Towards a Better Understanding of Side-Channel Analysis Measurements Setups

Davide Bellizia, Balazs Udvarhelyi, and François-Xavier Standaert

UCLouvain, ICTEAM/ELEN/Crypto Group

Abstract. The evaluation of side-channel measurement setups and the impact they can have on physical security evaluations is a surprisingly under-discussed topic. In this paper, we initiate a comprehensive study of such setups for embedded software and hardware (FPGA) implementations. We systematically investigate a design space including the choice of the probing method, the clock frequency of the device under test, its supply voltage and the sampling rate of the adversary's oscilloscope. Our results quantify the impact (i.e., the risk of security over-estimations) that suboptimal setups can cause and lead to easy-to-use guidelines for security evaluators. Despite some of our conclusions are device-dependent, we argue that the proposed methodology and some of the proposed guidelines are of general interest and could be applied to other setups.

Keywords: Probing Techniques · Frequency and Voltage Scaling · Sampling Rate · Signal-to-Noise Ratio · Perceived Information

1 Introduction

The design of a measurement setup is the first step in the evaluation of a cryptographic implementation against side-channel analysis. Due to its physical nature, this step inherently carries hard to quantify risks of security overstatements. Noisy setups may indeed lead evaluators to conclude that the measurements are less informative than they actually are, and this gap will then be increased in case a countermeasure aiming at noise amplification, like masking [7,13] or shuffling [15,29], is implemented. Surprisingly, and despite papers focused on practical side-channel attacks usually describe how they optimized their setups, especially when targeting challenging real-world devices [21,3], very few works are dedicated to the systematic evaluation of measurement setups and the impact of their optimization on security evaluations. Besides, and to the best of our knowledge, the most advanced (published) investigations of this topic were performed in specific settings such as the exploitation of static leakages, as recently investigated by Moos et al. [19], or the evaluation of physical effects such as couplings to reduce a masked implementation's security order [10,11,16]. But when it comes to the the impact of measurement setups on the noise level in the context of (standard) attacks exploiting the dynamic part of the leakage, the only works we are aware of are the one of Guilley et al. which puts forward the Signal-to-Noise Ratio (SNR) as a meaningful metric to quantify the quality of side-channel acquisitions [14], and the one of Merino del Pozo and Standaert that discusses the impact of different setups in the context of leakage detection [22]. In this respect, and despite these references are important first steps in specifying relevant comparison metrics and highlighting the existence of an interesting design space, they are still have a limited scope: [14] estimates its proposed (univariate) metric for a single measurement setup while [22] compares different analog amplifiers and filters for a single probing method.

Recognizing that the design space of measurement setups is broader than investigated in these previous works, this paper aims at analyzing four important parameters of actual measurement setups. Namely, our goal is to discuss and evaluate the impact of the probing method used in the setups, the clock frequency of the Device Under Test (DUT), its supply voltage and the sampling rate of the oscilloscope used to collect the measurements. We therefore study these parameters systematically for two DUTs: a software (ARM Cortex) target and a hardware (Xilinx FPGA) one. We additionally evaluate the effect of these different parameters for both univariate evaluation metrics like the SNR and multivariate evaluation metrics like the Perceived Information (PI).

We then use our investigations to extract useful observations regarding how to select the parameters of our design space. While most of these observations are admittedly present (implicitly or explicitly) in former experimental works, we hope their compilation for two different devices and the quantitative analysis of the losses a poor measurement setup may imply for security evaluators (which may reach orders of magnitude) make a useful consolidating effort.

2 Background

We will use Mangard's SNR [17] to evaluate the quality of first-order and univariate leakages, as suggested by Guilley et al., and the PI metric analyzed in [6] to evaluate the quality of higher-order or multivariate leakages. For the latter we profile Gaussian templates in a linear subspace. We next recall these different evaluation metrics and detail the profiling tools used in our analyzes.

2.1 Mangard's SNR

Introduced in the context of side-channel analysis by Mangard, the SNR intuitively captures the data-dependent signal as the variance of the mean traces and the noise as the mean of the variance traces, for each time sample [17]. As a result, for a target intermediate variable y, it is defined as the ratio:

$$\hat{\mathrm{SNR}} = \frac{\hat{\mathsf{Var}}_y\left(\hat{\mathsf{E}}_i\left(l_i^y\right)\right)}{\hat{\mathsf{E}}_y\left(\hat{\mathsf{Var}}_i\left(l_i^y\right)\right)}\,, \tag{1}$$

where $\hat{\mathsf{Var}}$ and $\hat{\mathsf{E}}$ are the sample variance and the sample mean estimated on $l_i^y \in \mathcal{L}$, which represents the i-th side-channel observation generated by a target variable y. It must be pointed out that the noise in Mangard's definition is the result of two contributions. First, physical noise is due to physical phenomena (e.g., thermal noise, flicker noise) and electrical conditions (e.g., impedance mismatch, unwanted coupling with unrelated equipment). Second, algorithmic noise is due to the presence of operations that are independent of the target ones and are processed in parallel to them (i.e., at the same time). As argued by Guilley et al., it is a good metric for assessing the quality of side-channel measurements to be exploited by first-order univariate attacks [14], since it can be related to the the complexity of popular attacks such as the Correlation Power Analysis (CPA) and (univariate Gaussian) Template Attacks (TA) [5,8,18].

2.2 Subspace based Gaussian templates

Gaussian template attacks are a standard method to exploit multivariate leakages [8]. We combine them with a dimensionality reduction step in order to reduce the possibly high number of informative dimensions d of the leakage traces to a lower value $d' < d$. The profiling consists of an estimation, using n leakage traces $\boldsymbol{l}$, of the parameters $\boldsymbol{\mu_x}$, $\boldsymbol{\Sigma_x}$ and $\boldsymbol{W}$ of a Probability Density Function (PDF) of the form:

$$\tilde{\mathrm{m}}_{\mathrm{n}}(\boldsymbol{l}|x) = \frac{1}{\sqrt{(2\pi)^{d'} \cdot |\boldsymbol{\Sigma_x}|}} \cdot \exp^{\frac{1}{2}(\boldsymbol{W}\boldsymbol{l}-\boldsymbol{\mu}_x)\boldsymbol{\Sigma_x}(\boldsymbol{W}\boldsymbol{l}-\boldsymbol{\mu}_x)'} \,, \tag{2}$$

where x is the value of the profiled variable, $\boldsymbol{\mu_x}$ the mean vector of length d', $\boldsymbol{\Sigma_x}$ the covariance matrix of size $d' \times d'$ and $\boldsymbol{W}$ is the projection matrix of size $d' \times d$. This projection matrix is determined thanks to Linear Discriminant Analysis (LDA)[25]. LDA aims to find the subspace that maximizes the inter-class variance (i.e., the signal of Mangard's SNR) and minimizes the intra-class variance (i.e., the noise of Mangard's SNR). In practice, we applied this dimensionality reduction to all the samples with sufficient SNR (which d ranging from 30 to 500 depending on the cases) and usually kept a dozen dimensions for d'. Next, in the online attack phase, the likelihood of x is obtained by applying Bayes' law to the leakage models estimated beforehand such that:

$$\tilde{\mathrm{m}}_{\mathrm{n}}(x|\boldsymbol{l}) = \frac{\tilde{\mathrm{m}}_{\mathrm{n}}(\boldsymbol{l}|x)}{\sum_{x^* \in \mathcal{X}} \tilde{\mathrm{m}}_{\mathrm{n}}(\boldsymbol{l}|x^*)} \,. \tag{3}$$

The estimated PDF and the likelihood of the profiled variable can then be used to calculate the amount of information contained in the leakages.

2.3 Information theoretic metrics & bounds

For higher-order or multivariate attacks, the SNR metric is not directly applicable and a more general information theoretic metric has to be used. In the context of side-channel attacks, the Mutual Information (MI) is the most frequently considered candidate [26]. It generalizes the SNR in the sense that it can be related to the the complexity of worst-case higher-order & multivariate attacks [12,9] (and it is essentially equivalent to the SNR in the first-order univariate case [18]). However, as recently discussed in [6], estimating the MI is in general a hard problem. Known estimators are biased and distribution-dependent. Perfect estimations would therefore require the exact knowledge of the leakage distribution. As a workaround, they proposed the use of the previously introduced PI metric, which represents the amount of information that can be extracted from a device thanks to an the adversary's model, possibly biased due to estimation and assumption errors. For a target secret variable X with leakage variable $\boldsymbol{L}$, and denoting the leakage model $\tilde{\mathrm{m}}_n(\boldsymbol{l}|x)$ as described in the previous section, the PI is expressed as:

$$\hat{\mathrm{PI}}_n(X;\boldsymbol{L}) = \mathrm{H}(K) + \sum_{x\in\mathcal{X}} \mathsf{p}(x) \sum_{\boldsymbol{l}\in\mathcal{L}} \mathsf{p}(\boldsymbol{l}|x)\cdot \log_2(\tilde{m}_n(x|\boldsymbol{l})), \tag{4}$$

with $\mathrm{H}(X)$ the Shannon entropy of the variable $S \in \mathcal{S}$. The PI is a lower bound to the worst-case MI and equality holds in case the adversary's model is perfect. It can be viewed as the amount of information extrated by the best practical attack tried by an evaluator. Concretely, the PI is usually estimated with k-fold cross-validation and we used $k = 10$ in our following experiments.

3 Setup model & design space

We now introduce our model and design space for measurement setups, alongside with the two devices we have adopted to conduct our investigations.

3.1 Setup model

The setup model is illustrated in Figure 1. Its goal is to highlight important parameters for the informativeness of the leakages such as the probing method, the DUT's parameters and the Digital Storage Oscilloscope (DSO)'s parameters. As reported in [27], the choice of those components and how they interact with each other impact sensibly on the final outcome of the practical side-channel security evaluation of a leaking implementation. A bit more precisely, the current absorbed by the DUT is first monitored by a probe, which has the role to convert the current signal into a voltage signal. This signal can then be amplified using a preamplifier stage in order to increase its magnitude, to mitigate noise in the measurements and to improve electrical characteristics for the following blocks. At the end of the so-called measurement chain, a DSO samples and quantizes the analog voltage signal, converting it in a digital representation. Usually, the sampling operation is handled following a specific timing, that exploits a trigger signal in order to synchronize different measurements.[1] The precise design space that we will consider for each block of the model will be detailed later.

[1] The availability of a good trigger may raise additional challenges [4].

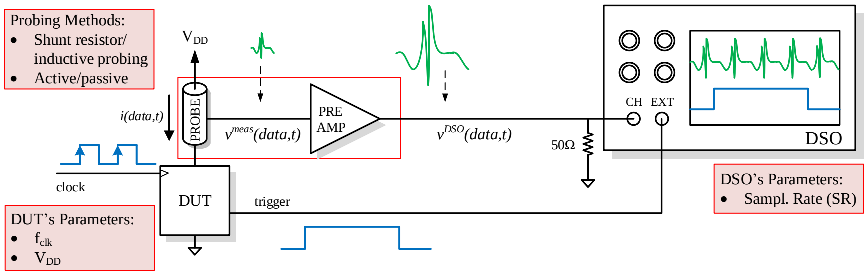


Fig. 1: Measurement setup model for power analysis evaluation.

We note that our investigations do not consider the question of filtering, which we view as an orthogonal one, since it can be performed after the measurements took place in order to compensate a too noisy setup.

3.2 Platforms

In our investigations, we have used two devices in order to cover both hardware and software implementations of cryptographic algorithms. This choice is motivated by the expected differences between the two types of targets. For example, hardware implementations generally allow better controlling the design aspects (from the level of parallelism to low-level implementation choices) while software implementations are usually more general purpose and serial.

Hardware DUT. Our target hardware DUT is a Xilinx Spartan-6 LX75 FPGA, mounted on a Sakura-G board, implementing an AES-128 processor with a 32-bit architecture. It is illustrated in Figure 2. In order to provide synchronization between measurements, we generate a trigger signal on one of the IO pins of the FPGA, rising to logical '1' one cycle before the starting of the encryption and set back to '0' one cycle after the end of the AES encryption. We used the integrated measurement point for our measurements.

Software DUT. Our target software implementation is running on a Cortex-M0 MCU from the STM32F0308 Discovery board. Small modifications were performed on the board. Namely, we added a crystal oscillator to provide a stable clock source for the measurements and decoupling capacitors were desoldered. The MCU is running tiny-AES [2], an open source AES-128 implementation. We used the same trigger methodology as for the hardware DUT. Our measurements were performed on the dedicated current measuring point for the MCU.

3.3 Design space

We explored our design space and DUTs by testing the following parameters:

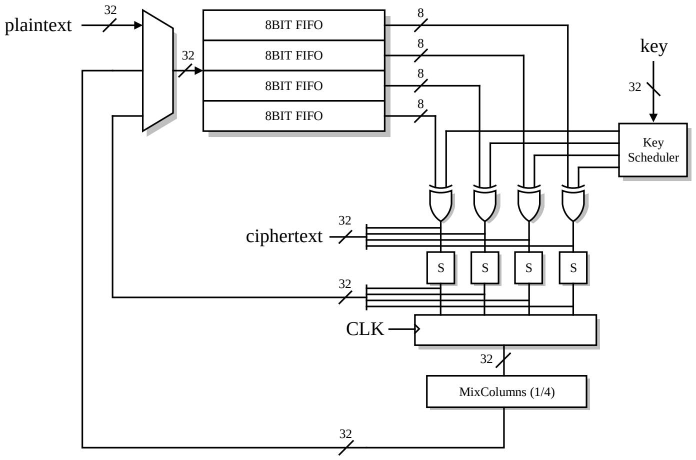


Fig. 2: Architecture of the 32-bit AES encryption co-processor.

Regarding the probing methodology, we used both a 2Ω resistor in series with the power supply voltage and an inductive probe (the Tektronix CT-1, which gives a transresistance of 5mV/mA in the frequency range 25kHz–1GHz [1]). When using the CT-1 current probe, the shunt resistor was short-circuited. We optionally used a preamplifier, namely a R&S HZ16 [24] providing a gain of 20dB with a noise figure of 4.5dB, in the frequency range 100kHz–3GHz.[2]

Next, the DUT's clock frequency is an important macroscopic feature of a side-channel trace, since it usually reflects the frequency spectrum where leakage can be found. We chose three clock frequency values (1MHz, 6MHz and 24MHz) for the hardware DUT and three clock frequencies (4MHz, 24MHz and 48MHz) for the software DUT. Note that 48MHz is the maximal clock frequency of the device. Those sets of values were chosen to observe the impact of the clock frequency on the shape and distinguishability of the leakage cycles.

Similarly, we chose three power supply voltage (0.8V, 1.2V and 1.4V) for the hardware DUT and three power supplies (2.6V, 3.0V and 3.6V) for the software DUT. Those sets of values were chosen in order to observe the impact of working at nominal supply voltage vs. in minimum and maximum corner cases.

Finally, we used a Picoscope 6424E providing a vertical resolution of 12 bits and running at three different sampling rates as DSO. We chose sampling rates values according to the clock frequency of the given DUT, to analyze the impact of the collected number of samples per clock cycles (which impacts the acquisition bandwidth and memory requirements). Precisely, we set the sampling rate of the DSO at approximately ×1 , ×5, ×25 the chosen DUT's clock frequency. We note

[2] The on-board Sakura amplifier was not used for consistency with the software setup.

that the sampling rate of our DSO is not an integer multiple to the DUT's clock frequency as it may induce correlated noise in the measurements.

In total, we performed $4 \times 3 \times 3 \times 3 = 108$ experiments on each DUT. In each experiment, we targeted the first key byte of the first AES round and collected 4×10^6 traces for the hardware DUT and 10^5 for the software one. Both the input plaintexts and keys have been picked up uniformly at random, in order to stimulate the combinational and sequential logics of both platforms.

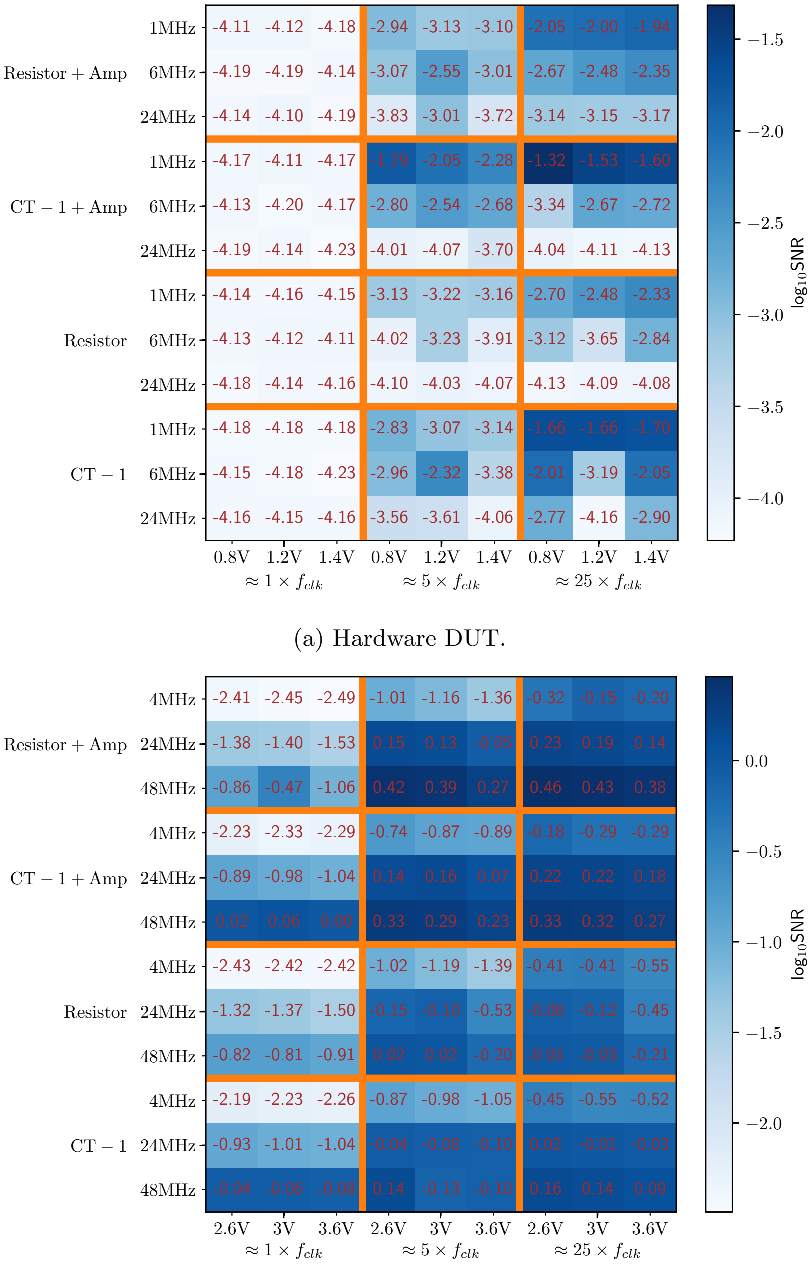
4 Experimental results and discussion

In this section, we present the results of our analyses for the proposed metrics throughout our design space. We first introduce the set of plots (e.g., for the SNR and PI) that summarize our experiments and will be the basis of our discussions. We then extract the best configurations for the measurement setup of both platforms. We finally propose general guidelines for the design of good measurement setups. Given the granularity of the explored design space, we organize this discussion according to the setup model in Section 3. We also evaluate the relevance of univariate evaluation metrics as predictors of multivariate ones.

Figure 3 shows the highest SNR value we found for each set of parameters for both platforms (in logarithmic scale). We present the results in the form of a matrix where the X-axis contains the different power supply values and sampling speeds, and the Y-axis contains the DUT clock frequencies and the probing method used in each experiment. The thick orange lines delimit the probing methods on the X-axis and the sampling speeds on the Y-axis. Darker blue blocks represent setup parameters where the SNR is higher. Figure 4 shows a similar matrix for the PI values obtained after LDA, in order to evaluate the impact of setup choices from a multivariate attack perspective.

A bit more in detail, the SNRs in Figure 3 were calculated on the whole leakage trace and the maximum value was then taken. In the hardware case, as shown in Figure 3a, the best SNR is obtained using the CT-1 current probe combined with the amplifier, setting the DUT to the slowest clock speed and lowest power supply value, and sampling at the highest rate. In the software case, as shown in Figure 3b, the differences are more subtle and many sets of parameters give a peak SNR value close to the best one. The latter is obtained using the resistor combined with the amplifier, setting the DUT to the highest clock speed and sampling at lowest rate (contrary to the hardware case) while still using the lowest power supply value (like in the hardware case).

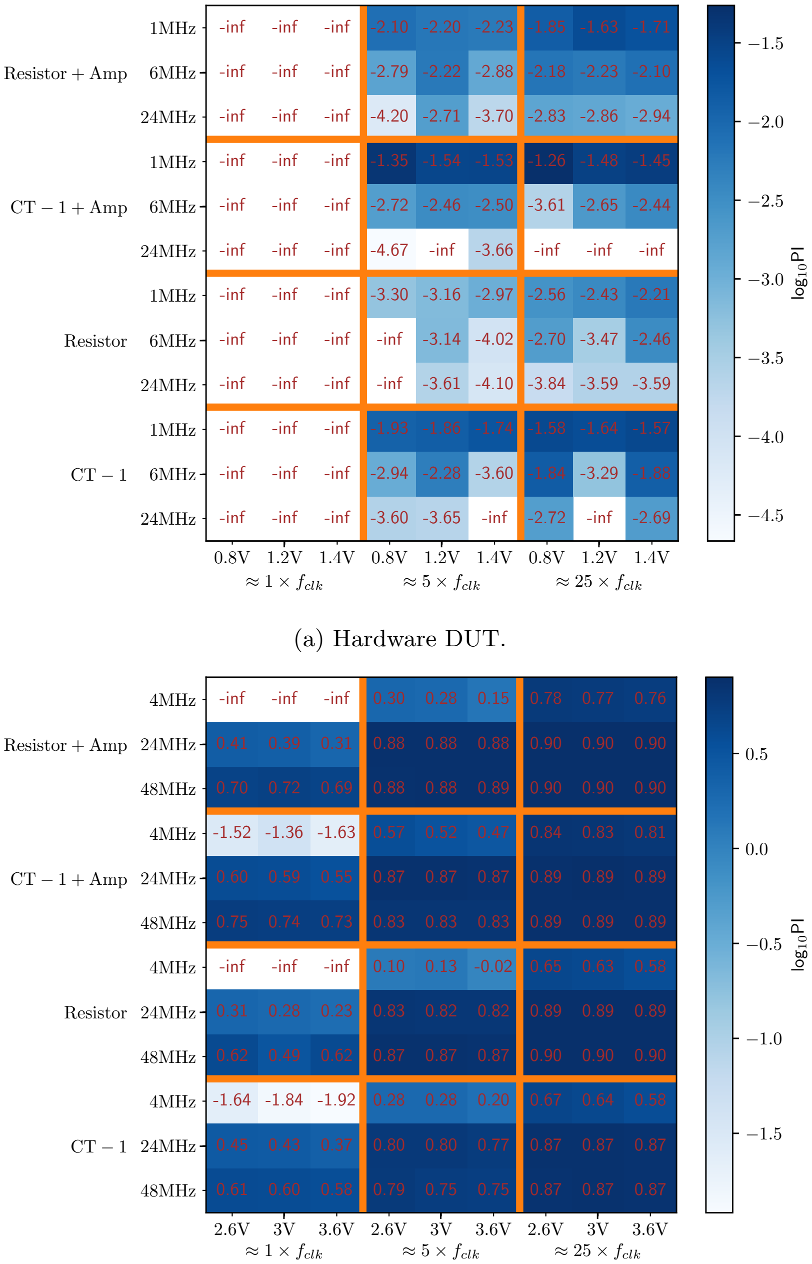
Regarding our multivariate analysis, we calculated the PI for each set of parameters. Concretely, for each experiment independently, we first pre-selected samples based on the SNR traces, keeping the ones above the noise floor for profiling. We then built Gaussian templates combined with LDA as presented in Section 2.2. We next analyzed the impact of the d' parameter, trying $d' = 1$ up to 25 for the hardware platform and 50 for the software one. We finally kept the d' leading to the highest PI, which is reported in Figure 4.



(a) Hardware DUT.

(b) Software DUT.

Fig. 3: Peak SNR values observed for the hardware (a) and software (b) DUTs.



(a) Hardware DUT.

(b) Software DUT.

Fig. 4: Peak PI values observed for the hardware (a) and software (b) DUTs.

For the hardware platform's results reported in Figure 4a, we observe that the experiment leading to the highest PI is obtained with the same set of parameters that leads to the highest SNR in Figure 3a. Generally speaking, comparing the two metrics, the PI follows the same trend as the SNR in this case.[3] For the software platform's results reported in Figure 4b, we see that the effect of the probing method and the power supply voltage are negligible, which is different from the univariate SNR analysis of Figure 3b. By contrast, observations regarding the clock frequency and sampling rate remain similar as in the univariate case. We also note that our highest PI value is 7.96 for an 8-bit bus.

As a complement, Figure 5 depicts exemplary SNR and leakage traces for both platforms, corresponding to the best cases in Figure 3. The SNR traces are in the upper subplots and mean leakage traces in the lower subplots.

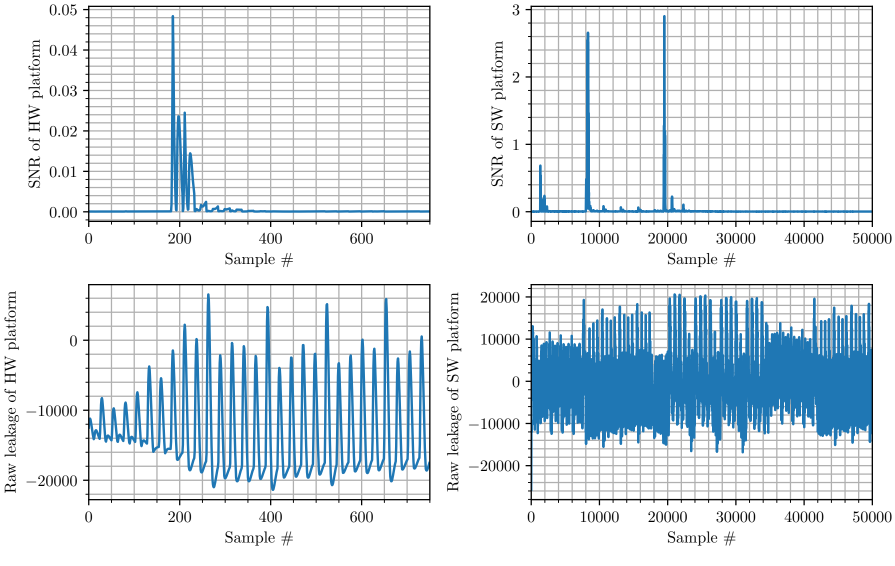


Fig. 5: Exemplary SNR and leakage traces: hardware (left) and software (right).

These experimental investigations lead to the following observations:

Probe. The choice of a probe was more critical for the hardware platform than the software one in our experiments. In the hardware case, the inductive probe gave better results than the resistor. A plausible explanation is that the CT-1 interferes less with the side-channel signal and is intrinsically less noisy than a shunt resistor. So as long as the target leakage is covered by the probe's bandwidth, it seems to be a good choice. In the software case, both the inductive probe

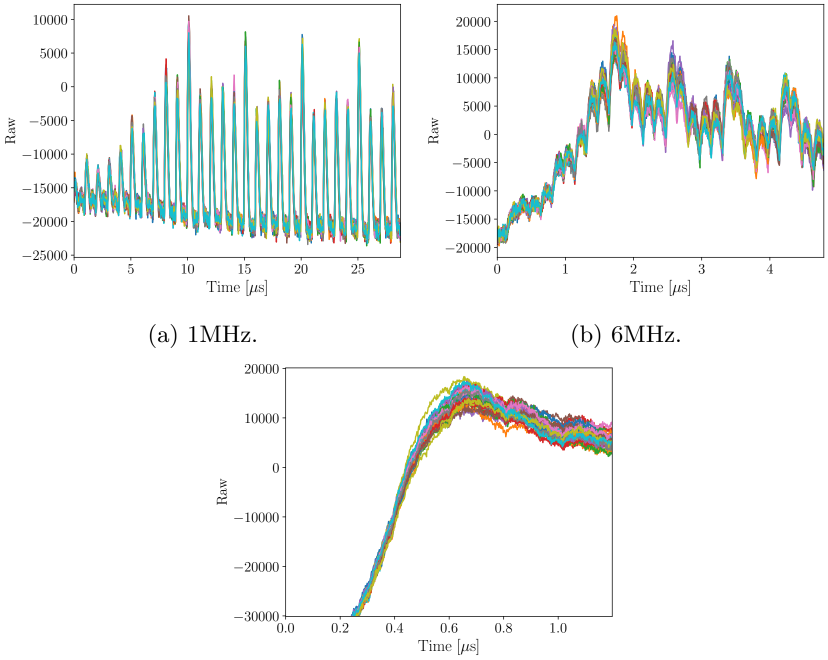
[3] The experiments where the $\log_{10} \mathsf{PI}$ is -inf correspond to a negative PI, indicating the no information could be extracted from the estimated model.

and the resistor gave good results, presumably due to the easier-to-exploit measurements (reflected by the higher SNR and PI values). As for the use of the amplifier, it does not show a significant impact as in most of our design space, the signal that we sample is within the vertical range of our DSO.

We posit that the observation regarding the inductive (CT-1) probe could change if targeting higher clock frequencies, and the observation regarding the amplifier could change if targeting more advanced technologies or a side-channel signal with lower amplitude (e.g., an electromagnetic one).

Clock frequency. This parameter is in general important for side-channel analysis. Whenever it can be controlled by the adversary, both our hardware and software results suggest the same rule-of-thumb: "*use the highest available clock frequency such that independent clock cycles are easy to distinguish*".

We first illustrate this rule-of-thumb with Figure 6. It shows the traces we recorded with the best parameter set and varying clock frequencies for the FPGA platform. At 1MHz, the independent peaks for each clock cycle are clearly distinguishable. At higher clock frequencies, the leakage traces are smoother and the overlapping between the clock cycles in the measurements increases.



(a) 1MHz.

(b) 6MHz.

(c) 24MHz.

Fig. 6: Clock frequency effect on the hardware setup.

We next turn to the software case study to explain the first part of the rule-of-thumb (i.e., why it is not advisable to reduce the clock frequency unconditionally). In this respect, we first note that for this software DUT, the clock cycles were clearly distinguishable even for the maximum clock frequency (so the second part of the rule-of-thumb was fulfilled). In this case, the best SNR and PI values are observed for higher clock frequencies. We explain this effect by observing that all the samples in a clock cycle are not equally informative. During an MCU clock cycle, most of the dynamic power is contained right after the rising edge of the clock as the effect of the registers changing state. The leakage from the remaining of the clock cycle is mostly due to static power and is usually less informative [23,20]. Therefore, the interest of decreasing the clock frequency can become detrimental when conditioned on a sampling frequency.

More precisely, and as illustrated in Figure 7, decreasing the clock frequency can lead the collected samples (represented by red diamonds in the figure) to correspond mostly to the static part of the leakage, and to miss the information of the dynamic part (represented by the green rectangles of the figures). Overall, this can lead to a collection of samples that is less informative: the univariate SNR can be lower by missing the most informative sample and the multivariate PI can be lower by cumulatively covering less relevant samples.

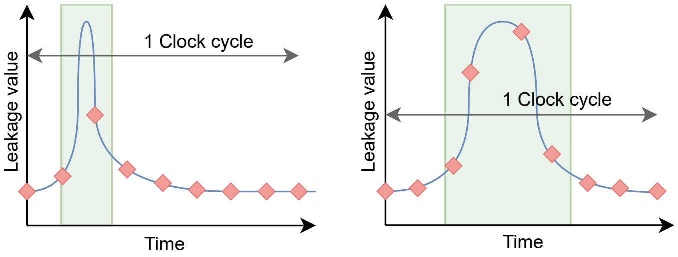


Fig. 7: Sampling with different DUT clock frequencies.

$\mathbf{V_{DD}}$. Despite less definitive than the clock frequency, the supply voltage also affects the shape of the leakage traces, as it increases the critical path and therefore spreads the information towards more samples. This naturally causes the multivariate PI to be improved when lowering the supply voltage below the nominal one. Interestingly, we also observed that for both targets and most sets of other parameters, decreasing $\mathrm{V_{DD}}$ is also beneficial to the univariate SNR.

A plausible reason for this observation is that both devices are based on CMOS technology (even though from different technology nodes and manufacturers) which generally exhibits smoother transient current when $\mathrm{V_{DD}}$ is lower than nominal, due to reduced transconductance of digital cells. This can reduce both the signal and, here more dominantly, the noise of the leakage.

We note that this observation is admittedly technology-dependent: see [28] for a report on several technology nodes. It is also not unconditional: as reported in the same paper, the output noise of a digital cell in subthreshold regime (that corresponds to extremely low $\mathrm{V_{DD}}$ values) is not minimal, as transistors exhibit higher resistance and thus contribute more to increase the noise level. So overall, our conclusion regarding the $\mathrm{V_{DD}}$ parameter is that reducing it below the nominal value can have marginal interest, especially for multivariate attacks, but is not expected to lead to significant gain/loss factors.

Sampling rate. This parameter is especially critical for the cost of the attacks as it affects the memory requirements to store the leakage measurements.

On the one hand, its selection is related to the clock frequency: as in general when quantizing signals, the sampling rate should at least be chosen larger than the Nyquist frequency. This requirement was confirmed in our experiments, and showed to be more critical (resp., less critical) in the hardware case (resp., software case). This is presumably due to the lower (resp., larger) amount of less (resp., more) informative samples of the harware (resp., software) case.

On the other hand, in the context of side-channel analysis, a natural question is whether increasing the sampling rate significantly beyond the Nyquist frequency can be useful. Namely, can it lead to more powerful multivariate attacks? By testing a sampling frequency of ×1, ×5 and ×25 the clock frequency, we observed that collecting more samples helps only to a limited extent. In particular, both for the software and the hardware platforms, the gains when moving from ×1 to ×5 the clock frequency are more significant than when moving from ×5 to ×25 the clock frequency, again with more incentive to increase the sampling rate in the hardware case than in the software case. A plausible reason for this difference is once more the more condensed and noisy nature of the hardware leakage (i.e., the fact that it is concentrated in less cycles with more algorithmic noise, rather than spread over more cycles in software).

Univariate vs multivariate evaluations. Eventually, our results indicate that whether the SNR is a good predictor of the PI is quite case-dependent.

If the SNR traces present a single peak or a set of peaks that are close to each other (e.g., within one cycle), they usually indicate correlated leakage coming from a single operation. In this case, which typically corresponds to our hardware experiments (see the left part of Figure 5), a good univariate SNR will generally be a good indicator of a good multivariate PI. Multivariate attacks will always be more powerful but the SNR can serve as a first-order comparison metric.

By contrast, if the SNR traces contain multiple peaks separated by several clock cycles, they rather indicate independent leakage coming from different operations. In this case, which typically corresponds to our software experiments (see the right part of Figure 5), multivariate attacks are expected to be significantly more powerful than univariate ones. So the direct estimation of the multivariate PI is in general a better (i.e., more conclusive) evaluation strategy.

5 Conclusions

This study aims at evaluating the risk of over-estimating the physical security of an implementation due inadequate parameter choices when configuring a measurement setup. We focus on four main parameters: the probing method, the clock frequency, the power supply voltage and the sampling rate. We apply our methodology to an embedded software and a hardware FPGA implementation of the AES-128 block cipher. It leads to 108 experiments for each DUT, that we analyze by means of univariate and multivariate evaluation metrics, namely the SNR and the PI. Our findings show that the losses due to a bad selection of parameters can be significant and lead to a strong over-estimations of an implementation's security level. We also use our experiments in order to consolidate general intuitions and recommendations regarding the good choice of parameters and to discuss their device and architecture dependencies.

Acknowledgments. François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the European Union through the ERC project SWORD (724725) and by the F.R.S.-FNRS equipment project SCALab.

References

1. Ac current probes - ct1, ct2, ct6 data sheet. `https://download.tek.com/manual/070795702web.pdf`
2. Tiny aes in c. `https://github.com/kokke/tiny-AES-c`
3. Balasch, J., Gierlichs, B., Reparaz, O., Verbauwhede, I.: Dpa, bitslicing and masking at 1 ghz. In: CHES. Lecture Notes in Computer Science, vol. 9293, pp. 599–619. Springer (2015)
4. Beckers, A., Balasch, J., Gierlichs, B., Verbauwhede, I.: Design and implementation of a waveform-matching based triggering system. In: COSADE. Lecture Notes in Computer Science, vol. 9689, pp. 184–198. Springer (2016)
5. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: CHES. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer (2004)
6. Bronchain, O., Hendrickx, J.M., Massart, C., Olshevsky, A., Standaert, F.: Leakage certification revisited: Bounding model errors in side-channel security evaluations. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 11692, pp. 713–737. Springer (2019)
7. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 398–412. Springer (1999)
8. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: CHES. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer (2002)
9. de Chérisey, E., Guilley, S., Rioul, O., Piantanida, P.: Best information is most successful mutual information and success rate in side-channel analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(2), 49–79 (2019)
10. Cnudde, T.D., Bilgin, B., Gierlichs, B., Nikov, V., Nikova, S., Rijmen, V.: Does coupling affect the security of masked implementations? In: COSADE. Lecture Notes in Computer Science, vol. 10348, pp. 1–18. Springer (2017)

11. Cnudde, T.D., Ender, M., Moradi, A.: Hardware masking, revisited. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(2), 123–148 (2018)
12. Duc, A., Faust, S., Standaert, F.: Making masking security proofs concrete - or how to evaluate the security of any leaking device. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 9056, pp. 401–429. Springer (2015)
13. Goubin, L., Patarin, J.: DES and differential power analysis (the "duplication" method). In: CHES. Lecture Notes in Computer Science, vol. 1717, pp. 158–172. Springer (1999)
14. Guilley, S., Maghrebi, H., Souissi, Y., Sauvage, L., Danger, J.L.: Quantifying the quality of side-channel acquisitions. In: COSADE 2011. pp. 16–28
15. Herbst, C., Oswald, E., Mangard, S.: An AES smart card implementation resistant to power analysis attacks. In: ACNS. Lecture Notes in Computer Science, vol. 3989, pp. 239–252 (2006)
16. Levi, I., Bellizia, D., Standaert, F.: Reducing a masked implementation's effective security order with setup manipulations and an explanation based on externally-amplified couplings. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(2), 293–317 (2019)
17. Mangard, S.: Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In: CT-RSA. Lecture Notes in Computer Science, vol. 2964, pp. 222–235. Springer (2004)
18. Mangard, S., Oswald, E., Standaert, F.: One for all - all for one: unifying standard differential power analysis attacks. IET Information Security **5**(2), 100–110 (2011)
19. Moos, T., Moradi, A., Richter, B.: Static power side-channel analysis - an investigation of measurement factors. IEEE Trans. Very Large Scale Integr. Syst. **28**(2), 376–389 (2020)
20. Moradi, A.: Side-channel leakage through static power - should we care about in practice? In: CHES. Lecture Notes in Computer Science, vol. 8731, pp. 562–579. Springer (2014)
21. Moradi, A., Barenghi, A., Kasper, T., Paar, C.: On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx virtex-ii fpgas. In: ACM Conference on Computer and Communications Security. pp. 111–124. ACM (2011)
22. Pozo, S.M.D., Standaert, F.: Getting the most out of leakage detection - statistical tools and measurement setups hand in hand. In: COSADE. Lecture Notes in Computer Science, vol. 10348, pp. 264–281. Springer (2017)
23. Pozo, S.M.D., Standaert, F., Kamel, D., Moradi, A.: Side-channel attacks from static power: when should we care? In: DATE. pp. 145–150. ACM (2015)
24. Rodhe&Schwarz: R&s hz-15, r&s hz-17 probe sets, r&s hz-16 preamplifier. `https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/service_support_30/HZ-15_16_17_bro_en_5213-6687-12_v0100.pdf`
25. Standaert, F., Archambeau, C.: Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In: CHES. Lecture Notes in Computer Science, vol. 5154, pp. 411–425. Springer (2008)
26. Standaert, F., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 5479, pp. 443–461. Springer (2009)
27. for Standardization, I.O.: It security techniques - test tool requirements and test tool calibration methods for use in testing non-invasive attack mitigation techniques in cryptographic modules - part 1: Test tools and techniques (Geneva (CH) 2019), iSO/IEC 20082-1

28. Veirano, F., Silveira, F., Navinery, L.: Is intrinsic noise a limiting factor for subthreshold digital logic in nanoscale cmos? In: 2015 International Workshop on CMOS Variability (VARI). pp. 45–50 (2015)
29. Veyrat-Charvillon, N., Medwed, M., Kerckhof, S., Standaert, F.: Shuffling against side-channel attacks: A comprehensive study with cautionary note. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 7658, pp. 740–757. Springer (2012)