How to Certify the Leakage of a Chip?

François Durvaux, François-Xavier Standaert, Nicolas Veyrat-Charvillon

UCL Crypto Group, Université catholique de Louvain.
Place du Levant 3, B-1348, Louvain-la-Neuve, Belgium.

Abstract. Evaluating side-channel attacks and countermeasures requires determining the amount of information leaked by a target device. For this purpose, information extraction procedures published so far essentially combine a "*leakage model*" with a "*distinguisher*". Fair evaluations ideally require exploiting a perfect leakage model (i.e. exactly corresponding to the true leakage distribution) with a Bayesian distinguisher. But since such perfect models are generally unknown, density estimation techniques have to be used to approximate the leakage distribution. This raises the fundamental problem that all security evaluations are potentially biased by both estimation and assumption errors. Hence, the best that we can hope is to be aware of these errors. In this paper, we provide and implement methodological tools to solve this issue. Namely, we show how sound statistical techniques allow both quantifying the leakage of a chip, and certifying that the amount of information extracted is close to the maximum value that would be obtained with a perfect model.

1 Introduction

Side-channel attacks aim to extract secret information from cryptographic implementations. For this purpose, they essentially compare key-dependent leakage models with actual measurements. As a result, models that accurately describe the target implementation are beneficial to the attack's efficiency.

In practice, this problem of model accuracy is directly reflected in the various distinguishers that have been published in the literature. Taking prominent examples, non-profiled Correlation Power Analysis (CPA) usually takes advantage of an a-priori (e.g. Hamming weight) leakage model [3]. By contrast, profiled Template Attacks (TA) take advantage of an offline learning phase in order to estimate the leakage model [5]. But even in the latter case, the profiling method is frequently based on some assumptions on the leakage distribution (e.g. that the noise is Gaussian). Furthermore, the model estimation can also be bounded by practical constraints (e.g. in terms of number of measurements available in the learning phase). Following these observations, the question "*how good is my leakage model?*" has become a central one in the analysis of side-channel attacks. In other words, whenever trying to quantify the security of an implementation, the goal is to reflect the actual target - not the evaluators' assumptions. Therefore, the main challenge for the evaluator is to avoid being biased by an incorrect model, possibly leading to a false sense of security (i.e. an insecure cryptographic implementation that would look secure in front of one particular adversary).

More formally, the relation between accurate leakage models and fair security analyses is also central in evaluation frameworks such as proposed at Eurocrypt 2009 [18]. In particular, this previous work established that the leakage of an implementation (or the quality of a measurement setup) can be quantified by measuring the Mutual Information (MI) between the secret cryptographic keys manipulated by a device and the actual leakage produced by this device. Unfortunately, the design of unbiased and non-parametric estimators for the MI is a notoriously hard problem. Yet, since the goal of side-channel attacks is to use the "best available" models in order to recover information, a solution is to estimate the MI based on these models. This idea has been precised by Renauld et al. with the notion of Perceived Information (PI) - that is nothing else than an estimation of the MI biased by the side-channel adversary's model [15]. Intuitively, the MI captures the worst-case security level of an implementation, as it corresponds to an (hypothetical) adversary who can perfectly profile the leakage Probability Density Function (PDF). By contrast, the PI captures its practical counterpart, where actual estimation procedures are used to profile the PDF.

Our contribution. The previous formal tools provide a sound basis for discussing the evaluation question "*how good is my leakage model?*". The answer to this question actually corresponds to the difference between the MI and the PI. Nevertheless, we remain with the problem that the MI is generally unknown (just as the actual leakage PDF), which makes it impossible to compute this difference directly. Interestingly, we show in this paper that it is possible to perform sound(er) security analyses, where the approximations used by the side-channel evaluators are quantified, and their impact on security is kept under control.

In this context, we start with the preliminary observation that understanding these fair evaluation issues requires to clearly distinguish between estimation errors and assumption errors, leading to three main contributions. First, we show how cross–validation can be used in order to precisely gauge the convergence of an estimated model. Doing so, we put forward that certain evaluation metrics (e.g. Pearson's correlation or PI) are better suited for this purpose. Second, we propose a method for measuring assumption errors in side-channel attacks, taking advantage of the distance sampling technique introduced in [20]. We argue that it allows detecting imperfect hypotheses without any knowledge of the true leakage distribution[1]! Third, we combine these tools in order to determine the probability that a model error is due to estimation or assumption issues. We then discuss the (im)possibility to precisely (and generally) bound the resulting information loss. We also provide pragmatic guidelines for physical security evaluators. For illustration, we apply these contributions to actual measurements obtained from an AES implementation in an embedded microcontroller. As a result and for the first time, we are able to certify that the leakage of a chip (i.e. its worst-case security level) is close to the one we are able to extract.

[1] By contrast, the direct solution for quantifying the PI/MI distance would be to compute a statistical (e.g. Kullback-Leibler) distance between the adversary's model and the actual leakages. But it requires knowing the true leakage distribution.

These results have implications for the certification of any cryptographic product against side-channel attacks - as they provide solutions to guarantee that the evaluation made by laboratories is based in sound assumptions. They could also be used to improve the comparison of measurement setups such as envisioned by the DPA contest v3 [6]. Namely, this contest suggests comparing the quality of side-channel measurements with a CPA based on an a-priori leakage model. But this implies that the best traces are those that best comply with this a-priori, independent of their true informativeness. Using the PI to compare the setups would already allow each participant to choose his leakage assumptions. And using the cross–validation and distance sampling techniques described in this work would allow determining how relevant these assumptions are.

Notations. We use capital letters for random variables, small caps for their realizations, sans serif fonts for functions and calligraphic letters for sets.

2 Background

2.1 Measurement setups

Our experiments are based on measurements of an AES Furious implementation[2] run by an 8-bit Atmel AVR (AtMega 644p) microcontroller at a 20 MHz clock frequency. Since the goal of this paper is to analyze leakage informativeness and model imperfections, we compared traces from three different setups. First, we considered two types of "power-like" measurements. For this purpose, we monitored the voltage variations across both a 22 Ω resistor and a 2 μH inductance introduced in the supply circuit of our target chip. Second, we captured the electromagnetic radiation of our target implementation, using a Rohde & Schwarz (RS H 400-1) probe - with up to 3 GHz bandwidth - and a 20 dB low-noise amplifier. Measurements were taken without depackaging the chip, hence providing no localization capabilities. Acquisitions were performed using a Tektronix TDS 7104 oscilloscope running at 625 MHz and providing 8-bit samples. In practice, our evaluations focused on the leakage of the first AES master key byte (but would apply identically to any other enumerable target). Leakage traces were produced according to the following procedure. Let x and s be our target input plaintext byte and subkey, and $y = x \oplus s$. For each of the 256 values of y, we generated 1000 encryption traces, where the rest of the plaintext and key was random (i.e. we generated 256 000 traces in total, with plaintexts of the shape $p = x||r_1||\dots||r_{15}$, keys of the shape $k = s||r_{16}||\dots||r_{30}$, and the r_i's denoting uniformly random bytes). In order to reduce the memory cost of our evaluations, we only stored the leakage corresponding to the 2 first AES rounds (as the dependencies in our target byte $y = x \oplus s$ typically vanish after the first round, because of the strong diffusion properties of the AES). In the following, we will denote the 1000 encryption traces obtained from a plaintext p including the target byte x under a key k including the subkey s as: $\mathsf{AES}_{k_s}(p_x) \rightsquigarrow l_y^i$ (with

[2] Available at `http://point-at-infinity.org/avraes/`.

$i \in [1; 1000]$). Furthermore, we will refer to the traces produced with the resistor, inductance and EM probe as $l_y^{\mathtt{r},i}$, $l_y^{\mathtt{l},i}$ and $l_y^{\mathtt{em},i}$. Eventually, whenever accessing the points of these traces, we will use the notation $l_y^i(j)$ (with $j \in [1; 10\ 000]$, typically). These subscripts and superscripts will omitted when not necessary.

2.2 Evaluation metrics

In this subsection, we recall a few evaluation metrics that have been introduced in previous works on side-channel attacks and countermeasures.

Correlation coefficient (non-profiled). In view of the popularity of the CPA distinguisher in the literature, a natural candidate evaluation metric is Pearson's correlation coefficient. In a non-profiled setting, an a-priori (e.g. Hamming weight) model is used for computing the metric. The evaluator then estimates the correlation between his measured leakages and the modeled leakages of a target intermediate value. In our AES example and targeting an S-box output, it would lead to $\hat{\rho}(L_Y, \mathsf{model}(\mathsf{Sbox}(Y)))$, where the "hat" notation is used to denote the estimation of a statistic. In practice, this estimation is performed by sampling (i.e. measuring) N_t "test" traces from the leakage distribution L_Y. In the following, we will denote the set of these N_t test traces as $\mathcal{L}_Y^t$.

Correlation coefficient (profiled). In order to avoid possible biases due to an incorrect a-priori choice of leakage model, a natural solution is to extend the previous proposal to a profiled setting. In this case, the evaluator will start by building a model from N_p "profiling" traces. We denoted this step as $\hat{\mathsf{model}}_\rho \leftarrow \mathcal{L}_Y^p$ (with $\mathcal{L}_Y^p \perp\!\!\!\perp \mathcal{L}_Y^t$). In practice, it is easily obtained by computing the sample mean values of the leakage points corresponding to the target intermediate values.

Signal-to-Noise Ratio (SNR). Yet another solution put forward by Mangard is to compute the SNR of the measurements [13], defined as:

$$\hat{\mathsf{SNR}} = \frac{\hat{\mathsf{var}}_y(\hat{\mathsf{E}}_i(L_y^i))}{\hat{\mathsf{E}}_y(\hat{\mathsf{var}}_i(L_y^i))},$$

where $\hat{\mathsf{E}}$ and $\hat{\mathsf{var}}$ denote the sample mean and variance of the leakage variable, that are estimated from the N_t traces in $\mathcal{L}_Y^t$ (like the correlation coefficient).

Perceived information. Eventually, as mentioned in introduction the PI can be used for evaluating the leakage of a cryptographic implementation. Its sample definition (that is most useful in evaluations of actual devices) is given by:

$$\hat{\mathrm{PI}}(S; X, L) = \mathrm{H}[S] - \sum_{s \in \mathcal{S}} \Pr[s] \sum_{x \in \mathcal{X}} \Pr[x] \sum_{l_y^i \in \mathcal{L}_Y^t} \Pr_{\mathtt{chip}}[l_y^i | s, x]. \log_2 \hat{\Pr}_{\mathtt{model}}[s | x, l_y^i],$$

where $\hat{\Pr}_{\mathtt{model}} \leftarrow \mathcal{L}_Y^p$. As already observed in several works, the sum over s is redundant whenever the target operations used in the attack follows a group operation (which is typically the case of a block cipher key addition).

Under the assumption that the model is properly estimated, it is shown in [12] that the three latter metrics are essentially equivalent in the context of standard univariate side-channel attacks (i.e. exploiting a single leakage point $l_y^i(j)$ at a time). By contrast, only the PI naturally extends to multivariate attacks [19]. It can be interpreted as the amount of information leakage that will be exploited by an adversary using an estimated model. So just as the MI is a good predictor for the success rate of an ideal TA exploiting the perfect model $\Pr_{\mathtt{chip}}$, the PI is a good predictor for the success rate of an actual TA exploiting the "best available" model $\hat{\Pr}_{\mathtt{model}}$ obtained through the profiling of a target device.

2.3 PDF estimation methods

Computing metrics such as the PI defined in the previous section requires one to build a probabilistic leakage model $\hat{\Pr}_{\mathtt{model}}$ for the leakage behavior of the device. We now describe a few techniques that can be applied for this purpose.

Gaussian templates. The seminal TA in [5] relies on an approximation of the leakages using a set of normal distributions. That is, it assumes that each intermediate computation generates samples according to a Gaussian distribution. In our typical scenario where the targets follow a key addition, we consequently use: $\hat{\Pr}_{\mathtt{model}}[l_y|s,x] \approx \hat{\Pr}_{\mathtt{model}}[l_y|s \oplus x] \sim \mathcal{N}(\mu_y, \sigma_y^2)$. This approach simply requires estimating the sample means and variances for each value of $y = x \oplus s$ (and mean vectors / covariance matrices in case of multivariate attacks).

Regression-based models. To reduce the data complexity of the profiling, an alternative approach proposed by Schindler et al. is to exploit Linear Regression (LR) [16]. In this case, a stochastic model $\hat{\theta}(y)$ is used to approximate the leakage function and built from a linear basis $\mathsf{g}(y) = \{\mathsf{g}_0(y), ..., \mathsf{g}_{B-1}(y)\}$ chosen by the adversary/evaluator (usually $\mathsf{g}_i(y)$ are monomials in the bits of y). Evaluating $\hat{\theta(y)}$ boils down to estimating the coefficients α_i such that the vector $\hat{\theta}(y) = \sum_j \alpha_j \mathsf{g}_j(y)$ is a least-square approximation of the measured leakages L_y. In general, an interesting feature of such models is that they allow trading profiling efforts for online attack complexity, by adapting the basis $\mathsf{g}(y)$. That is, a simpler model with fewer parameters will converge for smaller values of N_p, but a more complex model can potentially approximate the real leakage function more accurately. Compared to Gaussian templates, another feature of this approach is that only a single variance (or covariance matrix) is estimated for capturing the noise (i.e. it relies on an assumption of homoscedastic errors).

Histograms and Kernels. See appendix A.

3 Estimation errors and cross–validation

Estimating the PI from a leaking implementation essentially holds in two steps. First, a model has to be estimated from a set of profiling traces $\mathcal{L}_Y^p$: $\hat{\Pr}_{\mathtt{model}} \leftarrow \mathcal{L}_Y^p$. Second, a set of test traces $\mathcal{L}_Y^t$ is used to estimate the perceived information, corresponding to *actual* leakage samples of the device (i.e. following the

true distribution $\Pr_{\mathtt{chip}}[l_y^i|s,x]$). As a result, two main model errors can arise. First, the number of traces in the profiling set may be too low to estimate the model properly. This corresponds to the estimation errors that we analyze in this section. Second, the model $\hat{\Pr}_{\mathtt{model}}$ may not be able to predict the distribution of samples in the test set, even after intensive profiling. This corresponds to the assumption errors that will be analyzed in the next section. In both cases, such model errors will be reflected by a divergence between the PI and MI.

In order to verify that estimations in a security evaluation are sufficiently accurate, the standard solution is to exploit cross–validation. In general, this technique allows gauging how well a predictive (here leakage) model performs in practice [10]. In the rest of the paper, we use 10-fold cross–validations for illustration (which is commonly used in the literature [9]). What this means is that the set of acquired traces $\mathcal{L}_Y$ is first split into ten (non overlapping) sets $\mathcal{L}_Y^{(i)}$ of approximately the same size. Let us define the profiling sets $\mathcal{L}_Y^{p,(j)} = \bigcup_{i \neq j} \mathcal{L}_Y^{(i)}$ and the test sets $\mathcal{L}_Y^{t,(j)} = \mathcal{L}_Y \setminus \mathcal{L}_Y^{p,(j)}$. The sample PI is then repeatedly computed ten times for $1 \leq j \leq 10$ as follows. First, we build a model from a profiling set: $\hat{\Pr}_{\mathtt{model}}^{(j)} \leftarrow \mathcal{L}_Y^{p,(j)}$. Then we estimate $\hat{\mathrm{PI}}^{(j)}(S;X,L)$ with the associated test set $\mathcal{L}_Y^{t,(j)}$. Cross–validation protects us from obtaining too large PI values due to over-fitting, since the test computations are always performed with an independent data set. Finally, the 10 outputs can be averaged to get an unbiased estimate, and their spread characterizes the accuracy of the result[3].

3.1 Experimental results

As a starting point, we represented illustrative traces corresponding to our three measurement setups in Appendix B, Figure 8, 9, 10. The figures further contain the SNRs and correlation coefficients of a CPA using Hamming weight leakage model and targeting the S-box output. While insufficient for fair security evaluations as stated below, these metrics are interesting preliminary steps, since they indicate the parts of the traces where useful information lies. In the following, we extract a number of illustrative figures from meaningful samples.

From a methodological point of view, the impact of cross–validation is best represented with the box plot of Figure 1: it contains the PI of point 2605 in the resistor-based traces, estimated with Gaussian templates and a stochastic model using a 17-element linear basis for the bits of the S-box input and output. This point is the most informative one in our experiments (across all measurements and estimation procedures we tried). Results show that the PI estimated with Gaussian templates is higher - hence suggesting that the basis used in our regression-based profiling was not fully reflective of the chip activity for this

[3] Cross–validation can also apply to profiled CPA, by building models $\hat{\mathsf{model}}_\rho \leftarrow \mathcal{L}_Y^{p,(j)}$, and testing them with the remaining $\mathcal{L}_Y^{t,(j)}$ traces. By contrast, it does not apply to the SNR for which the computation does not include an a posteriori testing phase. We focus on the PI because of its possible extension to multivariate statistics.

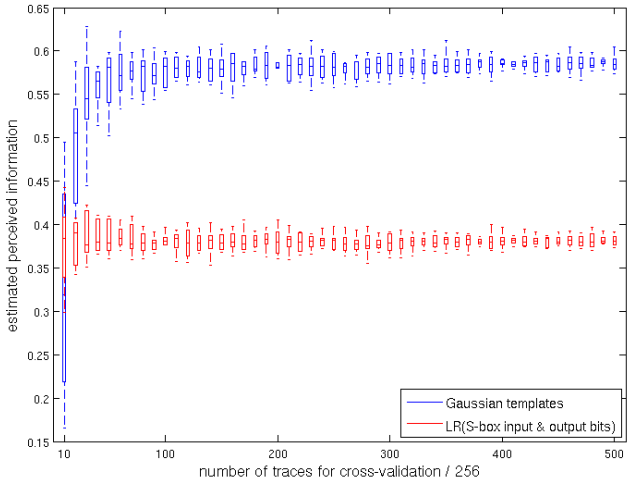

Fig. 1. Perceived information estimated from Gaussian templates and LR-based models, with cross–validation (target point 2605 from the resistor-based measurements).

sample. More importantly, we observe that the estimation converges quickly (as the spread of our 10 PI estimates decreases quickly with the number of traces). As expected, this convergence is faster for regression-based profiling, reflecting the smaller number of parameters to estimate in this case. Note that we also performed this cross–validation for the Kernel-based PDF estimation described in Appendix A (see Appendix B, Figure 11 for the results). Both the expected value of the PI and its spread suggest that these two density estimation techniques provide equally satisfying results in our implementation context.

A natural next step is to analyze the quantity of information given by alternative leakage points. An example is given in Figure 2 (where we only plot the expected value of the PI). The left part of the figure corresponds exactly to the most informative point of Figure 1. The right part of the figure is computed with a later sample (time 4978) that (we assumed) corresponds to the computation of the S-box output. Interestingly, we observe that while this second point is less informative, it is more accurately explained by a stochastic model using the S-box output bits as a basis, hence confirming our expectations. Eventually, we also investigated the additional information gathered when performing multivariate attacks in Appendix B, Figure 12. For this purpose, we considered both a couple of points (2605 and 4978) coming from the same setup in the left part of the figure, and a single point (2605) coming from two different setups in the right part of the figure. This experiment clearly suggests that combining information from different operations leads to more PI than combining information from different setups. It naturally fits with the intuition that two different block cipher operations (corresponding to different intermediate values) lead to more information leakage (i.e. less correlation) than the same operation mea-

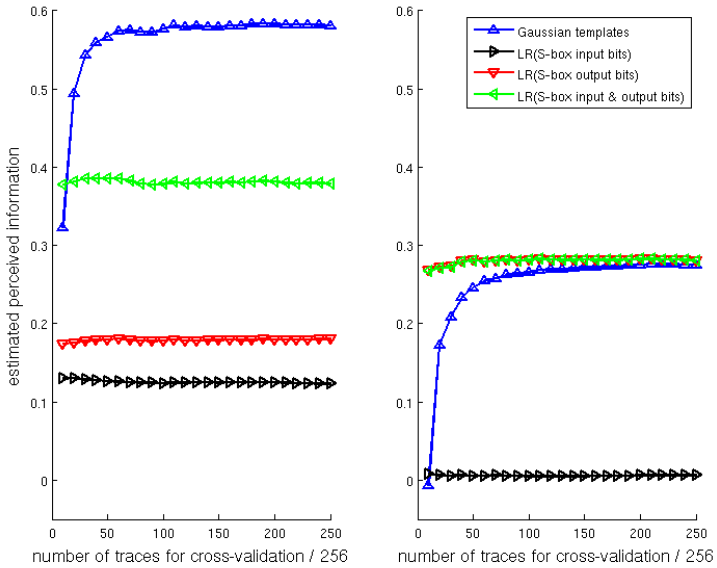

Fig. 2. PI for different PDF estimation techniques and two leakage (resistor-based) points. Left: most informative one (2605), right: other point of interest (4978).

sured with two different (yet similar) measurement setups. Many variations of such evaluations are possible (for more samples, estimation procedures, ...). For simplicity, we will limit our discussion to the previous examples, and use them to further discuss the critical question of assumption errors in the next section.

4 Assumption errors and distance sampling

Looking at Figures 1 and 2, we can conclude that our estimation of the PI is reasonably accurate and that Gaussian templates are able to extract a given amount of information from the measurements. Nevertheless, such pictures still do not provide any clue about the closeness between our estimated PI and the (true, unknown) MI. As previously mentioned in introduction, evaluating the deviation between the PI and MI is generally hard. In theory, the standard approach for evaluating such a deviation would be to compute a statistical (e.g. Kullback-Leibler) distance $\hat{\mathrm{D}}_{KL}(\hat{\Pr}_{\mathtt{model}}, \Pr_{\mathtt{chip}})$. But this requires knowing the (unknown) distribution $\Pr_{\mathtt{chip}}$, leading to an obvious chicken and egg problem.

Since standard probabilistic distances cannot be computed, an alternative solution that we will apply is to confront the test samples output by the device with estimated samples produced with the evaluator's model. In order to check their coherence, we essentially need a goodness-of-fit test. While several such tests exist in the literature for unidimensional distributions (e.g. Kolmogorov–

Smirnov [4] or Cramér–von–Mises [1]), much fewer solutions exist that generalize to multivariate statistics. Since we additionally need a test that applies to any distribution, possibly dealing with correlated leakage points, a natural proposal is to exploit statistics based on spacings (or interpoint distance) [14]. The basic idea of such a test is to reduce the dimensionality of the problem by comparing the distributions of distances between pairs of points, consequently simplifying it into a one-dimensional goodness-of-fit test again. It exploits the fact that two multidimensional distributions $\mathcal{F}$ and $\mathcal{G}$ are equal if and only if the variables $\mathbf{X} \sim \mathcal{F}$ and $\mathbf{Y} \sim \mathcal{G}$ generate identical distributions for the distances $\mathsf{D}(\mathbf{X_1}, \mathbf{X_2})$, $\mathsf{D}(\mathbf{Y_1}, \mathbf{Y_2})$ and $\mathsf{D}(\mathbf{X_3}, \mathbf{Y_3})$ [2, 11]. In our evaluation context, we can simply check if the distance between pairs of simulated samples (generated with a profiled model) and the distance between simulated and actual samples behave differently. If the model estimated during the profiling phase of a side-channel attack is accurate, then the distance distributions should be close. Otherwise, there will be a discrepancy that the test will be able to detect, as we now detail.

The first step of our test for the detection of incorrect assumptions is to compute the simulated distance cumulative distribution as follows:

$$f_{\mathrm{sim}}(d, s, x) = \Pr\left[L_y^1 - L_y^2 \leq d \,\middle|\, L_y^1, L_y^2 \sim \hat{\Pr}_{\mathtt{model}}[L_y|s, x]\right].$$

Since the evaluator has an analytical expression for $\hat{\Pr}_{\mathtt{model}}$, this cumulative distribution is easily obtained. Next, we compute the sampled distance cumulative distribution from the test sample set $\mathcal{L}_Y^t$ as follows:

$$\hat{g}_{N_t}(d, s, x) = \Pr\left[l_y^i - l_y^j \leq d \,\middle|\, \left\{l_y^i\right\}_{1 \leq i \leq N_t} \sim \hat{\Pr}_{\mathtt{model}}[L_y|s, x], \left\{l_y^j\right\}_{1 \leq j \leq N_t} = \mathcal{L}_Y^t\right].$$

Eventually, we need to detect how similar f_{sim} and g_{N_t} are, which is made easy since these cumulative distributions are now univariate. Hence, we can compute the distance between them by estimating the Cramér–von–Mises divergence:

$$\hat{\mathrm{CvM}}(f_{\mathrm{sim}}, \hat{g}_{N_t}) = \int_{-\infty}^{\infty} \left[f_{\mathrm{sim}}(x) - \hat{g}_{N_t}(x)\right]^2 dx.$$

As the number of samples in the estimation increases, this divergence should gradually tend towards zero provided the model assumptions are correct.

4.1 Experimental results

As in the previous section, we applied cross–validation in order to compute the Cramér–von–Mises divergence between the distance distributions. That is, for each of the 256 target intermediate values, we generated 10 different estimates $\hat{g}_{N_t}^{(j)}(d, s, x)$ and computed $\hat{\mathrm{CvM}}^{(j)}(f_{\mathrm{sim}}, \hat{g}_{N_t})$ from them. An exemplary evaluation is given in Figure 3 for the same leakage point and estimation methods as in Figure 1. For simplicity, we plotted a picture containing the 256 (average) estimates at once[4]. It shows that Gaussian templates better converge

[4] It is also possible to investigate the quality of the model for any given $y = x \oplus s$.

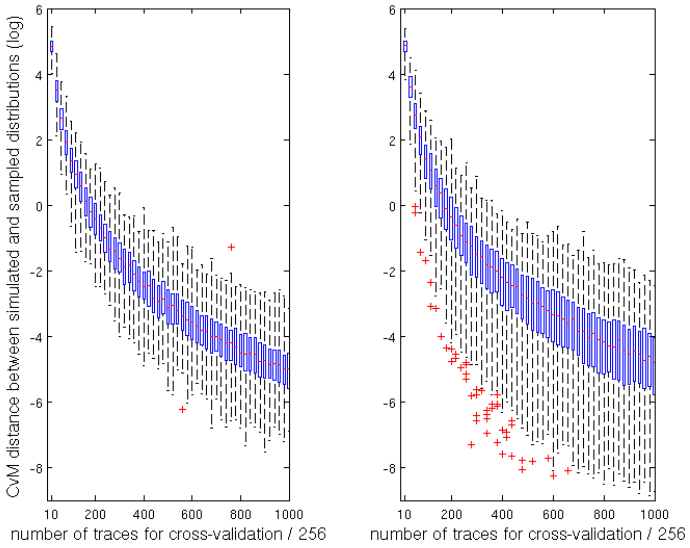

Fig. 3. Cramér–von–Mises divergence between simulated and sampled distributions, with cross–validation (target point 2605 from the resistor-based measurements). Left: Gaussian templates, right: LR-based estimation (S-box input and output bits).

towards a small divergence of the distance distributions. It is also noticeable that regression-based models lead to more outliers, corresponding to values y for which the leakage L_y is better approximated. Figure 4 additionally provides the quantiles of the Cramér–von–Mises divergence for both univariate and bivariate distributions (i.e. corresponding to the PIs in Appendix B, Figure 12). Interestingly, we observe that the better accuracy of Gaussian templates compared to regression-based models decreases when considering the second leakage point. This perfectly fits the intuition that we add a dimension that is better explained by a linear basis (as it corresponds to the right point in Figure 2). Note that any incorrect assumption would eventually lead the CvM divergence to saturate.

5 Estimation vs. assumption errors

From an evaluator's point of view, assumption errors are naturally the most damaging (since estimation errors can be made arbitrarily small by measuring more). In this respect, an important problem that we answer in this section is to determine whether a model error comes from estimation or assumption issues. For this purpose, the first statistic we need to evaluate is the *sampled* simulated distance cumulative distribution (for a given number of test traces N_t). This is the estimated counterpart of the distribution f_{sim} defined in Section 4:

$$\hat{f}_{\mathrm{sim}}^{N_t}(d,s,x) = \Pr\left[l_y^i - l_y^j \leq d \,\middle|\, \left\{l_y^i, l_y^j\right\}_{1\leq i\neq j\leq N_t} \sim \hat{\Pr}_{\mathtt{model}}[L_y|s,x]\right].$$

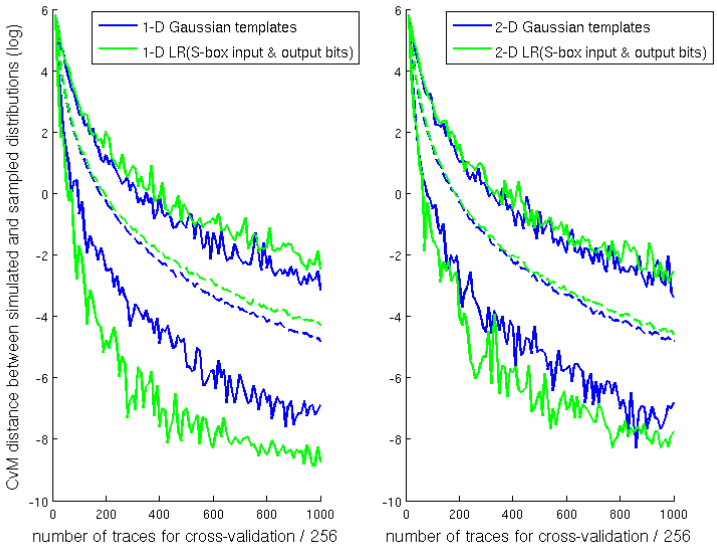

Fig. 4. Median, min and max of the CvM divergence btw. simulated and sampled distributions for Gaussian templates and LR-based models (resistor-based measurements). Left: univariate attack (sample 2605), right: bivariate attack (samples 2605 and 4978).

From this definition, our main interest is to know, for a given divergence between f_{sim} and $\hat{f}_{\text{sim}}^{N_t}$, what is the probability that this divergence would be observed for the chosen amount of test traces N_t. This probability is directly given by the following cumulative divergence distribution:

$$\hat{\text{Div}}_{N_t}(x) = \Pr\left[\hat{\text{CvM}}(f_{\text{sim}}, \hat{f}_{\text{sim}}^{N_t}) \leq x\right].$$

How to exploit this distribution is then illustrated in Figure 5. For each model $\hat{\Pr}_{\text{model}}^{(j)}$ estimated during cross–validation, we build the corresponding $\hat{\text{Div}}_{N_t}^{(j)}$'s (i.e. the cumulative distributions in the figure). The cross–validation additionally provides (for each cumulative distribution) a value for $\hat{\text{CvM}}^{(j)}(f_{\text{sim}}, \hat{g}_{N_t})$ estimated from the actual leakage samples in the test set: they correspond to the small circles below the X axis in the figure. Eventually, we just derive:

$$\hat{\text{Div}}_{N_t}^{(j)}\left(\hat{\text{CvM}}^{(j)}(f_{\text{sim}}, \hat{g}_{N_t})\right).$$

Computing this statistic is simply obtained by projecting the circles towards the Y axis in the figure. Large values indicate that there is a small probability that the observed samples follow the simulated distributions. More precisely, they correspond to large p-values when testing the hypothesis that the estimated model is incorrect. Thanks to cross–validation, we can obtain 10 such values,

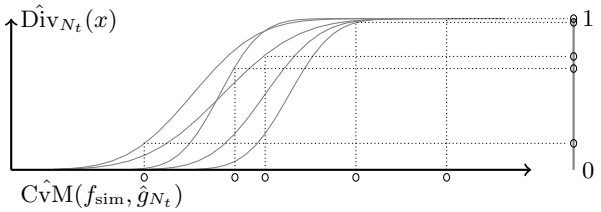

Fig. 5. Model divergence estimation.

leading to answers laid on a $[0; 1]$ interval, indicating the accuracy of each estimated model. Values grouped towards the top of the interval indicate that the assumptions used to estimate these models are likely incorrect.

An illustration of this method is given in Figure 6 for different Gaussian templates and regression-based profiling efforts, in function of the number of traces in the cross–validation set. It clearly exhibits that as this number of traces increases (hence, the estimation errors decrease), the regression approach suffers from assumption errors with high probability. Actually, the intermediate values for which these errors occur first are the ones already detected in the previous section, for which the leakage variable L_y cannot be precisely approximated given our choice of basis. By contrast, no such errors are detected for the Gaussian templates (up to the amount of traces measured in our experiments). This process can be further systematized to all intermediate values, as in Figure 7. It allows an evaluator to determine the number of measurements necessary for the assumption errors to become significant in front of estimation ones.

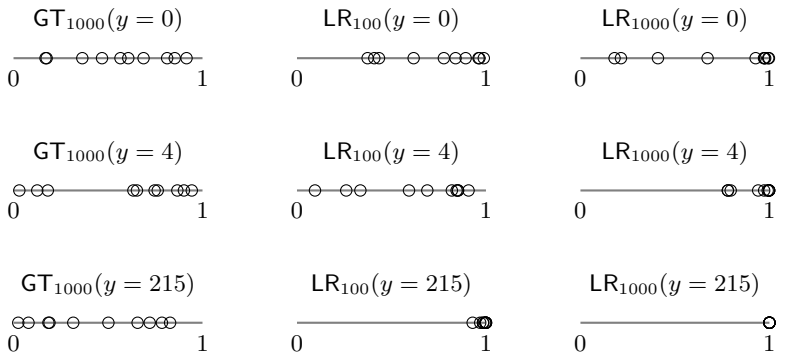

Fig. 6. Probability of assumption errors (p-values) for Gaussian templates (GT) and regression-based models (LR) corresponding to different target intermediate values y, in function of N_t (in subscript). Resistor-based measurements, sample 2605.

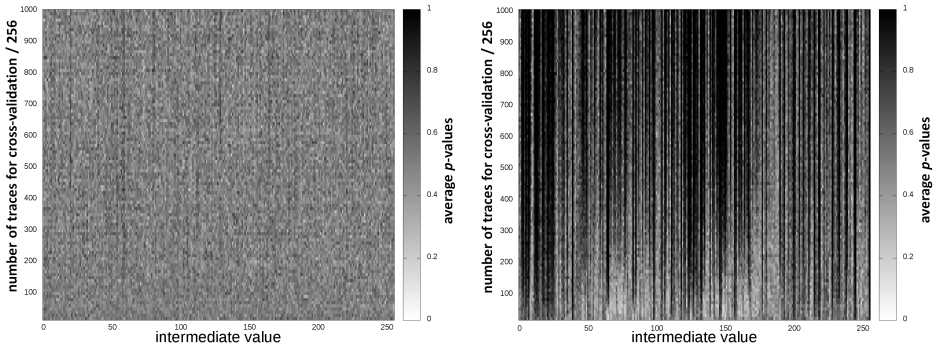

Fig. 7. Probability of assumption errors for Gaussian templates (left) and regression-based models with a 17-element basis (right) corresponding to all the target intermediate values y, in function of N_t. Resistor-based measurements, sample 2605.

6 Pragmatic evaluation guidelines & conclusions

Interestingly, most assumptions will eventually be detected as incorrect when the number of traces in a side-channel evaluation increases[5]. As detailed in introduction, it directly raises the question whether the information loss due to such assumption errors can be bounded? Intuitively, the "threshold" value for which they are detected by our test provides a measure of their "amplitude" (since errors that are detected earlier should be larger in some sense). In the long version of this paper [7], we discuss whether this intuition can be exploited quantitatively and answer negatively. In this section, we conclude by arguing that our results still lead to qualitatively interesting outcomes, and describe how they can be exploited in the fair evaluation of side-channel attacks.

In this respect, first note that the maximum number of measurements in an evaluation is usually determined by practical constraints (i.e. how much time is allowed for the evaluation). Given this limit, estimation and assumption errors can be analyzed separately, leading to quantified results such as in Figures 1 and 3. These steps allow ensuring that the statistical evaluation has converged. Next, one should always test the hypothesis that the leakage model is incorrect, as described in Section 5. Depending on whether assumption errors are detected "early" or "late", the evaluator should be able to decide whether more refined PDF estimation techniques should be incorporated in his analyses. As discussed in [7], Section 6, the precise definition of "early" and "late" is hard to formalize in terms of information loss. Yet, later is always better and such a process will at least guarantee that if no such errors are detected *given some measurement*

[5] Non-parametric PDF estimation methods (e.g. as described in Appendix A) could be viewed as an exception to this fact, assuming that the sets of profiling traces $\mathcal{L}_Y^p$ and test traces $\mathcal{L}_Y^t$ come from the same distribution. Yet, this assumption may turn out to be contradicted in practice because of technological mismatches [8, 15], in which case the detection of assumption errors remains critical even with such tools.

capabilities, an improved model will not lead to significantly improved attacks (since the evaluator will essentially not be able to distinguish the models with this amount of measurements). That is, the proposed methodology can provide an answer to the pragmatic question: "for an amount of measurements performed by a laboratory, is it worth spending time to refine the leakage model exploited in the evaluation?". In other words, it can be used to guarantee that the security level suggested by a side-channel analysis is close to the worst-case, and this guarantee is indeed conditional to number of measurement available for this purpose.

Acknowledgements. This work has been funded in parts by the ERC project 280141 (acronym CRASH), the Walloon Region MIPSs project and the 7th framework European project PREMISE. François-Xavier Standaert is an associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.).

References

1. T. W. Anderson. On the distribution of the two-sample Cramer-von Mises criterion. *The Annals of Mathematical Statistics*, 33(3):1148–1159, 1962.
2. Robert Bartoszynski, Dennis K. Pearl, and John Lawrence. A multidimensional goodness-of-fit test based on interpoint distances. *Journal of the American Statistical Association*, 92(438):pp. 577–586, 1997.
3. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, 2004.
4. Chakravarti, Laha, and Roy. *Handbook of methods of applied statistics*, Volume I, John Wiley and Sons, pp. 392-394, 1967.
5. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, 2002.
6. DPA Contest. http://www.dpacontest.org/v3/index.php, 2012.
7. Franois Durvaux, Franois-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? Cryptology ePrint Archive, Report 2013/706, 2013. `http://eprint.iacr.org/`.
8. M. Abdelaziz Elaabid and Sylvain Guilley. Portability of templates. *J. Cryptographic Engineering*, 2(1):63–74, 2012.
9. S. Geisser. *Predictive inference*, volume 55. Chapman & Hall/CRC, 1993.
10. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
11. Jen-Fue Maa, Dennis K. Pearl, and Robert Bartoszynski. Reducing multidimensional two-sample data to one-dimensional interpoint comparisons. *The Annals of Statistics*, 24(3):pp. 1069–1074, 1996.
12. S. Mangard, E. Oswald, and F.-X. Standaert. One for all – all for one: Unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.
13. Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *LNCS*, pages 222–235. Springer, 2004.

14. Ronald Pyke. Spacings revisited. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability (Univ. California, Berkeley, Calif., 1970/1971), Vol. I: Theory of statistics*, pages 417–427, Berkeley, Calif., 1972. Univ. California Press.
15. 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, *EUROCRYPT*, volume 6632 of *LNCS*, pages 109–128. Springer, 2011.
16. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, 2005.
17. B.W. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Taylor & Francis, 1986.
18. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, 2009.
19. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *LNCS*, pages 112–129. Springer, 2010.
20. Nicolas Veyrat-Charvillon and François-Xavier Standaert. Generic side-channel distinguishers: Improvements and limitations. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *LNCS*, pages 354–372. Springer, 2011.

A Histograms and kernels

The estimation methods of Section 2.3 make the assumption that the non-deterministic part of the leakage behaves according to a normal distribution. This may not always be correct, in which case one needs to use other techniques. For illustration, we considered two non-parametric solutions for density estimation, namely histograms and kernels. These allow one to finely characterize the non-deterministic part of the leakage. First, *histogram* estimation performs a partition of the samples by grouping them into bins. More precisely, each bin contains the samples of which the value falls into a certain range. The respective ranges of the bins have equal width and form a partition of the range between the extreme values of the samples. Using this method, one approximates a probability by dividing the number of samples that fall within a bin by the total number of samples. The optimal choice for the bin width h is an issue in statistical theory, as different bin sizes can have great impact on the estimation. In our case, we were able to tune this bin width according to the sensitivity of the oscilloscope. Second, *kernel* density estimation is a generalization of histograms. Instead of bundling samples together in bins, it adds (for each observed sample) a small kernel centered on the value of the leakage to the estimated PDF. The resulting estimation is a sum of small "bumps" that is much smoother than the corresponding histogram, which can be desirable when estimating a continuous distribution. In such cases it usually provides faster convergence towards the true distribution. Similarly to histograms, the most important parameter is the bandwidth h. In our case, we used the modified rule of thumb estimator in [17].

B Additional figures

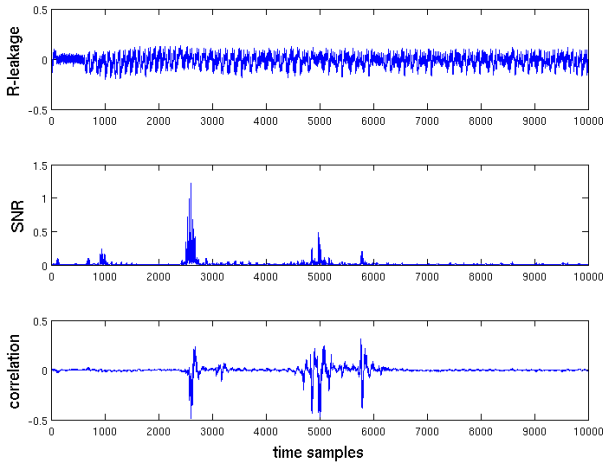

Fig. 8. Resistor-based measurements.

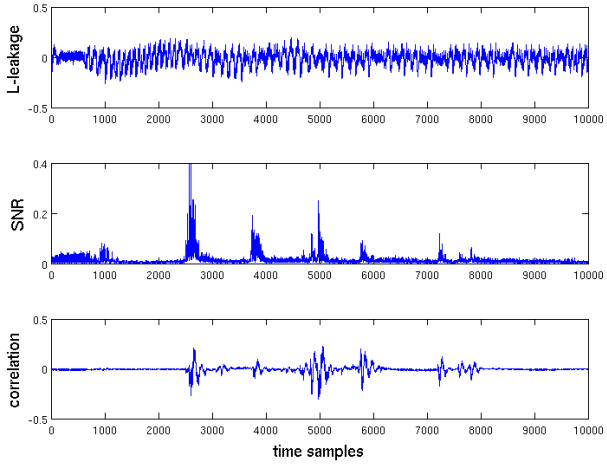

Fig. 9. Inductance-based measurements.

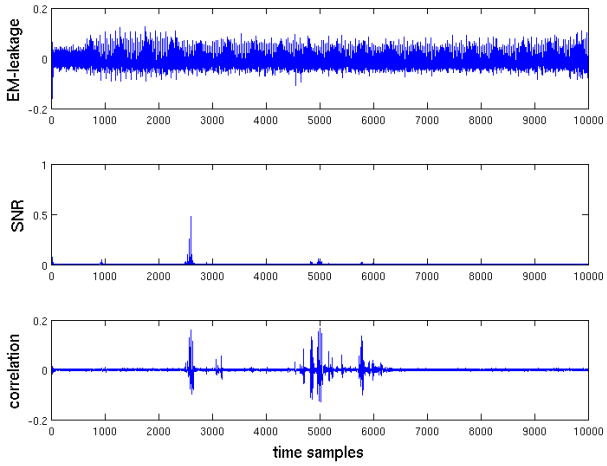

Fig. 10. Electromagnetic measurements.

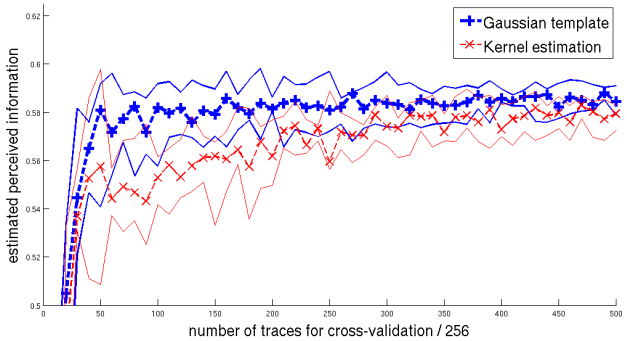

Fig. 11. Perceived information quantiles estimated from Gaussian templates and Kernels, with cross–validation (target point 2605 from the resistor-based measurements).

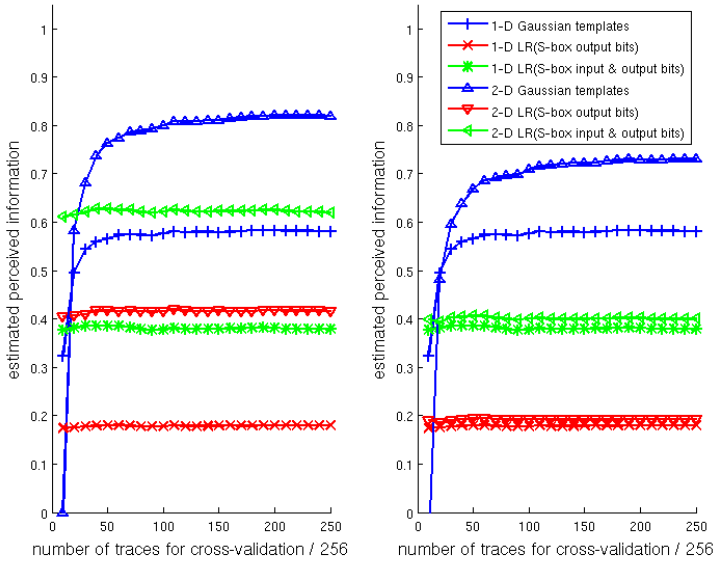

Fig. 12. PI for univariate and multivariate leakage models. Left: two points (2605, 4978) coming from the resistor-based measurements. Right: multi-channel attack exploiting the same point (2605) from resistor- and inductance-based measurements.

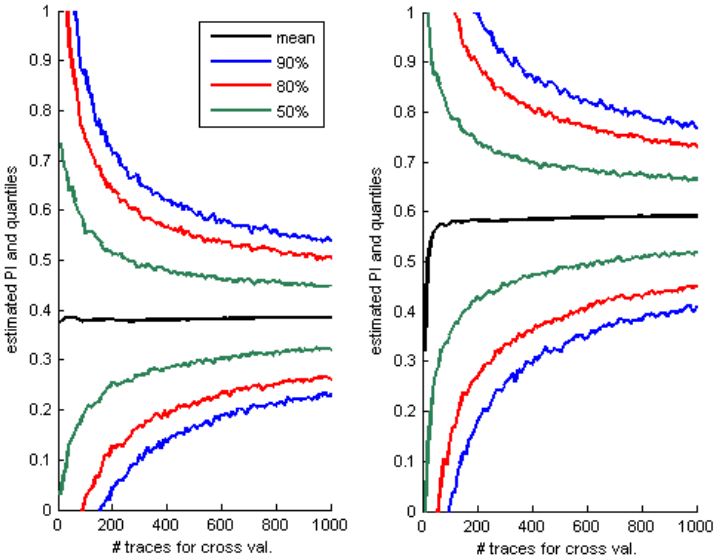

Fig. 13. Resistor-based measurements, sample 2605. Quantiles for the PI estimates obtained from the LR-based profiling (left) and Gaussian templates in Figure 1.