The Uncertainty of Side-channel Analysis: A Way to Leverage from Heuristics

UNAI RIOJA and SERVIO PAGUADA, Ikerlan Technology Research Centre, Spain and Digital Security Group, Radboud University Nijmegen, The Netherlands
LEJLA BATINA, Digital Security Group, Radboud University Nijmegen, The Netherlands
IGOR ARMENDARIZ, Ikerlan Technology Research Centre, Spain

Performing a comprehensive side-channel analysis evaluation of small embedded devices is a process known for its variability and complexity. In real-world experimental setups, the results are largely influenced by a huge amount of parameters, some of which are not easily adjusted without trial and error and are heavily relying on the experience of professional security analysts. In this article, we advocate the usage of an existing statistical methodology called Six Sigma ($6\sigma$) for side-channel analysis optimization. This well-known methodology is commonly used in other industrial fields, such as production and quality engineering, to reduce the variability of industrial processes. We propose a customized Six Sigma methodology, which allows even a less-experienced security analyst to select optimal values for the different variables that are critical for the side-channel analysis procedure. Moreover, we show how our methodology helps in improving different phases in the side-channel analysis process.

CCS Concepts: • Security and privacy → Side-channel analysis and countermeasures; Embedded systems security; Hardware reverse engineering; • Hardware → Emerging tools and methodologies;

Additional Key Words and Phrases: Cryptographic hardware, side-channel analysis, six sigma, deep learning

ACM Reference format:
Unai Rioja, Servio Paguada, Lejla Batina, and Igor Armendariz. 2021. The Uncertainty of Side-channel Analysis: A Way to Leverage from Heuristics. J. Emerg. Technol. Comput. Syst. 17, 3, Article 40 (May 2021), 27 pages.
https://doi.org/10.1145/3446997

1 INTRODUCTION
The process of obtaining data for Side-Channel Analysis (SCA) is a complex procedure in which not only the acquisition of thousands or even millions of power or EM traces is needed, but also the usage of signal processing techniques in combination with advanced statistical and mathematical
tools is most of the times mandatory. In every step of this path, many decisions have to be taken that are commonly based on the know-how of the people who have dealt with this kind of issues in the past. Obtaining data for SCA (and performing the SCA itself) is an unpredictable process in which those decisions are often reached via repeated "trial and error." Moreover, building a proper experimental setup is often a non-repetitive task. In other words, there is no guarantee that a setup that works properly for one device or traceset would work properly for another one. Thus, in such a varying process it is important to be structured and methodical, especially in keeping track of the whole process and its variables to justify the decisions that were made.

In a typical SCA scenario, security analysts deal with lots of parameters to tune and choices to make. Thus, it is no surprise that at the end of the day, when good results arise after a lot of changes applied on the spot, analysts often do not know exactly what decision or what parameter caused the improvement of the results. In practice, as the Pareto principle postulates [17], there are a few parameters that have a strong influence on the results (vital few) against lots of parameters whose impact on the results is negligible (trivial many). It is not easy to find the one parameter (or a group) that causes the biggest impact on the experiment. Taking all of the above into account, the importance of using a suitable methodology to perform an experimentation process with those characteristics is evident.

Six Sigma is a well-known statistical methodology targeted at improving industrial processes (production and quality engineering) by reducing their variability. To the best of our knowledge, it has not been used before to reduce the uncertainty of an SCA process. In this work, we develop a customized version of Six Sigma to make it fit into the SCA requirements and to be able to use it when optimizing each one of the SCA phases, e.g., acquisition, leakage assessment, and attacking phase. After applying the methodology, we were able to effortlessly select the best values for various parameters analyzed within distinct SCA scenarios. In addition, it helped us in finding the most relevant parameters for the analysis results.

Problem statement: Systematically keeping track of the parameters in an evaluation or an attack scenario is not trivial, and also not feasible without a well-defined procedure. The current state-of-the-art approach is often experimental and founded on previous experiences, especially on the acquisition and signal processing stages (see Sections 2 and 3). However, this strategy does not give much insight into the choice of the parameters that could have had the most impact on the results.

Our contribution: In this article, we do not claim having developed a new SCA evaluation/testing scheme. Instead, we present an approach that can complement existing schemes, in all the stages of a regular SCA evaluation or only in specific ones such as those typically troublesome to optimize due to their nature, e.g., measurement and preprocessing stages. This is done without the need for any new resources (in time or power). We claim that, by using this new methodology, one could reduce the uncertainty associated with those, often cumbersome, techniques. The main goal is to help lab technicians (or even product developers), who may not have a deep knowledge of all the statistical and signal processing concepts involved in these methods, to perform a sound side-channel evaluation and interpret the results properly. For a more experienced evaluator, using the methodology could also be beneficial, particularly in case of new experimental setups, new target devices, new cryptographic implementations, new side-channel techniques, new datasets, and so on.

The rest of the article is organized as follows: Section 2 gives a brief introduction to side-channel analysis, Section 3 explains how other works are related to this. Section 4 introduces the main concepts of the Six Sigma methodology, explains its main steps. Section 6 corresponds to the use case in experimental setup optimization. Section 7 develops a use case in an attack scenario. Sections 8 and 9 show two use cases that involve leakage assessment using both statistical and machine learning approach. Section 10 concludes the article.
The Uncertainty of Side-channel Analysis: A Way to Leverage from Heuristics

2 SIDE-CHANNEL ANALYSIS IN A NUTSHELL

The overall process of a typical side-channel analysis is depicted in Figure 1. The first stage is usually to acquire the side-channel information, i.e., leakage that becomes available from the Device Under Test (DUT) when computing a cryptographic algorithm. The leakage could come in the form of timing information, sound, electromagnetic emanations, or power consumption. A stage where one has to perform signal processing is not mandatory but can be helpful for the analysis. The final stage consists of statistical analysis or leakage assessment and it is performed by running an attack using "classical" SCA approaches such as Differential Power Analysis (DPA) [25] or even machine learning-based models.

All these stages involve many parameters, each of them defined in a specific numerical or non-numerical domain and range. Moreover, all the stages are strongly related to each other, so a parameter in one of them could significantly impact the other’s performance.

3 RELATED WORK

The need to integrate and validate countermeasures against side-channel attacks on embedded devices earned quite some attention in recent years. Current certification processes such as EMVCo [18] or Common Criteria (CC) [11] evaluate the robustness of a DUT by directly attacking it with different side-channels techniques (e.g., differential power analysis (DPA) [25], correlation power analysis (CPA) [7], mutual information analysis (MIA) [4, 19], template attacks (TA) [8, 10, 37], and deep learning-based attacks [26, 29, 34]). Nevertheless, the increasing number of attacks, different implementations, side-channel distinguishers, and leakage models make this evaluation not feasible in terms of resources and time, especially for less-experienced evaluators. Moreover, in such a scenario, we need to conduct various stages of a typical side-channel evaluation (see Figure 1), involving aspects from very different fields, such as cryptographic implementations, programming, signal processing, cryptanalysis, statistics, and so on. Thus, it is important to use a proper methodology to conduct an evaluation (or certification) process of this range.

Several methodologies, techniques, and solutions have been proposed to help evaluators; those are mostly focused on the analysis stage of the process, as the physical nature of the acquisition and signal processing steps make them strongly dependent on engineering expertise [3]. However, there are in fact some guidelines on how to carry out the acquisition and pre-processing phases. For instance, ISO/IEC 20085-1 specifies what elements the acquisition setup must be composed of and how they interact. Also, ISO/IEC 17825:2016 gives certain specifications and quality criteria the measurement setup must meet. Nevertheless, the effectiveness of the acquisition setup relies strongly on the evaluator. The same holds for preprocessing where several published solutions for filtering the noise [30, 33] and traces resynchronization exist [39, 43]. However, their success is typically application-dependent, which makes it difficult to apply in a generic way when looking
for an optimal solution. All this causes that a wrongly designed experimental setup usually increases the complexity of the problem and may have a strong negative impact on the success of the attack [3, 16, 27].

However, most of the related work focuses on the analysis phase, as mentioned above. For example, Test Vector Leakage Assessment (TVLA) by Cryptography Research (CRI) [12, 21, 23] uses a statistical test (commonly Welch’s $t$-test [41] or Pearson’s $X^2$-test [32]) to distinguish whether two sets of data (e.g., random vs. fixed) are significantly different. TVLA is a well-known leakage detection technique, but it merely points to the presence of leakage in the power traces, without stipulating whether that particular leakage is significant or giving any information on how to exploit it. The methodology also includes some recommendation on how one has to collect the leakage traces while the device processes specific values. Thus, while TVLA also discusses the side-channel acquisition stage, it mainly addresses the analysis stage. The same holds for the works that introduced classical attacks (i.e., SPA, DPA, CPA [7, 25]) and profiled attacks [8, 10, 37]; they also mainly cover the analysis stage. However, all kinds of attacks, profiled and non-profiled, rely strongly upon the way things are done in previous stages. Furthermore, machine learning approaches are not the exception; those methodologies focus on building the model’s architecture to find out the best performance when running evaluation [26, 29, 34, 46].

Additionally, it should be noticed that apart from the complex CC “attack driven” testing scheme, which exhaustively checks explicit security claims of the manufacturer, there exist another approach: “conformance style” testing. This approach for SCA testing is followed by NIST/FIPS (via ISO/IEC 17825:2016, the only publicly available standard of this kind) and relies on checking some minimum criteria regarding the security of the device. Concretely, for testing against basic power analysis attacks in the context of symmetric encryption, ISO/IEC 17825:2016 relies on using leakage detection procedures (TVLA) instead of attempting attacks. Nonetheless, as many authors claim [3, 41, 45], the goal of evaluations is commonly to “certify security,” and this is especially challenging to achieve based only on leakage detection techniques, as any missed leak may enable a device to pass certification. Thus, it is indeed crucial to “attack” a device comprehensively to certify its security with the complexity that this demands.

To conclude, after mentioning commonly used statistical tools and methodologies in the SCA field, we claim that there are no transverse methodologies like the one suggested in this work. By a transverse methodology, we consider those that could cover the whole process and monitor how the parameters in different stages influence the process itself. Thus, an evaluator can use our proposal on top of other methodologies and tools; we claim that it grants them an additional capability in evaluations. For the same reason, we consider this work as a way to optimize each of the different stages of a typical SCA, as shown in several use cases below. We also claim that our proposal is applicable to one single stage or a combination of several stages.

4 SIX SIGMA METHODOLOGY

The Six Sigma ($6\sigma$) methodology was created in 1986 by Bill Smith while working as an engineer at Motorola, as the company that registered the term as its trademark in 1993 [22]. The primary objective was to minimize the variability of the output of a process. To achieve this, different empirical quality management methods, along with statistical methods, are used. In this improvement process, some steps have to be repeated until the main goal is reached.

Six Sigma involves two main methodologies implicitly, which are the basis for the process management and optimization, and also for the guidelines proposed in this document. The methodologies are: Define-Measure-Analyze-Improve-Control (DMAIC) and Define-Measure-Analyze-Design-Verify (DMADV). The former aims at improving an existing process, while the latter aims at designing a new process. Both are based on Deming’s Plan-Do-Check-Act
Cycle [9, 40]. Although those two methodologies are similar, we focus on DMAIC, because our aim is to improve the selection of parameters for an SCA process and not to design the process itself. The steps of the methodology are shown in Figure 2, pointing out how they fit in an SCA evaluation use case (a detailed explanation can be found in Section 6).

- **Define** the system. Here, as the system’s inputs, we envision the client’s requirements and the goal of the project, i.e., evaluation. In our case, the “Define phase” is basically the study of the DUT. When a “black-box” evaluation is performed, one deals with a device with almost no information about its internals. Nevertheless, in a “white-box” evaluation there are also some variables with less certainty, e.g., working frequency, algorithm implementation details, location of the cryptographic operations, and so on. Depending on how evaluators define their goal, the “uncertain” variables are revealed systematically by the end of the whole process. In this step, one defines not only the main goal(s) but also the OK-criterium. The proper definition of those helps to minimize the uncertainties. We interpret this OK-criterium as the quantification of the goal we want to achieve. In other words, this implies the definition of a factor that allows us to decide if the experiments are conclusive or not. A more detailed explanation can be found in the “Define” section of use cases, i.e., Sections 6.2, 7.2, and 8.2.

- **Measure** the current process setup. To characterize the current state of the process, one collects its parameters and outputs. In our case, a few preliminary acquisitions of side-channel signals should be done to be used as a baseline for the results that are meant to be improved. The objective is to define the variables/parameters we are going to study. We need to define the system’s variables that could affect the quality of the experiments (e.g., number of traces, sampling frequency, filtering). For those variables, one prescribes two values for each working variable (i.e., low setting and high setting). Commonly, those values depend on a huge number of factors, so consequently, it might be necessary to perform several preliminary measurements to establish the right values. Once the variables are defined and bounded, we must choose three of them: the three variables that are most likely to affect the results of the experiment. The rest of them will be fixed to some constant values (e.g., low setting, high setting, or an intermediary value). To illustrate this, in the “Measure” section of our use cases (i.e., Sections 6.3, 7.3, and 8.3), the reader can find Tables 3, 11, and 18 with “Defined variables,” with all the variables considered, and Tables 4, 7, and 12 with the three variables chosen as “Working variables” along with their low and high settings for each respective case.

- **Analyze** the data obtained from the process, and determine its relationships with the problem. This step consists of experimentation i.e., crafting an experiment or **Design of Experiments (DoE)** [31]. The objective is to quantify which variables have more influence over the experiment and adjust them to the proper values. To do that, a DoE with the three selected working variables to perform eight experiments is chosen. The output of it gives the coefficient to each variable, which tells us if the effect is positive or negative (improves
or not the result of the experiment) and how strong each one in comparison with the others is. In Section 4.1, we briefly describe the usage of this technique inside Six Sigma. Also, in the “Analyze section” of use cases (Section 6.4, Section 7.4, and Section 8.4) the output of each DoE can be observed in more detail.

- **Improve** the current process using the analysis of the root causes done in the previous step to identify, test, and implement a solution for the problems that appeared. In this customized Six Sigma, the step consists of the analysis of the experimental design’s results. Here, one adjusts the identified working variables that have more influence over the experiments. Afterward, one performs many rounds of the experiment (at least two times), to ensure that the results are not altered due to any failure. If this is not the case, then a new setup is designed considering those results. If the results are not good enough (even after the eight experiments), then the process should be repeated from the previous step, considering to change the selected variables or adjust the value of their low setting and high setting. This is considered as one iteration. The idea is to perform several iterations between these two last steps until the main objective is reached. Practical examples of this analysis of the results and the readjustment of the DoE variables are shown in the “Improve” section of our use cases (Section 6.5, Section 7.5, and Section 8.5).

- **Control** the newly improved process to correct any undesired deviations of it. Repeat the steps until obtaining the desired quality level. This step does not strictly apply to our problem, but it can be understood as the action of taking notes of the results to apply in future experiments.

### 4.1 Design of Experiments Inside Six Sigma

Design of Experiments (a.k.a. Experimental design or Factorial Design) is a branch of applied statistics responsible for evaluating the factors (or variables) that influence a process. Six Sigma uses this tool to determine the effects that the inputs (variables) of a process have on its output. There are different designs or schemes to conduct a DoE, namely, full factorial, partial or fractional factorial, and some others [13, 31, 42]. Each of them establishes the number of variables’ interactions taken to perform one iteration or replication. When using a full factorial, an evaluator has no limit in the number of interactions that he can take. In fact, according to this scheme, the evaluator should take all of them in one single iteration. When the number of variables is significant, tracking down which variables’ interactions have more influence becomes impractical.

Fractional factorial uses a fraction of all possible variables’ interactions. Although this scheme is a feasible candidate to combine with our proposal, it is unclear what could be the impact of rejecting interaction from the experiment. Our best guess is that, due to the side-channel process’s intrinsic nature, a false positive could arise, identifying wrong variables as those that have more influence. Nevertheless, the analysis of fractional factorial design is left for future work.

Finally, when the evaluator uses our methodology, (s)he uses a variant of DoE as explained above. The evaluator should take all the variables’ interactions; however, our Six Sigma specifics restricts him in taking just three variables at a time, meaning that he has only \(2^3=8\) variables’ effects to consider. In each iteration of the steps Analyze and Improve, a new combination of three variables is chosen; the iterations between these two steps point out the influence of those three. Bear in mind that a new combination could imply the same variables but with their two values updated.

### 5 ADVERSARIAL CAPABILITIES

As mentioned above, the proposed methodology is oriented to the context of side-channel evaluation. In this use case, it is common to assume the attacker with the strongest possible capabilities
(worst-case adversary [3]). That implies, among other aspects, that the attacker has physical access to the DUT and the capability to send unlimited chosen messages to the device (we assume that the attacker knows the input and output data). Also, we assume that the attacker has limited information about the internal workings of the device.

6 USE CASE 1: EXPERIMENTAL SETUP OPTIMIZATION

In this section, the proposed method is presented and explained step-by-step, giving examples for the procedures done with our experimental setup to optimize the acquisition process over a real device. This use case is also taken as the explanatory one, meaning that its explanation is in more detail than the rest.

6.1 Use Case Description

In this use case, we aim to optimize the experimental setup used to acquire the leakage traces. That implies optimizing parameters from both the acquisition phase and the signal processing phase (Figure 1). A similar approach can be applied for optimizing only one of the phases, but we propose to optimize both at the same time, since they are strongly linked.

The target is an external I2C device (slave). We are storing data (8-bit values) in an external I2C memory using the STM32F4 32-bit microcontroller (master). During the storage operation, the I2C device’s power consumption is being measured with a current probe. As evaluators, we would like to know whether it is possible to find dependency between the stored data and the power consumption, allowing an attacker to perform SCA. We need a proper experimental setup to do that, and we apply our customized Six Sigma methodology to optimize it.

After a few preliminary acquisitions (taken with the setup shown in Table 2), we did not learn much about the point in time when the data was being stored, and we were not able to obtain any correlation between the stored data and the power traces. Then, we used our customized Six Sigma methodology to improve the acquisition setup resulting in a significant correlation spike, identifying the exact point of time in which the 8-bit value was being stored in memory (see Figure 5).

6.2 Define

First, the DUT should be studied by an analyst to get familiar with it. Although this process should be always done prior to any SCA experiment, it is crucial for the “black-box” testing use cases, since its results might be the only source of information about the DUT. From this analysis, enough information should be gathered to define the following: the acquisition setup’s requirements, the goal of analyzing the acquired data, and the OK-criterium. We propose several steps for this preliminary analysis. Note that this use case is for handling devices whose internals are mostly unknown while examining side-channel leakage from them (e.g., “black-box” testing or an attack scenario). Most of these steps can be avoided if there is enough information about the internal behavior (such as in a “white-box” testing scenario in which complete control of the device is assumed). So, the steps proposed are as follows:

a. Define the basis for the unknown system, assuming some basic information is known. Usually, there are some known characteristics, such as its purpose, manufacturer, inputs and outputs, how much power (externally) it consumes, and so on. Anything that is known (and not guessed), should be written down as a list, as it is a starting point for the questions about characteristics of the system that are (still) unknown. For instance: Does the manufacturer uses standard architectures? What are the operations that the power consumption depends on? What operations are performed with the input data? Part of the answers to those
Table 1. Goal and OK-criterium

| Requirements                                                                 | OK-criterium                                                                 |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Same setup and device operational parameters fixed and constant in each experiment | Significant correlation spike \(r\) [Where \(r \leq -0.1705\) \(\cap\) \(r \geq 0.1705\)] |
| Oscilloscope: LeCroy Waverunner 9104                                         |                                                                              |
| SW for the acquisition, signal processing and data analysis                  |                                                                              |
| Current probe                                                                |                                                                              |

questions can determine some of the initial parameters of the acquisition setup, like the voltage scale in which the measurements should be done.

b. Analyze all the official (device datasheet) and unofficial documentation that can be obtained, trying to infer the details of the internal architecture that are not explicitly mentioned by the manufacturer.

c. Apply non-invasive analysis techniques on it, like measuring the voltage and performing a Fourier analysis, to learn the working frequency. There are also invasive or more destructive techniques possible, but they are out of scope here, since we focus on passive analysis.

d. Finally, the experts’ knowledge on similar devices can be helpful, but it is not mandatory.

After the analysis, one has to summarize the gathered information, enhancing the basic knowledge about the system with the new information obtained through the comparison of documentation, the non-invasive analysis, and the opinion of the experts. In our use case, first, the datasheet provided by the manufacturer was read and thoroughly analyzed. After that, the process of getting the system’s details mainly consisted of voltage measuring and performing a frequency analysis, from which a few different frequencies were identified. Both processes were accomplished by a person with some expertise in the field, and the results were discussed with the expert in electronic devices of the team.

Once the DUT has been thoroughly studied, we can use this information to define the requirements of the acquisition setup, the main goal of the experiment, and the OK-criterium:

**Requirements of the setup:** We interpret this as the initial parameters of the acquisition setup, which are not supposed to change during all the steps of the process. For instance, these parameters could be (but exclusively): operational parameters of the target, acquisition/analysis tools, physical properties studied (e.g., power consumption, electromagnetic radiation, and timing), type of the SCA technique implemented, and so on. If, while applying the process something suggests to change this data, then this should be taken into account, continuing the process until the end, and adding those changes in a future iteration. The setup requirements for our use case are given in Table 1.

**Main goal:** Here, we define the goal that we want to achieve. In our case, the experiment was to obtain and visualize a dependency between the stored 8-bit values and the power consumption.

**OK-criterium:** This is the quantification of the goal, a factor that tells us whether the experiments are conclusive or not. During the experiments, one tries to tune several parameters obtaining different results through iterations, so the OK-criterium indicates when to stop the Six Sigma performance. Also, it tells when we have reached our objectives. In this use case, the OK-criterium (Table 1) is to obtain a significant correlation spike (in comparison to the correlation obtained in non-leaking parts of the signal), indicating that in fact there is a data-dependency with the power consumption of the device. Moreover, we can compute the Confidence intervals for a sample (Pearson) coefficient value [36] to establish a concrete threshold. With a sample size of 1k and an observed correlation of approximately ±0.05 (in the non-leaking parts), our confidence interval
Table 2. Tools List and Brief Description

| Tool                  | Description                                                                 |
|-----------------------|-----------------------------------------------------------------------------|
| LeCroy Waverunner 9104| • 2 channels (Power consumption & triggering)                                |
| Oscilloscope          | • 20 GS/s (Single sample capture mode)                                      |
|                       | • I2C bus-based triggering                                                 |
| The DUT               | • Handled by an STM32F411-DISCO developing board (I2C based communication)   |
|                       | • Power supplied through the STM32F411-DISCO developing board                |
| PC                    | • Communicates with the STM32F411-DISCO developing board                     |
|                       | • Controls the oscilloscope                                                 |
| Current Probe         | • Tektronix CT1 current probe                                               |
|                       | • Connected in series with the DUT power line                               |
|                       | • Measures the power consumption                                            |

(99.99%) is from $-0.1705$ to $0.1705$ (considering positive and negative correlation values). This means that if the observed correlation is out of that range, then we can assume that there exists a significant statistical difference.

After having defined the basic requirements for the acquisition setup, the goal of acquiring the data, and the OK-criterium to evaluate the improvement on the results, the measurement step can be started.

6.3 Measure

Following the requirements defined in the previous step, the usual measurement setup is parametrized, and a first preliminary round of acquisitions is started with this basic setup. This basic setup consists of the following tools configured as stated below (Table 2).

With this baseline setup, a round of random 8-bit values is sent to the DUT from the computer (through the STM32F4 developing board), while the oscilloscope triggers in the I2C clock line (SCL), just after finishing the communication, when the internal computation in the DUT is supposed to start. The main goal of this step is to define the system’s variables that we are going to study. To ensure quality traces, the evaluator acquires a couple of hundreds of them, and in those, he might also discover defects or inconvenient features. Accordingly, he/she might decide to tune further some parameters to improve the results. The parameters can be different in nature (environment, processing, measurement, etc.). Also, some parameters can be considered in most of the cases (Lowpass filtering, number of traces, compression techniques, etc.) while other parameters will depend on the specific use case.

After prompting all the possible parameters that need further tuning, we define two values (a.k.a. two levels; low setting and high setting) for each working variable in their appropriate domain. The idea is to identify later which setting throws the best results and adjust the working variable accordingly. It should be noticed that the working variables can be numerical (i.e., quantitative) and non-numerical (i.e., qualitative). Also, this list must be analyzed to avoid the selection of parameters that can be dependent on each other. Table 3 gives the parameters with their descriptions and ranges (low and high setting). The top three variables will be analyzed in the next step performing a DoE on them (Table 4). The rest of the parameters have to be fixed to some constant values (e.g., low setting, high setting, or an intermediary value). If the variable has been checked in any previous iterations, then the chosen values should be the best for fulfilling the OK-criterium. Otherwise, values could also be chosen according to the evaluator’s criterion (and be re-adjusted in further iterations if necessary).

An expert evaluator could sort the parameters by relevance based on his intuition and/or experience. It might speed up the process, since one typically considers the most relevant variables first for the DoE, and hence the OK-criterium could be reached sooner. Nevertheless, the parameters...
Table 3. Defined Variables

| Rank | Parameter | Description | Range | Fixed Value |
|------|-----------|-------------|-------|-------------|
| 1    | Point of Alignment | Two interesting places in the traces to search for correlation are observed | Align at the start vs. at the end | |
| 2    | LowPass Filter (SW) | Filtering the signal or not would affect the quality of the collected traces. We should eliminate high-frequency noise but without destroying the leakage. | Filtering vs. No filtering | |
| 3    | Standardization | Removing the mean of the dataset can help to improve the results. | Yes vs. No | |
| 4    | Nº of traces | The number of traces affects the confidence of the results but increases the computational effort | 1k vs. 100k | 1k |
| 5    | Compression | Compression can be used to reduce the dataset size and to improve the leakage (close points carry very similar information and noise can be reduced). Conversely, leakage can also be destroyed because of compression. | Compression vs. no compression | No compression |
| 6    | Sampling frequency | A high sampling frequency will improve the quality of the traces but also increase the data size | 50 MHz vs. 20 GHz | 1 GHz |

Table 4. Working Variables and Values

| ID | Parameter | 1st option (-) | 2nd option (+) |
|----|-----------|----------------|----------------|
| A  | Point of alignment | Align at the start | Align at the end |
| B  | LowPass filter (Software) | false | true |
| C  | Standardization | false | true |

could also be ranked randomly. Although this could imply more iterations, it is the right solution whenever the evaluator is less familiar with the task, since the process will, eventually, highlight the most relevant parameters anyway. New technicians can then interpret the results properly and re-establish the baseline knowledge for further experiments.

However, one could ask why an experienced technician needs to use our customized Six Sigma methodology, as he might already know which parameters can be the most relevant. The reason is that the methodology helps quantify each parameter’s impact in the experiment, keep control of the whole process, and justify decisions. However, it also helps identify some seemingly minor (but relevant) details that may otherwise go unnoticed. Usually, the more experienced a technician is, the quicker he (or she) will finish the task spending less time thinking about the decisions to take, which is not necessarily the best road map. Sometimes, experts tend to make hasty decisions, ignoring tiny but essential aspects of the experimental process. Similarly, in the Rubber Duck Debugging programming methodology [1], when you try to clearly explain the task you are performing (from the main goal to specific problems), you might be facing problems and inconsistencies you might have ignored previously. Since Six Sigma is based on Deming’s Plan-Do-Check-Act Cycle [9, 40], and it has shown its reliability in different applications in the industrial field, we show that using it in the SCA field can help to handle this kind of issue.

6.4 Analyze

From the ordered list, the three top parameters are chosen, and a simple DoE [31] process is carried out. In our case, we choose a two-level DoE with 3 variables (called factors in experimental design) for its simplicity and reliability. Therefore, 3 variables are investigated in two levels by performing $2^3$ experiments as follows:
(1) Create an experiment matrix for all possible combinations of parameters.
(2) Select the two most suitable values (or minimum and maximum) for each selected parameter.
(3) Enter the selected parameter values in the matrix.
(4) Proceed to acquire the traces of the eight experiments.
(5) Process these traces in the selected SCA suite and write down the results.
(6) Calculate the **Effect** and **Coefficient** of each parameter with the following formula:

\[
Effect = \sum_{i=1}^{4} \text{Maximum}_i - \sum_{i=1}^{4} \text{Minimum}_i
\]

\[
Coefficient = c = \frac{Effect}{2}. \tag{1}
\]

(7) Calculate the **Effect** and **Coefficient** of the interactions between the parameters with the following formula:

\[
Effect = \sum_{i=1}^{4} \text{positiveInteraction}_i - \sum_{i=1}^{4} \text{negativeInteraction}_i
\]

\[
Coefficient = c = \frac{Effect}{2}. \tag{2}
\]

Note: the sign of the interaction is the product of the code for the parameter levels: \(\text{Minimum} = -1, \text{Maximum} = 1\).

(8) Finally, calculate the results applying the DoE formula using the **Coefficients** for each factor:

\[
\text{DoE} = \sum_{i=1}^{8} \frac{R_i}{8} + c_A \ast A + c_B \ast B + c_C \ast C + c_{AB} \ast AB + c_{AC} \ast AC + c_{AC} \ast AC.
\]

(9) Optionally, the results can be plotted with a Pareto Chart for their better understanding (Figure 3, right).

(10) It is recommended to do more than one round of experiments with the same permutation list to compute the confidence of the results getting the variance \(\sigma\) and indicating the error. This will tell us whether the results are reliable or, on the contrary, they have been altered by some artifact during the experiments.

If the results of this process do not show a clear gain in any of the parameters, then the next parameters in the ordered list are selected and the DoE applied again. In the case of our experimental setup, we were not able to find any correlation with the basic setup (Figure 5). Some misalignments and very noisy signals were detected, so the three parameters shown in Table 4 were selected. The
Main effects and interactions

Fig. 4. Coefficients and interactions (left) and Pareto chart (right).

Fig. 5. Correlation results.

Table 5. Experiment Definition and Order

| Exp | Value | B Value | C Value |
|-----|-------|---------|---------|
| 1   | start | false   | false   |
| 2   | start | false   | true    |
| 3   | start | true    | false   |
| 4   | start | true    | true    |
| 5   | end   | false   | false   |
| 6   | end   | false   | true    |
| 7   | end   | true    | false   |
| 8   | end   | true    | true    |

DoE was applied again creating eight experiments with the limit in values as shown in Table 5. Note that the order of the experiment must follow exactly the one given in the table. Although the results will not be the optimal ones, it is important to finish the set of eight experiments without modifying any of the variables or their range. The results will be analyzed in the following step, taking into consideration whether it is mandatory to readjust them and make another iteration.
Table 6. The First Experiment Set Results

| Exp | A Value | B Value | C Value | AB Value | AC Value | BC Value | Round 1 | Round 2 | Round 3 | Std. Dev. | Average |
|-----|---------|---------|---------|----------|----------|----------|---------|---------|---------|-----------|---------|
| 1   | −1      | −1      | −1      | +1       | +1       | +1       | 0.0724  | 0.0808  | 0.0685  | 0.0063    | 0.0739  |
| 2   | −1      | −1      | +1      | +1       | −1       | −1       | 0.0726  | 0.0811  | 0.0612  | 0.0100    | 0.0716  |
| 3   | −1      | +1      | −1      | −1       | +1       | −1       | 0.0570  | 0.0748  | 0.0631  | 0.0090    | 0.0650  |
| 4   | −1      | +1      | +1      | −1       | −1       | +1       | 0.0597  | 0.0645  | 0.0664  | 0.0034    | 0.0636  |
| 5   | +1      | −1      | −1      | −1       | −1       | +1       | 0.1424  | 0.2098  | 0.1703  | 0.0339    | 0.1741  |
| 6   | +1      | −1      | +1      | −1       | +1       | −1       | 0.1428  | 0.2112  | 0.1707  | 0.0344    | 0.1749  |
| 7   | +1      | +1      | −1      | +1       | −1       | −1       | 0.1292  | 0.1634  | 0.1353  | 0.0182    | 0.1426  |
| 8   | +1      | +1      | +1      | +1       | +1       | +1       | 0.1294  | 0.1645  | 0.1351  | 0.0188    | 0.1430  |

Effect A 0.0901 B −0.0201 C −0.0006 AB −0.0116 AC 0.0012 BC 0.0001
Coefficient 0.0451 −0.0101 −0.0003 −0.0058 0.0006 0.0001

After acquiring and processing the traces, the “Round 1” column in Table 6 was filled in and the highest correlation level is obtained between the stored data and the power traces. It should be noticed that, as mentioned above, the set of eight experiments has been performed two times more (columns “Round 2” and “Round 3”) to ensure that the results are consistent. The coefficients are calculated with the averaged data (column “Average”).

After applying Equations (1) and (2), we derived the “Coefficients” of the tuning of the parameters and their interactions. The effect that each parameter has can be seen in Figure 3 (left). Also, in Figure 3 (right), we show how “vital few” variables have more influence over the results than the “trivial many” variables. This effect is also known as the 80/20 rule, stating that, for many events, the 80% of consequences come from 20% of causes. This has become a popular maxim in several fields such as economics, production, business, computer science, and so on. For instance, Lowell Arthur noted that, in computing, the 20% of the code has 80% of the errors. Also, he discovered that the 80% of a particular piece of software can be written in 20% of the total allocated time [35]. More examples can be found in literature [17].

In this case, the cumulative line crosses the line of 80% with the first variable, which basically means that the point of alignment is the only variable that is affecting the results in a significant way (it has almost the 80% of the influence in the results).

At this point, we have one set of the results and hence the factors of Equation (3) given by DoE. The parameters A, B, and C are given in Table 4, so the results given by Equation (3) mean that our experiment has better results when the Point of alignment is Align at the end and not when LowPass filter is applied. In this setup the Standardization effect is negligible.

\[
DoE = 0.101 + 0.035 \times A - 0.007 \times B + 0.0004 \times C + 0.0002 \times AB - 0.0003 \times AC + 0.0003 \times BC \tag{3}
\]

### 6.5 Improve

With the information derived in the previous steps, the expert should analyze the results of DoE, interpret them, and decide if they are good enough to implement them in the acquisition setup (the OK-criterium assists in this task). The expert can also dig deeper in his interpretation of the data and propose modifications in the range of some parameter(s), because he/she can derive from the results that the best approach might be keeping a better balance between the different parameters, instead of using the limited values for some of them. In this last case, the effect of the interactions can be very relevant to make a decision. However, if the decision is to discard the proposed changes, then two options are left: Go back to Step 3 (Analyze) and perform another iteration of DoE, making changes directly in the definition step based on the gathered knowledge; or perform the side-channel evaluation of the DUT with the baseline setup.
Table 7. Working Variables and Values

| ID | Parameter       | 1st option (−) | 2nd option (+) |
|----|----------------|----------------|----------------|
| A  | Number of traces | 3,000          | 5,000          |
| B  | Windowed resample| false          | true           |
| C  | Standardization | false          | true           |

Table 8. Results of the Three Experimental Rounds

| Exp | A | B | C        | Round 1 | Round 2 | Round 3 | Std. Dev. | Average |
|-----|---|---|----------|---------|---------|---------|-----------|---------|
| 1   | 3k| false | false | 0.5754  | 0.5649  | 0.6008  | 0.0185    | 0.5804  |
| 2   | 3k| false | true  | 0.5768  | 0.5691  | 0.6059  | 0.0194    | 0.5839  |
| 3   | 3k| true  | false | 0.5665  | 0.5805  | 0.6079  | 0.0210    | 0.5850  |
| 4   | 3k| true  | true  | 0.5708  | 0.5855  | 0.6146  | 0.0223    | 0.5903  |
| 5   | 5k| false | false | 0.5959  | 0.6031  | 0.6276  | 0.0166    | 0.6089  |
| 6   | 5k| false | true  | 0.5990  | 0.6078  | 0.6308  | 0.0164    | 0.6126  |
| 7   | 5k| true  | false | 0.6073  | 0.5978  | 0.6151  | 0.0086    | 0.6067  |
| 8   | 5k| true  | true  | 0.6115  | 0.6054  | 0.6178  | 0.0062    | 0.6116  |

As it can be noticed in Table 6, aligning at the end of the operation allows us to find better correlation values (approximately 0.1 larger values). We see some significant correlation spikes in comparison with the correlation obtained in non-leaking parts of the signal, but the correlation level is still too low (only the correlation in Experiments 5 and 6 is barely out of the confidence interval). Thus, we decided to perform another iteration (another DoE) modifying some variables, as shown in Table 7. We fix the variable “point of the alignment” in its maximum value (align at the end). In other words, we improved the alignment with the focus on the leaking part of the signal. As we obtained better results without applying a lowpass filter, we fix that variable (no lowpass filter), and we add two new variables to analyze (Number of traces and Compression technique). We keep the variable Standardization to discover whether it can improve the results with the new point of alignment. Repeating the same steps (but with the three new variables), we obtained the results shown in Table 8. In addition, Figure 4 shows the effect of each parameter in the results.

These results show that our experiment has better results when the number of power traces is 5,000, using a compression technique (Windowed resample = true) and using Standardization. Observe that the variable with more effect over the obtained correlation is the number of traces. Also, with this setup the results are slightly better when using standardization, contrary to the previous setup.

It must be mentioned that with this setup the alignment has been improved (due to the results of the previous iteration) and the OK-criterium has been accomplished in all the experiments. Figure 5 shows the differences between the traces taken with the preliminary setup and the traces obtained with the optimized setup. In the bottom chart named Correlation (improved setup) a high correlation spike can be seen at the end of the power trace, indicating the exact place in which the 8-bit value is stored in memory.

6.6 Control

Although there is no process running that should be controlled (strictly speaking), the recommendation is to document everything, every step taken, every guess done, every clue discovered, and so on. It is the key for having under control all the processes described above, and also for future improvements that can be done for this or other acquisition setups.
7 USE CASE 2: TEMPLATE ATTACK

In the previous section, we have shown how the proposed methodology is used for optimizing the acquisition setup. In this section, we describe how this methodology can be used to optimize an attack scenario. As in the previous section, we first briefly describe the presented use case, and then the 5 DMAIC scheme steps are discussed (except the control step). As the basis of the methodology have been already explained, we focus only on the results.

7.1 Use Case Description

In this use case, we use our customized Six Sigma methodology in an attack scenario. That implies optimizing parameters from both the signal processing phase and the analysis phase (Figure 1). A similar approach can be applied for optimizing only one of the phases, but we propose to optimize both at the same time, since signal processing parameters usually have a strong influence on the attack results (as shown in the current use case).

The target is an ATmega138P 8-bit microcontroller. Influenced by the result of the previous use case, we want to perform a template attack over a different device in a similar setup. We are storing data (8-bit values) in flash memory using a `memcpy()` operation (in a random address each time). During that operation, we take measurements of the power consumption of the device. As an attacker, our goal is to obtain the exact 8-value loaded in flash memory using template attacks [8, 10, 37]. Using our customized Six Sigma methodology, we have been able to successfully recover the 8-bit value performing template attacks. We performed two different phases: a profiling phase and an attack phase. In the profiling phase, we modeled the side-channel power consumption of the device during the `memcpy()` operation (taking measurements of the device loading random values into the memory), then in the attack phase, we were able to guess a (fixed) secret 8-bit value. Since performing a template attack is a complex process with lots of variables involved, we show how our customized Six Sigma methodology can help to optimize the attacking process.

7.2 Define

As mentioned above, the main goal of this phase is to define the setup requirements (see Table 9). For us, the main goal is to obtain an 8-bit value loaded into memory by using template attacks. The OK-criterium indicates that the correct key guess obtains a rank of 5 or less using template attacks. Table 9 shows the setup’s requirements.

7.3 Measure

With the baseline setup shown in Table 10, a round of 8-bit random values are sent to the DUT by the computer through the serial port, while the oscilloscope is triggered with a GPIO controlled by the ATmega platform, just after finishing the communication, when the internal computation in the DUT is supposed to start.

### Table 9. Goal and OK-criterium

| Goal | Successfully obtain the 8 bits loaded into memory using template attacks |
|------|------------------------------------------------------------------------|
| Requirements | - Same setup and device operational parameters fixed and constant in each experiment  
- Two sets of traces: one with random data (profiling phase) and other with constant data (attacking phase)  
- Oscilloscope: LeCroy Waverunner 9104  
- SW for the acquisition, signal processing, and data analysis  
- Current probe |
| OK-criterium | The correct candidate obtains a rank of 5 or less using template attacks |
Table 10. Tools List and Brief Description

| Tool                                      | Function                                                                 |
|-------------------------------------------|--------------------------------------------------------------------------|
| LeCroy Waverunner 9104 Oscilloscope       | 2 channels (Power consumption & triggering)                              |
|                                            | 20 GS/s (Single sample capture mode)                                     |
| The DUT                                   | Generates trigger trough GPIO                                            |
|                                            | Power source: continuous power supply                                    |
|                                            | Communicates with PC via serial port                                      |
| PC                                        | Communicates with the ATmega138P                                          |
|                                            | Controls the oscilloscope                                                |
| Current Probe                             | Tektronix CT1 current probe                                              |
|                                            | Connected in series with the DUT power line                              |
|                                            | Measure the power consumption                                           |

Table 11. Defined Variables

| Rank | Parameter       | Description                                                                 | Range          | Fixed Value |
|------|-----------------|------------------------------------------------------------------------------|----------------|-------------|
| 1    | Standardization | Removing the mean of the dataset can help to improve the results.           | Yes vs. No     |             |
| 2    | Points of Interest Nº | The selected POI will affect the templates and therefore the attack. We must select an optimal number of points of interest. | 1 vs. 3        |             |
| 3    | LowPass Filter (SW) | To filter the signal or not would affect the quality of the collected traces and the leakage. We should eliminate high-frequency noise but without destroying the leakage | With SW filter vs. Without SW filter |             |
| 4    | Nº Traces Profiling | Number of processed traces used for the profiling phase of the template attack. | 1k vs. 100k    | 20k         |
| 5    | Nº Traces Attack | Number of processed traces used for the attacking phase of the template attack. | 1k vs. 10k     | 1k          |
| 6    | Alignment       | When we align, we can choose different points as a reference.               | Start vs. End  | Start       |
| 7    | POI selection function | We can use different functions to select the points of interest of the traces ([SOST [20], SOSD [20], SNR [28], CORRELATION [24]]. | SOST vs. SNR   | SOST        |
| 8    | Compression technique (Windowed resample) | A compression technique can be used to reduce the dataset size and to improve the leakage (close points carry very similar information and noise can be reduced). Conversely, in some cases leakage can be destroyed. | With compression vs. without compression | No compression |
| 9    | Sampling Frequency | A high sampling frequency will improve the quality of the traces but also increase the data size. | 100MHz vs. 1GHz | 1 GHz       |

We set the range of the system’s variables and select three of them to check the effect they have on the success of the attack. Table 11 presents the variables considered for this experiment.

7.4 Analyze

Table 12 shows the three top variables to use in the DoE, creating 8 experiments. For each experiment 10k traces of random data were captured for the profiling phase and 1k traces of constant data were captured for the attack phase. The experiments and their results are given in Table 13. The parameters A, B, and C are given in Figure 6 (left), so the outcomes mean that our experiment has better results when the Number of Points of Interest is 3, and when we apply a LowPass filter.
Table 12. Working Variables and Values

| ID | Parameter                        | 1<sup>st</sup> option (−) | 2<sup>nd</sup> option (+) |
|----|----------------------------------|-----------------------------|---------------------------|
| A  | Standardization                  | false                       | true                      |
| B  | Nº POI                           | 1                           | 3                         |
| C  | Lowpass Filter (SW)              | false                       | true                      |

Table 13. Results of the Three Experimental Rounds

| Exp | A   | B   | C   | Round 1 | Round 2 | Round 3 | Round 4 | Round 5 | Round 6 | Std. Dev. | Average |
|-----|-----|-----|-----|---------|---------|---------|---------|---------|---------|-----------|---------|
| 1   | false | 1 | false | 46 | 63 | 42 | 27 | 47 | 27 | 13.65 | 42 |
| 2   | false | 1 | true  | 24 | 71 | 27 | 37 | 41 | 22 | 18.25 | 37 |
| 3   | false | 3 | false | 8  | 17 | 7  | 2  | 4  | 34 | 11.95 | 12 |
| 4   | false | 3 | true  | 5  | 13 | 3  | 2  | 3  | 4  | 4.05   | 5  |
| 5   | true  | 1 | false | 79 | 23 | 26 | 24 | 31 | 23 | 22.09 | 34.33 |
| 6   | true  | 1 | true  | 41 | 1  | 21 | 19 | 13 | 19 | 13.02 | 19 |
| 7   | true  | 3 | false | 23 | 16 | 2  | 5  | 5  | 32 | 11.96 | 13.83 |
| 8   | true  | 3 | true  | 9  | 9  | 5  | 1  | 2  | 2  | 3.61   | 4.67 |

Fig. 6. Coefficients and interactions (left) and Pareto chart (right).

Table 14. Working Variables and Values

| ID | Parameter                        | 1<sup>st</sup> option (−) | 2<sup>nd</sup> option (+) |
|----|----------------------------------|-----------------------------|---------------------------|
| A  | Strength of the lowpass filter   | 1                           | 10                        |
| B  | Nº POI                           | 3                           | 5                         |
| C  | Nº Traces Profiling phase        | 5 k                         | 15 k                      |

filter and Standardization. We can see that the number of POI is the variable with more effect in the results, but in this case variables A and C have also a significant effect (see Figure 6 (right)).

7.5 Improve

We reached the OK-criterium only in Experiment 8. Since we consider it a very poor result, we moved forward to increase the success rate and we performed another DoE iteration. Then, we redefined the three variables as shown in Table 14.
### Table 15. Results of the Three Experimental Rounds

| Exp | A  | B  | C  | Round 1 | Round 2 | Round 3 | Round 4 | Std. Dev. | Average |
|-----|----|----|----|--------|--------|--------|--------|----------|---------|
| 1   | 1  | 3  | 5k | 4      | 22     | 7      | 102     | 46.18    | 33.75   |
| 2   | 1  | 3  | 15k| 3      | 2      | 4      | 1       | 1.29     | 2.5     |
| 3   | 1  | 5  | 5k | 3      | 26     | 4      | 112     | 51.60    | 36.25   |
| 4   | 1  | 5  | 15k| 1      | 4      | 1      | 1       | 1.50     | 1.75    |
| 5   | 10 | 3  | 5k | 1      | 15     | 4      | 30      | 13.13    | 12.5    |
| 6   | 10 | 3  | 15k| 2      | 1      | 2      | 1       | 0.58     | 1.5     |
| 7   | 10 | 5  | 5k | 3      | 15     | 2      | 32      | 13.98    | 13      |
| 8   | 10 | 5  | 15k| 2      | 1      | 2      | 1       | 0.58     | 1.5     |

Fig. 7. Coefficients and interactions (left) and Pareto chart (right).

We modified the variable **Lowpass Filter** to **Strength of the lowpass filter** to obtain the proper value for characterizing the filter.\(^1\) Also, we wanted to know if adding more POI could improve the attack, so we modified the range (from 3 to 5). Moreover, we decided to check if by using more traces one could improve the attack significantly, so we added the parameter NO\(^0\) **Traces Profiling phase**. Table 15 shows the experiment definition and the results. Note that the **OK-criterium** was reached in the half of the experiments (experiments with 15k traces for profiling phase), so we can consider our main goal reached. Again, the parameters A, B, and C are given in Figure 7 (left), and this time the outcomes show that our experiment has better results with 15k **Traces for Profiling phase** and **Standardization**. Note that adding more POI does not improve the results in a significant way, so it is not worth increasing the computational effort required by adding more POI. In Figure 7 (right), we see that the variables with more impact to the results are the **Number of Traces for Profiling phase** (47.8%) and **Standardization** (24.8%).

### 8 USE CASE 3: LEAKAGE ASSESSMENT (STATISTICAL TESTS)

The following use case shows how our customized Six Sigma methodology can be used in a side-channel evaluation scenario to optimize the leakage assessment process with traditional statistical tests (e.g., \( t \)-test and \( X^2 \)-test). We first present a short overview of the current leakage assessment methodologies. Then, we describe our use case and the five DMAIC scheme steps.

---

\(^1\)The employed “Lowpass Filter” is a fast bidirectional filter that makes each sample of the power trace a weighted average of the previous sample and the current sample. The strength can be controlled with the weight \( W \): 
\[
\text{sample}_i = \frac{(W \cdot \text{sample}_{i-1} + \text{sample}_i)}{(W + 1)}.
\]
8.1 Use Case Description

In this use case, we use our customized Six Sigma methodology in a leakage assessment scenario (using the “classical” statistical test approach). Again, that implies optimizing parameters from both the signal processing phase and the analysis phase (Figure 1).

Our target is an STM32F417 32-bit microcontroller implementing a software AES-128 implementation [2, 5, 14]. As evaluators, our goal is to detect leakage with the least amount of resources possible (fast and efficient). Essentially, what we want to do is to prove that there exists a dependency between the data processed by the cryptographic algorithm and the power consumption. Commonly, the DUT is fed with two distinct types of data (fixed vs. fixed, fixed vs. random, semi-fixed vs. random, etc.) and the evaluator tries to confirm that there exists a significant statistical difference between both sets of traces.

As mentioned above, the TVLA is the main statistical tool used in side-channel leakage detection, since it is fast and versatile. However, it can also bring up false positives [41]. For this reason, in real scenario evaluations, a semi-fixed vs. random test is performed (instead of a fixed vs. random test), and also for the same reason new suggested evaluation techniques are published [15, 32, 38, 44].

In other words, instead of feeding the cryptographic device with fixed values, we generate specific test vectors that force a certain intermediate value (or its Hamming Weight) to remain constant. Then, we compare this power consumption, with the one generated when the device is encrypting fully random values. The usage of semi-fixed vectors makes the TVLA test more reliable, but it increases the complexity of the evaluation with more parameters to tune. For this reason, our customized Six Sigma fits also perfectly in this use case.

8.2 Define

Table 16 shows the main goal and OK-criterium of the phase. Here, the main goal is to detect leakage on an AES-128 bit software implementation running on an STM32F417 microcontroller. Thus, the OK-criterium in this use case is to obtain a $p$-value greater than $10^{-5}$ at the same time in two different sets of samples.

8.3 Measure

With the baseline setup parameters as in Table 17, a round of random or semi-fixed 128-bit values (randomly interleaved) is sent to the DUT by the computer through the serial port, while the oscilloscope is triggered with a GPIO controlled by the DUT just after finishing the communication (when the internal computation in the DUT is supposed to begin). We evaluate the system’s variables and select three of them to check the effect they have on the success of the attack (Table 18).
Table 17. Tools List and Brief Description

| Tool                        | Description                                                                 |
|-----------------------------|-----------------------------------------------------------------------------|
| LeCroy Waverunner 9104      | 2 channels (Power consumption & triggering)                                  |
| Oscilloscope                | 20 GS/s (Single sample capture mode)                                        |
| The DUT                     | Generates trigger trough GPIO                                                |
|                             | Power supplied through external batteries                                   |
|                             | Communicates with PC via serial port                                         |
| PC                          | Communicates with the STM32F417 microcontroller                             |
|                             | Controls the oscilloscope                                                   |
|                             | Generates random data to encrypt                                             |
| Current Probe               | Tektronix CT1 current probe                                                 |
|                             | Connected in series with the DUT power line                                 |
|                             | Measure the power consumption                                               |

Fig. 8. Coefficients and interactions (left) and Pareto chart (right).

In this case, we have selected the three variables shown in Table 19. We want to know which fixed intermediate value produces higher $p$-values, so we select variable A (Intermediate Value). Also, we want to select the proper Hamming Weight range for the semi-fixed set (for the selected intermediate value). Finally, we want to know which statistical test ($t$-test or $X^2$-test) works better in this evaluation.

8.4 Analysis

In this step, the DoE was applied on the three aforementioned variables creating eight experiments. The experiments and their results are presented in Table 20. The parameters A, B, and C are given in Figure 8, and the outcomes mean that our experiment has better results when the Intermediate value is SubBytes, the HW range is 80–100, and we use the $X^2$-test for the experiment.

8.5 Improve

Since the OK-criterium has been reached in most of the experiments, it is not necessary to perform another iteration of the DoE. We conclude that the device leaks information through its power consumption. From the results of the previous step, it is noticeable that the output of SubBytes is leakier than the one of AddRoundKey. Also, it can be seen that the higher Hamming Weight value we fix for the semi-fixed dataset, the larger statistical differences are observed in the traces. As it is observed in the Pareto chart (Figure 8) there are variables with more effect than others, but the particularity is that the three of them have a strong influence on the results.

Conversely, it is known that $X^2$-test gives better results in the cases where the leakage has multivariate behavior. Nevertheless, its authors confirmed that the technique is feasible also in univariate cases (our case) [32]. In our experiment, while using the $t$-test, we were not able to
Table 18. Defined Variables

| Rank | Parameter          | Description                                                                 | Range                      | Fixed Value               |
|------|--------------------|-----------------------------------------------------------------------------|----------------------------|----------------------------|
| 1    | Intermediate value | Semi-fixed traces are generated such that certain intermediate value Hamming Weight is always between a range. The intermediate value targeted can have an influence in the leakage detection. | SubBytes vs. AddRoundKey   |                             |
| 2    | HW range           | The HW range of the generated semi-fixed traces can have an influence in the leakage detection | 40-60 vs. 80-100           | t-test vs. X²-test         |
| 3    | Statistical test   | Although the t-test is considered to be the main statistical tool in leakage detection, recently the X²-test for the same purpose has been proposed. The usage of one statistical tool or another can have a clear influence on the obtained results. |                             |                             |
| 4    | Test Vector        | The nature of the test vector may affect the results. We could use the classical fixed vs. random approach, or we can generate specific vectors that force the HW of one intermediate value. | "Fixed vs. random" vs. "Semi-Fixed vs. random" |                             |
| 5    | Number of traces   | The number of traces affects the confidence of the results. In this case, in each experiment, we need two sets of traces taken with the same setup. | 1k vs. 100k                | 5k                         |
| 6    | Standardization    | Removing the mean of the dataset can help to improve the results.           | Yes vs. No                 | With standardization vs. No compression |
| 7    | Compression (Windowed resample) | A compression technique can be used to reduce the dataset size and improve the leakage (close points carry similar information and noise can be reduced). Conversely, in some cases leakage can be destroyed. | With compression vs. without compression |                             |
| 8    | Alignment          | When we align, we can choose different points as a reference.              | Start vs. End              | Start                      |
| 9    | LowPass Filter (SW) | To filter the signal or not would affect the quality of the collected traces and the leakage. We should eliminate high-frequency noise but without destroying the leakage | With SW filter vs. Without SW filter | Without LowPass filter     |
| 10   | Sampling Frequency | A high sampling frequency will improve the quality of the traces but also increase the data size. | 50MHz vs. 20GHz            | 1 GHz                      |

exceed the threshold by fixing the HW of the AddRoundKey operation with 5,000 traces. Surprisingly, using X²-test, we were able to find leakage with the same number of traces. In other words, in our experiments, the X²-test was more sensitive and allowed us to detect differences better than t-test. Thus, although X²-test requires more computational effort, using it improves the results in particular cases (as it can be seen in Figure 9).

9 USE CASE 4: LEAKAGE ASSESSMENT (DEEP LEARNING)

Recently, a new leakage assessment method based on deep learning has been proposed [44]. Their idea is to train a neural network that works as a classifier over the two sets of data (e.g., random vs. fixed). If the neural network is able to distinguish between these two sets of data, then it can be assumed that they are statistically different with a certain probability. Additionally, dealing
Table 19. Working Variables and Values

| ID | Parameter      | 1st option (-) | 2nd option (+) |
|----|----------------|----------------|----------------|
| A  | Intermediate value | SubBytes       | AddRoundKey    |
| B  | HW range        | 40–60          | 80–100         |
| C  | Statistical test | $t$-test       | $\chi^2$-test  |

Table 20. Results of the Six Experimental Rounds

| Exp | A            | B             | C             | Round 1  | Round 2  | Round 3  | Std. Dev | Average |
|-----|--------------|---------------|---------------|----------|----------|----------|----------|---------|
| 1   | SubBytes    | 40–60         | $t$-test     | 14.105   | 12.365   | 11.250   | 1.439    | 12.573  |
| 2   | SubBytes    | 40–60         | $\chi^2$-test| 9.889    | 13.256   | 11.143   | 1.702    | 11.429  |
| 3   | SubBytes    | 80–100        | $t$-test     | 64.930   | 51.837   | 51.198   | 7.750    | 55.988  |
| 4   | SubBytes    | 80–100        | $\chi^2$-test| 59.621   | 53.409   | 55.266   | 3.188    | 56.099  |
| 5   | AddRoundKey | 40–60         | $t$-test     | 3.886    | 3.907    | 3.620    | 0.160    | 3.804   |
| 6   | AddRoundKey | 40–60         | $\chi^2$-test| 6.802    | 6.025    | 6.182    | 0.411    | 6.337   |
| 7   | AddRoundKey | 80–100        | $t$-test     | 5.790    | 5.641    | 4.613    | 0.641    | 5.348   |
| 8   | AddRoundKey | 80–100        | $\chi^2$-test| 48.948   | 41.981   | 38.256   | 5.427    | 43.061  |

Fig. 9. Leakages of AddRoundKey intermediate value: $t$-test vs. $\chi^2$-test (Experiments 7 and 8).

with pre-processing problems like misalignment is becoming less problematic, if at all needed. Conversely, the inclusion of deep neural networks in the leakage assessment adds complexity to the problem. In this section, we show the viability of our customized Six Sigma methodology in deep learning leakage assessment, helping to discern which is the best setup for this purpose.

9.1 Use Case Description

In this use case, we use our customized Six Sigma methodology in a leakage assessment scenario (using the approach using deep learning purposed in Reference [44]). In this case, we are considering only parameters from the analysis phase (Figure 1), but a similar approach can be used combining several phases.

It should be noticed that the target, preliminary setup, and main goal are the same as in the previous use case. The neural network has the same architecture as defined in the original paper [44]. The set of “semi-fixed vs. random” power traces is divided into the training set and validation set. Then, from the training set, we took 1k and 3k traces to train the model, and from the validation set (10k traces), we compute a binomial test (as is suggested in the original paper) to obtain a
Table 21. Defined Variables

| Rank | Parameter | Description | Range | Fixed Value |
|------|-----------|-------------|-------|-------------|
| 1    | Standardizing | To reach a homogeneous range between all input points and weights, enabling efficient training. | No vs. Yes | |
| 2    | Nº Samples (per trace) | The number of samples will have an influence in the number of neurons of the neural network, which could have an impact in the results. | 2500 vs. 5000 | |
| 3    | Nº Traces (Training) | The number of traces used in the training phase may affect the obtained p-values. | 1000 vs. 3000 | |
| 4    | Nº Traces (Validation) | The number of traces used in the validation phase may affect the obtained p-values. | 1k vs. 100k | 10k |
| 5    | Neural Network Architecture | The architecture of the neural network can follow the architecture proposed in Reference [44] or design our own. | Neural Network Architecture | Neural Network Sequential Model vs. CNN |

probability (p-value) that indicates whether there is a significant statistical difference between the two sets of traces (“semi-fixed vs. random”).

Note that a comparison with the previous work is difficult, because we are using a microcontroller instead of an FPGA platform, and a different encryption algorithm (AES-128 instead of PRESENT [6]).

9.2 Define

As mentioned above, the setup requirements, main goal, and OK-criterium are the same as in the previous use case (Section 8.2) and can be found in Table 16.

9.3 Measure

The baseline setup (Table 17) is also the same as in the previous use case in Section 8.3. However, the leakage assessment method is completely different, and thus, the defined variables, too.

It should be noted that general signal processing variables that are (most of the time) considered in this step (Lowpass filtering, number of traces, compression techniques, etc.) were not applied, since deep learning approaches in SCA are often advocated by the needlessness of pre-processing techniques. However, the authors in Reference [44] propose standardizing the training and validation sets to obtain a homogeneous range between all input points and weights. Hence, we will consider standardizing as a variable in our experiments to check whether this method improves the results of our use case. Other considered variables are the number of traces used for training and validation and the number of samples per trace in each one of them. Table 21 shows the defined variables of this step.

9.4 Analysis

The DoE was applied to the three variables shown in Table 22 creating eight experiments. The experiments and their result are represented in Table 23. Figure 8 shows parameters A, B, and C. The outcomes mean that our experiment has better results with No Standardizing, using 2,500 samples per trace and 3k traces for training.
Table 22. Working Variables and Values

| ID | Parameter                  | 1st option (−) | 2nd option (+) |
|----|----------------------------|----------------|----------------|
| A  | Standardizing              | No Standardizing | Standardizing  |
| B  | N° Samples (per trace)     | 2,500          | 5,000          |
| C  | N° Traces (Training)       | 1,000          | 3,000          |

Table 23. Results of the Six Experimental Rounds

| Exp | A            | B | C  | Round 1 | Round 2 | Std. Dev. | Average |
|-----|--------------|---|----|---------|---------|-----------|---------|
| 1   | No Standardizing | 2k | 1k | 299.25  | 415.13  | 81.940    | 357.19  |
| 2   | No Standardizing | 2k | 3k | 910.65  | 956.40  | 32.350    | 933.53  |
| 3   | No Standardizing | 5k | 1k | 244.82  | 323.01  | 55.289    | 283.91  |
| 4   | No Standardizing | 5k | 3k | 763.51  | 910.65  | 104.044   | 837.08  |
| 5   | Standardizing      | 2k | 1k | 208.35  | 206.70  | 1.167     | 207.52  |
| 6   | Standardizing      | 2k | 3k | 667.50  | 729.90  | 44.123    | 698.70  |
| 7   | Standardizing      | 5k | 1k | 148.04  | 169.00  | 14.821    | 158.52  |
| 8   | Standardizing      | 5k | 3k | 638.05  | 686.51  | 34.266    | 662.28  |

Fig. 10. Coefficients and interactions (left) and Pareto chart (right).

9.5 Improve

Since the OK-criterium has been reached in all the experiments, it is not necessary to perform another iteration of the DoE. From the results of the previous step, we conclude that the number of training traces is the parameter that has more effect on the obtained p-value, as it can be seen in the Pareto Chart (Figure 10 (right)). Note that applying the standardizing technique does not improve the results. In this particular case, comparing the same experiments with and without standardizing the traces, we obtain slightly better results without standardizing as pre-processing. However, it is not necessarily conclusive that pre-processing is not required.

10 CONCLUSION

Our results when using this customized Six Sigma methodology demonstrate the suitability of this method for improving the SCA process in its different stages; from the basis of the process (which is improving the quality of the acquired side-channel measurement) to the performance of any kind of side-channel attack or leakage assessment technique.

Moreover, we have shown how our Six Sigma methodology can reduce the uncertainty associated with the SCA, helping technicians to interpret the results and discover the root causes of the phenomena that occurred during the process. During the process, the evaluator identifies the
parameters that have more influence on the results of a certain experiment and is able to adjust them to an optimal value. The methodology steps proposed are simple, methodical, and very helpful when dealing with security evaluations. This approach can be helpful to any researcher or security evaluator in a lab; it allows technicians without a deep knowledge of all the basics involved in these methods to implement and interpret side-channel evaluations properly. The methodology can also be used by experts when dealing with new tasks (e.g., regarding new setups, devices, attacks, or leakage assessment methods), as in such a case our methodology could guide the evaluator to find the best set of variables and speed up the evaluation process. In fact, it can be used by anyone who wants to develop an experimentation process in a structured and methodological way, identifying effects and root causes, keeping control of the whole process, and identifying essential aspects of the experimental process that could have been ignored otherwise.

REFERENCES

[1] Thiago Monteiro. 2019. *Improve how you code: Understanding Rubber Duck Debugging*. Retrieved from https://gitduck.com/blog/improve-how-to-code-with-rubber-duck-debugging/.

[2] Federal Information Processing Standards Publication 197. 2001. Announcing the Advanced Encryption Standard (AES). Retrieved from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[3] Melissa Azouaoui, Davide Bellizia, Ilieana Buhan, Nicolas Debande, Sebastien Duval, Christophe Giraud, Eliane Jaulmes, Francois Koeune, Elisabeth Oswald, Francois-Xavier Standaert, and Carolyn Whitnall. 2020. A Systematic Appraisal of Side Channel Evaluation Strategies. *Cryptology ePrint Archive, Report 2020/1347*. Retrieved from https://eprint.iacr.org/2020/1347.

[4] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, Francois-Xavier Standaert, and Nicolas Veyrat-Charvillon. 2011. Mutual information analysis: A comprehensive study. *J. Cryptol.* 24, 2 (2011), 269–291.

[5] Guido Bertoni, Luca Breveglieri, Pasqualina Fragno, Marco Macchetti, and Stefano Marchesin. 2003. Efficient software implementation of AES on 32-bit platforms. In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’02)*. Springer-Verlag, London, UK, 159–171. Retrieved from http://dl.acm.org/citation.cfm?id=648255.752733.

[6] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. 2007. PRESENT: An ultra-lightweight block cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’07)*. Springer-Verlag, Berlin, 450–466. DOI: https://doi.org/10.1007/978-3-540-74735-2_31.

[7] Eric Brier, Christophe Clavier, and Francis Olivier. 2004. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems (CHES’04)*, Marc Joye and Jean-Jacques Quisquater (Eds.). Springer Berlin, 16–29.

[8] Suresh Char, Josyula R. Rao, and Pankaj Rohatgi. 2002. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems (CHES’02)*. Springer Berlin, 13–28.

[9] Chuen-Sheng Cheng, S.-C. Lee, P.-W. Chen, and K.-K. Huang. 2012. The application of design for Six Sigma on high level smart phone development. *J. Qual.* 19 (1 2012), 117–136. DOI: https://doi.org/10.6220/jqj.2012.19(2).02.

[10] M. O. Choudary and M. G. Kuhn. 2018. Efficient, portable template attacks. *IEEE Trans. Inf. Forens. Secur.* 13, 2 (Feb. 2018), 490–501.

[11] Common Criteria. 2017. Common Criteria v3.1. Release 5. Retrieved from https://www.commoncriteriaportal.org/cc/index.cfm.

[12] Jeremy Cooper, Elke DeMulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, Pankaj Rohatgi, et al. 2013. Test vector leakage assessment (TVLA) methodology in practice. In *International Cryptographic Module Conference*, Vol. 20.

[13] Veronica Czitrom and Patrick D. Spagon. 1997. *Statistical Case Studies for Industrial Process Improvement*. Society for Industrial and Applied Mathematics. DOI: https://doi.org/10.1137/1.9780898719765.

[14] Joan Daemen and Vincent Rijmen. 2002. *The Design of Rijndael: AES—The Advanced Encryption Standard* (Information Security and Cryptography) (1st ed.). Springer.

[15] A. Ding, C. Chen, and Thomas Eisenbarth. 2016. Simpler, faster, and more robust T-test based leakage detection. In *COSADE*. Springer International Publishing.

[16] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. 2015. Making masking security proofs concrete. In *Advances in Cryptology (EUROCRYPT’15)*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer Berlin, 401–429.
[45] Carolyn Whitnall and Elisabeth Oswald. 2019. *A Critical Analysis of ISO 17825 ("Testing Methods for the Mitigation of Non-invasive Attack Classes Against Cryptographic Modules").* 256–284. DOI: https://doi.org/10.1007/978-3-030-34618-8_9

[46] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. 2019. Methodology for efficient CNN architectures in profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2020, 1 (2019).

Received June 2020; revised October 2020; accepted January 2021