Strong PUFs and their (Physical) Unpredictability - A Case Study with Power PUFs -

Michał Parusiński
michal.parusinski@uclouvain.be

Saloomeh Shariati
saloomeh.shariati@uclouvain.be

Dina Kamel
dina.kamel@uclouvain.be

François Xavier-Standaert
fstandae@uclouvain.be

Université Catholique de Louvain
ICTEAM - ELEN - Crypto Group
Bâtiment Maxwell, Place du Levant, 3
B - 1348 Louvain-la-Neuve, Belgium

ABSTRACT

Physically Unclonable Functions are more and more important in the design of secure hardware, as they can ensure properties that conventional cryptography can not. In this paper we clarify the relations between strong PUFs and their unpredictability. For this purpose we first introduce an alternative definition for physical unpredictability, where the adversary can probe the physical responses of the Physical Function. We then illustrate physical unpredictability with a new instance of a PUF, based on the variability of the power consumption of a 65-nanometer chip. For this new PUF, we also evaluate the relation between robustness, unclonability and physical unpredictability. Our new definitions highlights the importance for designers to take into account if physical probing is possible or not (since the power of modeling attacks highly depends on this assumption). It also suggests that physical unpredictability is a generally useful tool for evaluating the unclonability of PUFs (since it can generate warning signals regarding the independence assumption that is frequently exploited for this purpose).

1. INTRODUCTION

Physical(ly) Unclonable Functions (PUFs) are important primitives in secure hardware design. One of their main interest is that they can ensure properties that cannot be guaranteed by mathematical means only. As a result, PUFs can (ideally) be used as an interesting complement to traditional cryptographic objects. But quite naturally, a central question for this interest to materialize is to quantify these physical properties. It can be related to the long standing problem whether PUFs can be "strong", i.e. whether they remain secure after the observation of a large number of challenges by the adversary [1]. The evaluation framework introduced by Armknecht et al. and summarized in Figure 1 is a sound foundation for answering such questions [2]. Its main goal is to provide a set of minimum security definitions (namely robustness, unclonability and unpredictability) that can be evaluated by hardware engineers, and exploited in cryptographic protocols. Whenever considering weak PUFs with a limited challenge set (e.g. the image-based case in [3]), the resulting trade-off is quite well understood. Namely, the physical function will generate noisy responses for which the hope is to observe large inter-class and low intra-class variance. High robustness is then obtained at the output, by embedding an Error Correction Code (ECC) in the extraction step, which comes at the cost of a reduced unclonability (i.e. by lifting the noise, some small details of the response are also removed). By contrast, as soon as strong PUFs are considered (i.e. for larger challenge sets), the problem of the unpredictability of the corresponding outputs comes into play, leading to a more intricate situation.

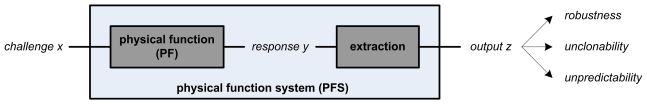

Figure 1: Physical Function System (simplified).

Our contribution. In this paper, we aim to clarify the relations between strong PUFs and their unpredictability. For this purpose, we first observe that in the definition of unpredictability by Armknecht et al., the adversary is only able to observe the output of the target Physical Function System (PFS). As a result, we first provide an alternative definition of *physical unpredictability*, where the adversary is additionally able to probe the Physical Function's (PF) noisy responses. We then illustrate this definition with a new instance of PUF, exploiting the variability of the power consumption traces of a chip (next denoted as Power PUF). Our main motivation for studying this instance is that it can directly take advantage of the vast literature on side-channel analysis in the evaluation of modeling attacks. Based on practical experiments with 65-nanometer chips, we finally evaluate the trade-off between robustness, unclonability and physical unpredictability for an exemplary Power PUF. It

allows us to put forward a first complete application of the evaluation framework in [2], leading to a realistic view of the security levels achieved. These physical unpredictability experiments also suggest two important messages for the security evaluation of PUF-based applications.

The first message relates to the practical relevance of the unpredictability provided by physical functions, and its connection with adversarial means. In this respect two situations can occur. On the one hand, it can be considered that the adversaries directly probe the PUF responses, in which case strong predicting attacks are usually possible (since these adversaries can directly model the physical signal on which the PUF is based). Physical unpredictability aims at capturing this (worst-case) situation. On the other hand, it can be assumed that adversaries can only probe the PUF outputs, in which case predicting the PUF behavior may be much more difficult (in particular in situations where its outputs are processed with a cryptographic hash function, e.g. as in [4]). The unpredictability defined in [2] captures this alternative scenario. Depending on the implementation choices of system designers and on the hardware assumptions made about where the probing of the PUFs can reasonably occur, one or the other definition will be preferred.

The second message relates to the relation between physical unpredictability and the evaluation of unclonability. Namely, it is a usual goal of PUF designers to extract the best possible unclonability with high robustness, and at the lowest hardware cost. In general, there are two main solutions to improve the physical unclonability of a (strong) PUF instance: either by increasing the challenge set and concatenating the corresponding outputs (which is the cheapest solution), or by using multiple - physically distinct - functions and concatenating the corresponding outputs (which is more expensive). In practice, the best option naturally is to exploit the first solution as much as possible. But this option usually faces the problem that estimating the unclonability of a PUF is a difficult and expensive task, as it requires many samples and depends on a probability distribution that is a priori unknown (since physical). As a result, a convenient solution is to rely on an independence assumption in the evaluation of unclonability. But this solution then faces the other problem that the PUF responses corresponding to multiple challenges may not be independent. Interestingly, we show in this work that the (easier to evaluate) physical unpredictability that we introduce can be used to generate warning signals in this respect. Namely, the concatenation of strong PUF outputs to increase unclonability should anyway be limited to the number of challenges for which physical unpredictability can be guaranteed. As a result, we believe it is a safe practice to always evaluate the physical unpredictability of a strong PUF, in order to properly assess the security level it provides.

The following of the paper is structured as follows: Section 2 briefly describes previous works. In Section 3, we bring the definitions of a Physical Function System and its important properties. Next, we present our new instance of Power PUF in Section 4 and experimentally evaluate its properties, namely robustness, physical unpredictability, and physical unclonability in Sections 5,6 and 7 respectively.

2. PREVIOUS WORK

PUFs were introduced by Pappu [5, 6]. Since then, many different physical objects have been proposed as PUF candidates, including Optical PUF [5,6], Coating PUF [7], Silicon PUF [8–10], SRAM PUF [11], Paper PUF [12–15], Laser-Written PUF [16,17], etc. Several application fields have also been proposed. Weak PUFs with a few number of challenge response pairs are mostly targeted for anti-counterfeiting [18–20] and key generation [21]. Strong PUFs with a potentially very large number of challenge response pairs have been mostly aimed for authentication [1, 8] or in the design of block ciphers [22]. For strong PUFs, unpredictability of their responses have been questioned through various modeling attacks [1, 23]. The modeling attacks try to build a numerical model which correctly predicts the PUF responses to arbitrary challenges with a high probability. Some machine learning techniques used as tools for modeling attacks on strong PUFs include Logistic Regression [1, 24], Artificial Neural Networks [23, 25], Support Vector Machines [25], etc.

3. DEFINITIONS

In this section, we define a Physical Function System (PFS) and its properties namely robustness, physical unclonability and unpredictability, based on the framework of Armknecht et al. [2]. We also provide an alternative definition of *physical unpredictability* that captures unpredictability of a PFS when the adversary can probe the PUF responses.

A *Physical Function System* PFS is a probabilistic procedure which takes as input a challenge $x \in \mathcal{X}$ and generates an output $z \in \mathcal{Z}$ (See Figure 1). A PFS is executed in two different modes of operation. In setup mode where the PUF is evaluated for the first time, in addition to the output z, PFS also generates as output some *helper data* h. In reconstruction mode, where the PUF is evaluated again, the helper data generated in setup mode is used as an input to produce the output. In this mode, the input helper data is returned unchanged to the output. The Physical Function System is thus defined as $\mathsf{PFS}(x, h) \rightarrow (z, h')$ such that helper data is an empty string $h = \epsilon$ in setup mode, and the input helper data is returned unchanged $h = h'$ in reconstruction mode. The important properties of the above Physical Function System are formally defined as follows:

1. **Robustness**. Robustness of a PFS is represented by the probability that for a given PUF, the output generated by reconstruction phase matches the value generated in setup phase using its corresponding helper data h. Formally, a PFS is ρ_{PFS}-robust if:

$$Pr\left[\mathsf{PFS}(x, h) \rightarrow (z, h) : \mathsf{PFS}(x, \epsilon) \rightarrow (z, h)\right] \geq \rho_{\mathsf{PFS}}, \tag{1}$$

   where ':' denotes the conditional probability.

2. **Physical Unclonability**. Physical Unclonability relates to the probability that an adversary can build (or find) a PFS' which shows the same behavior as another PFS. By the same behavior we mean that PFS' generates the same output as PFS providing that PFS' uses the helper data generated by PFS in setup mode. In general, this probability can be assessed for different

scenarios. For example, one can imagine a scenario where the adversary is confined to use facilities of the honest manufacturer to create the PUFs. Or, he can use more accurate creation tools than that of honest manufacturer. As the evaluation of unclonability in the second scenario is related to detail and cost of the manufacturing process, in practice the probability of cloning is usually evaluated for the first scenario by finding the probability that an honest manufacturer creates a clone by coincidence. In this case, *physical unclonability* β for the adversary who is limited to the facilities of an honest manufacturer can be defined as:

$$Pr\big[\mathsf{PFS}'(x,h) \to (z,h) : \mathsf{PFS}(x,\epsilon) \to (z,h)\big] \leq \beta.\rho_{\mathsf{PFS}}, \quad (2)$$

where PFS and PFS' have two different PUFs which are created by the honest manufacturer. The notion of robustness ρ_{PFS} is integrated into the definition of physical unclonability because it is a natural upper bound for the probability of generating a clone.

3. **Unpredictability**. Unpredictability is another property of a Physical Function System which is important specially for strong PUFs. Roughly speaking, unpredictability is related to the probability that an adversary wins a predicting experiment such that he predicts the output corresponding to a new challenge from previously observed challenge-output pairs. The predicting experiment to a Physical Function System is formally defined as follows. It is executed in learning and test phases. In learning phase, an adversary A learns the evaluations of PFS on a set of challenges $\{x_1, x_2, \ldots, x_q\} \in \mathcal{X}$. The adversary trains a model Model from the set of observed evaluations. Then in the test phase, he predicts $z = \mathsf{PFS}(x)$ for a new challenge $x \in \mathcal{X} : x \notin \{x_1, x_2, \ldots, x_q\}$ using Model.

   As mentioned in the introduction, either the adversary is only able to observe the output z and helper data h, or he can additionally observe the physical response y of the PUF. In the first scenario, the predicting experiment $\mathbf{Exp}_{\mathsf{A}}^{\mathrm{unp}}(q)$ is illustrated inside the plain box in Figure 2. In the second scenario, the "physical" predicting experiment $\mathbf{Exp}_{\mathsf{A}}^{\mathrm{phy-unp}}(q)$ also includes observing the PUF responses to the challenges and is illustrated inside the dashed box in Figure 2. The *unpredictability* η and *physical unpredictability* κ can thus be defined as the probability of success of the predicting experiments $\mathbf{Exp}_{\mathsf{A}}^{\mathrm{unp}}(q)$ and physical predicting experiment $\mathbf{Exp}_{\mathsf{A}}^{\mathrm{phy-unp}}(q)$, respectively:

$$Pr\big[\mathbf{Exp}_{\mathsf{A}}^{\mathrm{unp}}(q) \to z : \mathsf{PFS}(x,\epsilon) \to (z,h)\big] \leq \eta.\rho_{\mathsf{PFS}},$$

$$Pr\big[\mathbf{Exp}_{\mathsf{A}}^{\mathrm{phy-unp}}(q) \to z : \mathsf{PFS}(x,\epsilon) \to (z,h)\big] \leq \kappa.\rho_{\mathsf{PFS}}.$$

   Again, the robustness of PFS marks an upper bound on the predictability of its outputs. In following sections, we first introduce our instance of PFS, i.e. a Power PUF system, and then we evaluate its important properties, i.e. robustness, physical unpredictability and physical unclonability (and their relation).

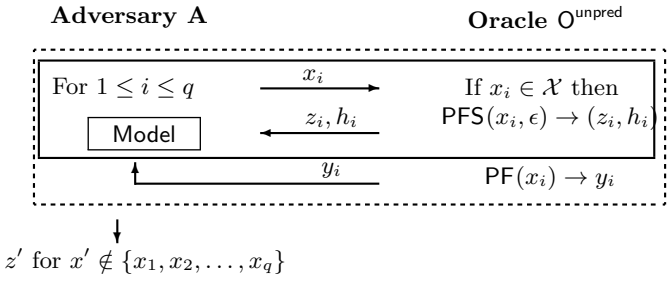

Figure 2: Description of the predicting experiment $\mathbf{Exp}_{\mathsf{A}}^{\mathrm{unp}}(q)$ in the solid box and physical predicting experiment $\mathbf{Exp}_{\mathsf{A}}^{\mathrm{phy-unp}}(q)$ in the dashed box.

4. POWER PUF INSTANCE

In this section we first describe the concept of power PUF and then introduce our instantiation. Power PUF is based on the variations of power traces between chips due to the inherent process variability in nanoscale technologies. It was shown in [26] that process variations imply increased inter-chip variability such that building a leakage model for one chip is not optimal anymore for being exploited against another chip. Considering the input transition as the challenge of the Power PUF, the response is the power trace of the chip. In Figure 3, the power traces of our PUF instance, described later in this section, are plotted for a single challenge and multiple chips. These traces can be divided into a dynamic part and a static part. By visual inspection, the dynamic power corresponds to the varying part of the trace which can further be divided into a nearly invariant part (part A) and a highly varying part (part B), while the static power corresponds to the constant part of the traces which is considered noise (part C). As can be seen from this figure, the most useful part of the traces for power PUF is part B. As discussed in [27], the impact of process variability on the dynamic power can be explained by the presence of random glitches generated by variability-induced unbalanced logic paths that are magnified when the computation delays increase at low voltage. Moreover, due to the accumulation effect of the random glitches along the logic paths, variations between chips are more significant in the late samples of the dynamic power trace, i.e. in part B. Glitch PUFs proposed in [28, 29] also use the same phenomenon and exploit the impact of process variations on glitches. The difference is that Glitch PUFs directly count the glitches to build the response, while Power PUFs use the indirect reflection of varying glitches on the power traces. In following, we describe the components of our instance of power PUF system according to the evaluation framework of [2] (See Figure 1).

Creation process: Creating Power PUF involves some key parameters relating to the circuit design, technology and the supply voltage. In following, we describe these parameters for our instance of Power PUF. We choose the circuit type to be the AES S-box and its implementation to be composite field arithmetic [30] using static CMOS logic style as described in [31]. We select the 65 nm CMOS technology as it features high process variations [32, 33]. Finally, the supply voltage is chosen to be 0.6 V in order to have a distinguishable signal from the noise (which benefits from a

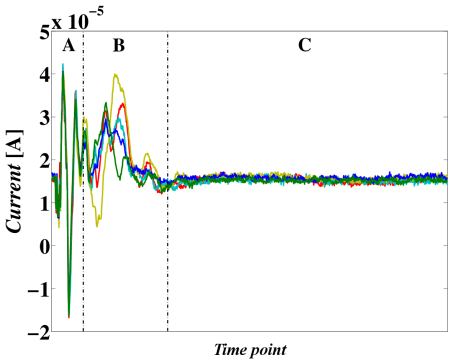

Figure 3: Power traces of the power PUF instance showing 5 different chips using a single challenge.

high supply) and at the same time to increase the impact of process variations by operating at low supply voltage. These creation choices allow the power PUF to take advantage of the randomness provided by the manufacturing technology through the magnified presence of random glitches.

Evaluation process: We performed measurements using a high sampling rate oscilloscope (1 Gsample/second), while running the chips at 2 MHz (motivated by interface constraints of our prototype board). We monitored the voltage drop on a resistor introduced in the supply circuit of the chips, then transformed it into power consumption (or current) in a post-processing step. Our dataset $\mathcal{D} = \{y_{pq},\ 1 \leq p \leq P,\ 1 \leq q \leq Q\} \subset \mathbb{R}^N$ contains $P = 18$ different PUFs evaluated $Q = 200$ times of $N = 1050$ time samples.

Extraction process: The goal of extraction process is mainly to gain robustness. The power trace has a large number of points (1050 in our instance). If we want to keep all points, in order to achieve a sufficiently high robustness, we have to use a very large ECC and also a very large helper data which is not efficient for the implementation. As a result, the main goal of the extraction process is to compress the power trace by selecting its Points of Interests (POIs). For that purpose, for each challenge, we first average over all traces from different measurements to remove the noise. Then, the POIs are selected such that the variance between (mean) traces of different PUFs are maximum. The detailed description of our selection of POIs based on maximum variance criteria is further described in Appendix-A.

Let us now define the extraction parameters. Let $\mathcal{P} \in \mathbb{N}^{N_\mathcal{P}}$ be the vector pointing out the selected POIs for each challenge based on the method of Appendix-A and N_b be the number of bits that is kept for each point after quantization. The size of the binarized response $\overline{y}$ is thus $M = N_p \times N_b$. Also, let ECC(M, ℓ, t) be the Error Correction Code applied in the extraction process where ℓ is the size of output, M is the size of binarized response and t is the number of corrected errors. Then, the extraction process (in setup mode) is given in Figure 4. The binarized response is first built from POIs as $\overline{y} = \mathsf{Q}_{\mathsf{N_b}}(y(\mathcal{P})) \in \mathcal{B}^M$ where $\mathsf{Q}_{\mathsf{N_b}}$ is a function that quantize the values and keep N_b most significant bits of each. The vector $\mathcal{P}$ is compressed in $h_\mathcal{P}$ and stored as the first part of helper data. The rest of the extraction

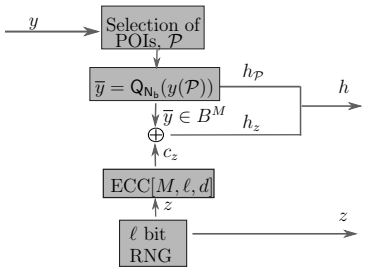

Figure 4: The extraction process in setup mode.

process is essentially the code-offset secure sketch scheme used in [3, 13, 21]. In summary, the output z is randomly generated and then encoded to codeword $c_z \in \mathcal{B}^M$ using the ECC(M, ℓ, t). In next step, the offset between $\overline{y}$ and c_z constitutes the second part of helper data $h_z = \overline{y} \oplus c_z \in \mathcal{B}^M$ and is stored in the database. The reconstruction mode of the above extraction algorithm (excluded in the figure) has the same procedure, except that it takes as input the first and second part of helper data to generate the output. The parameters of the extraction process are experimentally set to achieve the best possible results as follows: $N_p = 5$ points of interest are chosen and each are quantized to $N_b = 6$ bits. Error correcting code of code-offset secure sketch is realized using BCH (M, ℓ, t) code. The codeword size M must have the form $2^k - 1$ for some integer k, so we append 1 zero to $N_p \times N_b = 30$ bits to have a $M = 31$ bit binarized response $\overline{y}$.

Note that this instance of PFS is not claimed to be practically relevant as such (in view of its expensive and external measurements). As already mentioned in introduction, we consider it as a useful case study to illustrate our new definition of physical unpredictability, since we can take advantage of efficient side-channel attacks to build models in this context. As a result, the general conclusions we extract from our case study are more important than the instance we propose. Nevertheless, we also note that improvements could be obtained by embedding a measurement circuitry on chip (which would also reduce the noise in our measurements, and hence improve the robustness of this PUF - see next). In the next section, we show how to evaluate the important properties of a (power) PUF, namely robustness, physical unpredictability and physical unclonability. We first evaluate them individually, and then investigate their interrelations by particularly studying the trade-off between robustness and physical unclonability, and also the impact of physical unpredictability on the evaluation of unclonability.

5. ROBUSTNESS

In this section, we evaluate the first important property of our power PUF system i.e., robustness. We evaluate robustness of Eq. 1 after extracting outputs from dataset $\mathcal{D}$ using the extraction process described in Section 4.

Given the dataset $\mathcal{D}$ containing P different PUFs observed Q times each, the robustness is statistically estimated as follows. For each PUF, one observation of the PUF is used in set-up mode to produce output z_s and helper h. The remaining $(Q-1)$ observations, together with the initial helper data h, are then used in reconstruction mode, and the generated outputs are compared with z_s. This is repeated Q times,

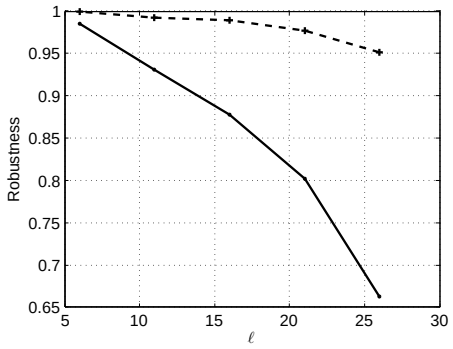

Figure 5: Robustness vs. output size ℓ using dataset $\mathcal{D}$ (solid) and after 10 times averaging (dashed).

with each observation being once in the set-up phase. The robustness is then estimated from $\Pr[z_r = z_s]$ that is equivalent to Eq. 1. Fixing code-word size to $M = 31$, robustness is evaluated for different error correction capabilities t, that lead to different output sizes ℓ. Note that by fixing the ECC code-word size, t and ℓ are inversely related, i.e. the increase of t leads to the decrease of ℓ and vice versa. The robustness versus output size ℓ is shown as a plain curve in Figure 5.

We observe that by increasing the number of corrected errors t (decreasing ℓ) robustness improves. In Section 7, we study how the number of corrected errors also affects the physical unclonability and thus makes a trade-off between robustness and physical unclonability. We also observe that we can achieve an acceptable robustness $\rho_{\mathsf{PFS}} = 0.98$ for the output size $\ell = 7$. It is similar to the output size of Glitch PUFs [28, 29]. In their scheme, for each 8 outputs of the S-box, glitches are counted and their parity acts as the one-bit output of the corresponding S-box output. So, for every challenge, the system generates in total an 8-bit output. As robustness depends on the measurement noise, by averaging traces or improving the measurement setup we can reach higher robustness. For example, the dashed curve of Figure 5, corresponds to a 10 times averaging process[1].

6. PHYSICAL UNPREDICTABILITY

In this section we discuss the notion of physical unpredictability. We propose a methodology for evaluating physical unpredictability, using the power PUF instance as illustration. We start by discussing the difference between modeling and predicting the physical response, and focus on the fact that a genuine model is required for prediction. We then finish with an interpretation of experimental results obtained from performing modeling and predicting experiments on power PUF instance. We also derive some results regarding physical unpredictability and the true randomness of the PFS.

[1]Besides averaging, another way to improve the noise level would be to embed the measurement circuitry on chip (which would naturally increase the practical relevance of the PUF instance as well). This solution would demand a fast enough current sensing scheme to capture the dynamic (changing) transient current. So, we have to use a high speed analog-to-digital converter [34,35] with 8-bit resolution (as the currently used oscilloscope). At the operating supply voltage, which is 0.6 V, the sampling rate of the required sampling scheme is expected to be in the 100's of MHz. Also to embed the measurements of the current traces on-chip, the selection of the POIs needs to be predefined. Therefore, if our extensive measurements off-chip / post-layout simulations led to fixed points in the trace, then the position of these points can be chosen as a helper data for the on-chip samplers. Finally, we could embed the ECC on-chip as well [36, 37].

6.1 Modeling versus Predicting

As mentioned in section 3, physical unpredictability is the probability of predicting PUF outputs z by an adversary who has access to the physical responses y. We argue that for the predicting experiment to be efficient, it requires the use of a precise physical model which is a true description of the underlying physics. To evaluate physical unpredictability, we proceed using the predicting experiment described in Figure 2. As a result we obtain a predicting success rate, percentage of PUF outputs z correctly predicted for challenges x not yet encountered, which is an estimator for the physical unpredictability. We contrast this with modeling success rate which is the percentage of PUF outputs z predicted over all challenges (i.e. x' in $\{x_1, ...x_q\}$ is allowed). The modeling success rate assesses the quality of the model we work with, whilst the predicting success rates estimates the physical unpredictability empirically. To illustrate the differences between modeling and predicting we focus on the power PUF instance. The vast literature on side-channel attacks provides insights about how to model the power consumption of a cryptographic device. As we are working in a 1^{st}-order side-channel type scenario (i.e. we exploit information lying in the means of the power traces), the stochastic model introduced by Schindler et al. in [38] is a natural candidate for evaluating the unpredictability of our power PUF. It provides a method to build a physical model from a restricted set of measurements. According to Schindler et al. the power consumption at any time can be written as:

$$p_t(x) = \sum_i \omega_i g_i(s_t(x)), \tag{3}$$

where $s_t(x)$ represents the internal state of the chip at time t for challenge x, g_i is a polynomial in the bits of $s(x)$ and ω_i is a weight characterizing the power consumption. The physical model is obtained from linear regression analysis. More precisely, given a set of (x, p) pairs we get a system of linear equations where the unknowns are the values ω_i. This system is solved using least squares and the solution gives us a model for all challenges. During modeling we have the choice of the *basis* $\{g_i\}_i$. We say the basis is *linear* when we consider all polynomials of degree one at most, we say the basis is *quadratic* when we consider all polynomials of degree two at most, ... Using the above method, we evaluate the predicting success rates and modeling success rates in two scenarios. In the first scenario the power PUF uses only one time point chosen in part B of the traces (See Figure 3) to produce the physical response (i.e. $N_{\mathcal{P}} = 1$). In the second scenario we use the PUF setup described in Section 4.

In both scenarios we use the S-Box output bits as state function $s_t(x)$ as it yields better results than using S-Box input bits. We show the results for 8 chips only for visibility reasons. The results for each chip is shown in various colors, and we always show in a black dotted line the robustness rate of the PUF. In the first scenario, where $N_{\mathcal{P}} = 1$, we only perform the modeling and predicting using a linear basis, and the results are shown in Figure 6. The figure shows

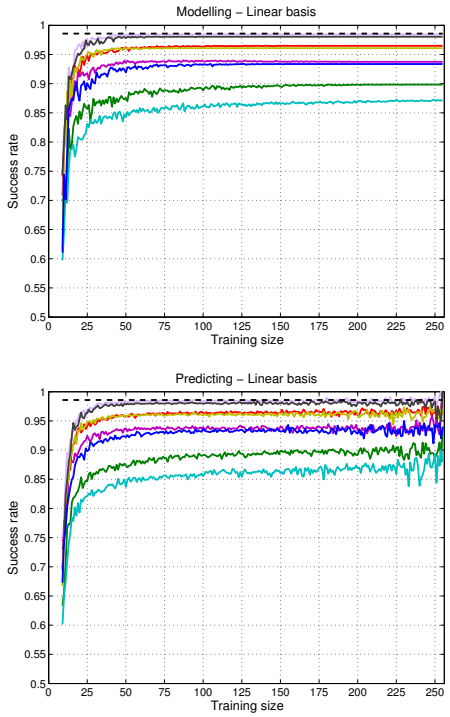

Figure 6: Modeling and unpredictability for a single point extraction process in the PUF setup.

modeling and prediction success rate for various sizes of the training sets. As we can see modeling and prediction success rates are similar in this scenario. Also, the figures validate the soundness of using stochastic model as a tool for prediction. For the second scenario, since the PUF setup described in Section 4 uses many time points, we adapt the attack procedure by creating a model for every time point used. This time we perform the experiment for various basis: linear, quadratic, cubic and quartic to show the variation in modeling and prediction success rates as the basis size increases. The results are shown in Figure 7. We see that as the size of basis increases the accuracy of the model improves, but we do not see the same effect for prediction due to over-fitting reasons. The experiments show that although an accurate model is necessary for accurate prediction, it is not sufficient. Thus an adversary performing the predicting experiments requires two conditions from the model to produce an effective physical attack. Namely, the model created has to fit well the observations (i.e. have a high modeling success rate), but it also has to be a genuine description of the physics so the predicting success rates is high.

6.2 Interpretation

In the previous subsection we presented two scenarios where we evaluated the modeling and predicting success rate for two different setups of a power PUF. In the first setup we only use one time point to produce the physical response. In the second setup we use 5 distinct point for each challenge (resulting in a hundred time points being used overall to produce the physical response). In the first scenario we achieved a prediction rate close to the robustness rate. In the second we achieved a prediction rate of 40 percent.

The main difference between the two scenarios is the number of explanatory variables. In the first scenario there is only one POI that is easily explained by a linear combination of the S-box output bits. It roughly indicates that there are as many unknowns to be evaluated in the system as elements in the basis $|\{g_i\}|$. In the second scenario the number of POIs is $|\{g_i\}| \times N_{\mathcal{P}}$, and they assumably correspond to a more complex function of the system state, hence corresponding to a larger number of unknowns. This explains why the predicting experiment is less successful in this case. However a 40 percent success rate for the second scenario shows that the power PUF cannot be considered as a very strong PUF when it comes to physical unpredictability. In fact, even the observation of 9 challenges (corresponding to the size of a linear basis) already leads to a non-negligible predictability. Therefore, already after 9 challenges, the power PUF outputs cannot be considered as independent anymore.

7. UNCLONABILITY

We now study physical unclonability and how it relates to the robustness and physical unpredictability evaluated in previous sections. Let us begin by evaluating physical unclonability as a function of the output size. It is computed from Eq. 2 (averaged over all challenges) after extracting outputs from dataset $\mathcal{D}$ using the extraction process described in section 4. Following the method of [3], the probability of Eq. 2 is statistically estimated as follows. First, one PUF is chosen as a "target", the response of which is used in set-up mode to generate helper data h and output z_s. The remaining $(P-1)$ PUFs, together with the initial helper data h, are then used in reconstruction mode, and the generated outputs are compared with z_s. This experiment is repeated P times, each PUF being selected once as target. The probability of Eq. 2 is thus estimated from $Pr\left[z_r = z_s : p \neq p'\right] \leq \beta$. Again with $M = 31$, physical unclonability is evaluated for different error correction capabilities t, that lead to different output sizes ℓ. The obtained physical unclonability represented by probability of cloning is plotted versus output size ℓ in Figure 8.

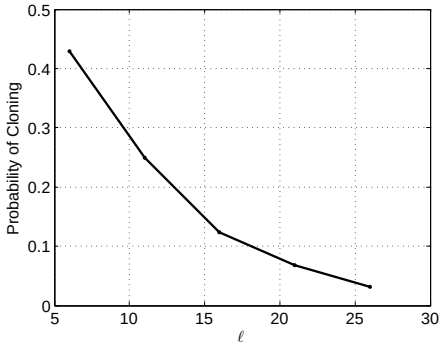

Figure 8: Physical unclonability versus output size.

By increasing the number of corrected errors t (hence by decreasing ℓ) physical unclonability gets worse (the probability of cloning increases). The reason is that correcting more errors results in removing more details (randomness) from the response. This fact causes the trade-off between robustness and physical unclonability illustrated in Figure 9.

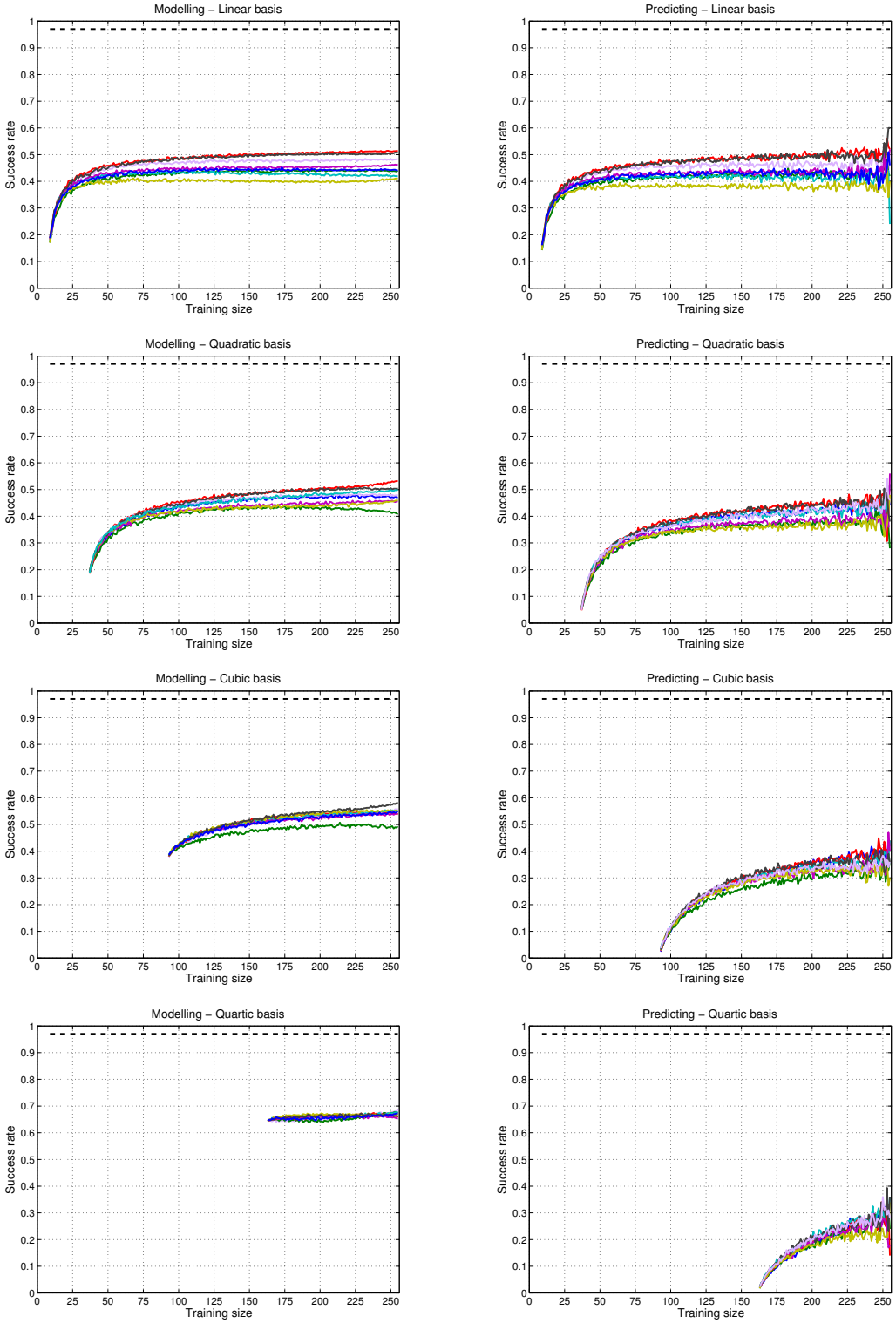

Figure 7: Modeling and unpredictability for PCA based PUF setup for various basis.

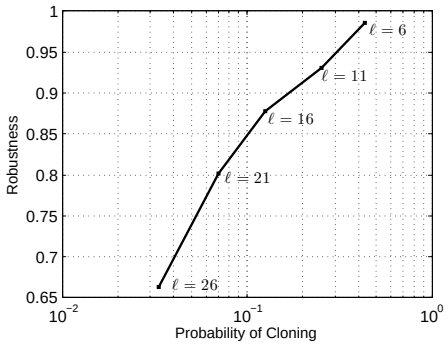

Figure 9: Robustness versus Physical unclonability.

Figure 9 shows that (as usual for PUFs) the physical unclonability with a single output is not enough for practical applications ($\sim 10^{-1}$). As mentioned in the introduction, there exist two solutions to improve the physical unclonability: either by increasing the challenge set and concatenating the corresponding outputs, or by using multiple physically distinct functions and concatenating the corresponding outputs. A typical example of the first approach is concatenating the outputs of multiple challenges for Arbiter PUFs [9,10]. For the second approach the solution can be using multiple S-boxes for the Power PUF or different memory cells for SRAM PUF [11]. As the second approach is more expensive, the goal in the design procedure is usually using the first approach as much as possible and rely on the second approach afterwards. Therefore, we now study how much this first approach can be used in our context, i.e. how much we can improve the physical unclonability of a power PUF by concatenating outputs of multiple challenges. In this case, the outputs of size $\ell = 6$ (with highest robustness in Figure 9) are concatenated to produce a larger output. The solid curve of Figure 10 srepresents the physical unclonability of the new output in function of its size.

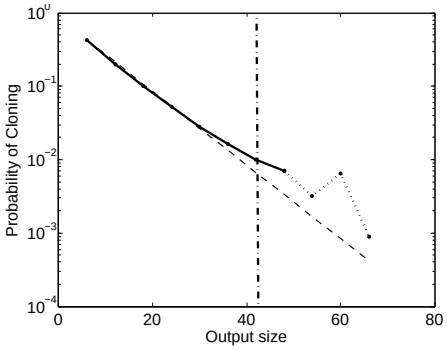

Figure 10: Physical unclonability vs. output size after concatenating outputs of different challenges by experiments (solid) and by assuming independent outputs (dashed). The dot-dashed vertical line shows the limit posed by physical unpredictability.

A first observation is that the estimation of the unclonability for large output sizes (i.e., bigger than 48 bits shown in the dotted line) is no longer accurate. The problem is that the number of samples available for evaluation is not sufficient to evaluate the probability of collisions (i.e. physical unclonability) for such large outputs. That is why in the literature, the physical unclonability is usually estimated assuming that the outputs are independent (as shown in the dashed line in Figure 10). However, the unpredictability experiments in the previous section showed that after 9 challenges, a model can be built for predicting the power PUF outputs. This implies that independence cannot be assumed anymore from that point on. As a result, taking some security margins, we could assume that the independence assumption is respected up to 7 challenges. The output size on this limit (vertical dashed-dot line in Figure 10) is 7 outputs $\times$ 6 bits per output = 42 bits. Interestingly, this threshold corresponds to the point where our empirical evaluation of the probability of cloning starts to deviate from the dashed curve.

These results suggest that the concatenation of PUF outputs should be limited to the number of challenges for which physical unpredictability can be guaranteed. This is an interesting observation for evaluators since the evaluation of physical unpredictability is in general easier than the one of physical unclonability. Indeed, the analysis of physical unclonability requires that many PUFs are available to the evaluator. By contrast, evaluating the physical unpredictability can be done from a few chips (in fact, even a single one may be sufficient). As a result, we believe our results give incentive to always evaluate the physical unpredictability of strong PUFs, together with their physical unclonability. Although not finding any model (and having high unpredictability) is naturally not a sufficient condition to guarantee the independence of PUF outputs, obtaining a good model can at least be considered as a warning signal that these outputs are not independent. Hence, and in view of the difficulty to evaluate PUFs, we believe this sanity check is a safe practice to avoid overestimating their security.

8. CONCLUSION

The physical unpredictability that we introduce serves in two different aspects in evaluating security of PUF-based systems. First, it captures the unpredictability of strong PUFs in the worst-case scenario where the adversary can probe the PUF responses. Second, it determines the safe region where we can rely on an independence assumption for the evaluation of physical unclonability. While only applied to a single PUF instance, we hope that these tools can be useful for a wide variety of PUFs, for which generalizing and applying our evaluation of robustness, physical unpredictability and physical unclonability is an interesting open problem. Eventually, the exploitation of physical parameters (such as the power supply) to increase the challenge set of strong PUFs is another scope for further research.

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

9. REFERENCES

[1] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 237–249. ACM, 2010.

[2] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, François-Xavier Standaert, and Christian Wachsmann. A formalization of the security features of physical functions. In *IEEE Symposium on Security and Privacy*, pages 397–412. IEEE Computer Society, 2011.

[3] Saloomeh Shariati, François-Xavier Standaert, Laurent Jacques, and Benoit M. Macq. Analysis and experimental evaluation of image-based PUFs. *J. Cryptographic Engineering*, 2(3):189–206, 2012.

[4] Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede. Pufky: A fully functional puf-based cryptographic key generator. In Prouff and Schaumont [39], pages 302–319.

[5] Ravikanth S. Pappu. *Physical one-way functions*. PhD thesis, MIT, March 2001.

[6] Ravikanth S. Pappu, Ben Recht, Jason Taylor, and Niel Gershenfeld. Physical one-way functions. *Science*, 297:2026–2030, 2002.

[7] Pim Tuyls, Geert Jan Schrijen, Boris Skoric, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In *CHES*, pages 369–383, 2006.

[8] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In *ACM Conference on Computer and Communications Security*, pages 148–160, 2002.

[9] Daihyun Lim, Jae W. Lee, Blaise Gassend, G. Edward Suh, Marten van Dijk, and Srinivas Devadas. Extracting secret keys from integrated circuits. *IEEE Trans. VLSI Syst.*, 13(10):1200–1205, 2005.

[10] Jae W. Lee, Daihyun Lim, Blaise Gassend, G. Edward Suh, Marten Van Dijk, and Srini Devadas. A technique to build a secret key in integrated circuits with identification and authentication applications. In *In Proceedings of the IEEE VLSI Circuits Symposium*, pages 176–179, 2004.

[11] Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In *CHES*, pages 63–80, 2007.

[12] James D. R. Buchanan, Russell P. Cowburn, Ana-Vanessa Jausovec, Dorothée Petit, Peter Seem, Gang Xiong, Del Atkinson, Kate Fenton, Dan A. Allwood, and Matthew T. Bryan. Forgery: 'fingerprinting' documents and packaging. *Nature*, 436(7050):475, July 2005.

[13] Philippe Bulens, François-Xavier Standaert, and Jean-Jacques Quisquater. How to strongly link data and its medium: the paper case. *Information Security, IET*, 4(3):125 –136, september 2010.

[14] Darko Kirovski. Toward an automated verification of certificates of authenticity. In *Proceedings of the 5th ACM conference on Electronic commerce*, EC '04, pages 160–169. ACM, 2004.

[15] Yuqun Chen, Kivanç Mihçak, and Darko Kirovski. Certifying authenticity via fiber-infused paper. *SIGecom Exch.*, 5:29–37, April 2005.

[16] Saloomeh Shariati, François-Xavier Standaert, Laurent Jacques, Benoit Macq, Mohamad Amine Salhi, and Philippe Antoine. Random Profiles of Laser Marks. *Proceedings of the 31th symposium on Information Theory in the Benelux*, pages 27–34, 5 2010.

[17] Saloomeh Shariati, Laurent Jacques, François-Xavier Standaert, Benoit M. Macq, Mohamed Amin Salhi, and Philippe Antoine. Randomly driven fuzzy key extraction of unclonable images. In *ICIP*, pages 4329–4332, 2010.

[18] Pim Tuyls, Boris Skoric, and Tom Kevenaar, editors. *Security with Noisy Data — On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. Springer Berlin/Heidelberg, April 2007.

[19] Cheun Ngen Chong, Dan Jiang, Jiagang Zhang, and Long Guo. Anti-counterfeiting with a random pattern. In *SECURWARE*, pages 146–153, 2008.

[20] Saloomeh Shariati, François Koeune, and François-Xavier Standaert. Security Analysis of Image-Based PUFs for Anti-counterfeiting. In Bart Decker and DavidW Chadwick, editors, *Communications and Multimedia Security*, volume 7394 of *Lecture Notes in Computer Science*, pages 26–38. Springer Berlin/Heidelberg, September 2012.

[21] Pim Tuyls and Boris Skoric. Secret key generation from classical physics: Physical uncloneable functions. In Satyen Mukherjee, RonaldM. Aarts, Raf Roovers, Frans Widdershoven, and Martin Ouwerkerk, editors, *AmIware Hardware Technology Drivers of Ambient Intelligence*, volume 5 of *Philips Research*, pages 421–447. Springer Netherlands, 2006.

[22] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls. Memory leakage-resilient encryption based on physically unclonable functions. In *ASIACRYPT*, pages 685–702, 2009.

[23] Blaise Gassend, Daihyun Lim, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Identification and authentication of integrated circuits. *Concurrency - Practice and Experience*, 16(11):1077–1098, 2004.

[24] Ulrich Rührmair, Jan Sölter, Frank Sehnke, Xiaolin Xu, Ahmed Mahmoud, Vera Stoyanova, Gideon Dror, Jürgen Schmidhuber, Wayne Burleson, and Srinivas Devadas. Puf modeling attacks on simulated and silicon data. *IACR Cryptology ePrint Archive*, 2013:112, 2013.

[25] Gabriel Hospodar, Roel Maes, and Ingrid Verbauwhede. Machine learning attacks on 65nm arbiter pufs: Accurate modeling poses strict bounds on usability. In *WIFS*, pages 37–42, 2012.

[26] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In *EUROCRYPT*, pages 109–128, 2011.

[27] Dina Kamel, Cédric Hocquet, Françoix-Xavier Standaert, Denis Flandre, and David Bol. Glitch-induced within-die variations of dynamic energy in voltage-scaled nano-CMOS circuits. In

Proceedings of European Solid-State Circuits Conference, ESSCIRC, pages 518 –521, Sept. 2010.

[28] Daisuke Suzuki and Koichi Shimizu. The glitch puf: A new delay-puf architecture exploiting glitch shapes. In *CHES*, pages 366–382, 2010.

[29] Koichi Shimizu, Daisuke Suzuki, and Tomomi Kasuya. Glitch puf: Extracting information from usually unwanted glitches. *IEICE Transactions*, 95-A(1):223–233, 2012.

[30] Nele Mentens, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. A systematic evaluation of compact hardware implementations for the Rijndael S-Box. In *CT-RSA*, pages 323–333, 2005.

[31] Dina Kamel, François-Xavier Standaert, and Denis Flandre. Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes. In *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS*, pages 1385 –1388, May 2009.

[32] Kerry Bernstein, David J. Frank, Anne E. Gattiker, Wilfried Haensch, Brian L. Ji, Sani R. Nassif, Edward J. Nowak, Dale J. Pearson, and Norman J. Rohrer. High-performance cmos variability in the 65-nm regime and beyond. *IBM Journal of Research and Development*, 50(4-5):433–450, 2006.

[33] Keith A. Bowman, Alaa R. Alameldeen, Srikanth T. Srinivasan, and Chris Wilkerson. Impact of die-to-die and within-die parameter variations on the clock frequency and throughput of multi-core processors. *IEEE Trans. VLSI Syst.*, 17(12):1679–1690, 2009.

[34] Zheng-Yu Wang, Hui Pan, Chung-Ming Chang, Hai-Rong Yu, and M.F. Chang. A 600 msps 8-bit folding adc in 0.18 /spl mu/m cmos. In *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*, pages 424–427, 2004.

[35] Seyed Danesh, Jed Hurwitz, Keith Findlater, David Renshaw, and Robert K. Henderson. A reconfigurable 1 gsps to 250 msps, 7-bit to 9-bit highly time-interleaved counter adc with low power comparator design. *J. Solid-State Circuits*, 48(3):733–748, 2013.

[36] Meng-Day (Mandel) Yu, David M'Raïhi, Richard Sowell, and Srinivas Devadas. Lightweight and secure puf key storage using limits of machine learning. In *CHES*, pages 358–373, 2011.

[37] Stefan Katzenbeisser, Ünal Koçabas, Vladimir Rozic, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (pufs) cast in silicon. In Prouff and Schaumont [39], pages 283–301.

[38] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.

[39] Emmanuel Prouff and Patrick Schaumont, editors. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*. Springer, 2012.

10. APPENDIX A

As stated in Section 4, POIs for each challenge are chosen such that they maximize the variance between the (mean) traces for different PUFs. To achieve that, we use a method adapted from Principal Component Analysis (PCA) as follows. In summary, the POIs are selected as the points which maximize the eigenvector of the covariance of the mean traces matrix. In particular, for each challenge, given a matrix of $P \times N$ mean traces, the eigenvectors ξ_i of the covariance of this matrix are sorted based on the value of their corresponding eigenvalues as $\{\xi^{(1)}, \xi^{(2)}, ..., \xi^{(N)}\}$. We select the POIs as the points that maximizes these eigenvectors. These points would have the main contribution on the principal components (which in PCA are obtained by projecting the trace to the eigenvectors). The POIs can either be selected as the $N_{\mathcal{P}}$ maximal points of the first eigenvector or the set of first maximal points of first $N_{\mathcal{P}}$ eigenvectors. In our experiments, we selected the latter approach as it leads to enough distance between POIs and resulting more randomness in the binarized response. The vector of our selected points of interest is thus given by $\mathcal{P} = \{\text{argmax}_n(\xi^{(1)}), \text{argmax}_n(\xi^{(2)}), ..., \text{argmax}_n(\xi^{(N_p)})\}$. It is worth noting that the above heuristic method to select the POIs is motivated but not optimal and can further be improved by other dimensionality reduction techniques.