Profiling DPA: Efficacy and efficiency trade-offs

Carolyn Whitnall and Elisabeth Oswald

University of Bristol, Department of Computer Science,
Merchant Venturers Building, Woodland Road, BS8 1UB, Bristol, UK.
{carolyn.whitnall, elisabeth.oswald}@bris.ac.uk

Abstract. Linear regression-based methods have been proposed as efficient means of characterising device leakage in the training phases of profiled side-channel attacks. Empirical comparisons between these and the 'classical' approach to template building have confirmed the reduction in profiling complexity to achieve the same attack-phase success, but have focused on a narrow range of leakage scenarios which are especially favourable to simple (i.e. efficiently estimated) model specifications. In this contribution we evaluate—from a *theoretic* perspective as much as possible—the performance of linear regression-based templating in a variety of realistic leakage scenarios as the complexity of the model specification varies. We are particularly interested in complexity trade-offs between the number of training samples needed for profiling and the number of attack samples needed for successful DPA: over-simplified models will be cheaper to estimate but DPA using such a degraded model will require more data to recover the key. However, they can still offer substantial improvements over non-profiling strategies relying on the Hamming weight power model, and so represent a meaningful middle-ground between 'no' prior information and 'full' prior information.

Keywords: side-channel analysis, profiled attacks, differential power analysis

1 Introduction

Attackers with the opportunity to *profile* an identical copy of a target device in a preliminary training phase are considered the strongest class of side-channel adversary. Many different strategies have been implemented—some (but not all) are multivariate, incorporating multiple points from a measurement trace; some characterise only the deterministic data-dependent leakage whilst others attempt to characterise the noise also; profiling may be followed by a DPA-style attack

phase, but need not be if the attacker has some other strategy in mind. Historically, the phrase 'template attack' denoted the multivariate Gaussian model variant with full noise characterisation [4]—regarded as the most powerful but also the most impractical method. Unsurprisingly, univariate attacks are much more feasible, and various simplifications make for relatively efficient template building [7]. One particularly interesting option for simplified profiling is to use linear regression [11]. Of course, as soon as more than one profiling method exists the natural question to ask is which is 'better' in practice? Previous studies evaluating linear regression relative to 'classical' templates [5,11,13] have demonstrated substantial efficiency gains in some typical leakage scenarios.

However, this previous work has some limitations. To begin with, comparisons have been predominantly experimental, and performed for devices conforming to Hamming weight (or otherwise close-to-linear) leakage assumptions. Such scenarios naturally favour linear regression from the outset, as the leakage functions may be approximated by very simple model equations (with few parameters and therefore low estimation complexity). Moreover, the comparisons have all been between simple linear regression equations (i.e. low degree polynomials) for intermediate values on the one hand and 'classical' templates for the inputs on the other. These are at opposite ends of a spectrum—'very simple' through to 'very complex' model specifications—leaving the middle ground largely unexplored. Hence we seek to evaluate a wider range of model specifications, in a broader, more varied, set of realistic leakage scenarios.

In an attempt to make unambiguous, like-for-like comparisons, which are not dependent on the estimation procedures used nor on the unknown underlying distributions arising in experimental scenarios, we follow the theoretic approach advocated in [15] in the context of *non-profiled* DPA. Namely, our analytic approach is (as far as possible) based on computed theoretic outcomes rather than estimated experimental outcomes, which entails focusing on fully-specified hypothetical leakage scenarios. We identify three key questions of interest:

1. How accurately does a particular model specification approximate the leakage function? For example, how well can an adversary hope to approximate a highly nonlinear function with a low-complexity model? The asymptotic goodness-of-fit of a model indicates its usefulness in DPA.
2. How many training samples are required in the profiling phase to estimate a particular model to an adequate degree of precision (relative to its asymptotic fit)?
3. How well does correlation DPA perform using a model built to a particular specification? Of most interest to an attacker or a designer/evaluator is the number of trace measurements needed for successful key recovery against the same or a sufficiently similar device.

In the following, we introduce 'classical' templates and the linear regression-based alternatives in Sect. 2 and present our evaluation methodology in Sect. 3.

We apply this methodology to a variety of realistic leakage scenarios and model specifications in Sect. 4. We confront our theoretic expectations with some example experimental analysis in Sect. 5, and conclude in Sect. 6.

2 Preliminaries

2.1 'Classical' templates

In 'classical' template attacks [4] separate multivariate Gaussian models are fitted to the leakage traces associated with each possible value of a particular key-dependent intermediate result V (which might be part of the key directly, or the output of some function that is dependent on part of the key). Supposing, then, that $\mathbf{Y}_v = \{Y_t | V = v\}_{t=1}^{T}$ is the random vector representing the leakage over time given that the associated intermediate target takes the value v; the profiling adversary assumes that $\mathbf{Y}_v \sim \mathcal{N}(\boldsymbol{\mu}_v, \Sigma_v)$ and fits the model by finding the $T \times 1$ sample mean $\hat{\boldsymbol{\mu}}_v$ and the $T \times T$ sample covariance $\hat{\Sigma_v}$ from N_v measurements $\{\mathbf{y}_{v,n}\}_{n=1}^{N_v}$ observed on the profiling device.

2.2 Linear regression-based templates

The approach proposed by [11] is to fit a linear regression model to the pooled data at each point in time: $Y_t = \sum_{j=0}^{p} \beta_{j,t} g_j(V) + \epsilon_t$, where Y_t is the leakage at time t, V is the intermediate value, $\{g_0, \ldots, g_p\}$ are $p+1$ functions of the intermediate value which form the *covariate set* for the model, and $\epsilon_t \sim \mathcal{N}(0, \sigma_t)$ is the residual noise at time t. In practice, g_0 is usually a constant (i.e. 1) and the remaining g_j are monomials of the form $\prod_{i \in \mathcal{I}} v[i]$ where $v[i]$ denotes the i^{th} bit of v and $\mathcal{I} \subset \{1, \ldots, m\}$ (with m the number of bits needed to represent V in binary), so that the model specification is of the form of a polynomial in function of the bits of the intermediate value. Ordinary Least Squares (OLS) is used to obtain the coefficients $\hat{\beta}_{j,t}$ and subsequently the model fitted values $\hat{Y}_t = \sum_{j=0}^{p} \hat{\beta}_{j,t} g_j(V)$. If all the influential terms are included in the model, the fitted values coincide asymptotically with the conditional means obtained via 'classical' templating ($\hat{\boldsymbol{Y}} = \boldsymbol{\mu}_v$). The noise profiling stage consists of estimating a single (pooled) covariance matrix $\hat{\Sigma}$ from the model residuals observed in a second independent sample.

2.3 Exploiting the fitted models for key recovery

Both of the methods output a fitted multivariate Gaussian model for the intermediate value-conditioned leakages:

– 'Classical' template for the T-dimensional leakage of intermediate value v: $\mathcal{N}(\hat{\boldsymbol{\mu}}_v, \hat{\Sigma}_v)$.
– Linear regression-based template for the T-dimensional leakage of intermediate value v:
$\mathcal{N}\left(\sum_{j=0}^{p} \hat{\boldsymbol{\beta}}_{\boldsymbol{j}} g_j(v), \hat{\Sigma}\right)$, where for each $j = 0, \ldots, p$, $\hat{\boldsymbol{\beta}}_{\boldsymbol{j}} = \{\hat{\beta}_{j,t}\}_{t=1}^{T}$ (i.e. $\hat{\boldsymbol{\beta}}_{\boldsymbol{j}}$ is the T-dimensional *vector* of estimated coefficients in the leakage function at each point in time).

Note that the $\hat{\beta}_{j,t}$, as well as $\hat{\Sigma}$, are estimated from the pooled data and are the *same* for all v, whilst $\hat{\boldsymbol{\mu}}_v$ and $\hat{\Sigma}_v$ are estimated from the v-partitioned data.

If the covariance matrix is symmetric and positive definite, a d-dimensional multivariate Gaussian distribution $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is said to be "non-degenerate", and has the following density function:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

(where A' denotes the transpose of matrix A). Otherwise, the distribution does not have a density—although, it is possible to get around this problem by restricting attention to a $\text{rank}(\Sigma)$-sized subset of the modelled vector (in our application, a reduced subset of trace points).

In the case, then, that the Gaussian models estimated in the profiling stage are non-degenerate, let us denote by $f_{CT,v}(\cdot)$ and $f_{LR,v}(\cdot)$ the densities of the 'classical' template and the linear regression-based template for the leakage distribution associated with intermediate value v.

(Bayesian) key recovery comprises acquiring N (T-dimensional) trace measurements $\{\boldsymbol{y}_n\}_{n=1}^{N}$ from the *target* device and selecting, from the set $\mathcal{K}$ of hypotheses on the key-part, the one under which the *likelihood* $\mathcal{L}$ (or, equivalently, log likelihood, to avoid numerical problems) of observing those measurements is maximised, according to the models obtained in the profiling stage.

$$\begin{aligned} k_{guess} &= \underset{k\in\mathcal{K}}{\text{argmax}}\, \mathcal{L}\left(k|\{\boldsymbol{y}_n\}_{n=1}^{N}\right) = \underset{k\in\mathcal{K}}{\text{argmax}} \prod_{n=1}^{N} f_{\cdot,v_{k,n}}(\boldsymbol{y}_n) \\ &= \underset{k\in\mathcal{K}}{\text{argmax}} \sum_{n=1}^{N} \log f_{\cdot,v_{k,n}}(\boldsymbol{y}_n) \end{aligned}$$

where $v_{k,n}$ is the key hypothesis-dependent prediction for the intermediate value corresponding to trace measurement $\boldsymbol{y}_n$.

Alternatively, the model fitted values (for a particular point in time t^*) may be used in a (univariate) correlation DPA [3]. The fitted model produced by 'classical' templates is simply the conditional means which comprise the first parameter of the fitted Gaussian distributions:

$$M_{CT}(v) = \mathbb{E}[Y_{t^*}|V = v] = \hat{\mu}_{v,t^*},$$

whereas the linear regression-based method returns the intermediate value-conditioned fitted values from the linear regression:

$$M_{LR}(v) = \mathbb{E}[Y_{t^*}|V = v] = \sum_{j=0}^{p} \hat{\beta}_{j,t^*} g_j(v).$$

The adversary proceeds in the usual way:

- For each key hypothesis $k \in \mathcal{K}$, predict the intermediate values $\{v_{k,n}\}_{n=1}^{N}$ associated with the set of (univariate) trace measurements $\{y_n\}_{n=1}^{N} = \{y_{n,t^*}\}_{n=1}^{N}$ (we drop the time index for notational convenience).
- Map the predicted intermediate values to a leakage prediction using the power model obtained from profiling $\{M_{k,n}\} = \{M_{\cdot}(v_{k,n})\}_{n=1}^{N}$.
- Compute (again for each key hypothesis $k \in \mathcal{K}$) the sample correlation coefficient between the actual trace measurements and the key-dependent model predictions:
$$r_k = \frac{\sum_{n=1}^{N}(y_n - \overline{y})(M_{k,n} - \overline{M_{k,n}})}{\sqrt{\sum_{n=1}^{N}(y_n - \overline{y})^2}\sqrt{(M_{k,n} - \overline{M_{k,n}})^2}}$$
(where $\overline{a}$ denotes the mean of a set of values $\{a_n\}_{n=1}^{N}$, i.e. $\overline{a} = \frac{1}{N}\sum_{n=1}^{N} a_n$).
- Choose as key guess the one which maximises the sample correlation: $k_{guess} = \underset{k \in \mathcal{K}}{\text{argmax}}\{r_k\}$.

In the following, we focus on the *goodness-of-fit* of each model specification—that is, the accuracy of the fitted values as approximations for the data-dependent deterministic part of the device leakage, leaving analysis of the noise characterisation as further work. Therefore, all evaluations of key recovery performance are made in the context of correlation DPA.[1]

2.4 Models for inputs vs. models for intermediate values

In the above, we have presented templates in the context of building models for *intermediate values*, but the original proposal [4] was to build them for (*input-part*,*key-part*) pairs without predicting or specifying any particular function. It was noticed that, as long as the algorithm possessed certain symmetry properties [13], the profiling workload could be reduced considerably. E.g., if a known

[1] Correlation DPA is generally accepted as the best performing strategy whenever a good (proportional) power model is available. Scenarios in which other strategies have the potential to outperform correlation DPA (see, e.g. [14]) have, to our knowledge, *all* so far been such that this was not the case—for example, those where the adversary only has access to a *nominal* approximation of the leakage function.

combining function (such as XOR) is used to mix the key bits with the plaintext bits, templates only need to be built for every possible *combination* (*input-part* $\oplus$ *key-part*)—in the case of 8-bit key-parts, this reduces the number of templates from 2^{16} to 2^8.

In 'classical' templating, a separate model for each combination amounts to the same thing as a separate model for each output of *any* (injective) component of the algorithm—an S-box, say—so that the particular intermediate values need not be specified by the attacker. This is useful because even when the full details of the algorithm are known, it may not be clear in advance at which points the device leakage is most vulnerable. Such a strategy recovers a model at each point in the trace which essentially maps the combined (*input-part*,*key-part*) value to the composition of the corresponding intermediate function and the leakage (performed in that order). One disadvantage is that, *without* knowing which intermediate values occur where in the trace, one does not actually learn the functional form of the leakage on its own so as to be able to use it in an attack against a different (specified) target function on a similar device (the templates can *only* be used to attack the same (sequence of) function(s) as the ones for which they were built). The pros and cons of different strategies for 'classical' templating are explored in more depth in Chapter 5 of [7].

It has been observed (e.g. in [13]) that linear regression-based methods do not have this capability. Fitting a model for the leakage of an *unspecified* target function—i.e., expressing the leakage in terms of the *input* bits similar to the above—will produce an approximation for the composition of the target and the leakage. If, then, the target is nonlinear (an S-box, for example) and the fitted model only includes linear or low-order terms, the approximation may be very poor. When all higher-order terms are included the approximation equates with that produced by 'classical' templates—with equally high profiling complexity and the same drawbacks of unportability. By contrast, when the model is specified in function of the *output* of a particular target, a transportable 'leakage-only' approximation is obtained, most likely requiring only low-order terms.

It is of practical interest, then, to consider the performance of linear regression-based templates of varying degree against unspecified targets. The 'best' model fit possible arises when a full set of polynomial terms is included in the regression equation (coinciding with the fitted values produced by 'classical' templates). But simplified models do capture *something* of the relationship between the target inputs and the leakage; the question is, how much, and is it useful? We will explore this as part of our analysis in Sect. 4.

3 Methodology

We want to know whether a given linear regression model specification will produce a 'good' DPA power model. We have identified the following criteria for a power model to be considered 'good':

1. Goodness-of-fit: The OLS-estimated fitted values are an asymptotically accurate approximation of the true data-dependent deterministic component of the device leakage.
2. Profiling complexity: The profiling phase to estimate the model is efficient (with respect to the amount of data required from the training device).
3. DPA performance: A DPA attack using the model is effective and efficient (with respect to the amount of data required from the target device).

Following the example of [15] we wish to carry out our evaluations as far as possible from a theoretic perspective, computing underlying theoretic quantities from fully-specified leakage distributions so that our evaluations are not contingent on the quality of our chosen estimation procedures. This also removes the element of 'guesswork' which inevitably accompanies attempts to evaluate experimental results, where the true underlying distributions arise from a real device and are therefore unknown.

Criterion 1 can be easily assessed by finding the least-squares solution (for $\boldsymbol{\beta}$) to the following system of equations representing the linear regression model in the absence of noise:

$$\{Y_v\}_{v\in\mathcal{V}} = \left\{\sum_{j=0}^{p} \beta_j g_j(v)\right\}_{v\in\mathcal{V}}.$$

The population[2] coefficient of determination ρ^2 represents the proportion of the variance in the data-dependent leakage function which is accounted for by the model. It is computed as the square of the correlation between the (asymptotic) fitted values $\{\hat{Y}_v\}_{v\in\mathcal{V}} = \{\sum_{j=0}^{p} \hat{\beta}_j g_j(v)\}_{v\in\mathcal{V}}$ and the actual values $\{Y_v\}$. This is our measure of goodness-of-fit.[3]

[2] 'Population' because we are considering computed theoretic quantities, not estimations from a sample. The *sample* coefficient of determination is the R^2, computed as the square of the correlation between the estimated fitted values and the sample.

[3] The 'perceived information' profiling metric proposed in [10] attempts to jointly capture model quality and device vulnerability, inspired by the 'mutual information' metric of [12]. For our purposes, we are interested in model quality distinct from device vulnerability, for which the coefficient of determination is a more appropriate natural indicator.

Criterion 2 is harder to evaluate theoretically. *Statistical power analysis*[4] [6] provides formulae for computing the sample sizes required for estimation, in straightforward scenarios where all relevant sampling distributions are known—applicable, perhaps, to the estimation of the conditional means in 'classical' templating, but not possible in general for complex estimation tasks like linear regression. Many (different) heuristics have been offered but remain *very* 'rule-of-thumb'—primarily designed as safeguards against over-ambitious use of data.

What *is* known is that the required sample size increases with the number of parameters to be estimated: we can assert with confidence that the simpler the polynomial expression for the leakage, the fewer trace measurements are needed to fit the model. Thus the appeal of linear regression model building, which is upper-bounded in complexity (as well as goodness-of-fit) by 'classical' templating. However, we go one step further than this intuition, and, in the absence of theoretic formulae, take an *empirical* approach—performing repeat random experiments to ascertain the average sample size needed to obtain a 'precise' fit as the degree of the model expression (and therefore the number of parameters to be estimated) varies.

The appropriate threshold for 'sufficient precision' depends on the context. We want our fitted models to be precise enough for distinct values to be separated, and so have selected precision margins based on 10 percent and 5 percent of the distance between unique values (0.1 and 0.05 respectively, in the case of Hamming weight leakage). These are arbitrarily chosen; our analysis later on (Sect. 4) indicates that any choice suffices to demonstrate *relative* profiling complexity.

We report the sample size at these two thresholds as the number of traces required so that the mean difference between the fitted values and their corresponding asymptotic values falls within those margins. These are obtained by averaging over 1,000 repeat experiments on randomly drawn balanced samples (i.e. comprising an equal number of replicates per intermediate value) with Gaussian noise at high (8), medium (1) and low (0.125) signal-to-noise ratios (SNRs)[5] as model degree ranges from 1 through to 8.[6]

Criterion 3 can be assessed straightforwardly by computing theoretic distinguishing vectors for correlation DPA using the asymptotically fitted model cor-

[4] 'Power' in this context refers to statistical power and should not be confused with the 'Power' in DPA.

[5] We define the SNR as $\frac{\mathrm{var}(L(V))}{\mathrm{var}(\varepsilon)}$, where L is the data-dependent leakage function (the variance of which is computed with respect to the distribution of the intermediate value V, which is uniform throughout in our analysis) and ε is the independent noise.

[6] To reduce computational complexity we take the usual strategy (see, for example, [1]) of fitting the models to intermediate value-conditioned mean traces rather than the increasingly large observation-level samples. For our purposes this is inconsequential, as the estimates on the coefficients are not affected and we are not concerned with statistical inference.

responding to a given specification, as per [15].

$$D_\rho(k) = \rho(Y, M_{LR}(V_k)) = \frac{\mathrm{cov}(Y, M_{LR}(V_k))}{\sqrt{\mathrm{var}(Y)}\sqrt{\mathrm{var}(M_{LR}(V_k))}} \tag{1}$$

(where Y is the actual device leakage, and V_k is the intermediate value predicted under key hypothesis k, viewed as random variables). This yields the nearest-rival distinguishing margin (the difference between the 'correct key' distinguisher value $D_\rho(k^*)$ and that relating to the highest-ranked alternative $D_\rho(k^{\mathrm{nr}})$), from which can be predicted the number of traces needed for a key recovery success, using the widely-adopted 'rule-of-thumb' suggested in chapters 4 and 6 of [7]:

$$N^* = 3 + 8 \cdot \frac{{z^2}_{1-\alpha}}{\left(\ln \frac{1+D_\rho(k^*)}{1-D_\rho(k^*)} - \ln \frac{1+D_\rho(k^{\mathrm{nr}})}{1-D_\rho(k^{\mathrm{nr}})}\right)^2}, \tag{2}$$

where $z^2_{1-\alpha}$ is the $(1-\alpha)$-level critical value in the standard Normal distribution. Such formulae originate in the practices of statistical hypothesis testing, where the aims are subtly different to those of DPA. It is difficult to determine the 'right' α (the 'false positive' rate—i.e. in our case the probability of deciding in favour of an incorrect key) since in practice DPA success is measured via crude 'correct/incorrect' criteria without consideration for statistical significance. Our computations are based on $\alpha = 0.1$—a comparatively lax threshold to reflect the key guess strategy employed in practical attacks—but we focus on relative attack complexity rather than the raw numbers. However, as we explore briefly in Sect. 5, the sensitivity of the analysis to the size of the α, and the overly-simplified assumptions inherent in the 'rule-of-thumb', can distort the theoretic predictions away from the relative complexity displayed in practice.

4 Analysis

In this section, we evaluate (via the methodology described above) linear regression model specifications of increasing polynomial degree, for a variety of (8-bit) leakage scenarios and attack assumptions. The hypothetical leakage functions we consider are the Hamming weight, a degradation of the Hamming weight in which interactions between adjacent wires also contribute, and a leakage function based on the toggle count of a VHDL description of the AES S-box. The independent noise is Gaussian in all cases and of the same magnitude for all inputs/intermediate values. We also consider models built for intermediate values (the AES S-box and AES AddRoundKey) vs. models built for unspecified targets via the inputs, as discussed in Sect. 2.4. We summarise key features of the different model specifications in Table 2, Sect. 4.3 (alongside the corresponding features of a non-profiled Hamming weight power model in relation to the same scenarios, for comparison).

4.1 Hamming weight leakage

We first consider the case that the device leaks the Hamming weight of the intermediate values processed internally. This is a popular context for research as it is both highly realistic (e.g., frequently observed in devices built using CMOS logic) and straightforward to analyse. Indeed, many previous works evaluating profiling methods [5,13] have focused on this scenario—either from a theoretic perspective, or as a consequence of carrying out experiments on typical devices.

Models for intermediate values (Scenario 1) Fitting a model for Hamming weight leakage in function of the bits of an intermediate value can be done very efficiently using OLS with a linear basis (so that you only need to estimate 8 coefficients and an intercept). Asymptotically, this will give a perfect approximation for the data-dependent leakage, as shown in the first panel of Fig. 1.

Since this strategy only requires estimating 9 parameters, the profiling phase requires minimal data from the training device. Table 1 shows the experimentally-obtained sample sizes required to achieve 5 percent and 10 percent precision relative to the asymptotic model fit as the SNR decreases. The data cost of estimating 256 separate means as per 'classical' templating ranges from 15 to over 30 times that of fitting the linear regression model with linear terms only, depending on the SNR levels and the margin threshold. Note that, since we are considering balanced samples only, the profiling complexity is lower bounded by 1 trace per intermediate value; in practice, OLS-fitted models in low-noise scenarios may well achieve adequate precision even when not all of the intermediate values are represented in the sample, so complexity in such cases may be over-estimated (hence, in Table 2, we report relative complexities based on the noisy scenario). However, the balanced sample approach is typical for 'classical' template building (e.g. [4]) and so we adopt it ourselves as being the most appropriate basis for like-for-like comparison.

Table 1: *Number of traces required* per intermediate value *for precise model fit in a Hamming weight leakage scenario.*

		5 percent margin			**10 percent margin**		
Method	**Params**	**SNR=8**	**SNR=1**	**SNR=0.125**	**SNR=8**	**SNR=1**	**SNR=0.125**
Classical	256	67	525	4206	17	134	1115
Degree 1	9	3	17	132	1	5	33
Ratio		22	31	32	17	27	34

It is already well-established in the literature [9] that the performance of any DPA attack depends not just on the form of the leakage and the quality of the

model but also on the target function. The second and third panels of the figure illustrate the nearest-rival margins and the required sample sizes for attacks against the AES S-box and AES AddRoundKey. The cryptanalytically robust properties of the S-box actually make it *more* vulnerable to DPA, as a small change in the input produces a large change in the output so that the correct hypothesis can be readily distinguished from the alternatives. Thus the theoretic distinguishing vectors for the S-box attacks have larger nearest-rival margins and the corresponding sample sizes are smaller than those for the attacks against AddRoundKey.

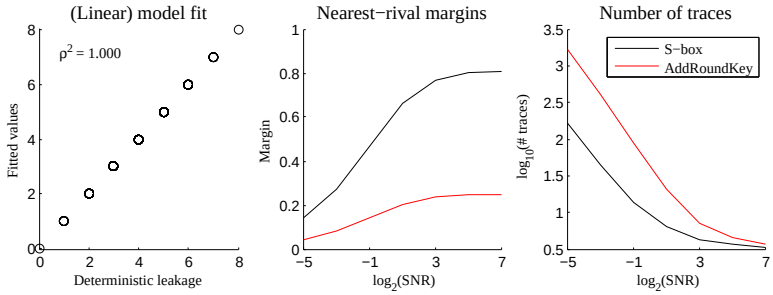

Fig. 1: *Asymptotic model fit and DPA performance of an OLS-estimated model specified as a linear function of the target bits, when the true leakage is Hamming weight.*

It is clear that this straightforward leakage scenario—which is the one investigated in [5] and [13]—lends itself very naturally to linear regression-based profiling, as the true data-dependent leakage function can be easily and precisely approximated with only linear terms. Our experiments indicate that the profiling stage requires around thirty times fewer training samples than 'classical' templates with *no trade-off* on model precision, fit, nor DPA performance. In the following sections we examine some more 'interesting' (but still realistic) scenarios in which simplified approximations may no longer be adequate.

Models for inputs (Scenario 2) We next suppose that the attacker attempts to build models without specifying the intermediate function, so that the linear regression function is expressed as a polynomial in the input bits (that is, the XOR between the input-part and the key-part), as per the discussion in Sect. 2.4. In such cases, the complexity of the model required to produce an asymptotically perfect fit will depend on the complexity of the target function (which might be a highly nonlinear S-box). This is the scenario to which we will pay most attention, as it is one in which the advantages and disadvantages of simplified approximations can be thoroughly explored.

Fig. 2 shows what happens when you build a model for the (Hamming weight) leakage of an AES S-box output in function of the input bits. The linear and quadratic models are very poor approximations (although, far better than simply taking the Hamming weight of the input). The degree 7 model gives a very close fit, which is unsurprising as only one term has been omitted.

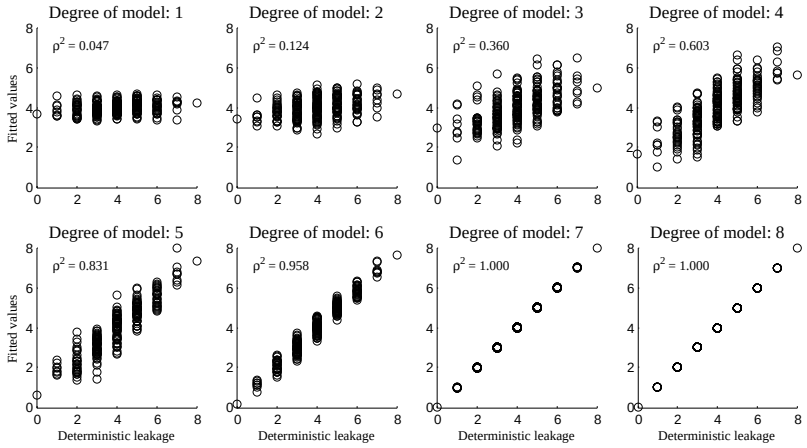

Fig. 2: *Asymptotic fitted values from OLS-estimated models for the leakage of an AES S-box output, in (increasing degree) polynomial function of the* inputs.

This scenario is a good test case for examining profiling complexity because the true (composite) leakage is highly non-linear so that all of the interaction terms are required to perfectly characterise it. It is one thing to show that the approximation improves as the model degree increases, but at what cost? By how much does the number of training traces need to increase to maintain an equivalent level of precision at each level of complexity?

The mean and the 10th and 90th percentiles of the sample size to achieve precision to within margins of 0.05 and 0.1 of the asymptotic values (as per Sect. 3) are reported in Fig. 3. As expected, the sample sizes required to estimate the maximum degree polynomials are much higher (around 30 times more) than the sample sizes required to estimate the linear polynomials. There is little difference in estimation complexity between degree 6 and degree 8 models, which is not surprising when we consider that there is only one degree 8 term and only 8 degree 7 terms, so the reduction in the number of parameters is small. Only models with degree 5 or lower begin to offer reasonable savings. Required sample size increases as signal decreases, as we would expect and in a consistent manner as model degree varies.

We now turn our attention to the performance of DPA attacks using the differently-accurate approximations as power models. Fig. 4 shows the distinguishing vector

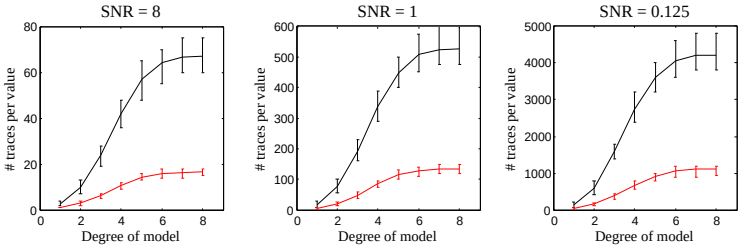

Fig. 3: *Mean sample size required (per intermediate value) to estimate model to 0.05 (black) and 0.1 (red) of the asymptotic model fit. Error bars depict 10th and 90th percentiles.*

nearest-rival margins and the corresponding estimates on the sample size required for key recovery, as the model specifications vary from linear terms only to maximum-degree polynomials.

The model built in maximum-degree polynomial function of the inputs approximates the data dependent leakage perfectly; the fitted values coincide with those from the simple model built in linear function of the intermediate value bits and, inevitably, it performs equivalently in key recovery (as we confirm by comparing Fig. 4 with Fig. 1). It has the advantage that the target function need not be specified for the model to be estimated, but the disadvantage that fitting the maximum-degree polynomial to the leakage has the same data complexity as estimating separate input-conditioned means, as is done for 'classical' templates. Under such circumstances there are no efficiency advantages to using linear regression-based profiling.

Lower degree specifications can only produce *less* accurate approximations, so inevitably incur a loss of DPA performance. It is evident that a trade-off between model-fitting complexity and key-recovery complexity is possible. The ballpark summary figures in Table 2 ('Scenario 2' column) help to get to grips with this. It is immediately clear that, for the lower degree models, the trade-offs are, in general, not of comparable magnitude—that is, small savings in the profiling phase can produce large costs in the attack phase. Nonetheless, the degree 4 model may be of interest: profiling complexity is reduced to just 63% of the traces required for 'classical' templates, at a cost of only around 3 times as many traces in the DPA attack phase. For adversaries with limited access to the training device but good access to the target device, even a degree 3 model may suffice: key recovery requires around 8 times as many attack traces, but profiling requires just a third of the number of training traces. Interestingly, even the models built to linear specifications are able to recover the key (unlike a non-profiled attack using the Hamming weight, as reported in the first row of the table), although with a large expected increase in attack data complexity relative to better fit models.

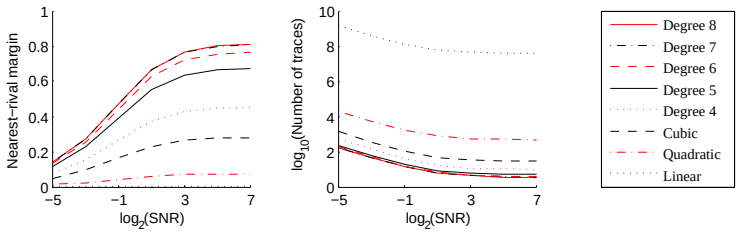

Fig. 4: *Nearest-rival margins and estimated data complexity of key-recovery correlation DPA attacks against the AES S-box output using OLS-fitted models expressed as (increasing degree) polynomials in the input bits.*

4.2 Other leakage scenarios

We have shown above that the attacker strategy (models for intermediate values vs. models for inputs) can influence the effectiveness of a linear regression-based templating phase, even when the true leakage function is very straightforward. In the case that the leakage function is *not* straightforward (i.e. is itself nonlinear) an attacker may be even more limited in what he can achieve using linear regression, as even intermediate value models will need to be increasingly complex in order to well-approximate the device leakage.

Leakage with adjacent bit interactions (Scenario 3) One realistic scenario we might consider is that adjacent wires in the device influence each other, so that the true function is quadratic in the targeted bits (see, e.g., [2]). The first panel of Fig. 5 (in Appendix A) shows the asymptotic fit of the linear and quadratic models (in function of the intermediate value bits) produced by OLS for an example such leakage distribution. The linear model, with 9 coefficients to estimate by comparison with the 256 conditional means required by 'classical' templates, is already a close fit (better than the Hamming weight), with a population coefficient of determination $\rho^2 = 0.96$. The quadratic model is (asymptotically) a perfect fit, and still only requires estimating $1 + 8 + 28 = 37$ coefficients total (or, $1 + 8 + 7 = 16$ if the adversary correctly assumes that only adjacent wires interact). We expect the number of traces required for precise profiling to be similar to those of the linear and quadratic models in the experimental results of Fig. 3—that is, around 3% and 13% of the number of traces required for 'classical' templates.

As before, we compute nearest-rival margins and the corresponding sample size requirements directly from the theoretic correlation DPA vectors. The second and third panels of Fig. 5 show that there is very little difference in attack capability between the linear and quadratic approximations (even the linear performs better than the Hamming weight), suggesting that—in this case—the

reduced covariate set would do just as well. (See section 'Scenario 3' of Table 2 for summary figures).

Toggle-count leakage (Scenario 4) The power consumption of hardware implementations have been shown to depend on the number of *transitions* that occur in the S-box, which can be computed from back-annotated netlists as in [8]. This produces leakages which are highly nonlinear in function of the input or the output bits of the S-box.

Our analysis of models built for the toggle-count based leakage function of [8] in function of the intermediate values (i.e. the AES S-box outputs) is summarised in section 'Scenario 4' of Table 2. The population coefficients of determination for the different model specifications (see also Fig. 6 in Appendix A) compare very similarly to those of the input-based models for Hamming weight leakage (Scenario 2), suggesting similar profiling trade-offs (again, we expect the sample sizes required for precise estimation to be comparable as model complexity varies).

Interestingly, although there is little difference in model fit between the two scenarios, the low degree approximations do much better in terms of attack phase performance (relative to 'classical' templates) than those in Scenario 2 (see also Fig. 7 in Appendix A). The linear model has a ρ^2 of 0.06 compared with 0.05 in Scenario 2, and yet the expected number of traces required relative to DPA attacks using the 'classical' templates is more modest than the increases expected in Scenario 2. Similarly, the quadratic model in Scenario 4 has a ρ^2 of 0.13 compared with 0.12 in Scenario 2, whilst the traces for key recovery are $\sim$20-30 times the number required by 'classical' templates in Scenario 4, compared with $\sim$120-140 in Scenario 2.

4.3 Summary

We have shown that approximating leakage functions with low degree polynomials via OLS estimation is extremely efficient and effective in the case that the leakage is linear or close to linear. The profiling phase requires only a fraction ($\sim 13\%$) of the number of traces needed to build 'classical' templates to the same degree of precision, with no increase in the traces required for successful key recovery in the attack phase. Even when faced with high degree leakage—either the composite of a highly nonlinear target function with a 'straightforward' leakage or the type of highly nonlinear leakage produced by hardware implementations—a low degree approximation can achieve substantially more than a non-profiled Hamming weight power model (as presented for comparison in the first row of Table 2)—demonstrating the value even of minimal profiling. However, in such cases only high degree model specifications—of similar profiling complexity to 'classical' templates—are able to achieve similar attack-phase efficiency.

Table 2: *Summary of linear regression models relative to 'classical' templates.*

			Scenario 1		Scenario 2		Scenario 3		Scenario 4	
Model	**#Params**	**C2**	**C1**	**C3**	**C1**	**C3**	**C1**	**C3**	**C1**	**C3**
HW	–	0	1	1	0.00006	N/A	0.88	1.2–1.3	0.04	930–1,270
Deg. 1	9	0.03	1	1	0.05	8×10^6–1×10^7	0.96	1.0–1.1	0.06	136–220
Deg. 2	37	0.13	1	1	0.12	117–142	1	1	0.13	19–29
Deg. 3	93	0.33	1	1	0.36	7.6–8.3	1	1	0.35	3.6–5.2
Deg. 4	163	0.63	1	1	0.60	2.7–3.3	1	1	0.65	1.7–2.2
Deg. 5	219	0.83	1	1	0.83	1.4–1.5	1	1	0.85	1.2–1.4
Deg. 6	247	0.90	1	1	0.96	1.1	1	1	0.96	1.0–1.1
Deg. 7	255	1	1	1	1	1	1	1	1	1
Deg. 8	256	1	1	1	1	1	1	1	1	1

Notes: *C1: Population coefficient of determination (ρ^2) of asymptotic model fit; C2: Number of traces required (per intermediate value/input) in the profiling phase as a proportion of the number required to build 'classical' templates (based on the 'noisy' scenario); C3: Number of traces required for successful correlation DPA for every one trace required when 'classical' templates are used (as the SNR ranges from 2^{-5} to 2^7). Scenario 1: Hamming weight leakage, models built for intermediate values (Sect. 4.1); Scenario 2: Hamming weight leakage, models built for inputs (Sect. 4.1); Scenario 3: Adjacent bit interactions (Sect. 4.2); Scenario 4: Toggle-count leakage (Sect. 4.1).*

5 Some experimental results

To see how the expected outcomes play out in practice, we performed experimental profiling attacks against simulated leakage of an AES S-box under scenario 4 with an SNR of 1. Table 3 shows the numbers needed to achieve a 99 per cent success rate as model complexity and the number of traces for profiling varies. It is clear from the last column of the table that even with an asymptotic profiling phase the ratio between the 'low degree' end, where the distinguishing margins are small, and the 'high degree' end, where they are large, is rather more modest than that implied by the analysis in Table 2. This highlights the imperfect nature of the heuristic rule-of-thumb—which is widely relied upon as an appealing means of quantifying attack complexity without performing the attacks, but may produce distortions in cases like this where the simplifying assumptions of bivariate normality are met to different degrees for the models being compared, and where (we conjecture) the over-exacting requirements of statistical significance impose a greater relative divergence from practice when the margins are small. We concede that Table 2 should be interpreted with caution; experimental analysis may be required to produce more true-to-life results for poor quality power models.

Model	256 ×1	256 ×2	256 ×5	256 ×10	Asymptotic fit	Ratio to 'classical'
HW	–	–	–	–	13500	281.3
Deg. 1	0	0	6800	4250	2900	60.4
Deg. 2	0	1550	1000	875	750	15.6
Deg. 3	550	370	310	270	230	4.8
Deg. 4	230	170	120	110	95	2.0
Deg. 5	170	120	80	70	60	1.3
Deg. 6	140	100	70	60	50	1.0
Deg. 7	130	95	65	55	48	1.0
Deg. 8	130	95	65	55	48	1.0

Table 3: *Number of traces needed to achieve a success rate of 99 percent in 2,000 experiments against simulated AES S-box leakage (scenario 4) with an SNR of 1. Where a 99 percent success was not achieved because of model inadequacy we have reported the asymptotic success rate.*

6 Conclusion

Models built to over-simplified specifications may be estimated more cheaply than maximum-complexity 'classical' templates but incur greater data costs in the DPA attack phase than they save in the profiling phase. However, they may represent a 'middle ground' for attackers with limited access to a training device (but relatively free access to the target device), or for whom it is more convenient to build models for the inputs rather than particular intermediate values. That is, lower degree models still capture enough of the data-dependent variation to succeed in a DPA phase, so long as they are supplied with sufficient measurements from the attacked device. In particular, even very minimal profiling can substantially improve on what is possible for a completely uninformed attacker relying on the Hamming weight power model (although we find that the *magnitudes* of the differences in complexity implied by the common rule-of-thumb may be exaggerated at the 'minimal' end).

References

1. The DPA Contest. `http://www.dpacontest.org/`. (Accessed 5th September 2012).
2. M.L. Akkar, R. Bevan, P. Dischamp, and D. Moyart. Power Analysis, What is Now Possible... In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT '00*, volume 1976 of *LNCS*, pages 489–502, 2000.
3. E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In M Joye and J-J Quisquater, editors, *Proceedings of CHES 2004*, volume 3156 of *LNCS*, pages 135–152. Springer Berlin / Heidelberg, 2004.
4. Suresh Chari, Josyula Rao, and Pankaj Rohatgi. Template Attacks. In Burton Kaliski, Çetin Koç, and Christof Paar, editors, *Proceedings of CHES 2002*, volume 2523 of *LNCS*, pages 51–62. Springer Berlin / Heidelberg, 2003.

5. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In Louis Goubin and Mitsuru Matsui, editors, *Proceedings of CHES 2006*, volume 4249 of *LNCS*, pages 15–29. Springer, 2006.
6. Helena C. Kraemer and Sue Thiemann. *How Many Subjects?: Statistical Power Analysis in Research*. Sage Publications, Inc, 1st edition, September 1987.
7. S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007.
8. Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In Josyula R. Rao and Berk Sunar, editors, *Proceedings of CHES 2005*, volume 3659 of *LNCS*, pages 157–171. Springer, 2005.
9. E. Prouff. DPA Attacks and S-Boxes. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption*, volume 3557 of *LNCS*, pages 424–441. Springer Berlin / Heidelberg, 2005.
10. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT '11*, volume 6632 of *LNCS*, pages 109–128. Springer, 2011.
11. Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In Josyula Rao and Berk Sunar, editors, *Proceedings of CHES 2005*, volume 3659 of *LNCS*, pages 30–46. Springer Berlin / Heidelberg, 2005.
12. F-X Standaert, T. G. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT '09*, volume 5479 of *LNCS*, pages 443–461, Berlin, Heidelberg, 2009. Springer–Verlag.
13. François-Xavier Standaert, François Koeune, and Werner Schindler. How to Compare Profiled Side-Channel Attacks? In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *ACNS*, volume 5536 of *LNCS*, pages 485–498, 2009.
14. Carolyn Whitnall and Elisabeth Oswald. A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO '11*, LNCS. Springer Berlin / Heidelberg, 2011.
15. Carolyn Whitnall and Elisabeth Oswald. A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. *Journal of Cryptographic Engineering*, 1(2):145–160, August 2011.

A Figures

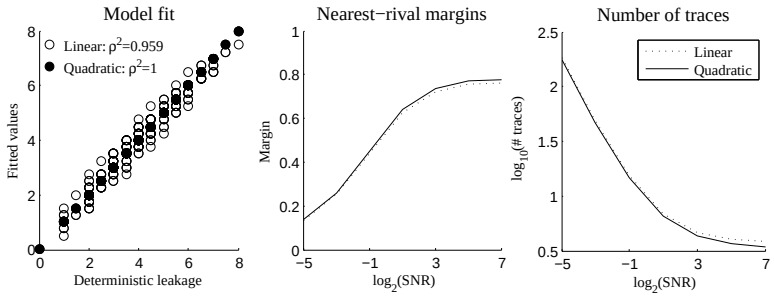

Fig. 5: *The asymptotic fit and DPA performance of OLS-fitted models specified as linear and quadratic functions of the target bits, when the true leakage has adjacent bit interactions.*

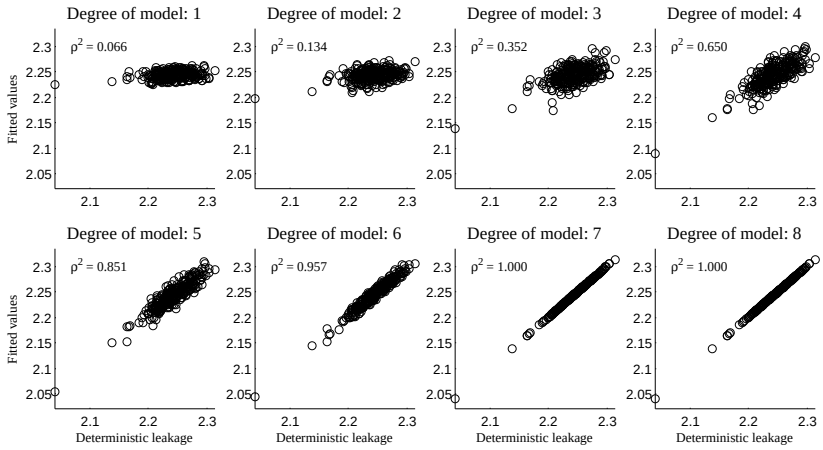

Fig. 6: *The asymptotic fit of OLS-fitted models specified as increasingly high degree polynomials of the intermediate value bits, when the true leakage is highly nonlinear (based on the toggle-count).*

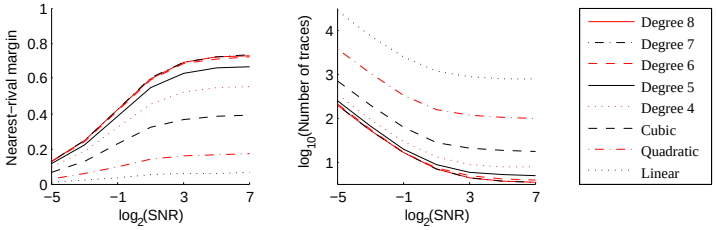

Fig. 7: *Nearest-rival margins and estimated data complexity of key-recovery correlation DPA attacks against highly nonlinear (toggle-count) leakage of the AES S-box, using OLS-fitted models of increasing polynomial degree.*