Enhancing the Resiliency of Multi-bit Parallel Arbiter-PUF and Its Derivatives Against Power Attacks

Trevor Kroeger[1](✉), Wei Cheng[2], Sylvain Guilley[2,3], Jean-Luc Danger[2], and Naghmeh Karimi[1]

[1] CSEE Department, University of Maryland Baltimore County,
Baltimore, MD 21250, USA
{trevor.kroeger,naghmeh.karimi}@umbc.edu
[2] LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France
{wei.cheng,sylvain.guilley,jean-luc.danger}@telecom-paris.fr
[3] Secure-IC S.A.S., 35510 Cesson-Sévigné, France
sylvain.guilley@secure-ic.com

Abstract. Embedded systems utilize Physically Unclonable Functions (PUFs) for authentication and identification purposes. However, modeling PUFs' behavior via machine-learning methods has received utmost attention. Current research on modeling PUFs mainly targets a single PUF instance (PUF producing a single-bit response per query). It is admittedly more challenging to attack multi-bit parallel PUFs (with $M > 1$ PUF instances). In this work, we first target a multi-bit (mainly $M = 2$-bit) parallel arbiter-PUF using its power traces, then introduce a hybrid countermeasure, combining Dual Rail Logic and Randomized Initialization Logic mechanisms, to thwart such attack. In addition, we explore Randomized Arbiter Swapping and Randomized Response Masking mitigation techniques for providing further protection for parallel PUFs against modeling attacks. To mimic the PUFs' behavior in real silicon, we add noise artificially in our simulations. The results confirm the high success of the launched attack for the unprotected-PUF, and the resiliency of our countermeasures.

Keywords: Physically unclonable functions · Side-channel attack · Machine learning · Modeling attack · Parallel PUFs

1 Introduction

The fabrication of Integrated Circuits (IC) suffers from unavoidable imperfections. Such drawbacks can be leveraged within carefully designed circuits which harness these distinct variations for generating unpredictable unique values despite the similarity of their gate-level netlists. These circuitries, aka Physically Unclonable Functions (PUFs), map their input bits (referred to as a challenge hereafter) to a unique bit vector, so-called response. A PUF can be embedded in

S. Bhasin and F. De Santis (Eds.): COSADE 2021, LNCS 12910, pp. 303–321, 2021.
https://doi.org/10.1007/978-3-030-89915-8_14

an IC to generate hardware fingerprints and in turn to enable device authentication. Such reproducible fingerprints, so-called Challenge-Response Pairs (CRPs) can also be used for security parameters, for example secret keys in cryptographic modules [18]. Test and methodologies to assess the security of PUFs have recently been standardized at international level, in ISO/IEC 20897 [16].

Thanks to their ease of implementation and small size, PUFs are broadly deployed in radio-frequency identifiers (RFIDs), smart cards, and low-cost internet of Things (IoT) devices. Indeed, PUFs have been proven to be highly useful such that they have found their way into critical systems such as autonomous vehicles [8] as well as cryptocurrencies [22], and in chip onboarding schemes [14].

PUFs are categorized into strong and weak groups. A weak PUF simply refers to a PUF with highly limited number of (or no) CRPs, e.g., SRAM-PUF. Weak PUFs are typically used for key generation. However, they are susceptible to invasive attacks that monitor the internal structure of the PUF [23], as well as cloning attacks where the response can be easily replicated due to the limited number of responses [17]. On the other hand, strong PUFs have an exponential number of challenge response pairs. The arbiter-PUF [4] is an archetype of strong PUFs suitable for authentication purposes [18].

Although PUFs are abundantly useful they are not infallible. One such drawback is their susceptibility to the modeling attacks where an adversary tries to build a model that mimics the behavior of the target PUF. PUFs can be modeled through their CRPs [24] or via their side-channel leakage, in particular their power consumption [3,12]. To alleviate such vulnerability, several countermeasures have been proposed in literature including new PUF designs like the XOR-PUF [32], the Feed-Forward PUF [13], and Challenge Obfuscation schemes [9,29]. Besides, it has been suggested to use a fake PUF along with the genuine PUF and sneakly query the fake PUF intermittently [5]. However, these countermeasures are tailored against the attacks that exploits CRPs, and fall short when the adversary takes the power side-channel into account since the exploited leakage in power consumption is independent of the challenges [2,11].

Arbiter-PUFs and their derivatives are highly popular as a candidate for device authentication. To decrease their area and power overhead, they can be implemented as single instance, i.e., as a 1-bit response generator. This instance is queried multiple times with different challenges to generate an R-bit response. However, such implementation has been shown to be vulnerable against power analysis attacks [11]. Thereby, this paper expands the knowledge of power side-channel based modeling attacks by investigating the vulnerability of the arbiter-PUFs composed of multiple single-bit response arbiter-PUFs ($M = 2$-bit in particular) operating in parallel as parallelization may contribute to hinder the modeling of a PUF behavior through observing its power side-channel. To the best of our knowledge, there is no research in open literature on these multi-bit response parallel PUFs with respect to their assailability to power side-channel modeling attacks. This paper shows that these PUFs are in fact vulnerable (in the presence of realistic noise) especially if simple attack enhancements are made such as averaging multiple repeated power trace captures to increase the signal-to-noise ratio (SNR) of the target device's traces. Finally, to mitigate such vulnerability we propose a hybrid countermeasure that benefits from hiding the current leakage through the use of complementary Dual Rail Logic and response confusion with Randomized Initialization Logic. We also evaluate swapping and masking countermeasures. The contributions of this paper are as follows:

– Successful power-based modeling attacks on parallel multi-bit response arbiter-PUFs;
– Investigation into the attack validity in the presence of realistic noise and data extraction methods;
– Presentation of a number of lightweight countermeasures to thwart the modeling attack in the parallel PUFs based on equalizing the power consumption using Dual-Rail Logic, as well as randomizing the response;
– Assessment of the efficiency of the proposed countermeasures at different noise levels.

2 Background on Arbiter-PUFs

The arbiter-PUF is broadly used due to its ease of implementation, its effectiveness in producing unique values and its large space of its CRPs [7]. This PUF creates a base for many other PUF variants such as XOR-PUF [32], Feed-Forward PUF [13], etc. Each instance of an arbiter-PUF is composed of a pair of delay chains and one arbiter (as simple as an SR latch), and generates one response bit per challenge, in a single query [4] based on the process-variation induced race between two identical paths (top and bottom paths in Fig. 1). The difference in the propagation delay of these paths determines the PUF response to each challenge. Only the sign of this difference (not the exact amount) determines the response bit for each challenge.

Note that a full implementation of a PUF, embedded in a chip for generating keys or authentication purposes, would contain a storage mechanism following the PUF's output; denoted as system component in Fig. 1 and is mainly realized as a Flip-Flop to store the result of the PUF before the downstream components use the response. However, these system components create power side-channels that can be exploited to extract information, which we call *power leakages* in this work, and accordingly may be exploited by an adversary in order to model the PUF's behaviors. These leakages play an important role in the overall power consumption of the PUF, and in turn the underlying chip [6]. The derivatives of arbiter-PUFs, e.g., XOR-PUFs and Feed-Forward PUFs all follow the same scenario regarding the inclusion of system components which jeopardizes their security against power side-channel attacks.

3 Related Works

Most modeling countermeasures are for CRP-based modeling attacks which attempt to predict the PUF's response for previously unseen challenges: they can consist in design-level protection [9,28,29] or in mode-of-operation hindering [30]

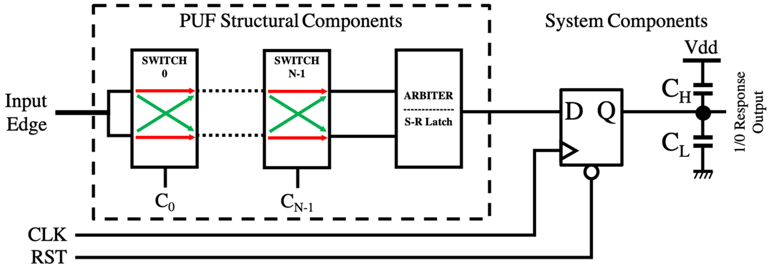


Fig. 1. Structure of an arbiter-PUF [4]. This includes both the PUF structural components as well as the system components (the target of the modeling attacks).

(here: a lockdown technique). However these methods fall short when confronted with power-based modeling attacks performed in this work [10,11,20]. In fact due to the implementation of power-based modeling attacks (namely: the leakage upon response sampling is spied on), the challenge is unused in the modeling of the PUF's behavior and therefore the size of the PUF does not matter [20].

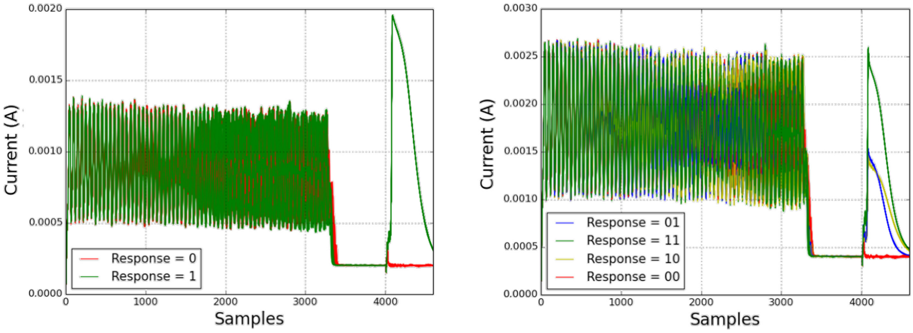
Literature is not devoid of power-based modeling countermeasures. In [25] the authors share a method for a duplicate arbiter implementation on the arbiter-PUF. In this proposed work a second arbiter is used with the top and bottom traces reversed so that it creates the leakage from the opposite response value simultaneously. This hides the leakage from the arbitration unit of the PUF. The authors of [2] describe a mitigation technique which utilize overlapping delay chains to obfuscate the side channel information. The goal of this is to introduce algorithmic (or intentional) noise such that the modeling algorithm will not be able to coherently model the PUF.

These mitigations fail to consider the system components, shown in Fig. 1, used for storing the response for usage which as we will discuss later on are the principle component of attack. Therefore these countermeasures provide scant protection from what we discuss here.

4 Motivation

As mentioned earlier, arbiter-PUFs (and their derivatives) are mainly realized as a single-bit response circuitry (Fig. 1) which is queried multiple times to generate multi-bit responses. However, such implementation not only imposes low throughput but also an individual response leakage which is discernible as only one PUF is active in each point of time which is devoid of algorithmic noise.

To create greater algorithmic noise in the critical components designers can choose parallel multi-bit response PUFs. As shown in Fig. 2, deploying multi-bit parallel PUFs makes the attack more difficult as in this case there are more variations in the output traces that have to be discerned for an attack to be successful compared to a single-bit PUF. Parallelization of the PUF chains does not affect the uniformity and uniqueness of PUFs, and its effect on PUFs' reliability is negligible.



(a) Single Bit Power Trace

(b) Parallel 2-Bit Power Trace

Fig. 2. The noiseless power traces segregated into their response values of a single-bit response arbiter-PUF and that of a parallel multi-bit ($M = 2$-bit) response arbiter-PUF.

PUFs are valued for their small implementation size and power consumption in comparison to traditional cryptographic algorithms [18]. Designers hoping to use these to their advantage have limitations on the number of parallel instances. Therefore a smaller number of parallelizations are more likely to be used.

In the following sections, we target a 2-bit parallel PUF and show that attacking is indeed feasible. Accordingly, we introduce potential countermeasures to thwart such an attack.

5 Threat Model and Attack Methodology

We assume that the adversary has physical access and the ability to record the power traces of a device with an embedded multi-bit arbiter-PUF. It is also assumed that the attacker knows the number of parallel PUF instances. The attacker typically models the chip in the enrollment phase when it is still “open” to readily query with known challenges (for CRP-based modeling attacks). He launches a power side-channel based modeling attack subsequently to retrieve valuable secrets in the post-customization phase. To improve the SNR, the adversary can replay each challenge multiple times, collect the corresponding traces and average them to reduce the noise, and in turn increase the focus on the power associated solely with the usage of the PUF [2]. Since actual hardware runs relatively fast, the adversary can amass many traces to average and increase the SNR. After creating the model, this attacker reintroduces the PUF into the supply chain so that it can be deployed in a critical system and compromise it afterwards.

Attack Methodology: Power-based modeling attacks opt to characterize the target PUF's behavior via its power side-channel. These attacks involve the collection of a series of power traces corresponding to the operation of the target PUF when it is queried , i.e. when the input transition is propagating through the switches (in case of targeting an arbiter-PUF). These traces are then used to train a Machine Learning (ML) algorithm such that the resulting model mimics the target PUF's behavior [3,12]. Modeling a PUF via its power traces is more applicable than via its CRPs as the former requires fewer traces compared to the large number of the CRPs that the latter requires to model the PUF accurately [11]. Moreover, the PUF output is typically cut through anti-fuses following the Enrollment phase of the PUF [15] making CRP-based attacks almost impossible. Such limiting access to the CRPs is performed as a countermeasure against the adversary who aims at using the CRPs to model the PUF behavior. Accordingly, in this paper we focus on the power-based modeling attacks. It is noteworthy to mention that in the power based attack when the system component is targeted, the size of the challenge bitstream does not matter to the attacker since the challenge is not used when building the PUF model through its power side-channel. This means that the relation between the challenge and response is unnecessary to discern but rather it is the relation between the power and the response that is of interest. More details can be found in [20].

We assume that the adversary exploits the leakage produced from the Flip-Flop, shown as the system component in Fig. 1, which is used to store the PUF's result before the downstream components use it. There are several advantages for targeting this leakage namely that the Flip-Flop is sequential, synchronized, and has considerable capacitive loading which in turn induces higher observable leakage. The presence of such leakage has already been confirmed in real silicon devices [31].

After collecting enough power traces, we use Support Vector Machine (SVM), a supervised ML scheme, to train a model that mimics the PUF's behavior. As mentioned earlier, before training the model we apply the averaging technique to increase the SNR by repeating each measurement multiple times and computing the average of all measurements related to the same challenge. The averaged traces are used for training the model. We repeat the averaging scheme during the evaluation phase to increase the accuracy of predicting the response.

6 Proposed Countermeasures

We propose two sets of countermeasures, each including two schemes, to mitigate the power side-channel based modeling attacks discussed earlier. The first set opts to reduce the SNR of the power trace leakage from the Flip-Flop, while the second set consists of random masking or switching the PUF's response bits before storing in the Flip-Flop, thereby confusing/poisoning the model during training phase.

Reducing the SNR of Flip-Flop Leakage: To thwart power based modeling attacks by reducing the SNR of the Flip-Flops' leakage, we propose the marrying of two mitigation techniques, namely Dual Rail Logic (DRL) and Randomized Initialization Logic (RIL) implementations.

The DRL makes use of two complimentary Flip-Flops connected to the Q and $\bar{Q}$ output pins of the PUF's arbiter (i.e., the S-R latch in Fig. 1). Indeed, the standard implementation (unprotected) would have one Flip-Flop fed with the Q output of the arbitration unit to feed the system circuitry that utilizes the PUF's response. However, as discussed, this Flip-Flop produces unavoidable leakage. Placing a second Flip-Flop after the $\bar{Q}$ output of the arbitration unit, balances the leakage and prevents exploiting such leakage for modeling the PUF. This countermeasure is inspired by [21, § 7.3]. The loading on the outputs of the Flip-Flops also needs to be balanced. Accordingly, in this research we consider differences between the output capacitors of the deployed Flip-Flops, i.e., C_{H1}, C'_{H1}, C_{L1}, C'_{L1} in Fig. 3. The capacitance values were chosen regarding the relatively high load of the DFF which is generally a system bus, and the process mismatch. The same specification holds for the lower PUF as well. To increase the randomness of the leakage in the response we propose the RIL countermeasure. Increased randomization is a common technique for thwarting modeling attacks [27]. To do so, we initialize each Flip-Flop with a random value before querying the PUF. Such random initialization hides the leakage as monitoring the switching from "0" to "1" or "1" to "0" (which can be exploited by the adversary to predict the PUF's response) may not benefit any more since observing a transition or not depends on the initial random value of the Flip-Flop (which is unknown to the adversary) as well. In parallel PUFs the random value for each PUF should be unique to prevent revealing the response to an adversary inadvertently. *Note that we use the above two methods together and refer to them together as DRILL.* This method can be used in both cases of single and parallel PUFs. However, in case of parallel PUFs, we propose to equip the circuits with the following countermeasures on top of the ones mentioned above.

It can also be surmised that the parallelization of the PUF can be considered as a countermeasure itself. For instance increasing the parallelization will increase the algorithmic noise that occurs during the PUF operation, thus decreasing the SNR and accordingly making the output less discernible (particularly in presence of noise).

Confusing and Poisoning the Modeling Algorithm: To enhance the resiliency of the parallel PUFs against the power-based modeling attack even more, we propose 2 countermeasures aiming at confusing and poisoning the inputs to the ML algorithms. The first method is referred to as Randomized Arbiter Swapping (RAS) hereafter, and the second is introduced as Randomized Response Masking (RRM), where both methods utilize a Random Number Generator (RNG).

Specifically, the RAS scheme poisons the PUF output via swapping the different bits of PUFs responses which other based on a value generated with a RNG. This makes differentiating the "01" from "10" responses highly difficult,

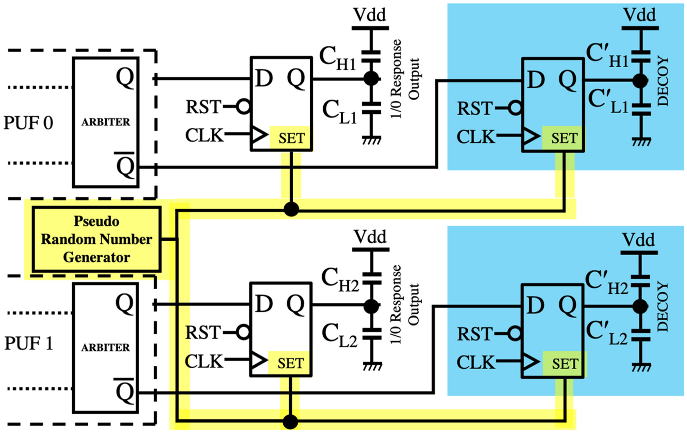


Fig. 3. The DRL (highlighted in blue) and RIL (highlighted in yellow) implemented on a standard 2-bit response parallel arbiter-PUF. (Color figure online)

if not impossible, in our 2-bit parallel PUF as depending on the generated random value, the outputs of the PUFs may swap for some challenges but remain intact for other ones. This countermeasure is shown in Fig. 4 and is realized with multiplexers which swap the outputs (shuffle them when $M > 2$) of the arbitration units based on the RNG value. Note that by swapping the outputs the entropy of the response increases but the uniformity is unchanged, provided that the PUF is unbiased.

In the RRM, the arbiters' responses are masked as being XORed with random values generated by an on-chip RNG. Figure 5 shows this implementation. Since this scheme is applied to both response bits of our 2-bit parallel PUF, it can diminish the modeling success rate significantly, i.e., the ML algorithm has 1 out of 4 chance for predicting the response correctly. Since the unmasking of the response is intended to be reversed the result will have the same response as PUF.

Note that in both of the above countermeasures, the randomization effects are reversed in software to extract the true PUF's response before using it for authentication. The software and hardware should follow the same RNG schemes. In this paper, we assume that the mask generator is secure and the method for sharing with the trusted hardware is secure as well. This means that there is no second order attack focused on the mask and the response.

7 Experimental Setup

Simulation Details: We targeted various instances of the single bit arbiter-PUF, 2-bit, and 4-bit response parallel arbiter-PUF circuitries, each with 64 challenge bits, using 15,000 random challenges and recording both responses and power traces. The capacitance values used were considered to be the worst case

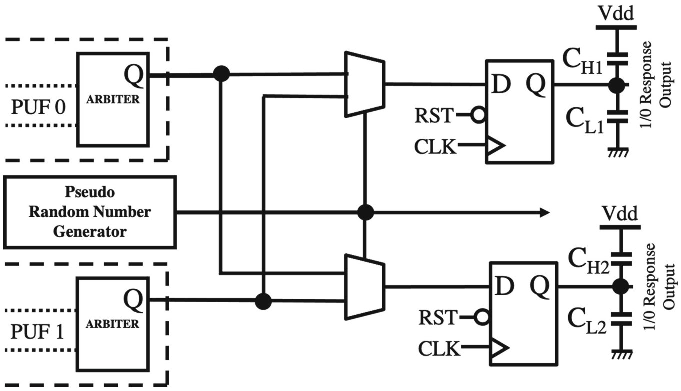


Fig. 4. The RAS countermeasure implemented within the PUF. Software is used to descramble the outputs.

loading condition for attacking, this was done to ensure that the side-channel modeling countermeasures were effective in this scenario. The capacitance values are shown in Table 1. If the capacitances are perfectly balanced then the attack will be extremely difficult as there is little differentiation in the leakages; here we consider imbalances which produce a worst case for our protections.

Table 1. Loading Capacitance Values for our PUFs. All have 2-bit responses except the single-bit one. For the PUF equipped with DRILL, each line shows the capacitors of one rail.

	C_{H1}	C_{L1}	C_{H2}	C_{L2}
Single-Bit PUF	200 fF	250 fF	N/A	N/A
Unprotected PUF	200 fF	250 fF	150 fF	200 fF
PUF + DRILL	200 fF	250 fF	150 fF	200 fF
	150 fF (′)	200 fF (′)	100 fF (′)	150 fF (′)
PUF + RAS	200 fF	250 fF	150 fF	200 fF
PUF + RRM	200 fF	250 fF	150 fF	200 fF

For the parallel PUFs both unprotected and protected circuitries were simulated in the transistor level using Synopsys HSPICE and a 45nm NANGATE technology [1]. Process variation was realized through Monte-Carlo simulations with Gaussian distributions: transistor gate length L: $3\sigma = 10\%$, threshold voltage V_{TH}: $3\sigma = 30\%$, and gate-oxide thickness t_{OX}: $3\sigma = 3\%$ reflecting a 45 nm process in commercial use.

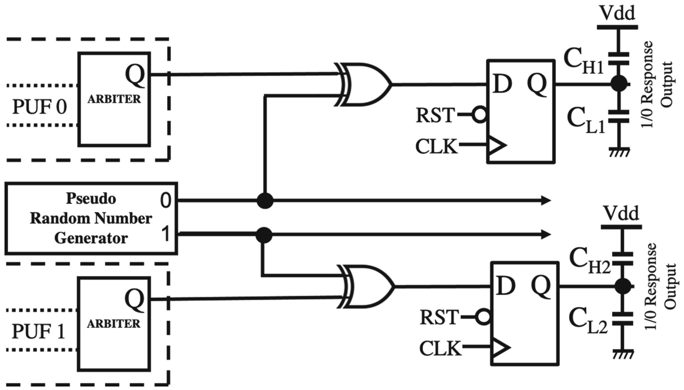


Fig. 5. The RRM countermeasure implemented on the outputs of the PUF arbiters. Software is used to unmask the outputs.

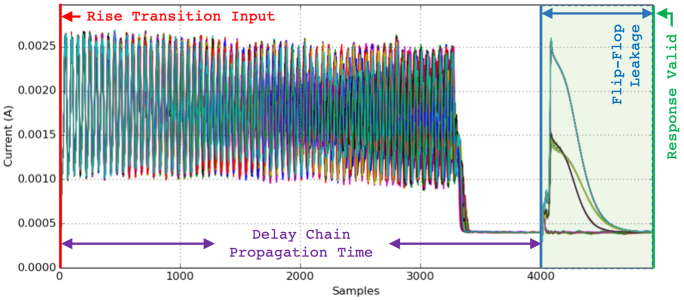


Fig. 6. Timing of the sampling window used to collect the power traces of the Unprotected 2-bit response parallel arbiter-PUF (with a 64-bit challenge). The Flip-Flop's leakage is highlighted.

Data Extraction: A set of collected power traces (sampled at 1ps) from the simulated 2-bit unprotected arbiter-PUF are shown in Fig. 6. Before sampling, the arbiter-PUF is given its challenge and the circuit has the opportunity to settle to a steady state before being queried for a response. The sampling of the power trace starts when the arbiter-PUF is given a rising transition and continues whilst the transition propagates through the chain of switches. Sampling is pursued during arbitration and the registration of the response in the Flip-Flop. Once registered, the response bit becomes valid and the sampling of the power trace ceases.

Adding Noise: In real silicon, noise occurs naturally as the PUF is embedded in a chip which may contain multiple other IP blocks producing their own current draw and fluctuations. Those circuitries entail additional "algorithmic" noise, which increases the difficulty of exploiting the target PUF's current leakage to build a model that mimics its behavior. Accordingly, in this paper to realize traces that reflect the effects of real silicon experiments more precisely, artificial noise is added to the power traces post simulation. The noisy trace (Y) is produced by adding the Gaussian noise N to the original trace (X):

$$Y = X + N \qquad where\ N \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

where we realize different standard deviations as $\sigma \in \{2.5e{-}4, 9.5e{-}4, 16e{-}4, 32e{-}4, 64e{-}4\}$. The signal-to-noise ratio (SNR) decreases with higher level of noise. The SNR is considered as a measure of attack feasibility, and is commonly used in side-channel analysis for this reason. It is assessed as the ratio of inter-variance and intra-variance [21, § 4.3.2] as below:

$$SNR = \frac{Var(Signal)}{Var(Noise)}. \tag{2}$$

For a 2-bit parallel PUF, the SNR can be assessed via the following equation, where $\mathcal{L}_{00}$, $\mathcal{L}_{01}$, $\mathcal{L}_{10}$, and $\mathcal{L}_{11}$ relate to the cases with '00', '01', '10', and '11' responses, respectively.

$$SNR = \frac{Var([\mathbb{E}(\mathcal{L}_{00}), \mathbb{E}(\mathcal{L}_{01}), \mathbb{E}(\mathcal{L}_{10}), \mathbb{E}(\mathcal{L}_{11})])}{\mathbb{E}([Var(\mathcal{L}_{00}), Var(\mathcal{L}_{01}), Var(\mathcal{L}_{10}), Var(\mathcal{L}_{11})])}. \tag{3}$$

Indeed, Eq. 3 presents the full SNR between all response possibilities. Recent research targeting a real arbiter-PUF shows that a plausible SNR is 1.81 [7]. We refer to this as a comparison point in our experiments.

Modeling Accuracy: In this paper, the accuracy of the modeling attack is defined as:

$$\text{Accuracy} = \frac{\text{Predicted Correctly}}{\text{Total Tested}}. \tag{4}$$

Note that the ideal accuracy when modeling a resilient PUF is equal to the probability of each response occurrence, i.e., 50% for a single PUF and 25% in case of 2 parallel PUFs.

All experiments are based on using 1000 traces for training and 5000 traces for testing.

8 Experimental Results

Single PUF Results: As a baseline for the parallel multi-bit PUF, the results for attacking a single-bit PUF are shown in Fig. 7. As shown, the accuracy of the attack is *approx* 100% until the noise level becomes quite high. The SNR for $\sigma = 32e{-}4$ (the noise for the last successful attack) is 0.079 which is far below the SNR of 1.81 seen in a real circuit [7]. This confirms the high vulnerability of single-bit PUFs against power based modeling attacks and motivates using parallel PUFs.

Parallel Multi-bit PUF Results: Figure 8 depicts the resiliency of the 2-bit response parallel PUF against modeling attack in different noise levels. At a first glance, the 2-bit parallel PUF seems secure as for the noise level of 32e−4 and

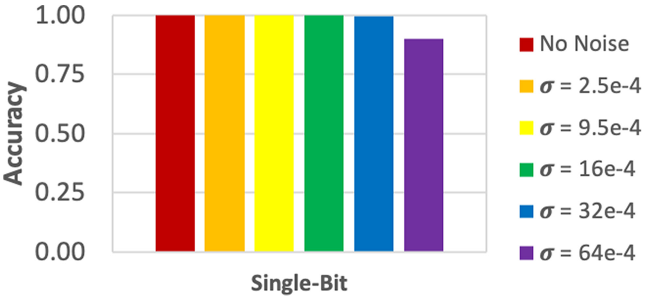


Fig. 7. The attack accuracy targeting a 1-bit response arbiter-PUF for various noise levels. Ideally the accuracy, from a design standpoint, for a 1-bit PUF is 50%.

beyond, the modeling accuracy is significantly lower than the single PUF counterpart shown in Fig. 7. However, we resemble the case in which the adversary uses a simple averaging technique in which the same challenge is fed multiple times and the recorded traces are averaged. In that case, as shown in the right side of Fig. 7, the 2-bit response PUF can be compromised as simple as the single PUF, i.e., the 2-bit PUF can be modeled with ≈100% accuracy for SNR of $\sigma =$ 16e−4 or less, and the drop occurs at $\sigma =$ 32e−4 which presents 94.8% accuracy. Investigating the SNR values gives a better picture in this case. The maximal SNR values for these cases are displayed in Table 2; when sigma is 16e−4 or lower (with averaging) the SNR is higher than then 1.81 (our baseline in real silicon) and as expected the attack accuracy is 100%. For sigma beyond this value, although not 100% but our attack was still quite successful. As expected, with averaging of 10 traces the SNR increases around 10 times. This confirms the ease of attack when averaging technique is applied.

Table 2. The maximum SNR when the Flip-Flops are queried in the unprotected parallel PUF with & without averaging.

	σ = 2.5e−4	σ = 9.5e−4	σ = 16e−4	σ = 32e−4	σ = 64e−4
Non-averaging	9.713419	0.658853	0.232567	0.062253	0.016153
Averaging	94.948925	6.666043	2.344949	0.589082	0.146669

The takeaway from these results is that the parallel multi-bit PUF is indeed vulnerable to power side-channel based modeling attacks when the SNR is even lower than that seen in real silicon. This is because the entropy introduced by the side-channel leakage of an M-bit response PUF is not really M bit as expected but rather is equal to the smaller quantity (where $X \in \{0,1\}^M$ and w_H is the Hamming weight function):

$$H(w_H(X)) = -\sum_{i=0}^{M} \frac{1}{2^M}\binom{M}{i} \log_2\left(\frac{1}{2^M}\binom{M}{i}\right) \text{ bit}, \tag{5}$$

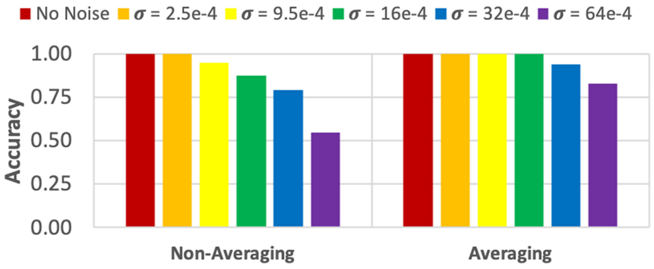


Fig. 8. The attack accuracy targeting an unprotected 2-bit response PUF for various noise levels with & without using averaging scheme during the attack. Ideally the accuracy, from a design standpoint, for a 2-bit PUF is 25%.

resulting in the entropy of 1.5 for $M = 2$. Moreover, although an accuracy of 75% would be expected for the 2-bit PUF corresponding to entropy of 1.5, it is only the imbalance and mismatch of the Flip-Flops and their capacitive loads which allow the adversary to discriminate the state "01" from "10". The results also show that the averaging scheme is highly effective in improving SNR as one would expect.

Reducing the SNR of the Target Flip-Flop's Leakage: To mitigate the attack, the proposed countermeasure shown in Fig. 3 was implemented. The modeling attack accuracy in presence of this DRILL countermeasure is shown in Fig. 9. As depicted the DRILL countermeasure is effective at mitigating the attack when the noise level is $\sigma = 9.5\text{e}{-4}$ and beyond if averaging scheme is not applied, while benefiting from averaging scheme results in a more successful attack and increasing the accuracy to 100% for $\sigma = 2.5\text{e}{-4}$ and to 95.7% for $\sigma = 9.5\text{e}{-4}$, respectively. Even with averaging, our DRILL countermeasure is highly successful for noises with $\sigma > 9.5\text{e}{-4}$. Note that all SNR values reported in Table 3, are much lower than the real-silicon baseline (i.e., 1.81) we referred to earlier.

Table 3. The maximum SNR for the proposed DRILL Protected PUF with and without averaging.

	$\sigma = 2.5\text{e}{-4}$	$\sigma = 9.5\text{e}{-4}$	$\sigma = 16\text{e}{-4}$	$\sigma = 32\text{e}{-4}$	$\sigma = 64\text{e}{-4}$
Non-averaging	0.034776	0.00476	0.002327	0.001941	0.001105
Averaging	0.185653	0.025787	0.011142	0.004322	0.001901

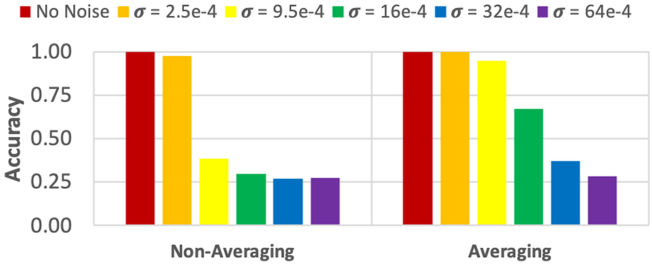


Fig. 9. The attack accuracy when the 2-bit response parallel PUF is protected with the proposed DRILL countermeasure. Ideally the accuracy, from a design standpoint, for a 2-bit PUF is 25%.

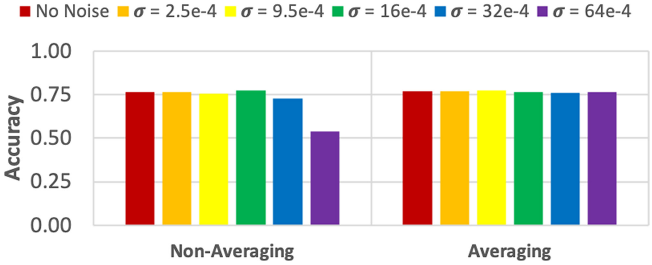


Fig. 10. The modeling accuracy when the 2-bit response PUF is only protected with RAS scheme. Ideally the accuracy, from a design standpoint, for a 2-bit PUF is 25%.

The takeaway point from these results is that the proposed DRILL countermeasure does mitigate the attack. However, we need to improve its resiliency at low noise levels.

Confusing and Poisoning Countermeasure Results: To further mitigate the modeling attacks, we implemented the proposed countermeasures of RAS (shown in Fig. 4) and RRM (depicted in Fig. 5). The accuracy of modeling attack when the PUF is only equipped with RAS is shown in Fig. 10. As depicted, in almost all cases, regardless if the averaging scheme is used or not, the accuracy is ≈75%. This is because, with such swapping, only the cases in which the responses equal to "01" or "10" are protected, but the "11" and "00" cases are still differentiable. Although this protection may seem not effective in 2-bit response PUFs, its inclusion in the PUFs with more response bits is promising. For example, for a 3-bit response PUF, the leakage of 6 out of 8 response cases (all 3-bit combinations of responses except "000" and "111") is reduced when the RAS scheme is adopted.

To improve the resiliency of the parallel-PUF against modeling attacks, we inserted our RAS scheme on top of the DRILL protection. Figure 11 depicts the related modeling accuracies. As shown the results are very promising. Even when the attacker uses the averaging scheme, the accuracy does not exceed 60%.

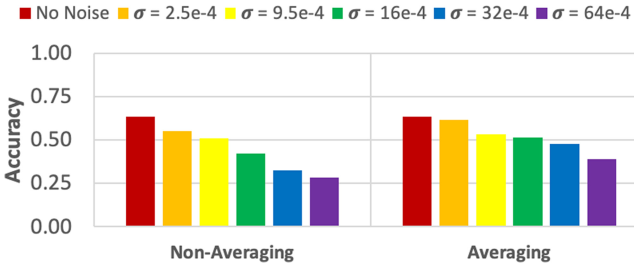


Fig. 11. The accuracy of modeling the 2-bit response PUF when RAS scheme is applied on top of DRILL countermeasure. Ideally the accuracy, from a design standpoint, for a 2-bit PUF is 25%.

The takeaway point from these observation is that the combination of RAS and DRILL countermeasures can highly protect the multi-bit PUF against modeling attacks.

We further applied the RRM (shown in Fig. 5). The results (not shown for the sake of space) confirm that this countermeasure is highly successful in thwarting the modeling attack, with the accuracy consistently being at 25% in all noise levels.

4-bit Parallel PUF Results: To assess the parallelization as a natural countermeasure, we launched similar attacks on a 4-bit parallel PUF. The power traces when the Flip-Flops are registering their responses is shown in Fig. 12. As shown the hamming weight of the response is clearly discernible, while the trace for the individual response is less distinguishable.

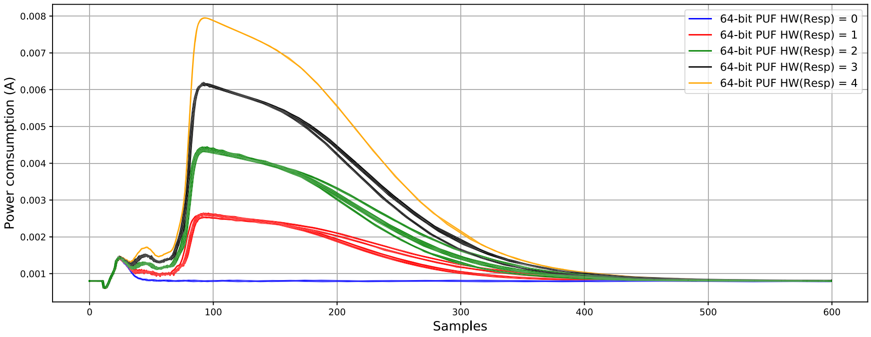


Fig. 12. Superimposing 50 traces of the 64-bit PUF in 4-bit parallel settings. Note that *HW(Resp)* denotes the Hamming weight of a 4-bit response *Resp*.

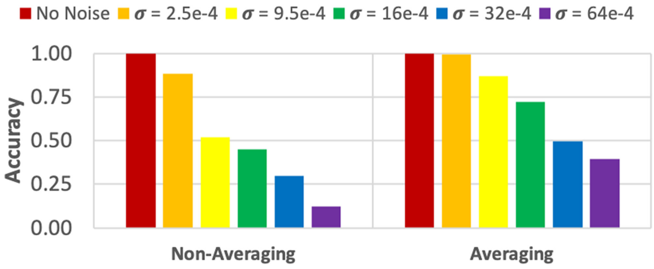


Fig. 13. The attack accuracy targeting a 4-bit response arbiter-PUF for various noise levels. Ideally the accuracy, from a design standpoint, for a 4-bit PUF is 6.25%.

The results of attacking are shown in Fig. 13. Compared to the 2-bit results in Fig. 8, we can clearly see that the individual 4-bit responses were less distinguishable, thus the attack is less accurate at predicting the responses of a 4-bit response parallel PUF. In fact, with noise levels' greater than $\sigma = 2.5\text{e}{-4}$ the response is not modelable, and the accuracy of modeling is less than 75%.

9 Discussions

In the attack of the multi-bit arbiter PUF, we considered slight variations between PUFs, realized as small differences between output capacitances of the response flip-flops due to process mismatch. Indeed, these variations increase the adversary's ability to successfully attack the PUF through its power traces. It is interesting to note that the technological imbalance which is the essence of the PUF, can reduce the efficiency of the protections, notably when using multi-bit responses. For the simulation results, we used variation values that are typical in real designs. Moreover, even though the capacitances C_{H1} and C_{H2} are rigorously equal, an attacker resorting to small-sized electromagnetic probes (instead of powerline fluctuations measurements with an ammeter/oscilloscope) could make a difference depending on the provenance (H1 vs H2) of the leakage [19]. Thus, in the sequel, we should consider that somehow, the adversary manages to collect even slightly different signals originating from either capacitance.

The use of multi-bit arbiter PUF does not only provide intrinsic security enhancement against side-channel observations but also against invasive attacks which manage to monitor the PUF while it is challenged [26]. Such attacks are very powerful as they require the knowledge of the PUF layout and its position within the chip, as well as a perfect synchronization with the challenges. Few PUFs (and actually few security IPs in general) resist such attacks. However, we notice that the use of parallel PUFs, provided the M instances are sufficient apart, allow to thwart this attack, since the imaging sensor has a small aperture size.

Finally, we wish to remind the reader that in this work the randomly generated variables used in mask generation and arbiter swapping are not exploited by the attacker. This is the case if they are generated before the arbitration time and the attack is at first order.

10 Conclusion and Future Directions

We investigated the resiliency of parallel multi-bit response arbiter-PUFs against power side-channel based modeling attacks. The results confirm the vulnerability of such PUFs against the power based modeling attacks, especially when the adversary benefits from the averaging technique. We proposed a number of countermeasures based on hiding the power consumption by equalizing the power for different response values as well as randomizing the relation between the power consumption and response bits. We also showed that increasing the number of bits in multi-bit responses naturally improve the security against power modeling attacks. The results confirmed the efficacy of the proposed countermeasures in thwarting the power based modeling attacks in parallel arbiter-PUFs. All findings can be extended to the arbiter-PUF derivatives considering their architecture. We plan to extend this research by investigating the findings on real silicon, and for larger multi-bit response implementations.

Acknowledgments. This work has benefited from a funding via the bilateral project APRIORI (*Advanced Privacy of IOT Devices through Robust Hardware Implementations*), from FR-DE cybersecurity 2020 call (MESRI-BMBF), managed by ANR from the French side.

References

1. Nangate 45nm open cell library. http://www.nangate.com
2. Aghaie, A., Moradi, A.: TI-PUF: toward side-channel resistant physical unclonable functions. TIFS **15**, 3470–3481 (2020)
3. Mahmoud, A., et al.: Combined modeling and side channel attacks on strong PUFs. IACR Crypt. ePrint Arch. **2013**, 632 (2013)
4. Gassend B., et al.: Silicon physical random functions. In: CCS, pp. 148–160 (2002)
5. Gu, C., et al.: A modeling attack resistant deception technique for securing PUF based authentication. In: AsianHOST, pp. 1–6 (2019)
6. Merli, D., et al.: Side-channel analysis of PUFs and fuzzy extractors. In: Trust and Trustworthy Computing, pp. 33–47 (2011)
7. Fukushima, K., et al.: Delay PUF assessment method based on side-channel and modeling analyzes: the final piece of all-in-one assessment methodology. In: IEEE Trustcom/BigDataSE/ISPA, pp. 201–207 (2016). https://doi.org/10.1109/TrustCom.2016.0064
8. Jiang, Q., et al.: Two-Factor Authentication Protocol Using Physical Unclonable Function for IoV. In: IEEE/CIC ICCC, pp. 195–200 (2019)
9. Zalivaka, S.S., et al.: Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response. IEEE TIFS **14**(4), 1109–1123 (2019)
10. Kroeger, T., et al.: Cross-PUF attacks on arbiter-PUFs through their power side-channel. In: ITC (2020)

11. Kroeger, T., et al.: Effect of aging on PUF modeling attacks based on power side-channel observations. In: DATE, pp. 454–459 (2020)
12. Rührmair, U., et al.: Efficient power and timing side channels for physical unclonable functions. In: CHES, pp. 476–492 (2014)
13. Alkatheiri, M.S., Zhuang, Y.: Towards fast and accurate machine learning attacks of feed-forward arbiter PUFs. In: IEEE Conference on Dependable and Secure Computing, pp. 181–187 (2017). https://doi.org/10.1109/DESEC.2017.8073845
14. Danger, J.-L., Guilley, S., Pehl, M., Senni, S., Souissi, Y.: Highly reliable PUFs for embedded systems, protected against tampering. In: Vo, N.-S., Hoang, V.-P., Vien, Q.-T. (eds.) INISCOM 2021. LNICSSITE, vol. 379, pp. 167–184. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77424-0_14
15. Gao, Y., et al.: Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices. In: PerCom Workshops, pp. 1–6 (2016)
16. Guilley, S., Hamaguchi, S., Kang, Y.: ISO/IEC 20897–1:2020. Information security, cybersecurity and privacy protection - Physically unclonable functions - Part 1: Security requirements (2020). https://www.iso.org/standard/76353.html
17. Helfmeier, C., Boit, C.: Cloning Physically Unclonable Functions. In: HOST, pp. 1–6 (2013). https://doi.org/10.1109/HST.2013.6581556
18. Herder, C., et al.: Physical unclonable functions and applications. Tutorial. Proc. IEEE **102**(8), 1126–1141 (2014)
19. Immler, V., Specht, R., Unterstein, F.: Your rails cannot hide from localized EM: how dual-rail logic fails on FPGAs. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 403–424. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_20
20. Kroeger, T., Cheng, W., Guilley, S., Danger, J., Karimi, N.: Making obfuscated PUFs secure against power side-channel based modeling attacks. In: DATE (2021)
21. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer, Cham (2006)
22. Mars, A., Adi, W.: New concept for physically-secured e-coins circulations. In: Adaptive Hardware and Systems, pp. 333–338 (2018)
23. Nedospasov, D., Seifert, J., Helfmeier, C., Boit, C.: Invasive PUF analysis. In: FDTC, pp. 30–38 (2013). https://doi.org/10.1109/FDTC.2013.19
24. Rührmair, U., Sölter, J.: PUF modeling attacks: an introduction and overview. In: DATE, pp. 1–6 (2014). https://doi.org/10.7873/DATE.2014.361
25. Rührmair, U., et al.: Power and timing side channels for PUFs and their efficient exploitation. IACR Cryptol. ePrint Arch. **2013**, 851 (2013)
26. Tajik, S., et al.: Photonic side-channel analysis of arbiter PUFs. J. Cryptol. **30**(2), 550–571 (2017)
27. Tebelmann, L., et al.: Self-secured PUF: protecting the loop PUF by masking. In: COSADE, Lugano, 5–7 October (2020)
28. Vijayakumar, A., Kundu, S.: A novel modeling attack resistant PUF design based on non-linear voltage transfer characteristics. In: DATE, pp. 653–658 (2015). https://doi.org/10.7873/DATE.2015.0522
29. Wang, Q., Gao, M., Qu, G.: A machine learning attack resistant dual-mode PUF. In: Great Lakes Symposium on VLSI, pp. 177–182 (2018). https://doi.org/10.1145/3194554.3194590
30. Yu, M.M., Hiller, M., Delvaux, J., Sowell, R., Devadas, S., Verbauwhede, I.: A lockdown technique to prevent machine learning on PUFs for lightweight authentication. IEEE Trans. Multi Scale Comput. Syst. **2**(3), 146–159 (2016). https://doi.org/10.1109/TMSCS.2016.2553027

31. Yu, Y., et al.: Profiled deep learning side-channel attack on a protected arbiter PUF combined with bitstream modification. Cryptology ePrint Archive, Report 2020/1031 (2020). https://eprint.iacr.org/2020/1031
32. Zhou, C., Parhi, K.K., Kim, C.H.: Secure and reliable XOR arbiter PUF design: an experimental study based on 1 trillion challenge response pair measurements. In: Proceedings of the 54th Annual Design Automation Conference 2017. DAC 2017, Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3061639.3062315