bias-variance tradeoff is consistent with the results of the theoretical analysis in Section IV-A1: the bias is reduced and variance is increased in relation to the constraint, and the intersection of bias and variance curves implies the optimal choice of the constraint and best trade-off of bias and variance. Moreover, both of the bias and variance curves of ridge-based profiling are much lower than the ones of SLR-based profiling, which indicates that the bias and variance of ridge-based profiling are lower than that of SLR-based profiling.

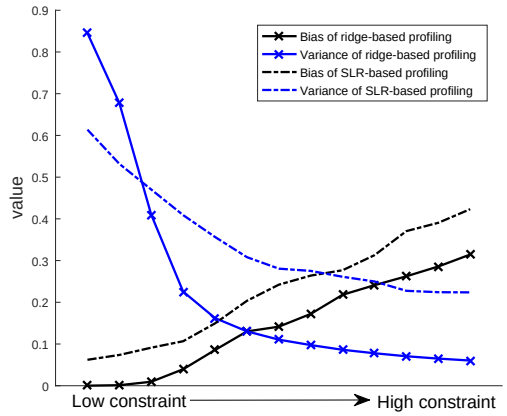

Fig. 17. The bias-variance tradeoff of SLR-based and ridge-based profilings for degree 4 in simulation-based experiments. We set SNR to 0.4, and the profiling trace number is 2000.

B. The results of the experiments with more traces.

Fig. 18 shows the guessing entropies for the robust setting in simulation-based experiments, with more exploitation traces than the ones in Section V-A2. We can see that the guessing entropies of the clustered-based profiling are generally decreased towards 0, whereas the other profiling methods all along perform much better.

Fig. 19 shows the guessing entropies for the robust setting in FPGA-based experiments, with more exploitation traces than the ones in Section V-B1. We can see that the results consistent with the ones of simulation-based experiments.

C. Comparison between LDA and PCA in FPGA-based experiments

Fig. 20 shows (in our FPGA-based experiments of 1st setting) the comparison between LDA and PCA for profiled ridge-based and LR-based DPAs with orders 2 and 1 respectively (which enjoy the best performances). We can see that, in this setting, the PCA outperforms the LDA for both (profiled) ridge-based and LR-based DPAs. Additionally, the ridge-based

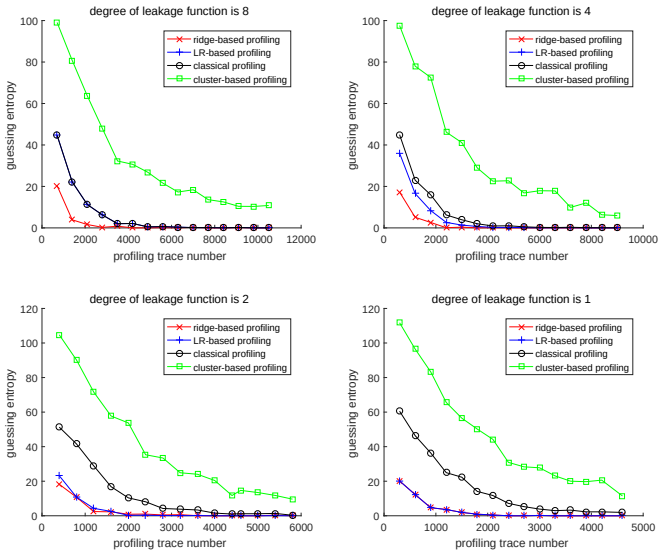

Fig. 18. The guessing entropies of the profiled DPAs of simulation-based implementation with more profiling traces, and exploitation traces number is 20000, SNR = 0.1.

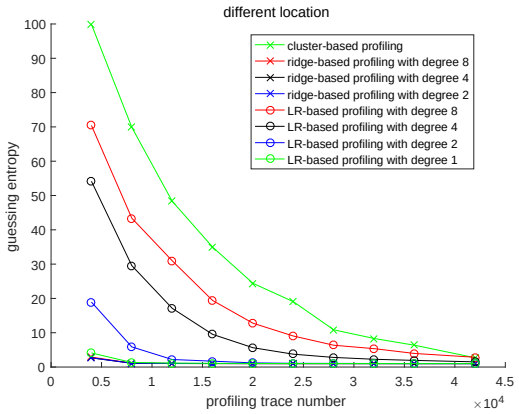

Fig. 19. The guessing entropies of the profiled DPAs in 2nd setting of FPGA implementation with more profiling traces, and exploitation traces number is 15000.

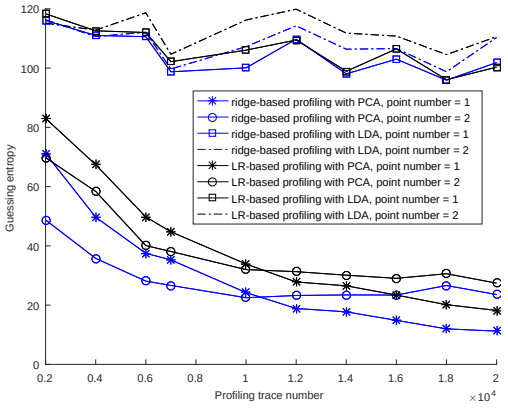

Fig. 20. The comparison of LDA and PCA in profiled case.

DPA still performs better than the LR-based one even with LDA pre-processing.

Fig. 21 shows the comparison between LDA and PCA for

non-profiled ridge-based and LR-based DPAs in our FPGA-based setting. The result is consistent with the profiled case: the PCA outperforms the LDA for both (non-profiled) ridge-based and LR-based DPAs, and the ridge-based DPA still performs better than the LR-based one even with LDA pre-processing.

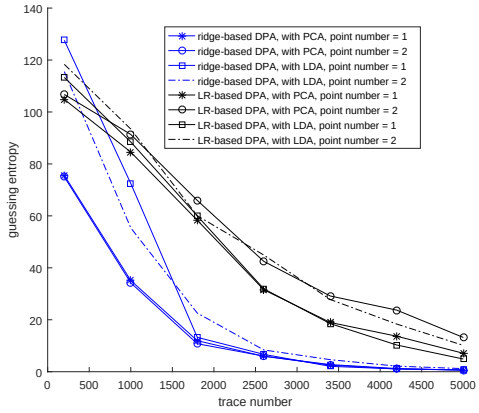

Fig. 21. The comparison of LDA and PCA in non-profiled case.

ACKNOWLEDGMENT

Yu Yu is supported by the National Natural Science Foundation of China (Grant Nos. 61472249, 61572149), the National Cryptography Development Fund MMJJ20170209, and International Science & Technology Cooperation & Exchange Projects of Shaanxi Province (2016KW-038). François-Xavier Standaert is an associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by European Commission through the ERC project 724725 (acronym SWORD) and the H2020 project REASSURE. Junrong liu is supported by the National Natural Science Foundation of China (Grant No. U1536103). Zheng Guo is supported by the National Natural Science Foundation of China (Grant Nos. 61402286, 61572192), Shanghai Minhang Industry-University-Research Cooperation project (No. 2016MH310). Dawu Gu is supported by the National Natural Science Foundation of China (Grant No. 61472250), the Major State Basic Research Development Program (973 Plan) (2013CB338004)

REFERENCES

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, 1999, pp. 388–397.

[2] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, 2004, pp. 16–29.

[3] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, 2002, pp. 13–28.

[4] C. Whitnall and E. Oswald, "Profiling DPA: efficacy and efficiency trade-offs," in *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, 2013, pp. 37–54.

[5] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, 2005, pp. 30–46.

[6] L. Batina, B. Gierlichs, and K. Lemke-Rust, "Differential cluster analysis," in *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, 2009, pp. 112–127.

[7] J. Doget, E. Prouff, M. Rivain, and F. Standaert, "Univariate side channel attacks and leakage modeling," *J. Cryptographic Engineering*, vol. 1, no. 2, pp. 123–144, 2011.

[8] C. Whitnall, E. Oswald, and F. Standaert, "The myth of generic DPA...and the magic of learning," in *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, 2014, pp. 183–205.

[9] N. Veyrat-Charvillon and F. Standaert, "Generic side-channel distinguishers: Improvements and limitations," in *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, 2011, pp. 354–372.

[10] M. Renauld, F. Standaert, N. Veyrat-Charvillon, D. Kamel, and D. Flandre, "A formal study of power variability issues and side-channel attacks for nanoscale devices," in *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, 2011, pp. 109–128.

[11] C. Whitnall and E. Oswald, "Robust profiling for DPA-style attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, 2015, pp. 3–21.

[12] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, 2008, pp. 426–442.

[13] N. Veyrat-Charvillon and F. Standaert, "Mutual information analysis: How, when and why?" in *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, 2009, pp. 429–443.

[14] W. Wang, Y. Yu, J. Liu, Z. Guo, F. Standaert, D. Gu, S. Xu, and R. Fu, "Evaluation and improvement of generic-emulating DPA attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, 2015, pp. 416–432.

[15] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: Data mining, inference, and prediction second edition." Springer New York, 2009, vol. 1, no. 1, pp. 43–94.

[16] K. Lemke-Rust and C. Paar, "Gaussian mixture models for higher-order side channel analysis," in *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, 2007, pp. 14–27.

[17] W. Schindler, "Advanced stochastic methods in side channel analysis on block ciphers in the presence of masking," *J. Mathematical Cryptology*, vol. 2, no. 3, pp. 291–310, 2008.

[18] G. Dabosville, J. Doget, and E. Prouff, "A new second-order side channel attack based on linear regression," *IEEE Trans. Computers*, vol. 62, no. 8, pp. 1629–1640, 2013.

[19] W. Wang, Y. Yu, F. Standaert, D. Gu, S. Xu, and C. Zhang, "Ridge-based profiled differential power analysis," in *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, 2017, pp. 347–362.

[20] C. Carlet, "Boolean functions for cryptography and error correcting codes," *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, vol. 2, pp. 257–397, 2010.

[21] O. Choudary and M. G. Kuhn, "Efficient template attacks," in *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, 2013, pp. 253–270.

[22] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, 1999, pp. 398–412.

[23] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, 2003, pp. 463–481.

[24] M. Rivain and E. Prouff, "Provably secure higher-order masking of AES," in *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, 2010, pp. 413–427.

[25] J. Balasch, S. Faust, and B. Gierlichs, "Inner product masking revisited," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, 2015, pp. 486–510.

[26] K. Schramm and C. Paar, "Higher order masking of the AES," in *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, 2006, pp. 208–225.

[27] B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede, "Revisiting higher-order DPA attacks:," in *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, 2010, pp. 221–234.

[28] E. Oswald, S. Mangard, C. Herbst, and S. Tillich, "Practical second-order DPA attacks for masked smart card implementations of block ciphers," in *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, 2006, pp. 192–207.

[29] F. Durvaux, F. Standaert, and N. Veyrat-Charvillon, "How to certify the leakage of a chip?" in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014, pp. 459–476.

[30] L. Lerman, G. Bontempi, and O. Markowitch, "The bias-variance decomposition in profiled attacks," *J. Cryptographic Engineering*, vol. 5, no. 4, pp. 255–267, 2015.

[31] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[32] C. Archambeau, E. Peeters, F. Standaert, and J. Quisquater, "Template attacks in principal subspaces," in *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, 2006, pp. 1–14.

[33] L. Batina, J. Hogenboom, and J. G. J. van Woudenberg, "Getting more from PCA: first results of using principal component analysis for extensive power analysis," in *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings*, 2012, pp. 383–397.

[34] F. Standaert and C. Archambeau, "Using subspace-based template attacks to compare and combine power and electromagnetic information leakages," in *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, 2008, pp. 411–425.

Weijia Wang was born in China in 1988. He is currently a Ph.D. student of Computer Science and Engineering at Shanghai Jiao Tong University. He received his Master and BSc degree from Tongji University and Chongqing University of Technology respectively. His research interests include side channel analysis, leakage resilient, cryptographic implementations and hardware security.

Yu Yu was born in China in 1981. He received his BSc from Fudan University in 2003 and his PhD from Nanyang Technological University in 2006 respectively. His research interests include theoretical aspects of cryptography such as leakage-resilient cryptography and post-quantum cryptography.

François-Xavier Standaert was born in Brussels, Belgium in 1978. He received the Electrical Engineering degree and PhD degree from the Universite catholique de Louvain, respectively in June 2001 and June 2004. His research interests include digital electronics, FPGAs and cryptographic hardware, low power implementations for constrained environments, the design and cryptanalysis of symmetric cryptographic primitives, physical security issues in general and side-channel analysis in particular.

Junrong Liu is a research assistant at Shanghai Jiao Tong University. He received his Ph.D degree from Shanghai Jiao Tong University in 2016. He received his BSc from Xidian University of China in 1992 and his Master degree from Beijing University of Posts and Telecommunications in 2001. His research interests include cryptography and side channel attack.

Zheng Guo received B.S degree in Electronic Engineering from Shanghai Jiao Tong University in 2002, received M.S. degree in Communication and Information System from Shanghai Jiao Tong University in 2005, and received Ph.D in Comuter Science from Shanghai Jiao Tong University in 2016. He is now an engineer in Shanghai Jiao Tong University. He is also a consultant in Shanghai Viewsource Information Science & Technology Co., Ltd. His research interests include side channel attack and IC design.

Dawu Gu is a full professor at Shanghai Jiao Tong University in Computer Science and Engineering Department. He received from Xidian University of China his B.S. degree in applied mathematics in 1992, M.S. in 1995, and Ph.D. degree in 1998 both in cryptography. His current research interests include cryptography, side channel attack, and software security. He leads the Laboratory of Cryptology and Computer Security (LoCCS) at SJTU.